Track fitting, vertex fitting and Detector alignment

Salvador Martí-García
IFIC-Valencia

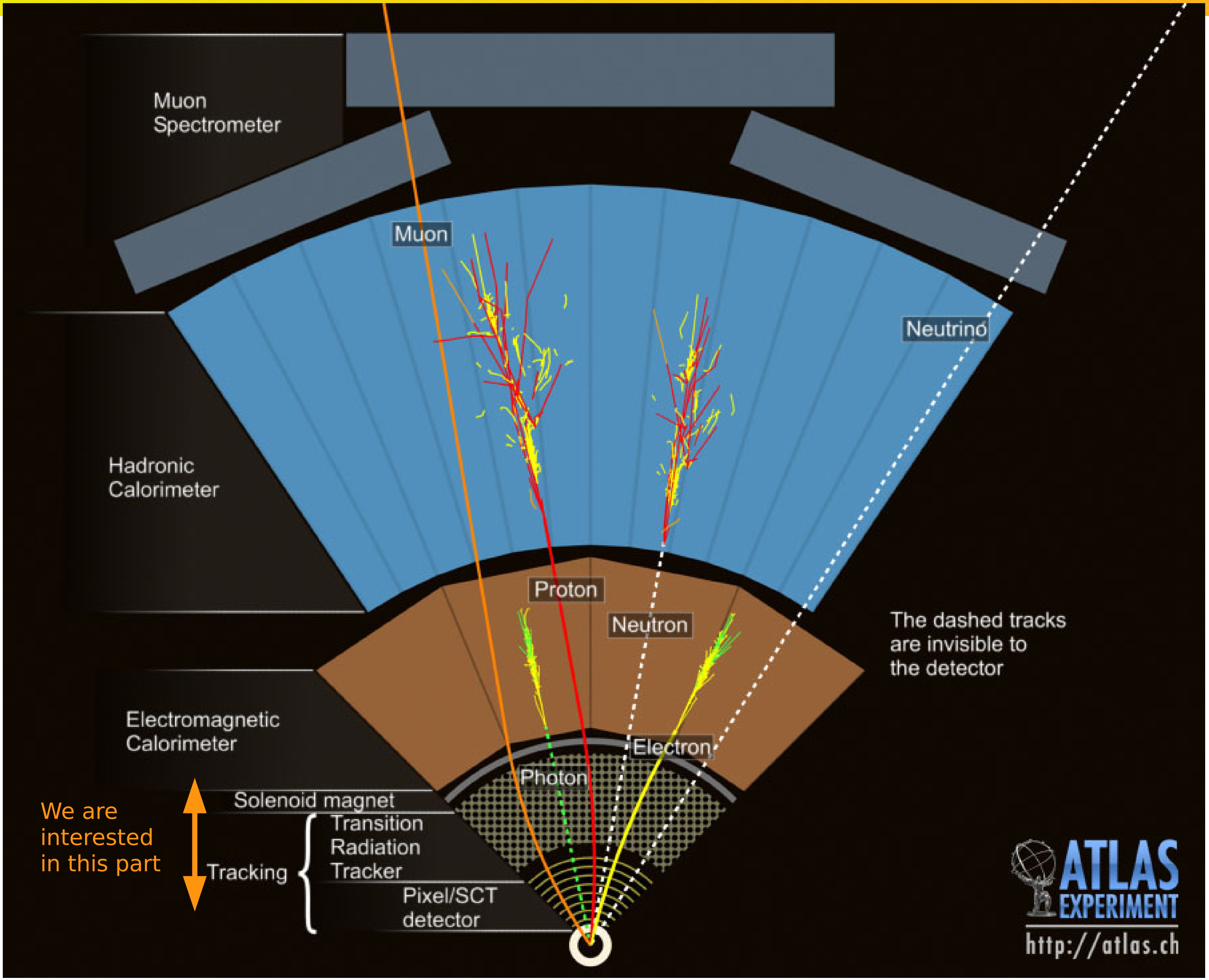
[Usain Bolt (JAM) in Berlin 2009]

Outline

- Track fitting
  - Basic ideas & concepts
  - Basic formulae
  - Signal processing
  - Global and local reference frames
  - Pattern recognition
  - Track fitting with χ^2 and Kalman filter techniques
  - Multiple Coulomb Scattering
- Vertex fitting
  - Basic ideas & concepts
  - Billoir vertex fitting
  - Addaptive vertex fitting
- Detector alignment
  - Basic ideas & concepts
  - Basic formulae
  - Alignment strategy
  - Alignment systematics

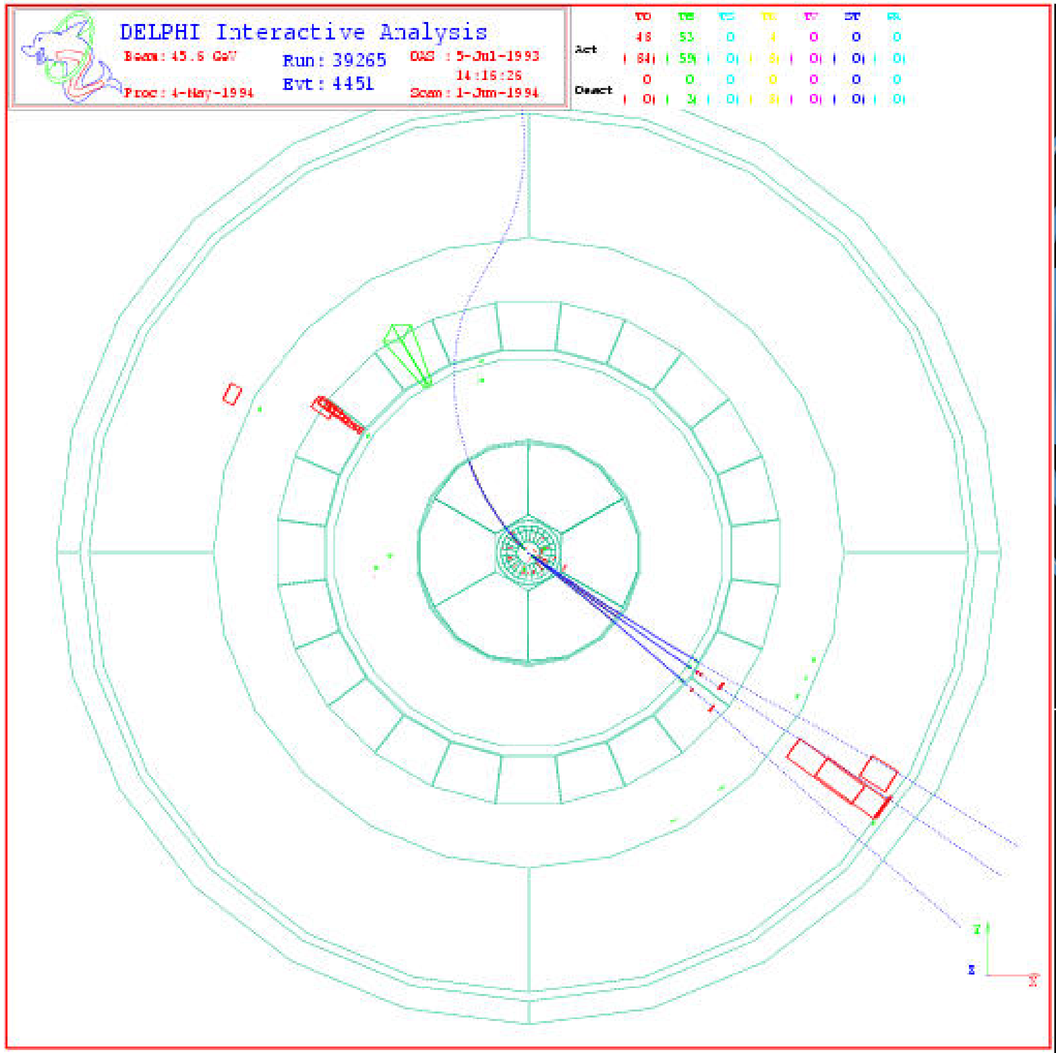
Disclaimer: the geometry description is an important issue that is not treated in this lecture

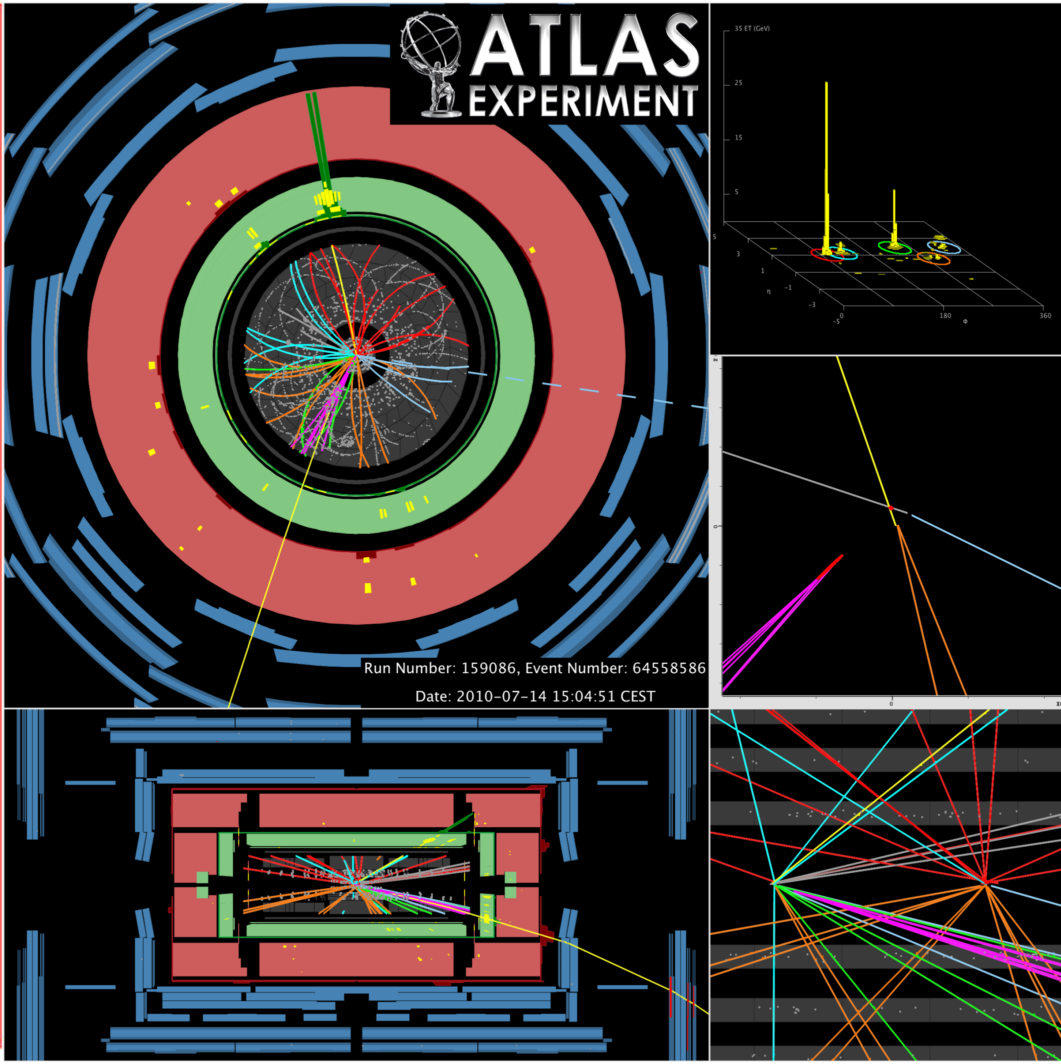
Particles and detectors



Introduction: tracking what for ?

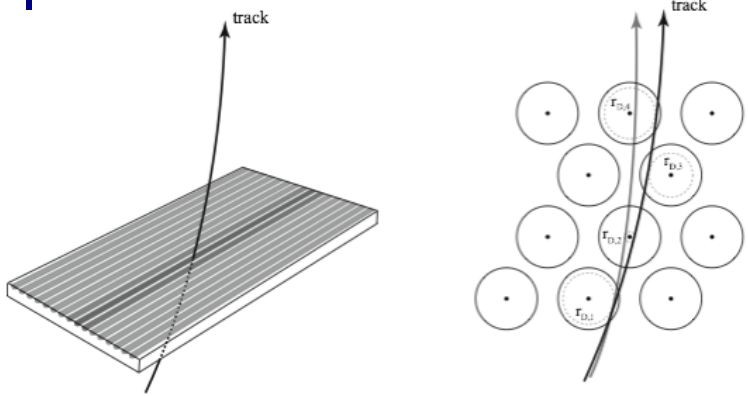
- Tracking allows to determine the properties of those charged particles present in an experiment





Introduction: tracking what for ?

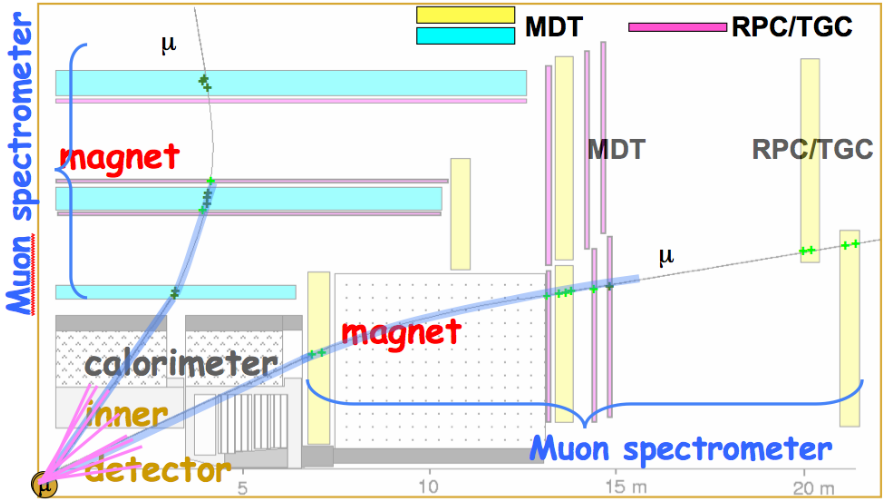
- Tracking allows to determine the properties of those charged particles present in an experiment
  - Where is the particle ?
  - Where does it go ?
  - At which speed travels ?
- Tracking is possible because charged particles interact with detector material
  - Energy loss by ionization
    - Bethe-Bloch formula



- A good performance of the Track Fitting is a key ingredient of the success of the physics program of the HEP experiments
  - An accurate determination of the charged particles properties is necessary
    - Invariant masses have to be determined with optimal precision and well estimated errors
    - Secondary vertices must be fully reconstructed: evaluate short lifetimes
    - Kink reconstruction: on flight decays

Introduction: tracking what for ?

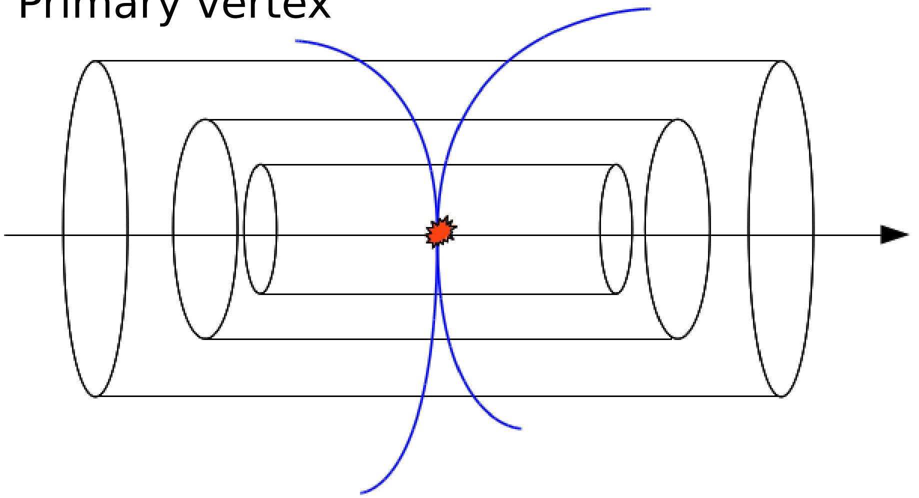
- Challenges for the tracking systems of the LHC detectors
  - Momenta of particles in the final state ranging from MeV to TeV
  - High multiplicity of charged particles (up to 1000 for $\mathscr{L} \sim 10^{34} cm^{-1} s^{-1}$)
    - Even higher for heavy ion collisions
  - Large background from secondary activities of the particles
  - Multiple Coulomb Scattering in detector frames, supports, cables, pipes...
  - Complex modular tracking systems combining different detecting technologies, different resolutions
  - Resolutions that vary as a function of the momentum (p), polar angle (θ) or pseudorapidity (η)
  - Very high event rates leading to large amount of data
    - with demanding requirements of CPU and storage → Tracking CPU budget



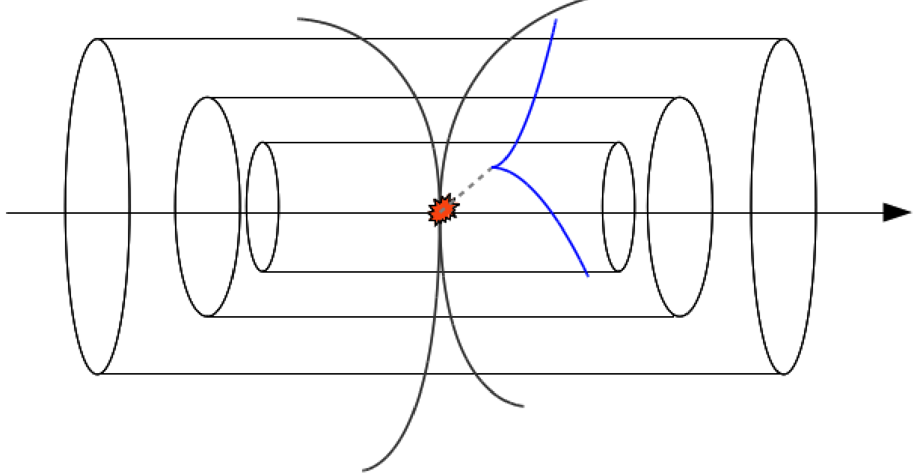
- Finding where the particle was originated tell us much about the physics: primary vertex, secondary vertex or material interactions

Vertex fitting capabilities depend
on tracking performance
(specially in impact parameter resolution)

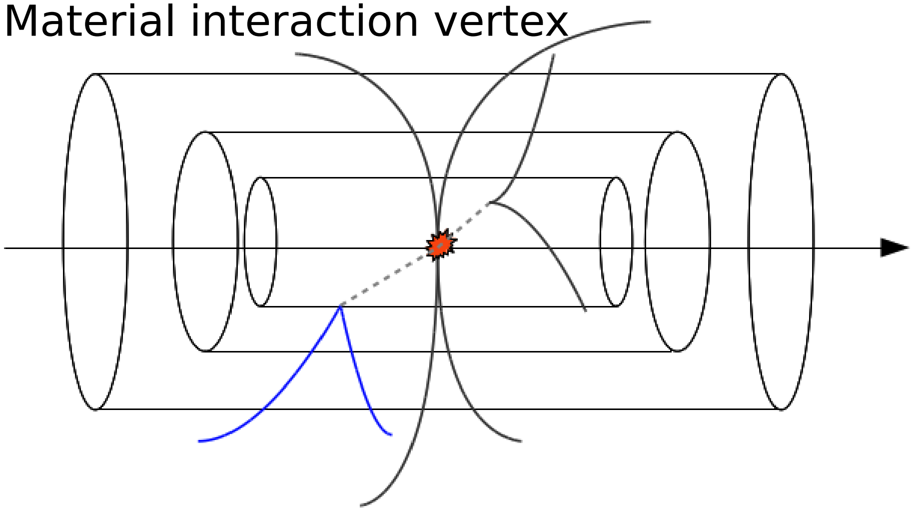
Primary vertex



Secondary vertex: particle decay

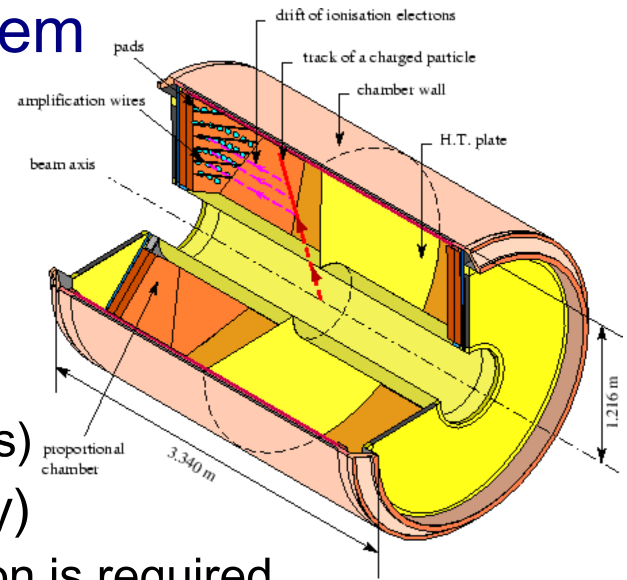


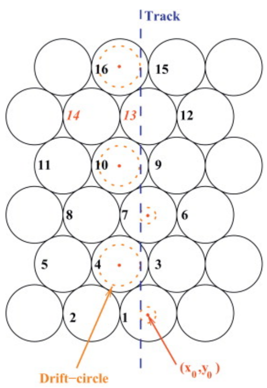
Material interaction vertex

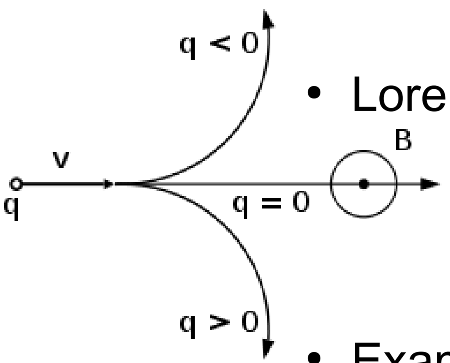


Basic ingredients

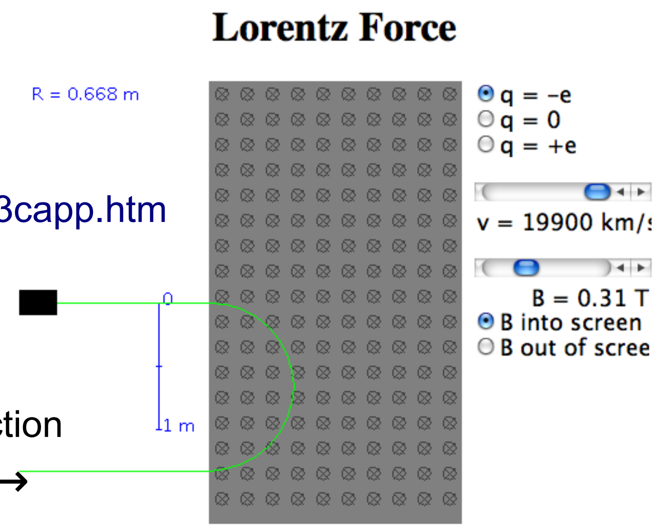
- Basic ingredients of the tracking system
  - Charged particles (+ve or -ve)
    - |q| = 1, 2
    - e, μ, π, k, p, α, d,...
  - Ionization detector
    - Continuous (e.g.: gas detectors)
    - Discrete (e.g.: silicon planar detectors)
  - Magnetic field (no strictly necessary)
    - Necessary if momentum determination is required
      - Some times experiments runs with magnets switched off
    - Lorentz force







$$\vec{F} = q\left(\vec{E} + \vec{v} \times \vec{B}\right)$$

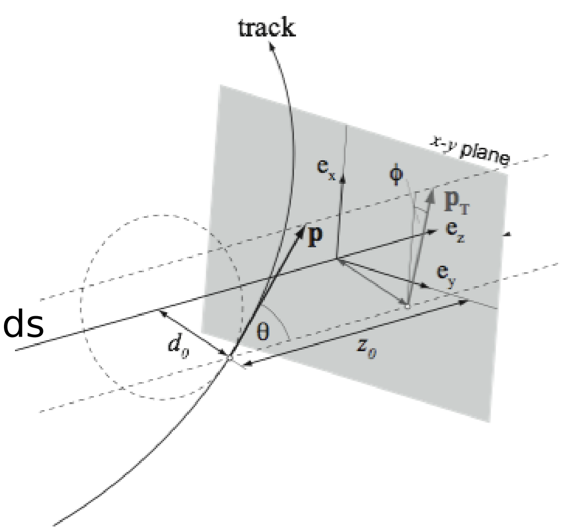


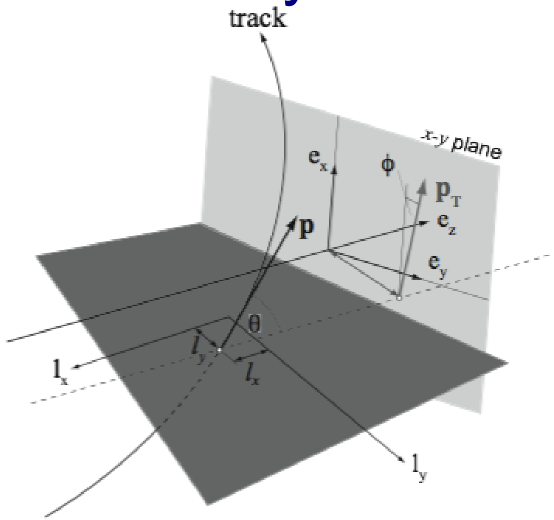
- Example: Nice Java applet
  - http://www.lon-capa.org/~mmp/kap21/cd533capp.htm
- Usually $\vec{E}$=0 inside detectors
  - Or quite small
  - Negligible effects on tracks
  - E > 0 necessary for ionization charge collection
- The bending of the trajectory is due to B field →

Track parameters

- A trajectory can be parametrized with just 5 parameters at a surface
  - x, y, φ, θ, v
- The track extrapolation to detector surfaces usually requires a different parametrization
  - Optimization
    - Track parameters given in the local reference frame of the surface
  - Error matrix propagation !
- The track is characterized by 5 its parameters as given at the "perigee surface" & using the global reference coordinate system
  - d_0, z_0, φ_0, θ_0, q/p
  - d_0, z_0, φ_0, $\cot\theta_0$, $q \cdot p_T$
  - d_0, z_0, φ_0, η, q/p

the choice of parametrization depends on the detector layout

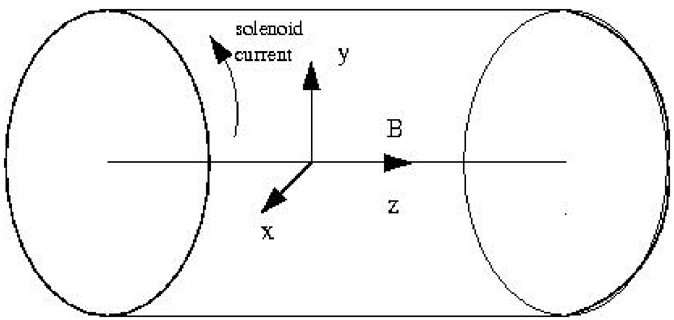


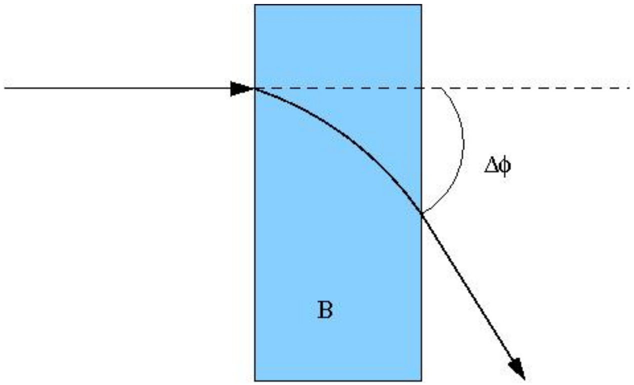


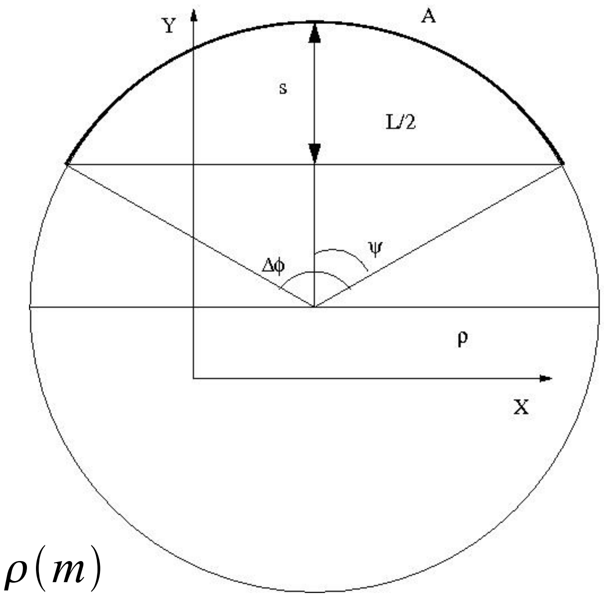
- Track extrapolation
    - Heavily used in tracking code and alignment code

Basic track formulæ

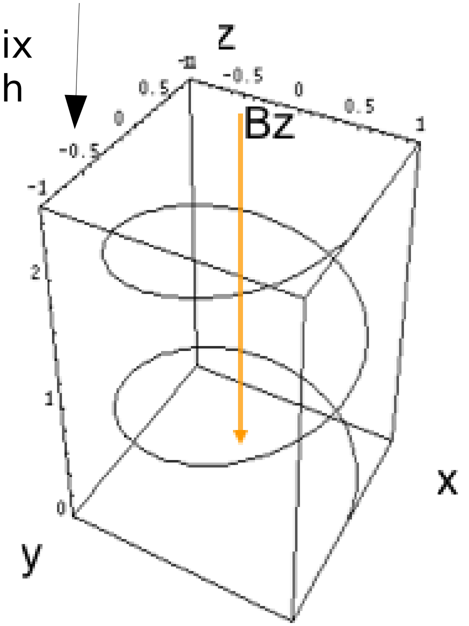
- Consider axial (along Z) and uniform B field
  - From a solenoid field as in most of the HEP experiments trackers.
  - Charged particles follow a helicoidal path
    - Describe circles in the XY (transverse plane) due to Lorentz force
    - Move uniformly along Z







Helix path



$$\vec{F} = q\vec{v} \times \vec{B}$$

$$p_T(GeV/c) = 0.3\, q\, B(T)\, \rho(m)$$

$$\rho = \frac{L^2}{8s} + \frac{s}{2}$$

$$\frac{\delta p_T}{p_T} = \frac{\delta s}{s}$$

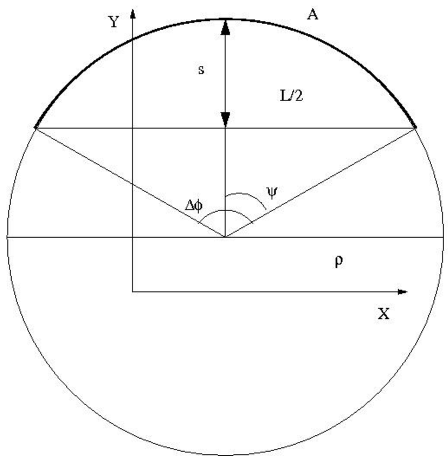
$$\frac{\delta p_T}{p_T} \propto \frac{\delta s}{B L^2} p_T$$

← Momentum resolution

s is the sagitta. It tells us how much the track has deviated from a straight trajectory

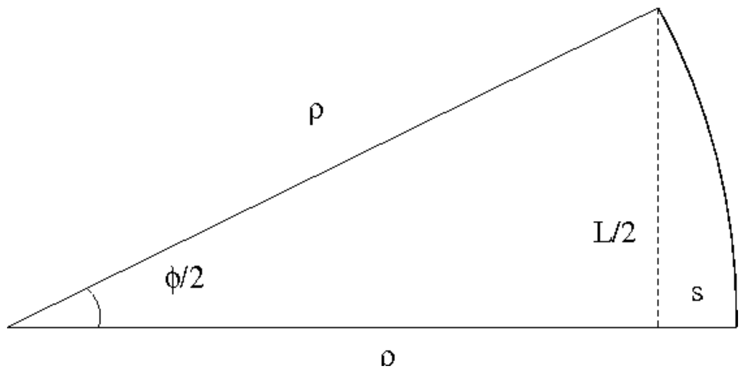
Sagitta

- The sagitta is a measure of the bowing (bending) of the trajectory
- It is the basic parameter that gives information about the momentum
  - Actually,it gives information about the transverse momentum with respect to the B field axis



$$\sin\frac{\phi}{2}=\frac{L}{2\rho}$$

$$\cos\frac{\phi}{2}=\frac{\rho-s}{\rho}$$



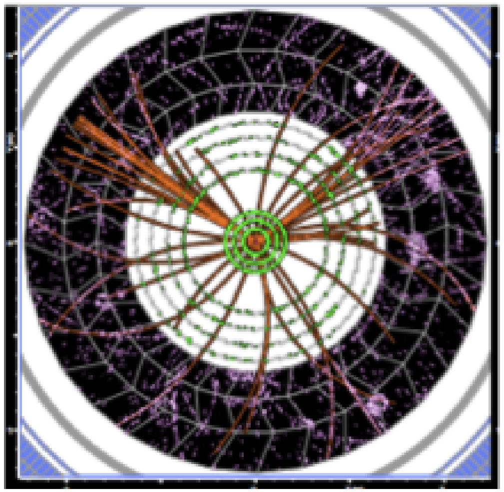
$$\sin^2\frac{\phi}{2}+\cos^2\frac{\phi}{2}=1 \;\rightarrow\; \frac{L^2}{4\rho^2}+\frac{(\rho-s)^2}{\rho^2}=1 \;\rightarrow\; \rho=\frac{L^2}{8s}+\frac{s}{2}$$

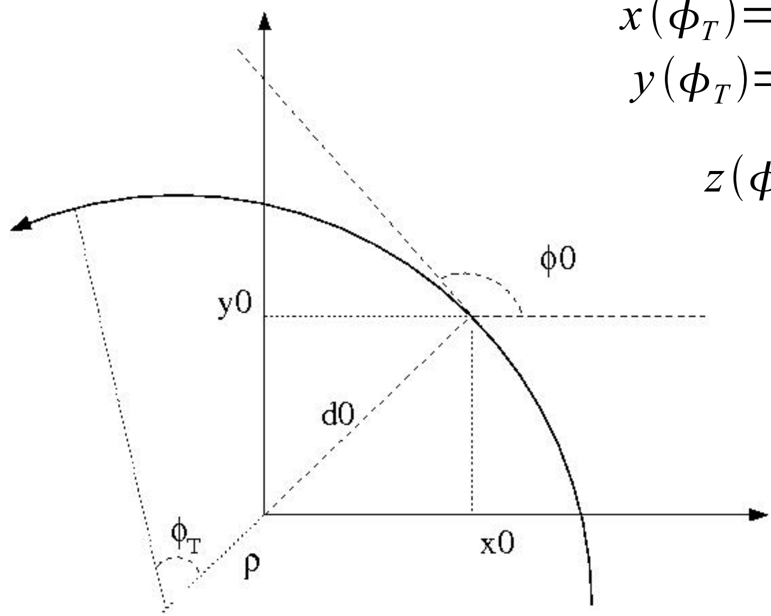
  - Usually L≫ s $\quad \rho\approx\frac{L^2}{8s} \;\rightarrow\; d\rho=\frac{-L^2}{8s^2}ds=-\rho\frac{ds}{s} \;\rightarrow\; \frac{d\rho}{\rho}=-\frac{ds}{s}$
  - That means: when the radius increases, the saggita decreases
    - Large momentum particles, have large radius and small sagita
  - The precision on the saggita measurement is the limiting factor of the momentum resolution
    - Sagitta resolution is closely linked to detector resolution

$$\frac{\delta p_T}{p_T^2}\propto\frac{\delta s}{B L^2}$$

Basic track formulae

- Helix trajectory of charged particles
  - Parametrization of the helix: (x,y,z) of a trajectory point as a function of a single path parameter

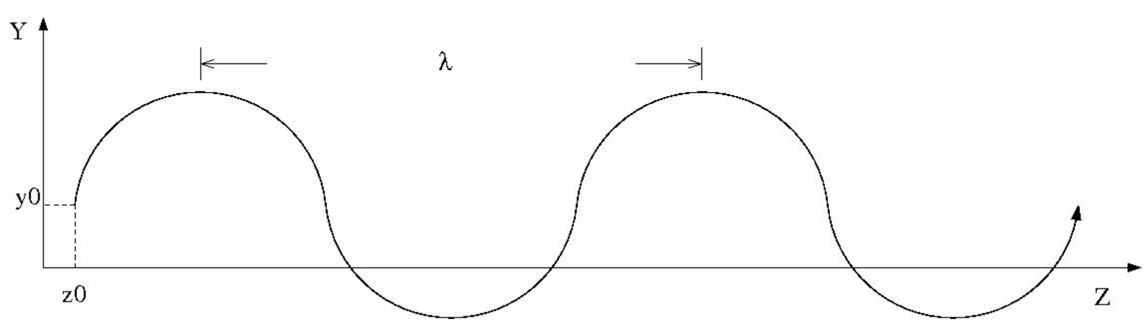




$$x(\phi_T) = -q\rho\sin(\phi_0 - q\phi_T) + (d_0 + q\rho)\sin\phi_0$$
$$y(\phi_T) = q\rho\cos(\phi_0 - q\phi_T) - (d_0 + q\rho)\cos\phi_0$$
$$z(\phi_T) = z_0 + \lambda\frac{\phi_t}{2\pi} = z_0 + (\rho\cot\theta_0)\phi_T$$

$$x_0 = -d_0\sin\phi_0$$
$$y_0 = d_0\cos\phi_0$$

$$\rho = \frac{p_T}{0.3\,B}$$
$$p_T = p\sin\theta_0$$
$$\rho\cot\theta_0 = \frac{p}{0.3\,B}\cos\theta_0$$

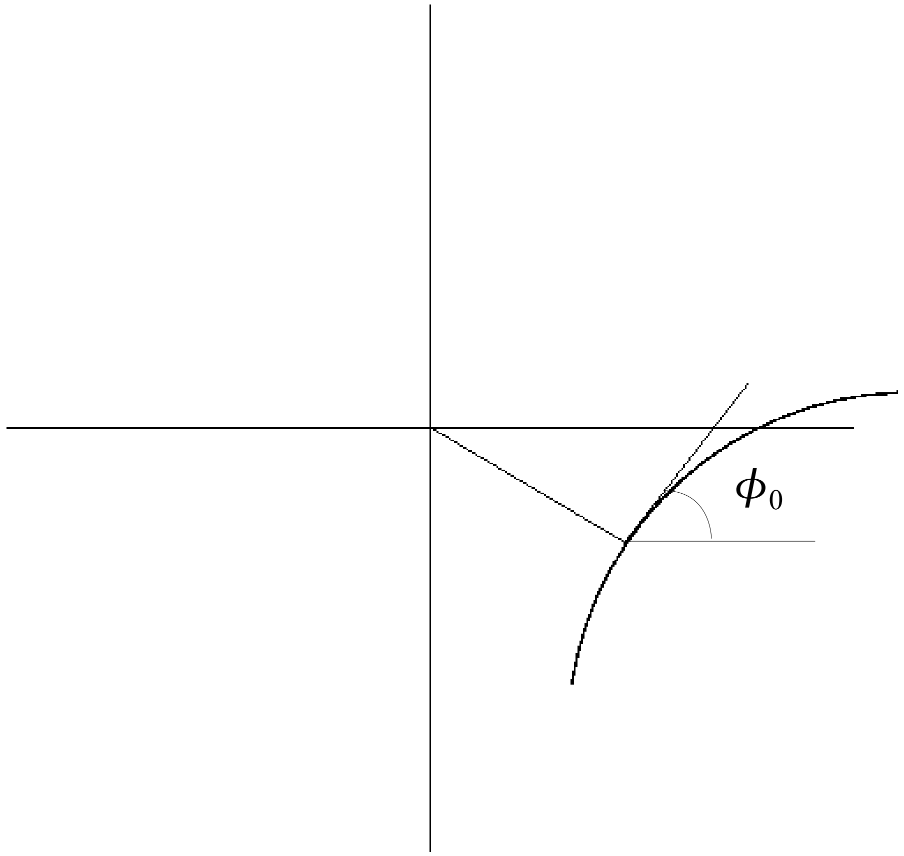


Units: ρ [m], B [T] & p [GeV]

See example at: http://www-jlc.kek.jp/2003oct/subg/offl/lib/docs/helix_manip/node3.html

Signed impact parameter

- It is convenient to give a sign to the impact parameter
  - That helps to compute the perigee point (x_0,y_0) with d_0 and ϕ_0
    - Otherwise a two fold degeneracy occurs

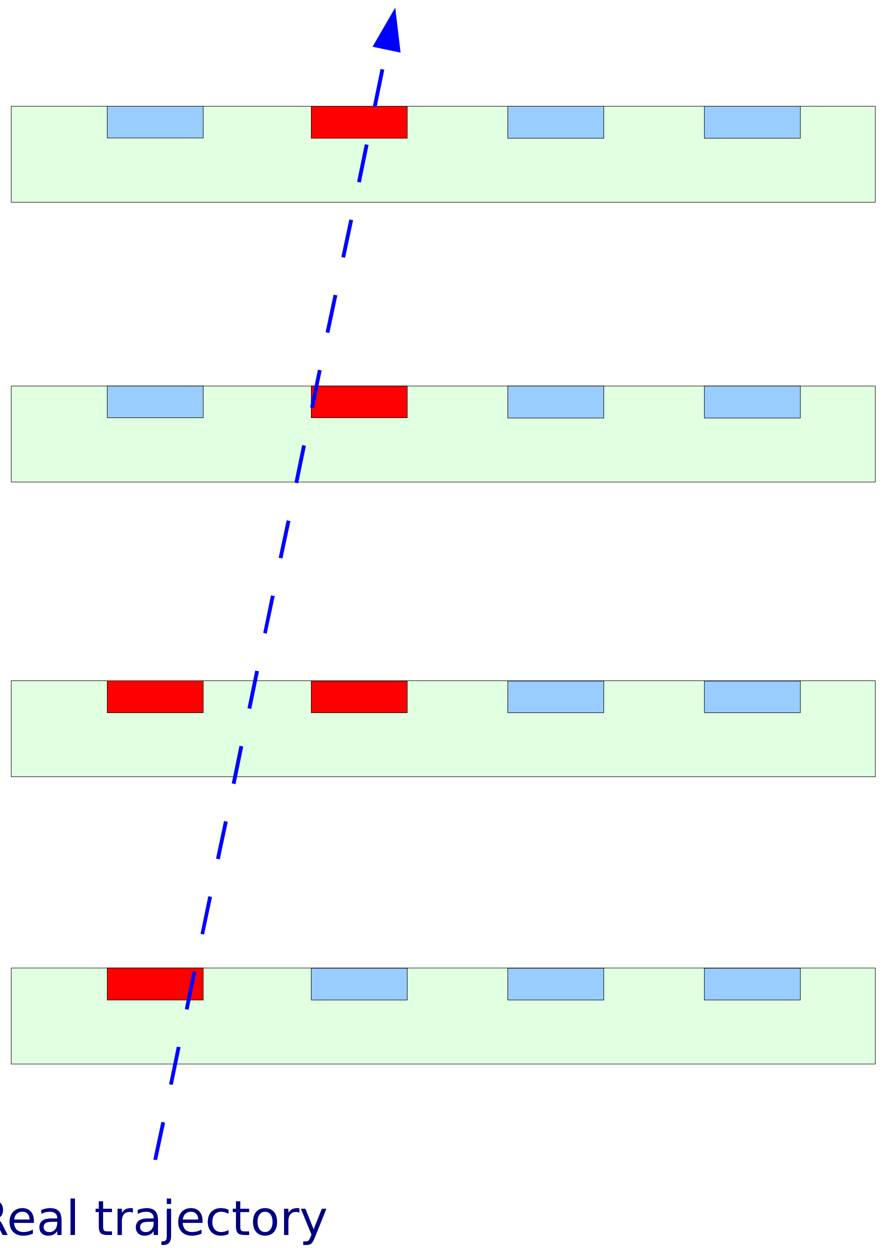


$$x_0 = -d_0 \sin\phi_0$$
$$y_0 = d_0 \cos\phi_0$$

Home work !

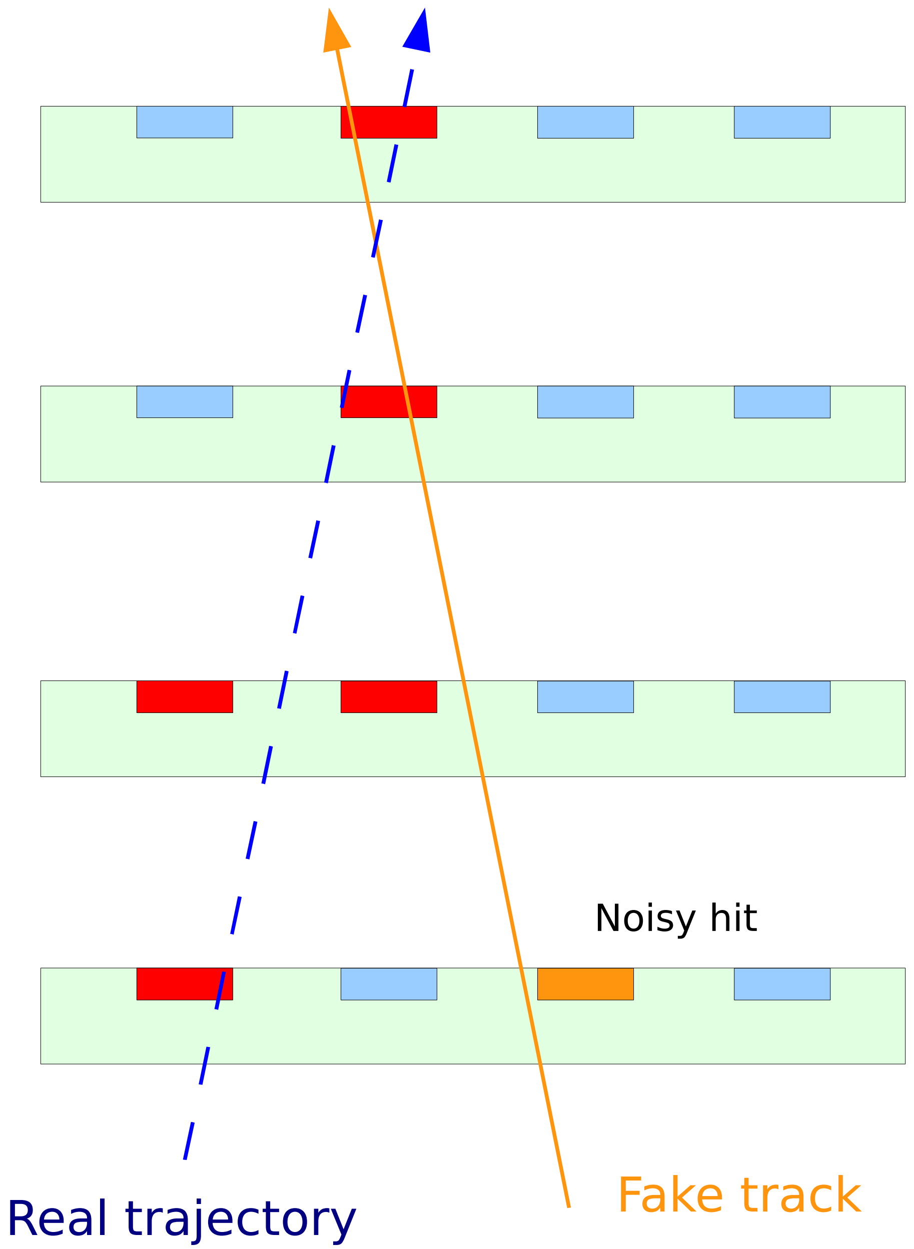
Signal processing for track fitting: hits

- First step is to collect the detector hits → “raw data”
- Need to distinguish genuine signals from noise
- Flag “bad channels” (noisy)
  - Main problem: fake tracks
  - Difficulty: the set of bad channels may not be static
  - Operational conditions may change the bad channels
- Dead channels
  - Main problem → Ineficiency
  - Tracking resolution may be affected: example d0 gets worse when problems in innermost layer



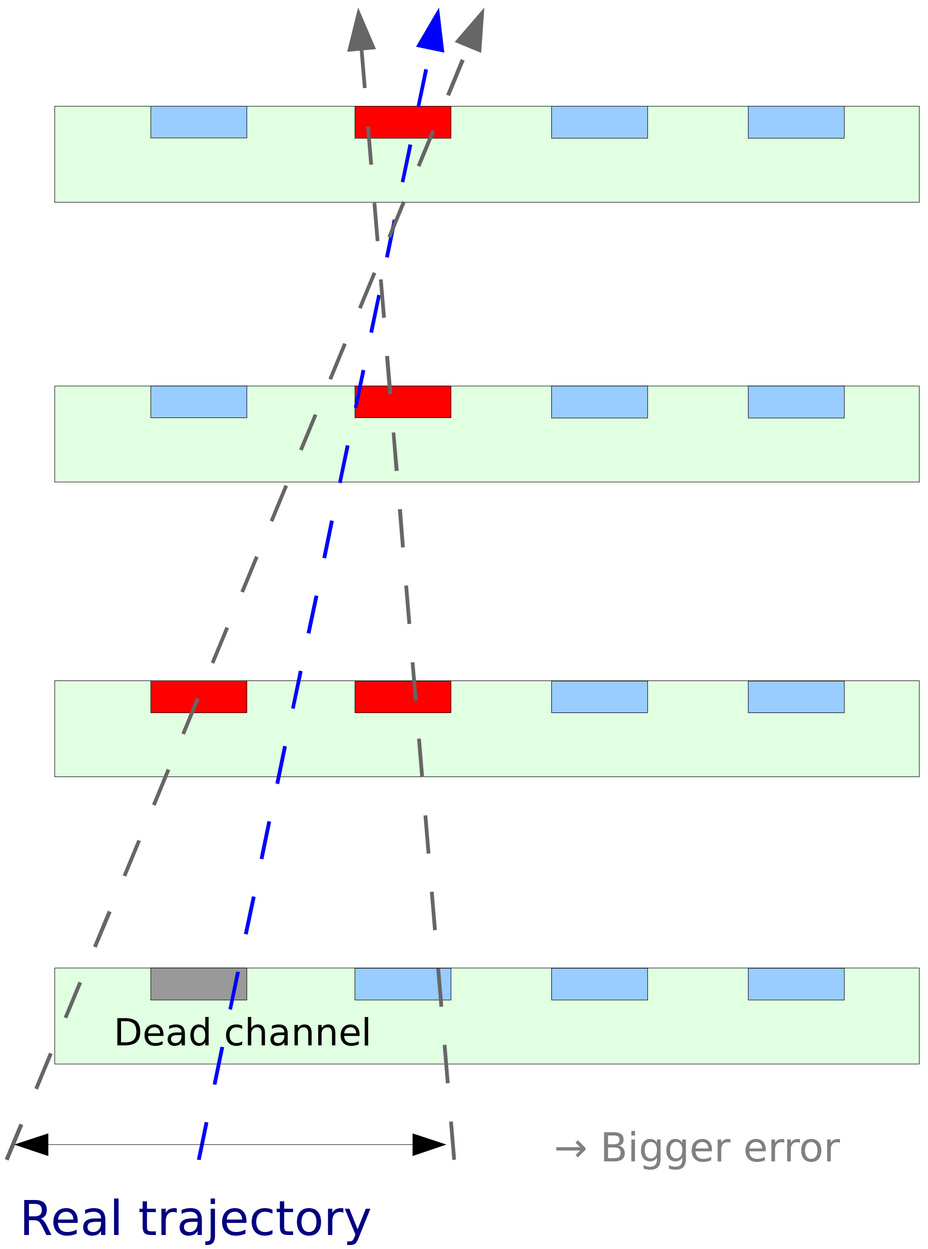
Signal processing for track fitting: hits

- First step is to collect the detector hits → "raw data"
- Need to distinguish genuine signals from noise
- Flag "bad channels" (noisy)
  - Main problem: fake tracks
  - Difficulty: the set of bad channels may not be static
  - Operational conditions may change the bad channels
- Dead channels
  - Main problem → Ineficiency
  - Tracking resolution may be affected: example d0 gets worse when problems in innermost layer



Signal processing for track fitting: hits

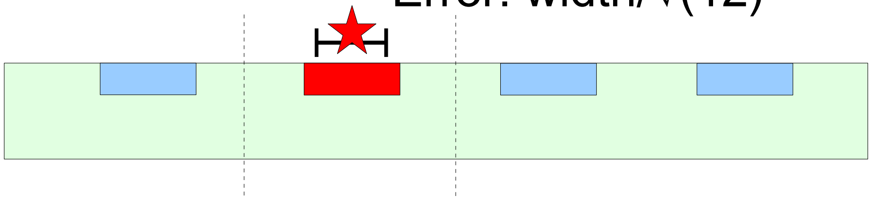
- First step is to collect the detector hits → "raw data"
- Need to distinguish genuine signals from noise
- Flag "bad channels" (noisy)
  - Main problem: fake tracks
  - Difficulty: the set of bad channels may not be static
  - Operational conditions may change the bad channels
- Dead channels
  - Main problem → Ineficiency
  - Tracking resolution may be affected: example d0 gets worse when problems in innermost layer



Signal processing for track fitting: clusters

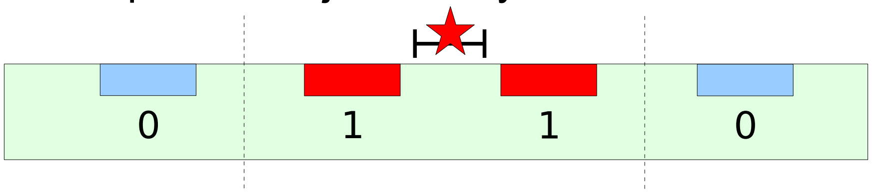
- Hit ↔ channel with signal
  - Detector specific
  - Channel ID & pulse height
- Cluster → group of channels
  - From 1 channel to many
  - 3D information:
    - Global or local coordinates
  - Position: (x, y, z)
  - Error: (δx, δy, δz) in a covariance matrix form
- Cluster position may depend:
  - Hit data: binary or analog
  - Center of gravity
  - Lorentz angle corrections
    - Charge carriers drift
  - Track incident angle
  - MCS corrections

Single channel → Position: channel center
Error: width/√(12)

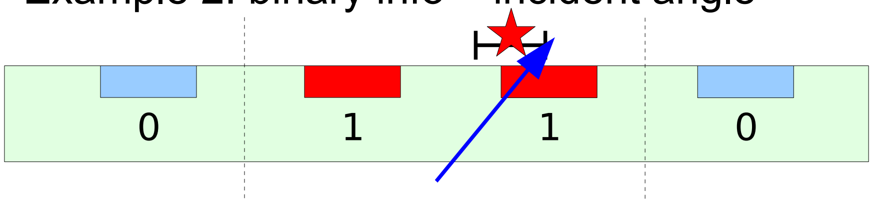


Many channels → Position and error depend on clustering algorithm, hit info (analog or binary), strategy and conditions

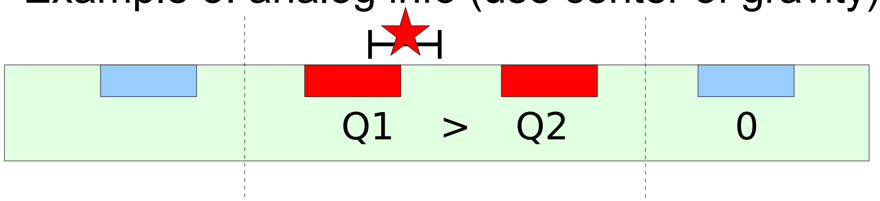
Example 1: use just binary info



Example 2: binary info + incident angle



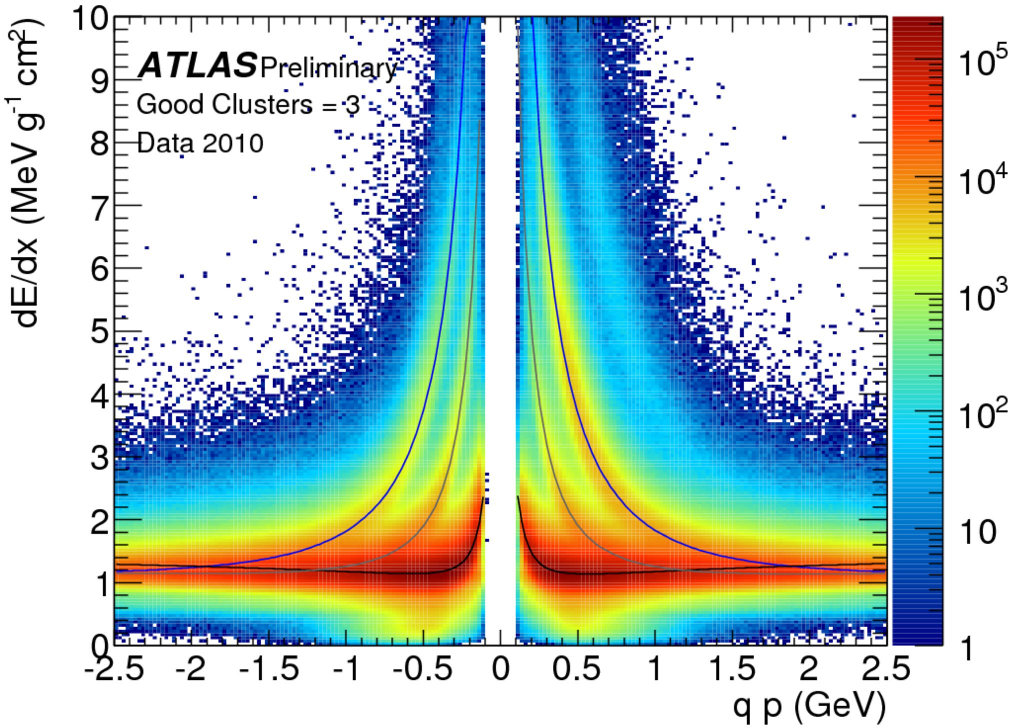
Example 3: analog info (use center of gravity)



Signal processing for track fitting: clusters

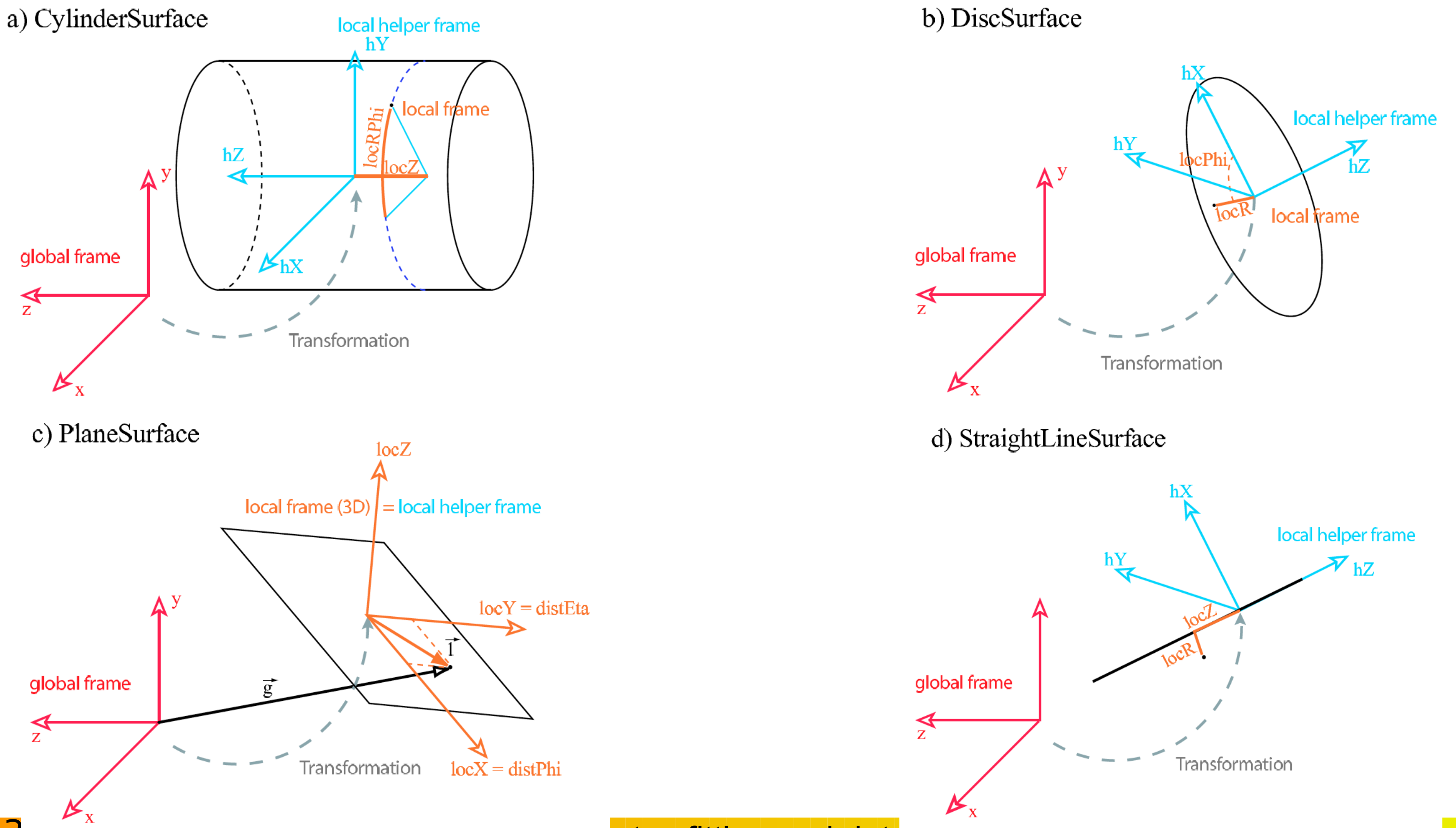
- Hit ↔ channel with signal
  - Detector specific
  - Channel ID & pulse height
- Cluster → group of channels
  - From 1 channel to many
  - 3D information:
    - Global or local coordinates
  - Position: (x, y, z)
  - Error: (δx, δy, δz) in a covariance matrix form
- Cluster position may depend:
  - Hit data: binary or analog
  - Center of gravity
  - Lorentz angle corrections
    - Charge carriers drift
  - Track incident angle
  - MCS corrections

- Cluster charge (signal)
  - Computed from hits
  - Correct for:
    - gain & noise
    - Track path within tracking volume
  - Allow to compute dE/dx
    - Analog (pulse height) data



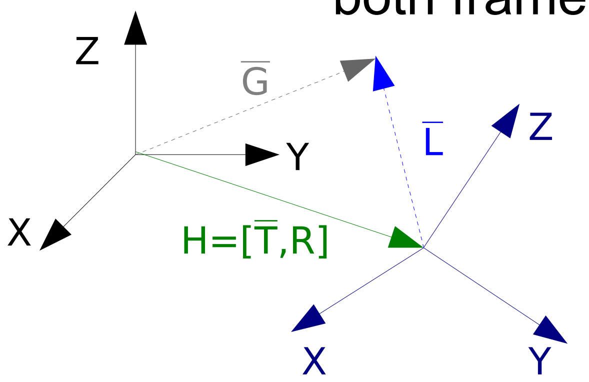
Global and local coordinates

- Track parameters are usually given in the experiment (global) reference frame
- Measurements (hits or clusters) are usually given in the sensor (local) reference frame
  - In local frame, the covariance matrix usually has a diagonal form
  - coordinates depend on detector geometry: cylinder, disk, plane, wire....



Global and local coordinates

- Reference frame transforms are heavily used in track fitting
  - Points (detector measurements or track extrapolations) are needed/computed in the global and local frame
  - Of course, the same happens with their errors
  - Track residuals are usually given in the local (sensor) frame
- The change of global-to-local reference frame includes:
  - A translation (of the origin of the local frame w.r.t. the global frame)
  - A rotation (orientation of the axis of the local frame w.r.t. the global frame)
  - Still the possibility to use cartesian, cylindric, spherical, ... coordinates in both frames



$$H=[\vec{T},R] \qquad \vec{G}=H\vec{L} = \vec{T}+R\vec{L} \qquad \vec{L}=H^{-1}\vec{G}$$

$$\vec{T}=\begin{pmatrix}T_x\\T_y\\T_z\end{pmatrix} \qquad R(\alpha,\beta,\gamma)=R_x(\alpha)\cdot R_y(\beta)\cdot R_z(\gamma)$$

$$R_x(\alpha)=\begin{pmatrix}1 & 0 & 0\\ 0 & \cos\alpha & -\sin\alpha\\ 0 & \sin\alpha & \cos\alpha\end{pmatrix} \qquad R_y(\beta)=\begin{pmatrix}-\cos\beta & 0 & \sin\beta\\ 0 & 1 & 0\\ \sin\beta & 0 & \cos\beta\end{pmatrix} \qquad R_z(\gamma)=\begin{pmatrix}\cos\gamma & -\sin\gamma & 0\\ \sin\gamma & \cos\gamma & 0\\ 0 & 0 & 1\end{pmatrix}$$

Global and local coordinates

- R, R_x, R_y & R_z are change of base matrices
  - Unitary matrices and $R^T = R^{-1}$
  - The norm of the vectors is kept
- The translation T, is just a 3D vector
- Therefore the covariance matrix of the measurements (local) can be expressed in the global frame as:

$$V_L = \begin{pmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{pmatrix}$$

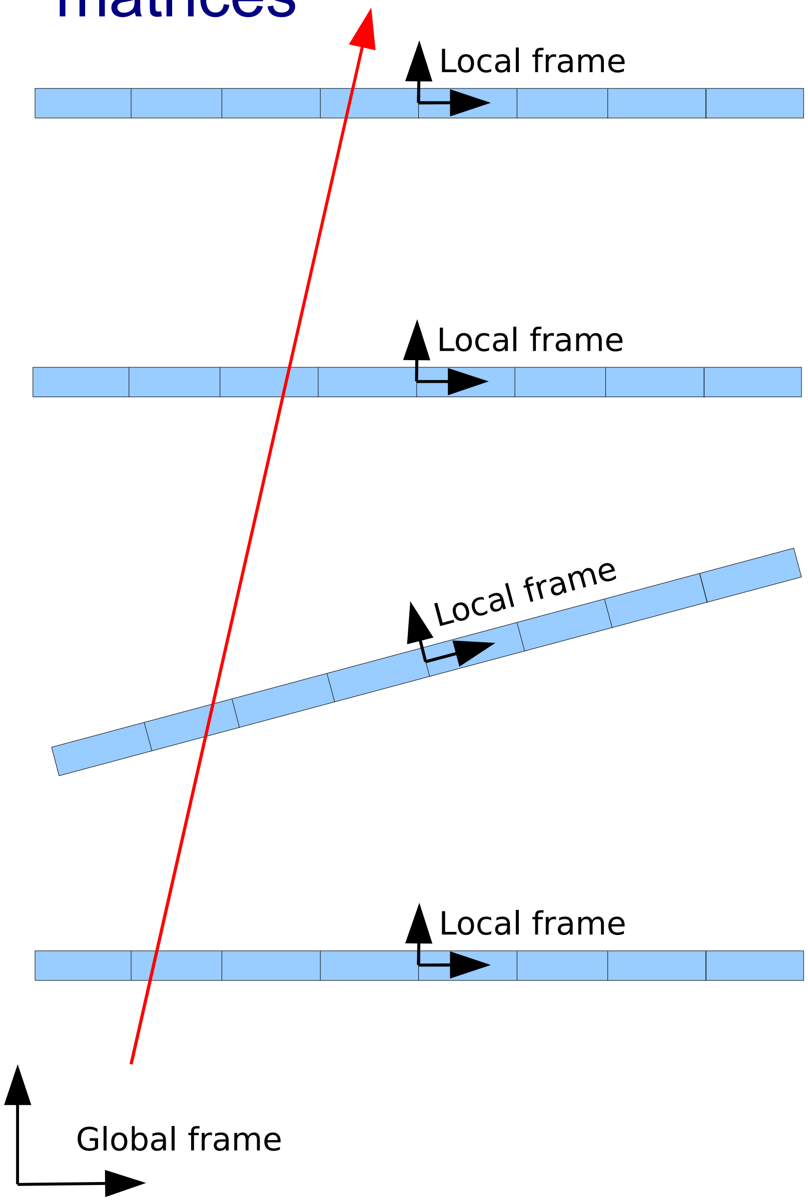
$$V_G = R^T V_L R = R^T \begin{pmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{pmatrix} R$$

- Similar expressions can be used to express the extrapolated point of a track & its error (global coordinates) in local coordinates of a detector element

$$V_L = R V_G R^T$$

Global and local coordinates

- The choice of local coordinates allows to handle diagonal covariant matrices



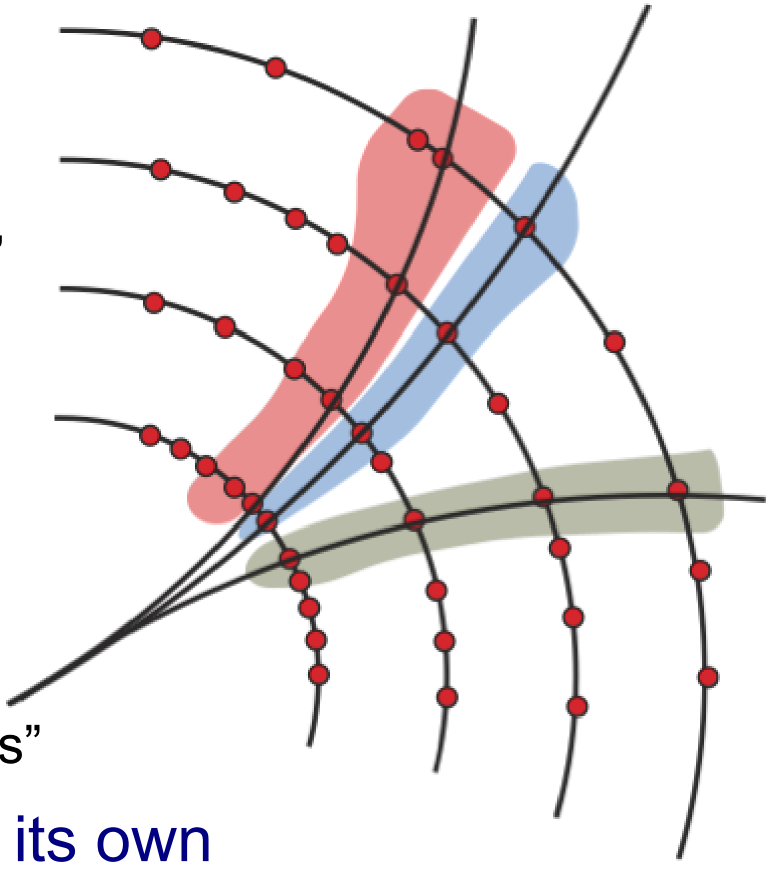
$$V_L = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix} \qquad V_G = R^T V_L R = I\, V_L\, I = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}$$

$$V_L = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}$$

$$V_G = R^T V_L R = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix} \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix}$$

Pattern recognition

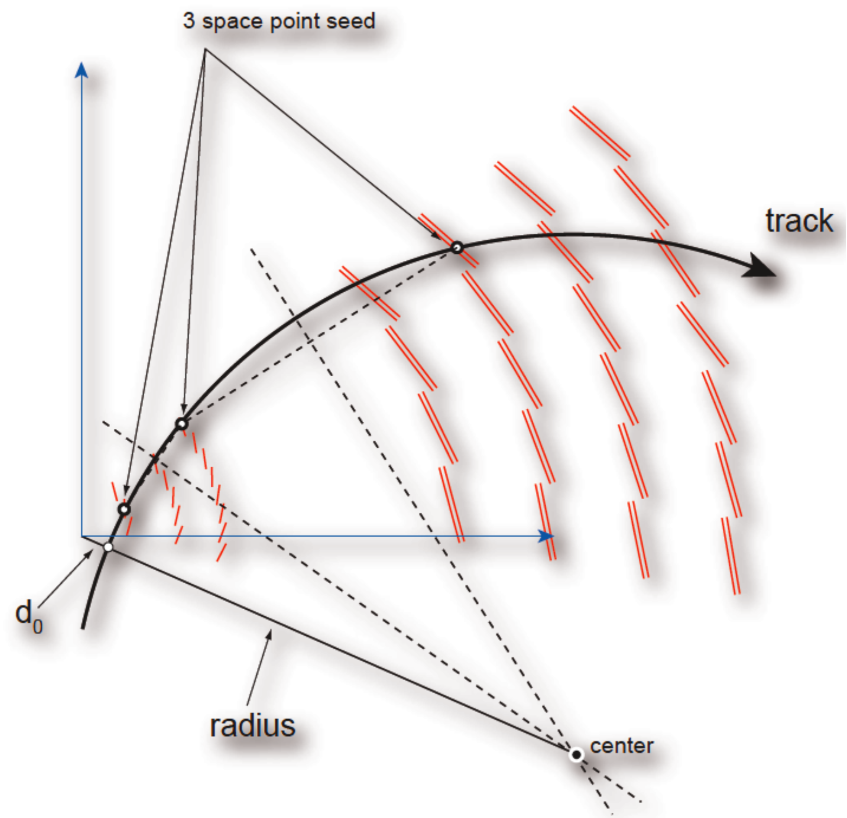
- The main goal of the pattern recognition is to associate hits to tracks
  - Efficient: all hits
  - Robust: no noise and no hits from other tracks
- Pattern recognition is a field of applied mathematics
  - It makes use of statistics, cluster analysis, combinatorial optimization, etc
  - The choice of the algorithm depends heavily in the type of measurements
    - 2D vs 3D points
  - And in the track model
    - Detector shape and B field
  - Hough space transform, template matching, minimum spanning tree, local pattern recognition

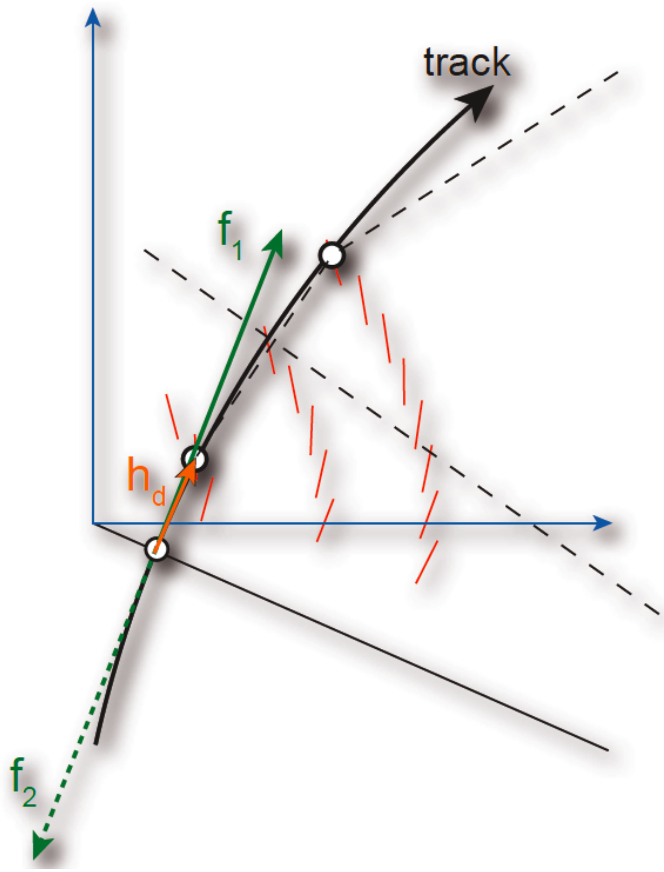


- Hit-to-track association
  - Defined by pattern recognition
  - Later altered by tracking
    - Removing bad hits & outliers
  - Noisy channels tend to be the "party spoilers"
- In summary: pattern recognition is an art on its own

Pattern recognition:

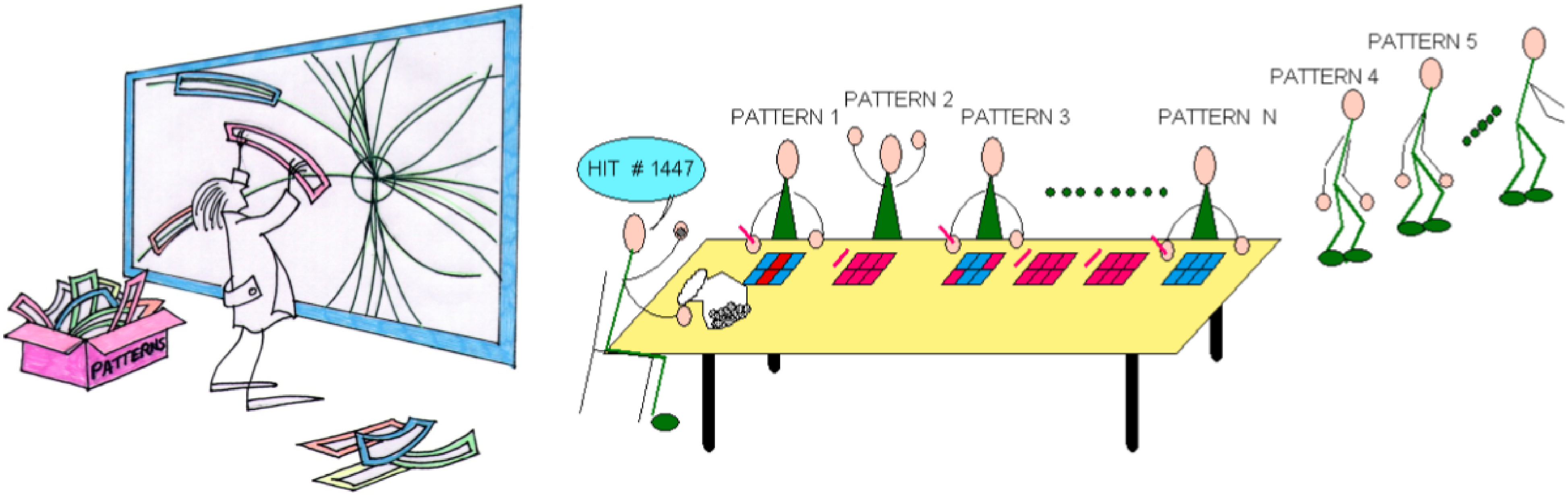
- 3 points seed:
  - Adding other measurements: (inside-out or outside-in) may use 3 consecutive measurements (compute a circle) and extrapolate the track (outwards or inwards) attaching near-by measurements



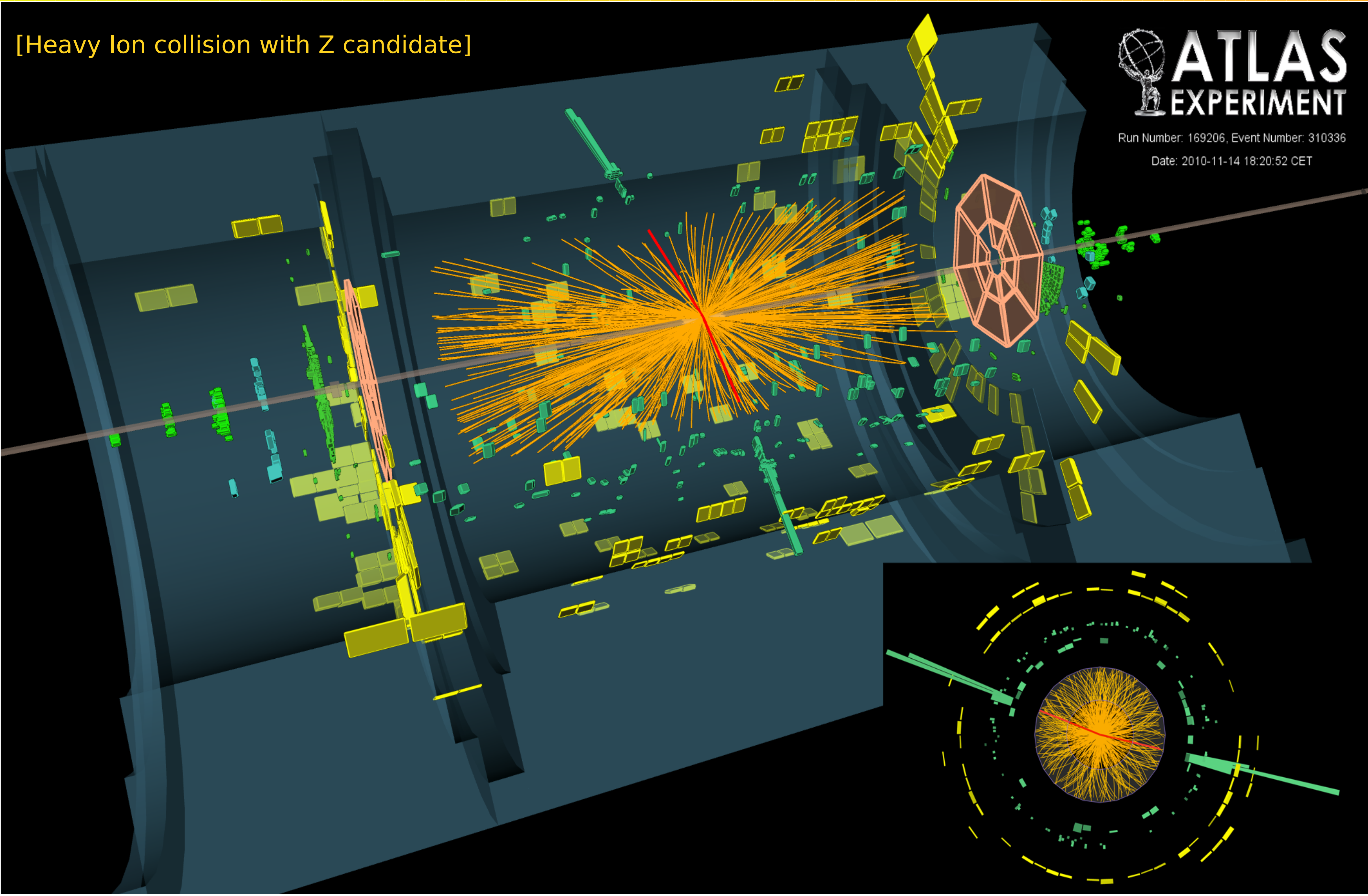


Pattern recognition

- It is possible to perform an *online* pattern recognition for a fast online tracking
  - Why fast tracking ?
    - Online one has a limited time to decide if the event is stored or discarded
    - A finite set of track topologies is used
  - Possibility to implement a “fast tracking” based trigger
    - Trigger on secondary vertices → online B-tagging



Example of event with many tracks



Track fitting with X^2 minimization

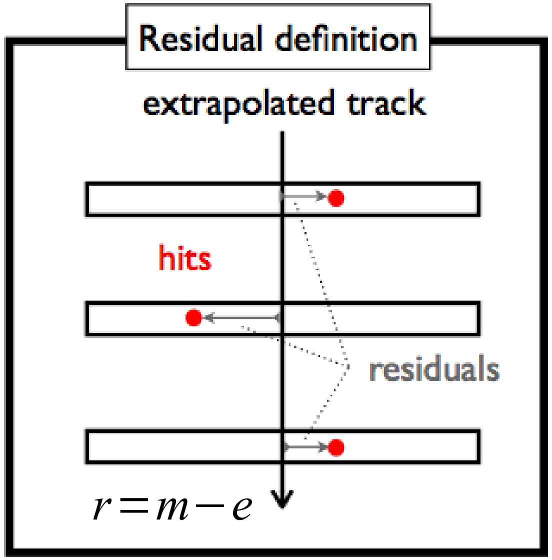
- The Least Squares Method best meets the requirements of track fitting:
  - Rather simple formulation (usual X^2 definition) and statistical properties
  - Easy implementation of measurements (hits) and their errors
  - Quite fast numerically (even for large number of degrees of freedom)
  - Provides solution to track parameters and their errors

- An important ingredient is the track model which has to:
  - Be well approximated by a linear model in the neighborhood of the measurements (→ second and higher order derivatives negligible)

- The track model requires that (generic to all track fitters):
  - The equation of motion can be solved with sufficient precision
    - When B is in use, this implies that B is well known
  - The material traversed by the particles is well known
    - Allow the accurate evaluation of energy loss and multiple scattering
  - No wrong measurements (noisy hits) have been associated to the track during the pattern recognition

Track fitting with X^2 minimization

- Use well known technique of residual minimization for track parameters determination via X^2 function
  - Usual X^2 definition
    - Residuals (r) and their errors (σ)
  - X^2 minimization w.r.t. track parameters (τ)



$$X^2 = \sum_{i=1}^{N_R} \left(\frac{r_i}{\sigma(r_i)} \right)^2 \qquad \frac{d X^2}{d\tau} = 0 \quad \rightarrow \quad \sum_{i=1}^{N_R} \frac{r_i}{\sigma(r_i)^2} \frac{dr_i}{d\tau} = 0$$

- Rewrite the X^2 using the matrix algebra:

$$r = \begin{pmatrix} r_1 \\ \vdots \\ r_{N_R} \end{pmatrix} \qquad V = \begin{pmatrix} \sigma^2(r_1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma^2(r_{N_R}) \end{pmatrix} \quad \rightarrow \quad X^2 = r^T V^{-1} r$$

1) V may contain correlations terms as well. Therefore V is not necessarily diagonal
2) The residuals errors are taken as the intrinsic errors of the detector elements. Each hit may come from a different tracking device, therefore each one has its own error

  - Apply the X^2 minimization w.r.t. track parameters ( )

$$\frac{d X^2}{d\tau} = 0 \quad \rightarrow \quad 2 \left(\frac{dr}{d\tau} \right)^T V^{-1} r = 0 \qquad \tau = \begin{pmatrix} \tau_1 \\ \vdots \\ \tau_{N_T} \end{pmatrix} = \begin{pmatrix} d_0 \\ z_0 \\ \phi_0 \\ \theta_0 \\ p \end{pmatrix} \qquad \frac{dr}{d\tau} = \begin{pmatrix} dr_1/d\tau_1 & \dots & dr_1/d\tau_{N_T} \\ \vdots & \ddots & \vdots \\ dr_N/d\tau_1 & \dots & dr_N/d\tau_{N_T} \end{pmatrix}$$

Track fitting with X^2 minimization

- Taylor's expansion up to first order derivatives: $r = r(\tau_0) + \left|\frac{dr}{d\tau}\right|_{\tau_0} \delta\tau$
  - Computed at initial track parameter (π_0) estimation
  - Neglect second and higher order derivatives: $\frac{d^2 r}{d\tau_i d\tau_j} = 0$
- The minimum condition equation becomes:

$$\frac{dX^2}{d\tau} = 0 \quad \rightarrow \quad \left(\frac{dr}{d\tau}\right)^T V^{-1} r = 0 \quad \rightarrow \quad \left[\left(\frac{dr}{d\tau}\right)^T V^{-1} \left(\frac{dr}{d\tau}\right)\right] \delta\tau + \left[\left(\frac{dr}{d\tau}\right)^T V^{-1} r\right] = 0$$

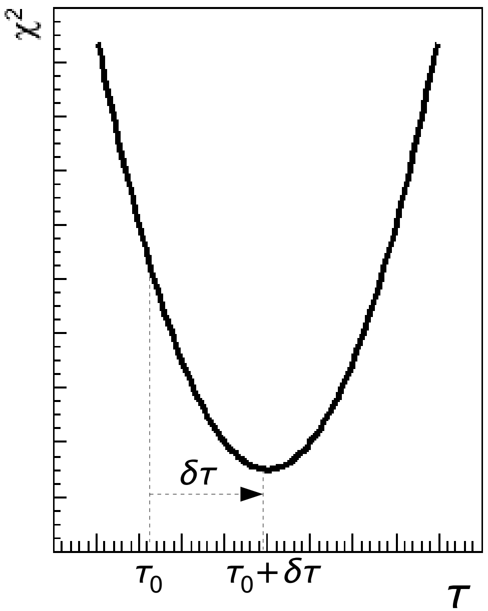
- Solving the above matrix equation requires to invert a N_T x N_T matrix

$$\delta\tau = -\left[\left(\frac{dr}{d\tau}\right)^T V^{-1} \left(\frac{dr}{d\tau}\right)\right]^{-1} \left[\left(\frac{dr}{d\tau}\right)^T V^{-1} r\right] \quad \rightarrow \quad \tau = \tau_0 + \delta\tau$$

- Pros & cons:
  - pros:
    - The covariance matrix of the track parameters is just the inverse of the track derivatives matrix. So track parameters errors are computed for free :)
    - If the problem is linear then the solution is exact
  - Cons:
    - The derivatives of the residuals wrt track parameters may be hard to compute
    - If the problem is not linear then one needs to iterate

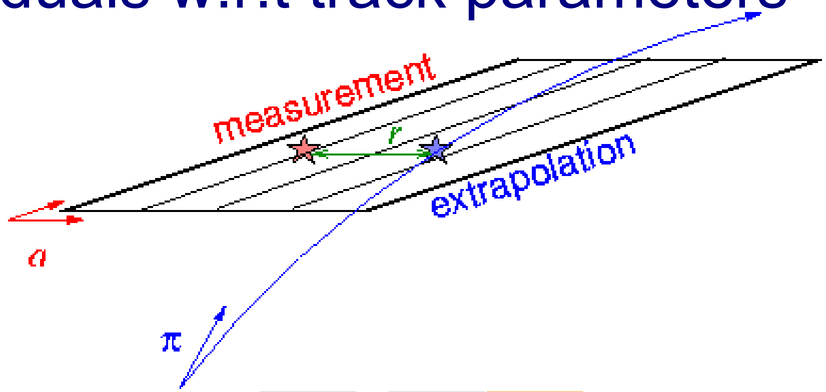
Track fitting with χ^2 minimization

- The calculation of the derivatives of residuals w.r.t track parameters

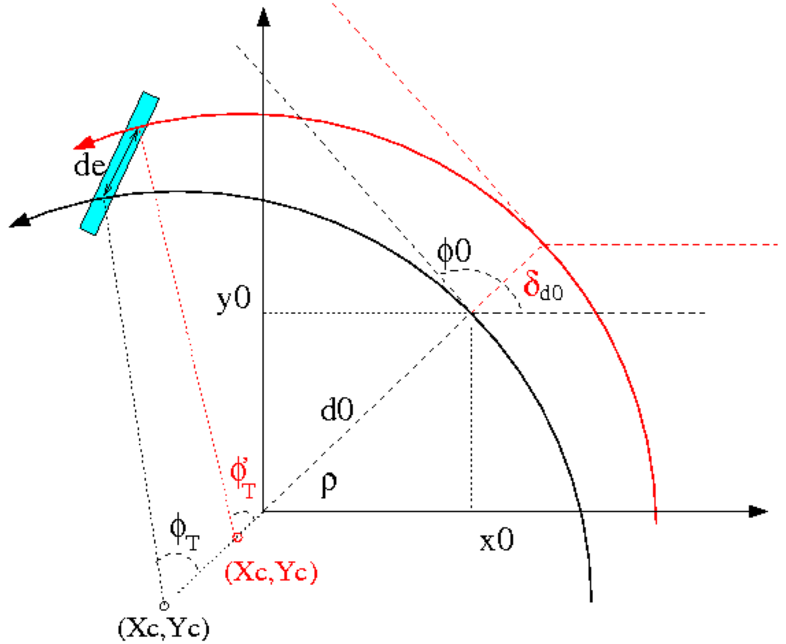


$$r = m - e \quad \rightarrow \quad \frac{dr}{d\tau} = -\frac{de}{d\tau}$$

$$m = \begin{pmatrix} m_x \\ m_y \\ m_z \end{pmatrix} \qquad e = \begin{pmatrix} e_x \\ e_y \\ e_z \end{pmatrix} = \begin{pmatrix} x(\phi_T) \\ y(\phi_T) \\ z(\phi_T) \end{pmatrix}$$



$$dx = \frac{\partial x}{\partial \tau_i} d\tau_i + \frac{\partial x}{\partial \phi_T} d\phi_T \quad \rightarrow \quad \frac{dx}{d\tau_i} = \frac{\partial x}{\partial \tau_i} + \frac{\partial x}{\partial \phi_T} \frac{d\phi_T}{d\tau_i}$$



- Intersection of the track with the detector:
  - Changes with changing track parameters
- Analytic calculations make assumptions:
  - On track model and detector conditions
    - e.g.uniform B & material description
  - Fast and reliable
- Numerical calculations
  - Time consuming, reliable & heavy use of the track extrapolation package

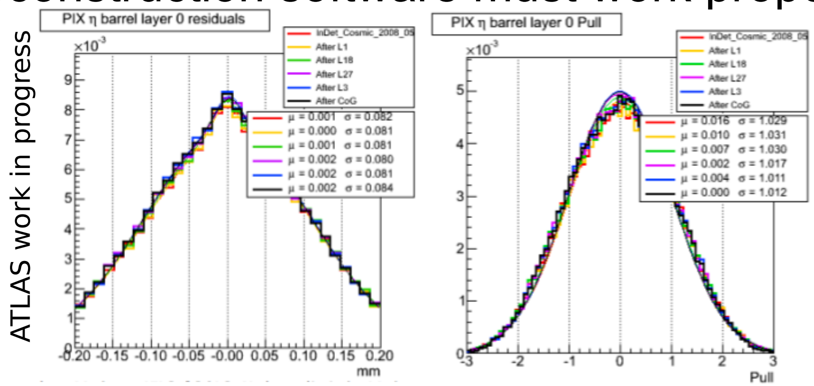
Track fitting with X² minimization

- Track fit with constrained track parameters
  - Beam spot, secondary vertices, invariant masses, …

$$R=\begin{pmatrix} d_0-\hat{d}_0 \\ \vdots \\ p-\hat{p} \end{pmatrix} \qquad W=\begin{pmatrix} \sigma^2(d_0) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma^2(p) \end{pmatrix} \quad \rightarrow \quad X^2=r^T V^{-1} r+R^T W^{-1} R$$

$$\delta\tau=-\left[\left(\frac{dr}{d\tau}\right)^T V^{-1}\left(\frac{dr}{d\tau}\right)+\left(\frac{dR}{d\tau}\right)^T W^{-1}\left(\frac{dR}{d\tau}\right)\right]^{-1}\left[\left(\frac{dr}{d\tau}\right)^T V^{-1} r+\left(\frac{dR}{d\tau}\right)^T W^{-1} R\right] \quad \rightarrow \quad \tau=\tau_0+\delta\tau$$

- Goodness of the fit: evaluate the pull quantities
  - When fit is correct: pulls follow a Normal distribution (μ=0,σ=1)
  - Three conditions must be fulfilled
    1) The track model must be correct
    2) The covariance matrix of the measurement errors must be correct
    3) The reconstruction software must work properly



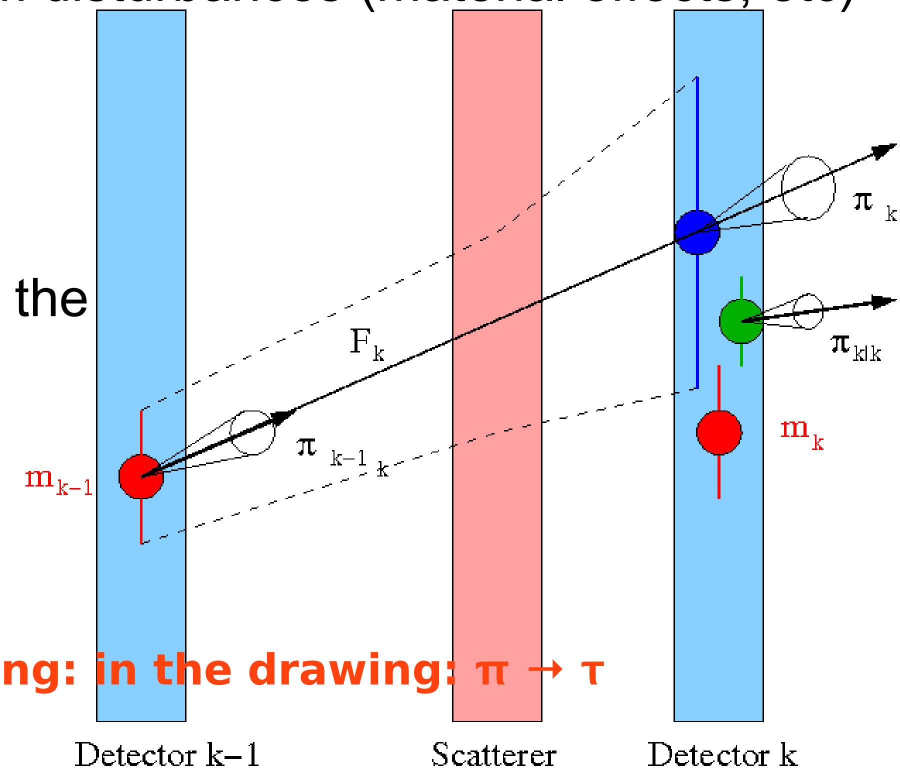
Track fitting with Kalman filter

- The Kalman filter was developed by R.E. Kalman during the 1950's
  - To solve differential matrix equations without matrix inversions
  - It is a method of estimating the states of dynamic systems
    - Applied by the NASA in the rocket trajectory control for the Apollo program
    - Military applications: compute plane trajectory by radar tracking
- Assumption:
  - The trajectory of a particle between two adjacent surfaces is described by a deterministic function plus random disturbances (material effects, etc)
  - The system equation: propagates the estate in one surface to the next

$$\tau_k = F_k(\tau_{k-1}) + P_k \boldsymbol{\delta}_k \qquad \langle \boldsymbol{\delta}_k \rangle = 0 \qquad Cov(\boldsymbol{\delta}_k) = Q_k$$

  - The measurement equation: mapping the track in the surface and considers some measurement error

$$\boldsymbol{m}_k = H_k(\boldsymbol{\tau}_k) + \varepsilon_k \qquad \langle \boldsymbol{\varepsilon}_k \rangle = 0 \qquad Cov(\boldsymbol{\varepsilon}_k) = V_k$$



Warning: in the drawing: π → τ

Track fitting with Kalman filter

- The aim is to estimate the track parameters from the observations
  - From j observations and a k^{th} measurement: obtain a new k estimate

$$\left\{\{\boldsymbol{m}_{\mathbf{1}},\ldots,m_j\},\quad \boldsymbol{\tau}_{\boldsymbol{j}}\right\} + \boldsymbol{m}_{\boldsymbol{k}} \rightarrow \boldsymbol{\tau}_{\boldsymbol{k}}$$

  - **Prediction** $\tau_{k|k-1}=F_k(\tau_{k-1})+P_k\boldsymbol{\delta}_{\boldsymbol{k}}$
    - and its covariance matrix (error): $C_{k|k-1}=F_k C_{k-1|k-1}F_k^T+P_k Q_k P_k^T$
  - **Filtering**, based on $\tau_{k/k\text{-}1}$ and m_k:
    - It consists in minimizing the following:

$$L(\boldsymbol{\tau}_{\boldsymbol{k}})=(\boldsymbol{m}_{\boldsymbol{k}}-H_k\boldsymbol{\tau}_{\boldsymbol{k}})^T V_k^{-1}(\boldsymbol{m}_{\boldsymbol{k}}-H_k\boldsymbol{\tau}_{\boldsymbol{k}})+(\boldsymbol{\tau}_{\boldsymbol{k|k-1}}-\boldsymbol{\tau}_{\boldsymbol{k}})^T C_{k|k-1}(\boldsymbol{\tau}_{\boldsymbol{k|k-1}}-\boldsymbol{\tau}_{\boldsymbol{k}})$$

    - The solution should be well known by now:

$$\boldsymbol{\tau}_{\boldsymbol{k|k}}=\boldsymbol{\tau}_{\boldsymbol{k|k-1}}+\left[(H_k^T V^{-1}H_k)+C_{k|k-1}\right]^{-1}\left[H_k^T V^{-1}(\boldsymbol{m}_{\boldsymbol{k}}-H_k\boldsymbol{\tau}_{\boldsymbol{k}})\right]$$

    - And its covariance matrix (error):

$$C_{k|k}=\left[(H_k^T V^{-1}H_k)+C_{k|k-1}\right]^{-1}$$

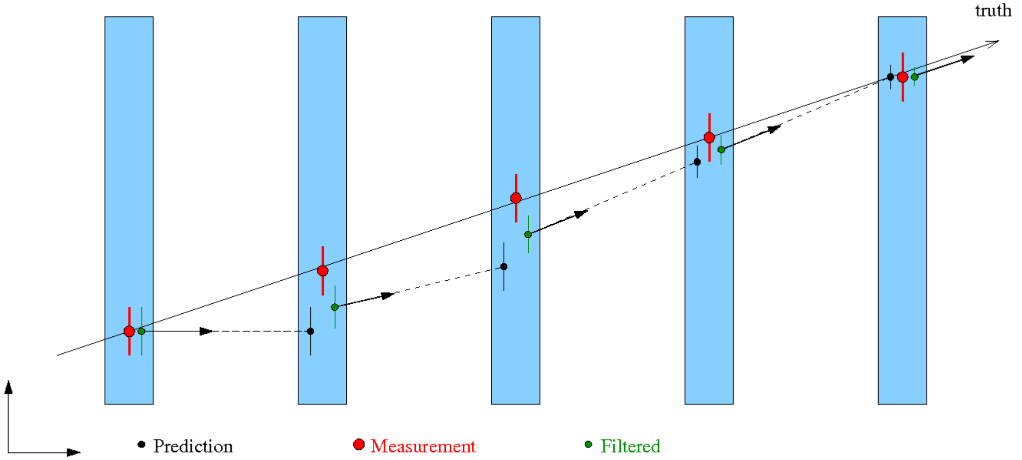
  - The residual is thus: $\boldsymbol{r}_{\boldsymbol{k|k}}=\boldsymbol{m}_{\boldsymbol{k}}-H_k\boldsymbol{\tau}_{\boldsymbol{k|k}}$
    - Which allows to compute a χ^2 in order to test the goodness of the fit

$$\chi^2_{k|k}=\boldsymbol{r}_k^T V_k^{-1}\boldsymbol{r}_k \qquad \chi^2=\sum_k \chi^2_k$$

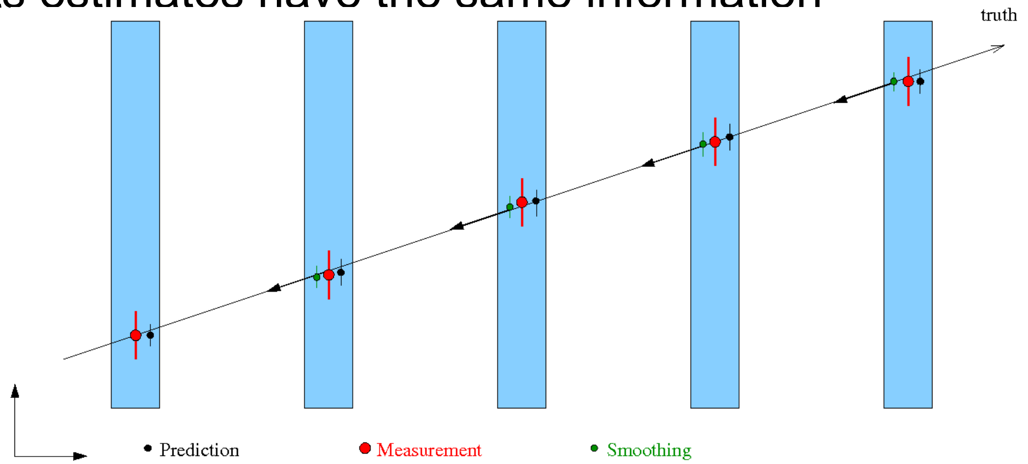
      that needs some smoothing.

Track fitting with Kalman filter

- Estimate of the track parameters and state at the detector surfaces
  - Filtering from estimate *k*-1 to *k*
    - Outer points estimates have more information than inner points

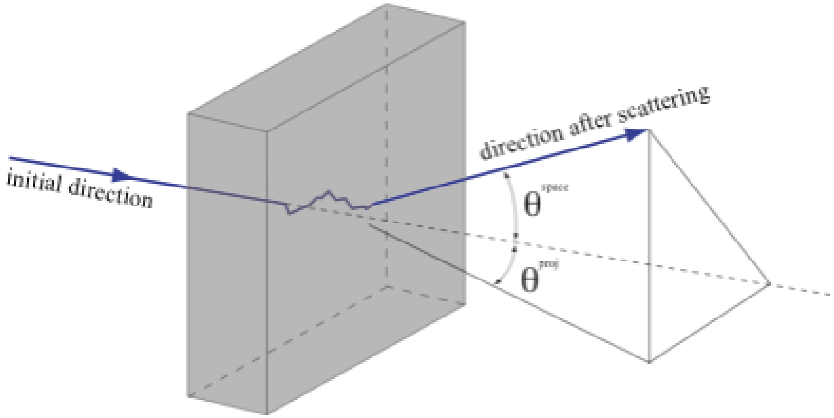


  - Smoothing: from estimate k to k-1 (sort of backward filter)
    - All points estimates have the same information



Treatment of the MCS

- The Multiple Coulomb Scattering must be included in the track fitting
  - Particle traversing material undergoes successive deflections
    - In main tracking algorithms the assumption is that the MCS angles follow a Gaussian distribution. It is know that the tails are larger than just Gaussian tails



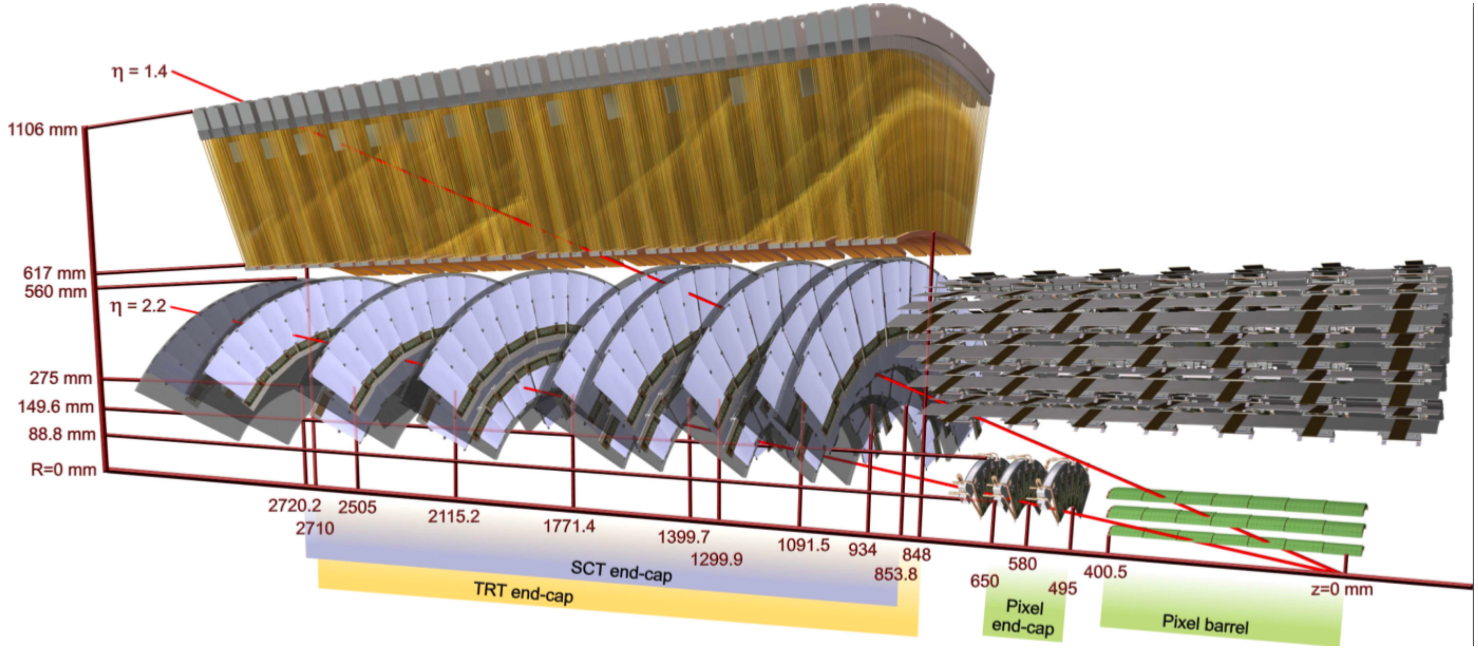
$$\theta_{MCS}=\theta_{rms}=\frac{13.6\,MeV}{\beta c\, p}\, z\sqrt{\frac{x}{X_0}}\left[1+0.038\ln\left(\frac{x}{X_0}\right)\right]$$

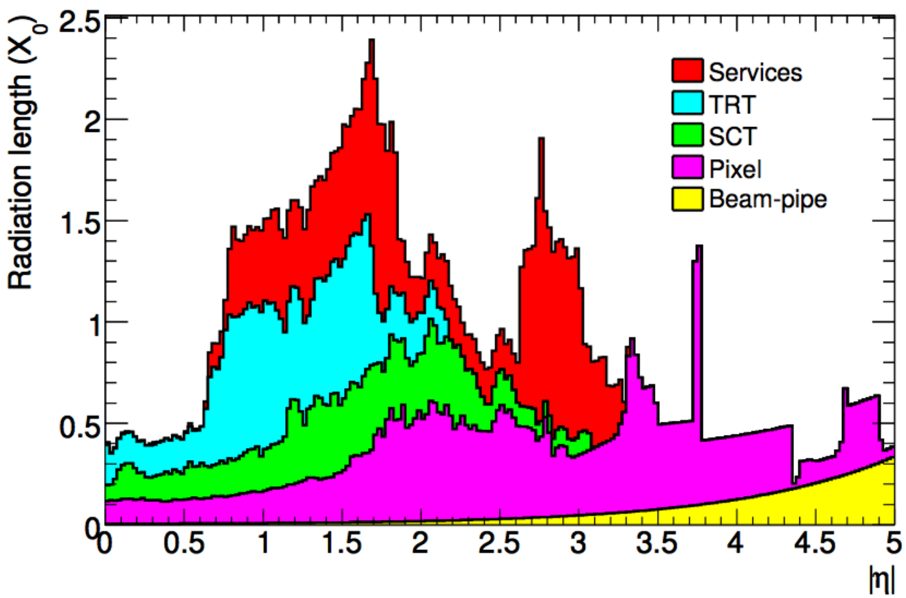
- Energy loss: charged particles loss energy due to it Coulomb interaction with charged particles in matter (detector)
  - Ionization energy that may be used to detect the particle and identify it
  - Bethe-Block formula

$$\frac{dE}{dx}=-2\pi N_a r_e^2 m_e c^2 \rho \frac{Z}{A}\frac{z^2}{\beta^2}\left[\ln\left(\frac{2 m_e \gamma^2 v^2 W_{max}}{I^2}\right)-2\beta^2\right]$$

Treatment of the MCS

- The amount of material affects the track reconstruction

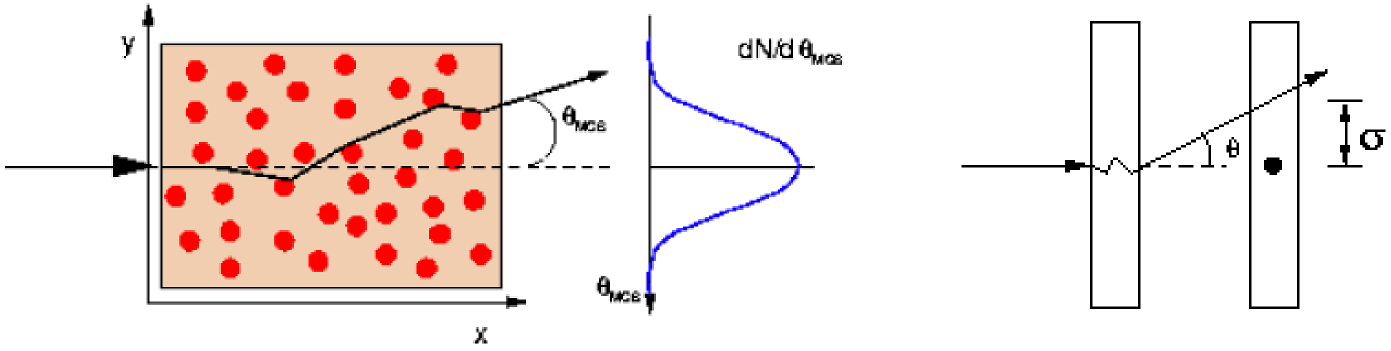


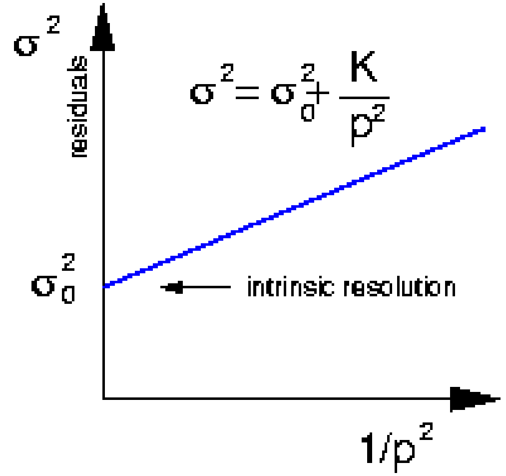


Material in the ATLAS Inner Detector expressed in units of radiation length and given as a function of the pseudorapidity

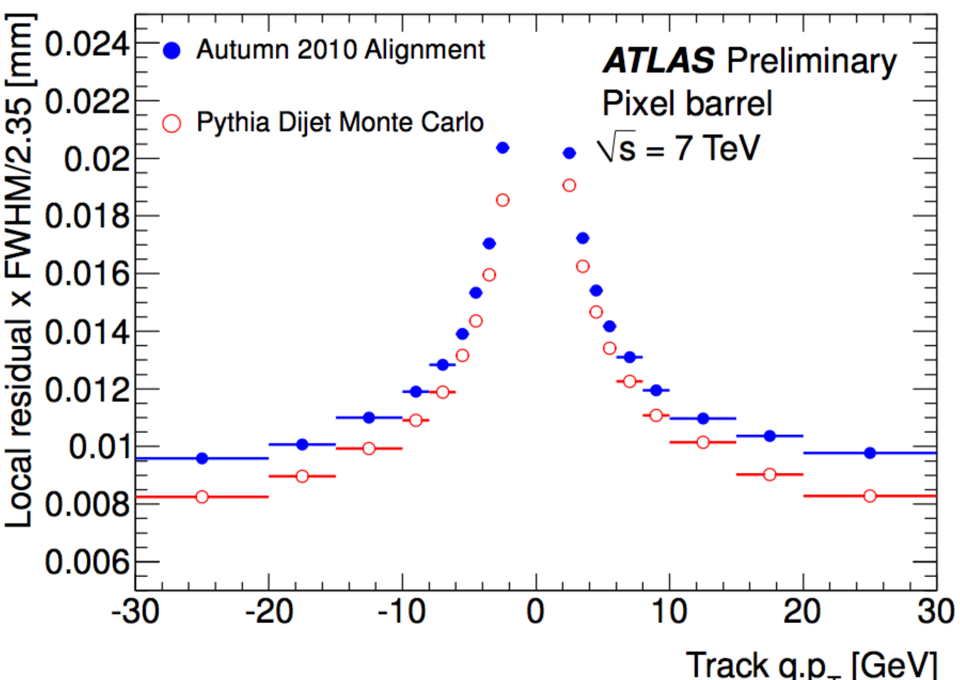
Treatment of the MCS

- The MCS deflects the tracks and it affects the detector residuals
  - Residuals become momentum dependent



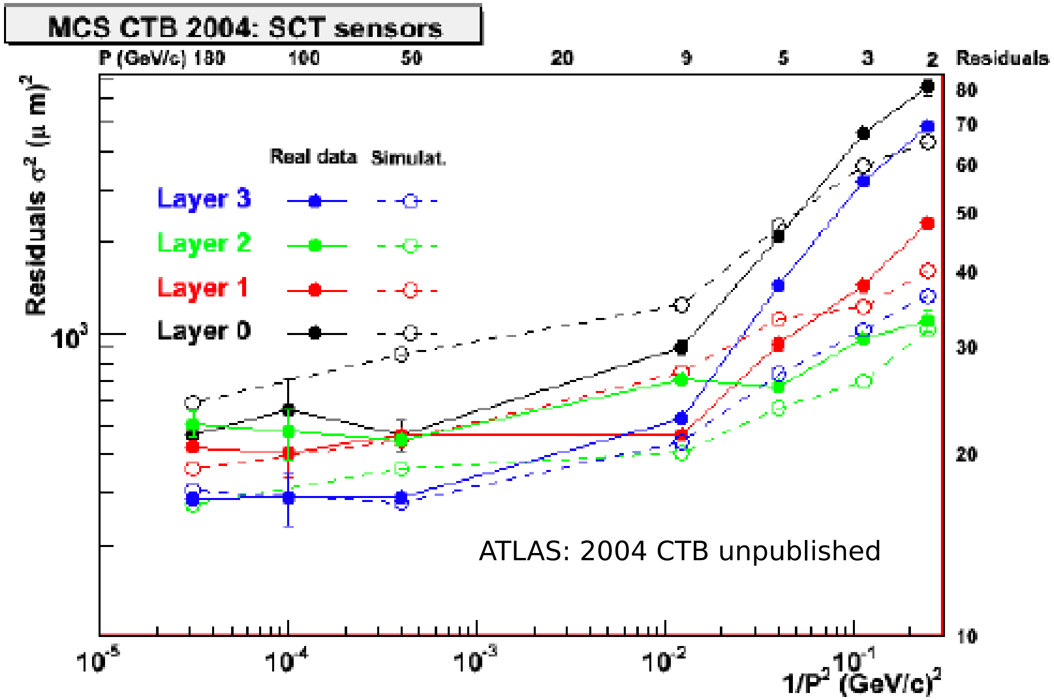


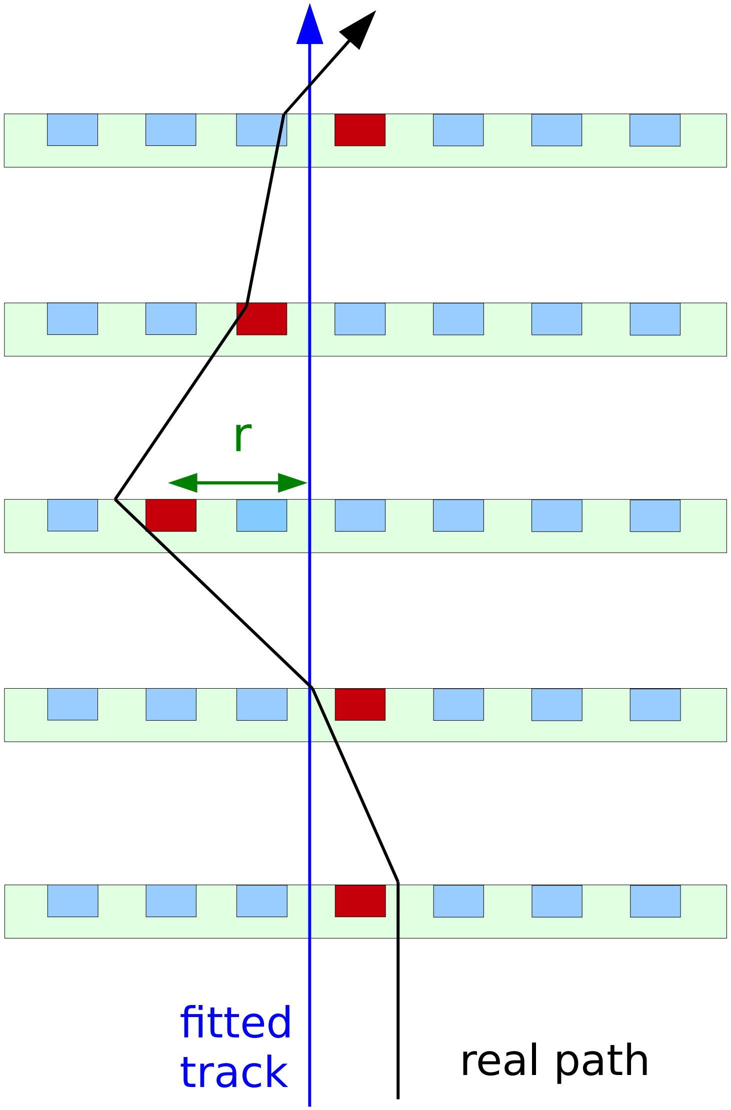
In principle, it is possible to remove the MCS contribution to the residuals.
This requires an almost perfect description of the material budget.
Besides, the MCS is a statistical process.



Treatment of the MCS

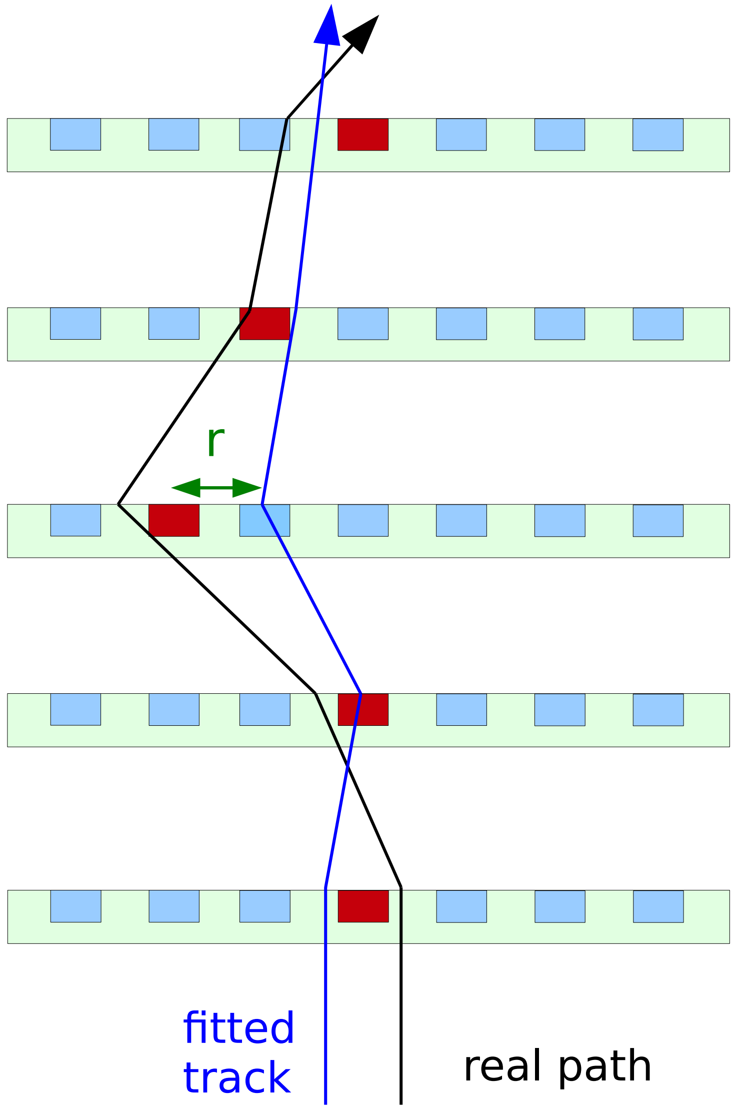
- Option 1: keep track model
  - Use same track parameters
    - Straight line, helix, ...
  - Larger residuals
    - Residuals become momentum dependent
  - One may include MCS through correlations in the covariance matrix





Treatment of the MCS

- Option 2: include MCS terms in tracking model
  - Precise knowledge of the material budget
    - Weight and components of the detector and its services (cooling, support,...)
    - Their precise location
- Use extra track parameters
  - Allow for scattering angles
    - 2 angles per surface
    - Track kinks
  - Energy loss
  - Momentum dependent
- Track fitting may need extra iterations
  - Initial momentum assumption
  - Refit with 1st fitted momentun



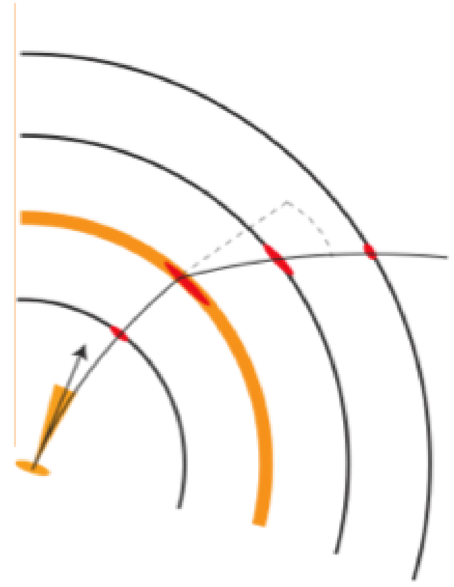
Treatment of the MCS

- Practical implementation in the algorithm
  - As non diagonal correlation matrix

$$V = V_{hit} + V_{MCS} = \begin{pmatrix} \sigma^2(r_1) & \dots & corr(r_1, r_{N_R}) \\ \vdots & \ddots & \vdots \\ corr(r_1, r_{N_R}) & \dots & \sigma^2(r_{N_R}) \end{pmatrix}$$

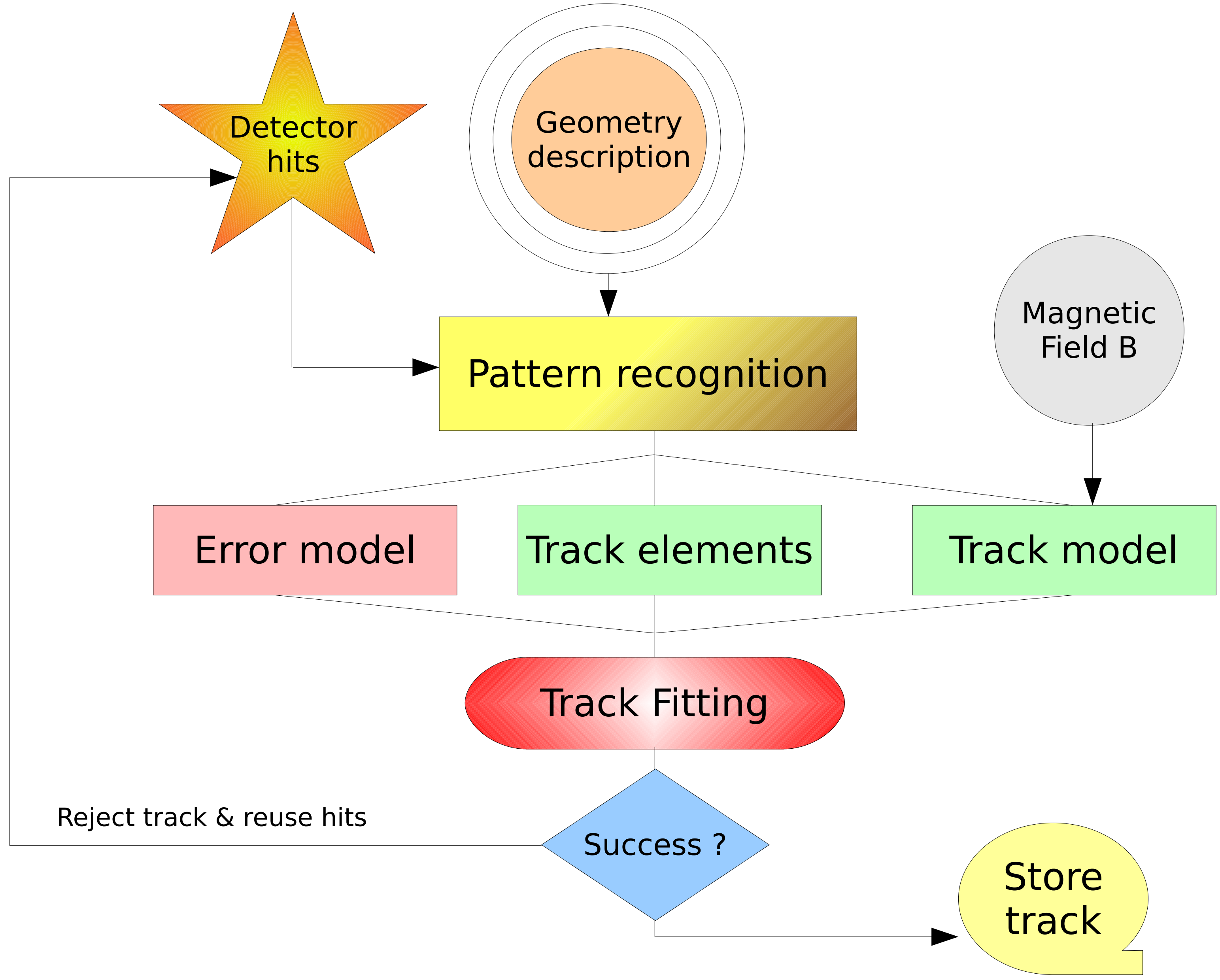
  - As extra track parameters that are fitted

$$r_\theta = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_{Nscat} \end{pmatrix} \qquad r_{\Delta_E} = \begin{pmatrix} \Delta_{E1} \\ \vdots \\ \Delta_{ENscat} \end{pmatrix} \qquad \tau = \begin{pmatrix} \pi_i \\ \theta_j \\ \Delta_{Ek} \end{pmatrix} \qquad X^2 = r^T V^{-1} r + r_\theta^T V_{MCS}^{-1} r_\theta + r_{\Delta_E}^T V_{\Delta_E}^{-1} r_{\Delta_E}$$



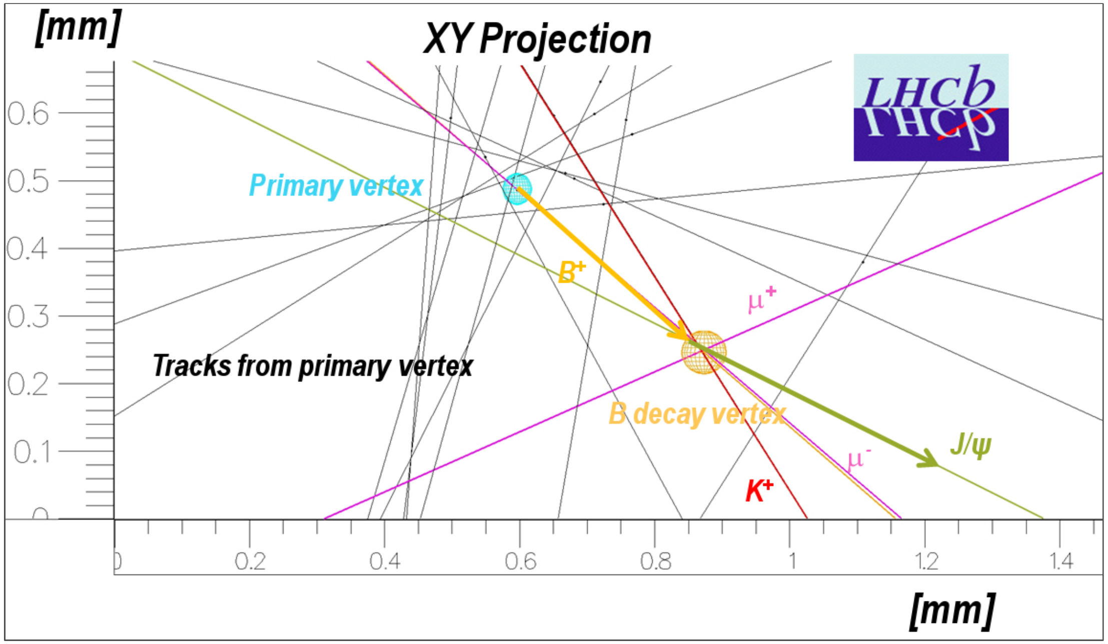
Track fitting summary

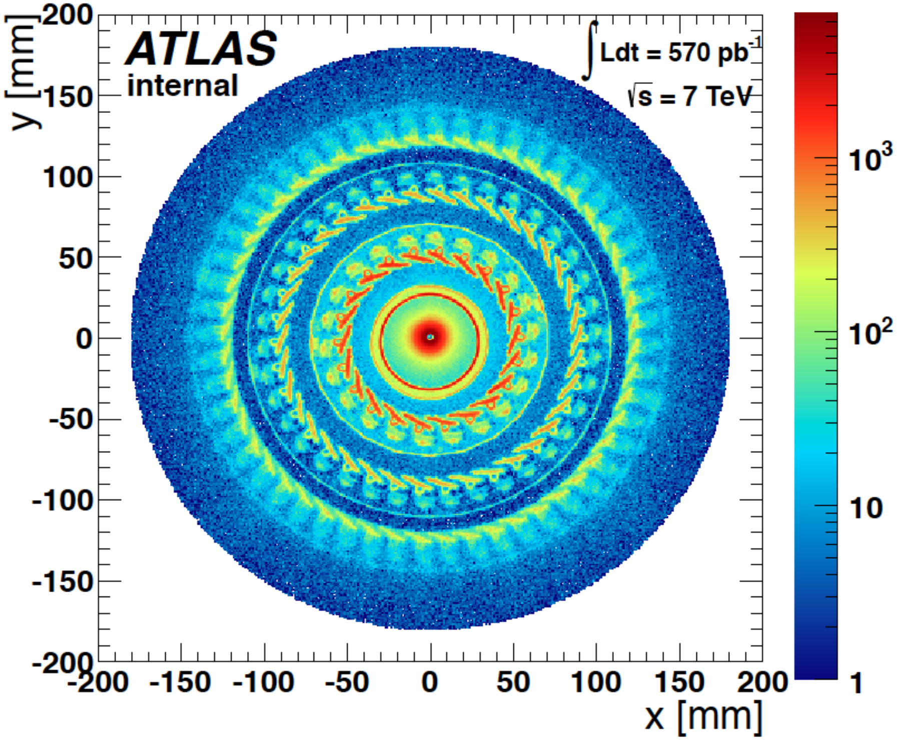
- From detector hits to particle trajectories



Vertex fit: basic ideas and concepts

- From the physics point of view, vertexing helps to
  - identify decaying particles from their products
  - Study the properties of those particles (mass, lifetime, couplings, …)
- From the instrumentation point of view, vertexing serves to characterize the detector
  - Positioning resolution
  - Find where and how much material is in the inner detector layers





Vertex fit: basic ideas and concepts

- The basic geometric idea of vertex is the point where many (≥ 2) particles where originated
- Mathematically: one can apply a minimization technique and set the vertex as the point that minimizes simultaneously the distance of the bunch of particles under question.
- Refinements:
  - Preselection of the particles to combine
    - Similar to pattern recognition in tracking
    - Saves great amount of CPU
    - Try and error may work but combinatorics will ingest precious CPU time
  - Constraints:
    - Easy to apply under the X2 formalism
    - But be ware and think twice: they may bias the result
- Vertex fitting methods:
  - Billoir [P. Billoir and S. Quian, Fast vertex fitting with a local parametrization of tracks, Nucl. Instrum. Meth. A319 (1992) 139.]
  - Adaptive [R.Früwirth et al. CMS Note 2007/008]

Billoir vertex fitting method

- This is a X^2 minimization based method.
- Track parametrization is amended to include the vertex location
- The fitting accounts for the effects of the vertex (extra point) in the track parameters

$$\boldsymbol{v}=\begin{pmatrix} x_v \\ y_v \\ z_v \end{pmatrix} \qquad \tau=\begin{pmatrix} \theta \\ \phi \\ q/p \end{pmatrix} \quad \Rightarrow \quad \boldsymbol{t}=\begin{pmatrix} \boldsymbol{v} \\ \tau \end{pmatrix}$$

- The X^2 is built as:

$$X^2=\sum_{i=1}^{N} \Delta \boldsymbol{t}_i^T V^{-1} \Delta \boldsymbol{t}_i \qquad V=\begin{pmatrix} cov(v) & cov(v,\tau) \\ \sigma(\tau,v) & cov(\tau) \end{pmatrix}$$

[covariance matrix]

- Minimization: $\frac{d X^2}{d \boldsymbol{v}}=0 \quad \rightarrow \quad 2\left(\frac{d \Delta \boldsymbol{t}}{d \boldsymbol{v}}\right)^T V^{-1} \Delta \boldsymbol{t}=0$

$$\frac{d \Delta \boldsymbol{t}}{d \boldsymbol{v}}=\frac{\partial \Delta \boldsymbol{t}}{\partial \boldsymbol{v}} d \boldsymbol{v}+\frac{\partial \Delta \boldsymbol{t}}{\partial \boldsymbol{\tau}} \frac{d \boldsymbol{\tau}}{d \boldsymbol{v}}$$

- This has two nested fits:
  - track parameters as local parameters that vary when including the vertex point
  - The vertex point as global parameters that is common for all tracks

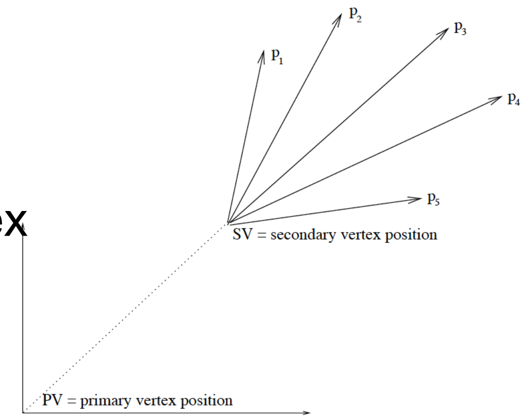
Billoir fitting method

- The solving provides the vertex position, the track parameters and their errors and correlations (via the weight matrix: V^{-1})
- It introduces correlations among tracks as now, all have a common point (vertex).
- Adding constraints:

$$\boldsymbol{b}=\begin{pmatrix} b_v \\ b_v \\ b_v \end{pmatrix}$$

  - Beam spot position $\boldsymbol{b}$. Add extra term to minimize: $(\boldsymbol{v}-\boldsymbol{b})^T V_{BS}^{-1}(\boldsymbol{v}-\boldsymbol{b})$
    - Useful for primary vertex only
  - Pointing constraint
    - Helpful for secondary vertices.
    - The sum of momenta of particles in the secondary vertex must be parallel to the vector joining the vertices

$$\sum_{i=1}^{N} \boldsymbol{p}_i \times (\boldsymbol{v}_p - \boldsymbol{v}_s) = 0$$



    - This can be added as extra X^2 term or with Lagrange multiplier
  - Other constraints as: mass constraint
    - This helps to reject combinatorial background

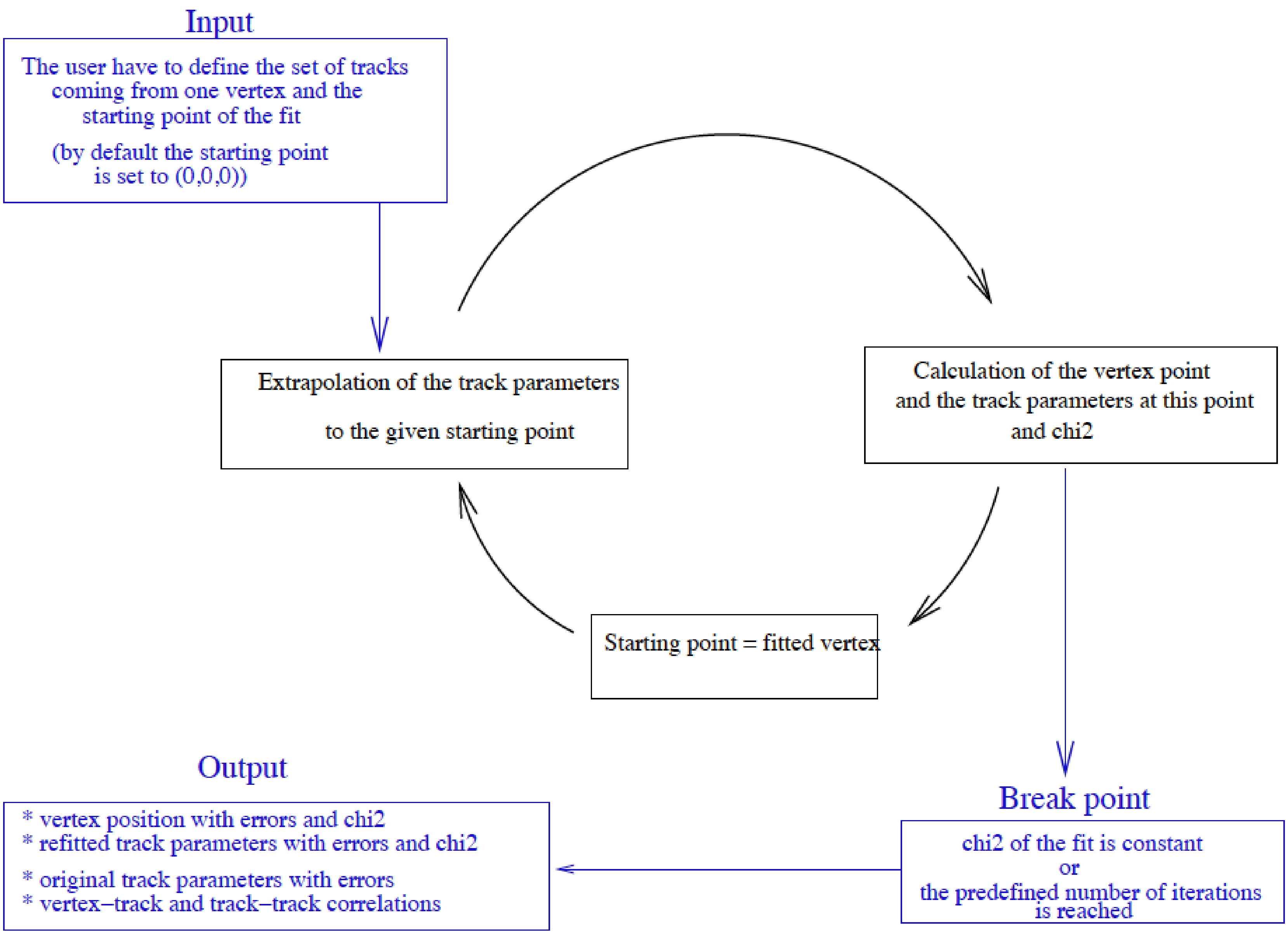
Adaptive vertex fitting method

- The X^2 technique work well with Gaussian errors
- However one has to deal with contaminated data:
  - mis-associated tracks (to vertex) and mis-measured tracks (that include wrong/noisy hits) leading to badly estimated track errors
- Adaptive vertex fitting method implements the Kalman-filter routine to the vertex finding
  - Tracks are given weights (according to their χ^2 probability or the probability that a track belongs to that vertex).

$$X^2 = \sum_{i=1}^{N} w_i(\chi^2) \Delta \boldsymbol{t}_i^T V^{-1} \Delta \boldsymbol{t}_i \qquad \frac{d X^2}{d \boldsymbol{v}} = 0 \quad \rightarrow \quad 2 \sum w(\chi^2) \left(\frac{d \Delta \boldsymbol{t}}{d \boldsymbol{v}} \right)^T V^{-1} \Delta \boldsymbol{t} = 0$$

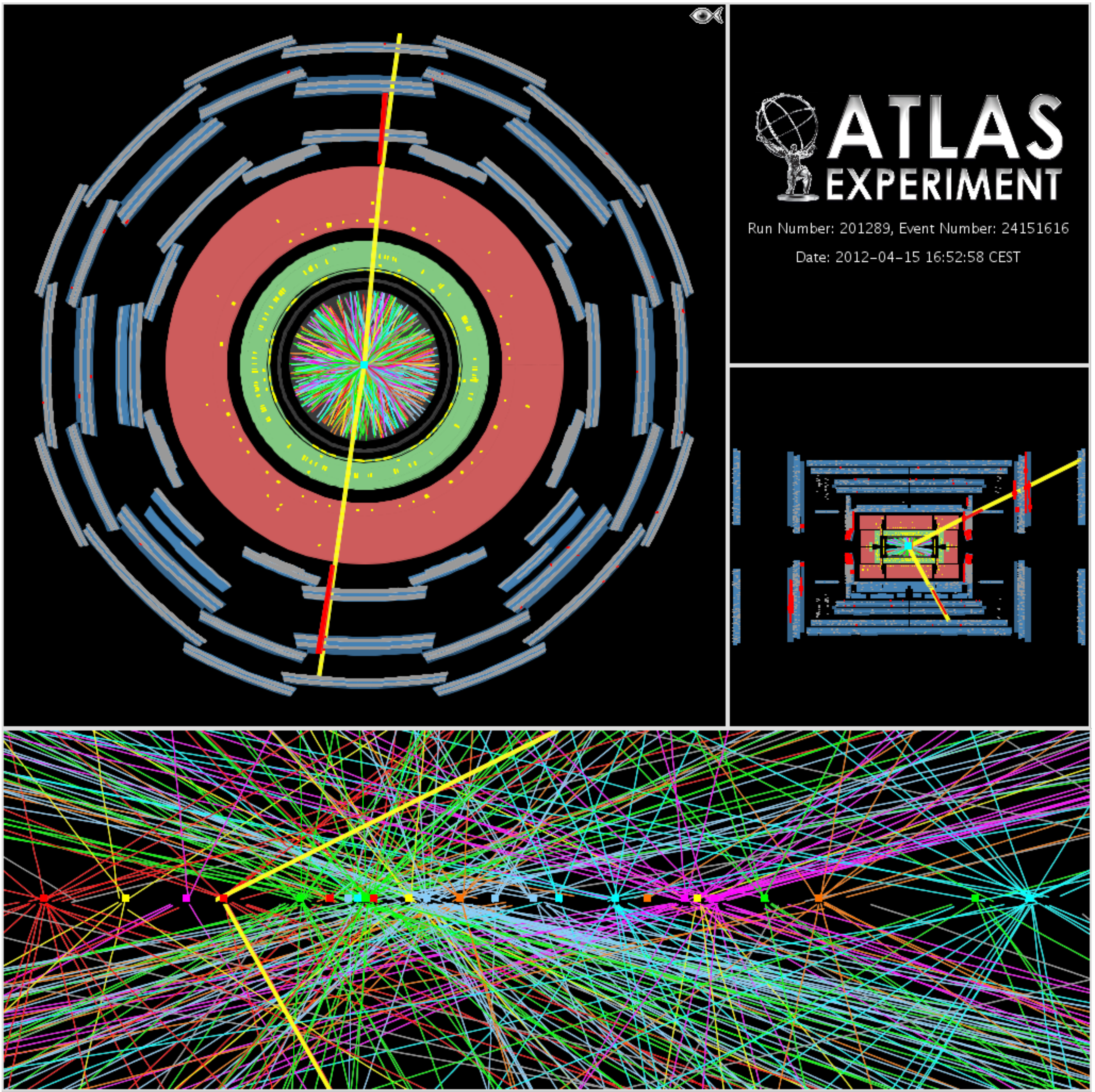
  - As tracks are fed in the vertex, the weights may change. The filter step incorporates the track info in to the current vertex status
  - Incompatible tracks end up with ~0 weight.
- Proven very useful method in a wide range of applications
  - For example. Many vertices in LHC pile-up collisions.

Vertex fitting summary



[Tatjiana Lenz Thesis. U. Wuppertal. 2006]

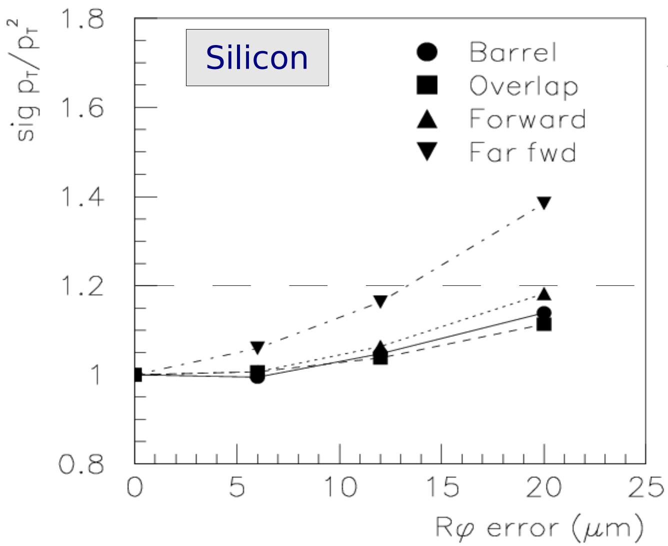
Vertex fitting summary

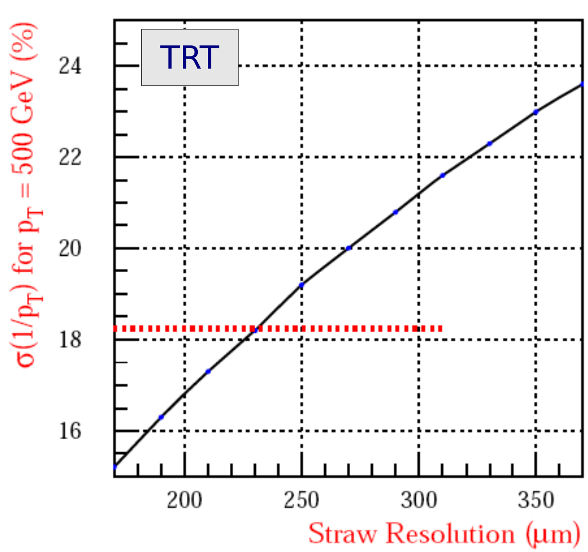


Basic ideas & concepts for alignment

- The aim of the detector alignment is to provide an accurate description of the detector geometry
  - In straight words: to know where the modules are
- The point is: the limited knowledge of the alignment constants should not lead to a significant degradation of the track parameters, beyond that of the intrinsic tracker resolution
  - In ATLAS and for the "initial physics analysis" the requirement is that the degradation should be kept below the 20%

	pixels		SCT	
	barrel	endcap	barrel	endcap
rΦ(µm)	7	7	12	12
z (µm)	20	100	50	200



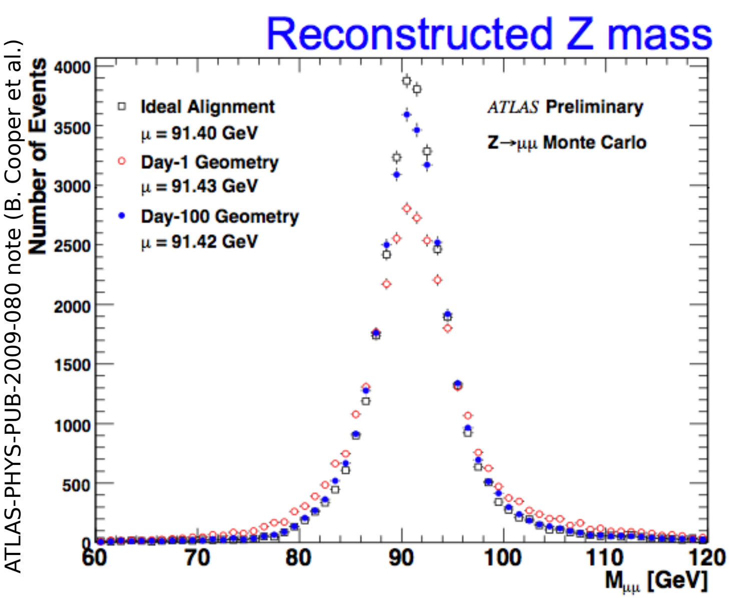


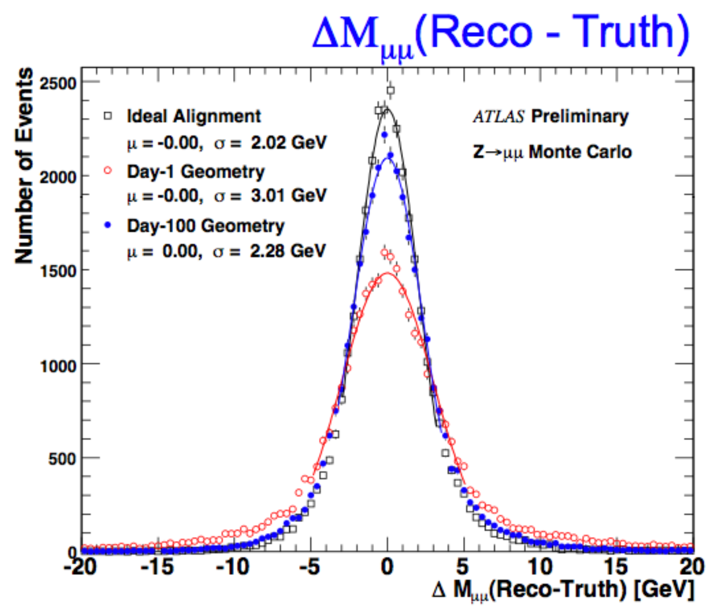
Basic ideas & concepts for alignment

- High accuracy is required for precision measurements
  - A W-mass measurement accuracy of 15-20 MeV/c^2 requires 1μm alignment precision (S. Haywood, ATL-INDET-2000-2005)
  - Higgs mass: if 180 < m_h < 400 GeV/c^2. H→ZZ→ 4l
  - B-tagging: impact parameter & mass
- Example: $Z\to\mu^+\mu^-$ analysis
  - random misalignment
  - Day-1: expected alignment accuracy for Day-1 from cosmic data
  - Day-100: estimate of situation after 100 days of collision data

	Day-1 Barrel	Day-1 Endcap	Day-100 Barrel	Day-100 Endcap
Pixel	20 μm	50 μm	10 μm	10 μm
SCT	20 μm	50 μm	10 μm	10 μm
TRT	100 μm	100 μm	50 μm	50 μm

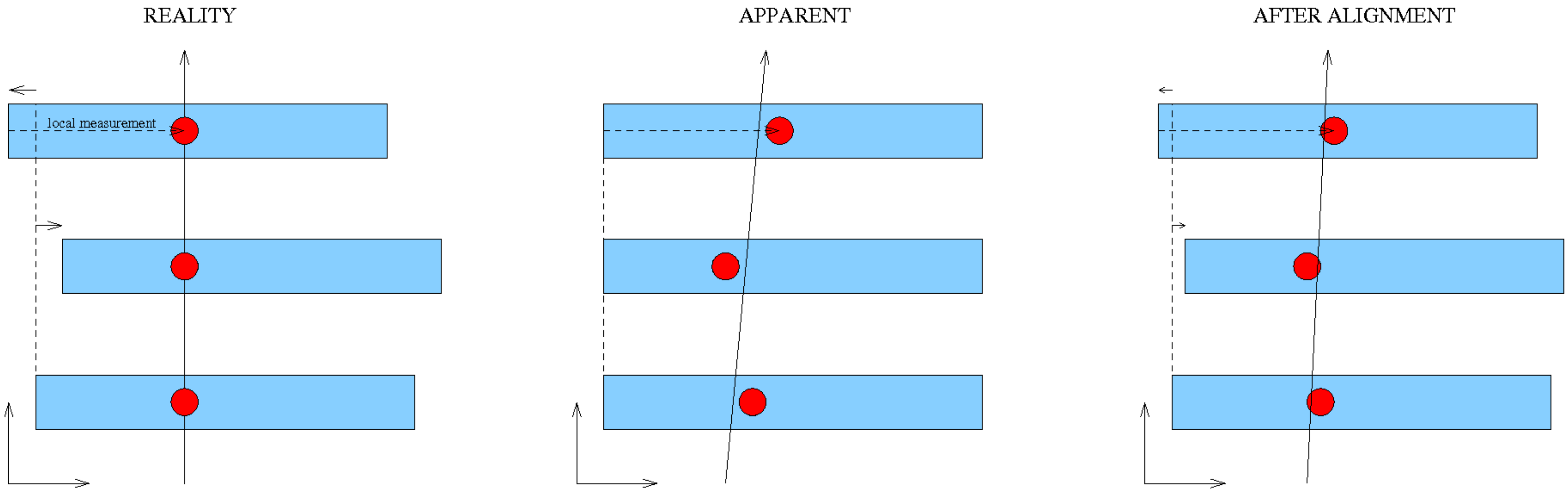
ATLAS-PHYS-PUB-2009-080 note (B. Cooper et al.)





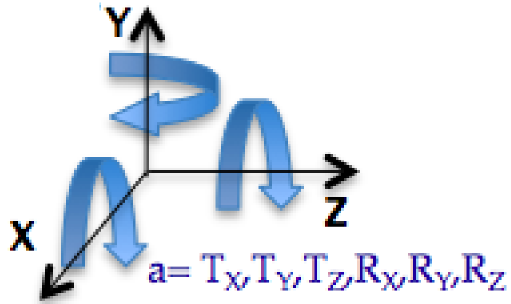
Basic ideas & concepts for alignment

- Basic visualization of the alignment problem
  - Modules are at “unknown” positions. Real hit coordinates are generated by particles that crosses the detector at their “true” location
  - Reconstruction without knowing the real module location. Hits are located at “apparent” positions. Track reconstructions is not accurate
  - After alignment it is possible to have a “residual” misalignment. It will affect the hit positions and the track reconstruction. Hopefully the effect is small



Alignment by χ^2 minimization

- Need to determine 6 alignment parameters per module



$$\boldsymbol{a} = \begin{pmatrix} a_1 \\ \vdots \\ a_{N_A} \end{pmatrix} = \begin{pmatrix} Tx_1 \\ \vdots \\ Rz_1 \\ \vdots \\ Tx_{N_M} \\ \vdots \\ Rz_{N_M} \end{pmatrix}$$

- Define an alignment χ^2 function built from all tracks and hits

$$\boldsymbol{r}_t = \begin{pmatrix} r_{t\ 1} \\ \vdots \\ r_{t\ N_R} \end{pmatrix} \qquad V = \begin{pmatrix} \sigma^2(r_1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma^2(r_{N_R}) \end{pmatrix} \rightarrow \chi^2 = \sum_{\forall t} \boldsymbol{r}_t^T V^{-1} \boldsymbol{r}_t$$

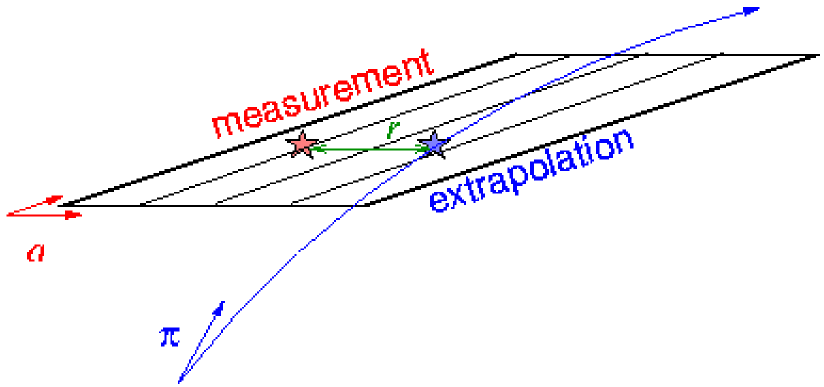
  - Require the minimum condition w.r.t. the alignment parameters

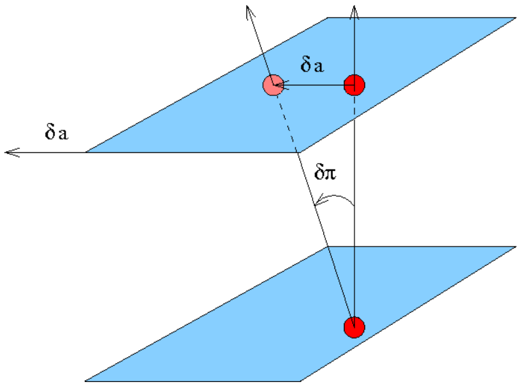
$$\frac{d\chi^2}{d\boldsymbol{a}} = 0 \rightarrow \sum_{\forall t} \left(\frac{d\boldsymbol{r}_t}{d\boldsymbol{a}} \right)^T V^{-1} \boldsymbol{r}_t = 0$$

$$\frac{d\boldsymbol{r}}{d\boldsymbol{a}} = \begin{pmatrix} dr_1/da_1 & \dots & dr_1/da_{N_A} \\ \vdots & \ddots & \vdots \\ dr_N/da_1 & \dots & dr_N/da_{N_A} \end{pmatrix}$$

Alignment by χ^2 minimization

- Now... the residuals derivative contain a nested dependence
  - Residuals depend on track parameters and alignment parameters
  - And track parameters depend on their turn on alignment parameters





  - Mathematically this means:

$$d\,\boldsymbol{r} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{a}} d\,\boldsymbol{a} + \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\pi}} d\,\boldsymbol{\pi} \quad \rightarrow \quad \frac{d\,\boldsymbol{r}}{d\,\boldsymbol{a}} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{a}} + \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\pi}} \frac{d\,\boldsymbol{\pi}}{d\,\boldsymbol{a}}$$

    - Actually this is equivalent to a track refit when alignment parameters change

$$\frac{d\,\boldsymbol{\pi}}{d\,\boldsymbol{a}} = -\underbrace{\left[\left(\frac{d\,\boldsymbol{r}}{d\,\boldsymbol{\pi}}\right)^T V^{-1} \left(\frac{d\,\boldsymbol{r}}{d\,\boldsymbol{\pi}}\right)\right]^{-1}}_{\text{track fit matrix}} \left[\left(\frac{d\,\boldsymbol{r}}{d\,\boldsymbol{\pi}}\right)^T V^{-1} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{a}}\right]$$

  - Again, the derivatives can be computed analytically or numerically

Alignment by χ^2 minimization

- Now... use the first order Taylor expansion
  $$\boldsymbol{r}=\boldsymbol{r}(\boldsymbol{a}_0)+\left.\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{a}}\right|_{\boldsymbol{a}_0}\delta \boldsymbol{a}$$
  - Neglect second order derivatives
  - Compute track parameters, residuals and derivatives with an initial set of alignment constants a_0
- The alignment solution:

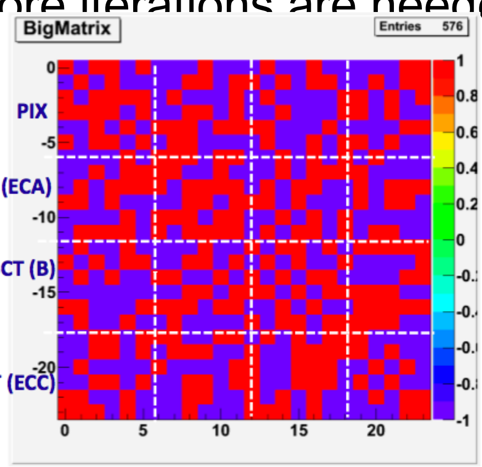
$$\delta \boldsymbol{a}=-\left[\left(\frac{d\,\boldsymbol{r}}{d\,\boldsymbol{a}}\right)^T V^{-1}\left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{a}}\right)\right]^{-1}\left[\left(\frac{d\,\boldsymbol{r}}{d\,\boldsymbol{a}}\right)^T V^{-1}\boldsymbol{r}\right] \quad\rightarrow\quad \boldsymbol{a}=\boldsymbol{a}_0+\delta \boldsymbol{a}$$

- The alignment matrix can be huge !
  - Size is N_A x N_A
    - ATLAS silicon tracker (pixel + microstrips) 36K x 36K → 4.5 GB
    - CMS tracker: ~100K x 100K (size grows a N_A^2)
  - Inversion time:
    - Tests in ALINEATOR (4-core, 32 GB, parallel) @ IFIC-Valencia
      - Full & dense matrix > 1 day (time grows as ~N_A^3)
      - Correlation matrix of ***a*** available
    - In a commercial PC:
      - Fast inversion of sparse matrix ~1 min
      - No correlation matrix available

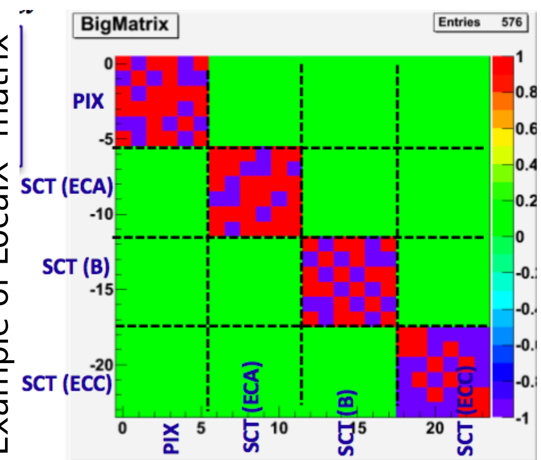
Alignment by χ^2 minimization

- Solving the alignment. Two approaches: Globalχ^2 vs Localχ^2
  - Globalχ^2 :module correlation is taken into account by $d\pi/da$
    - Alignment matrix becomes dense
  - Localχ^2: $d\pi/da = 0$ module correlation is not considered
    - Alignment matrix becomes block diagonal
    - Alignment matrix inversion is not an issue
    - More iterations are needed

Example of Globalχ^2 matrix



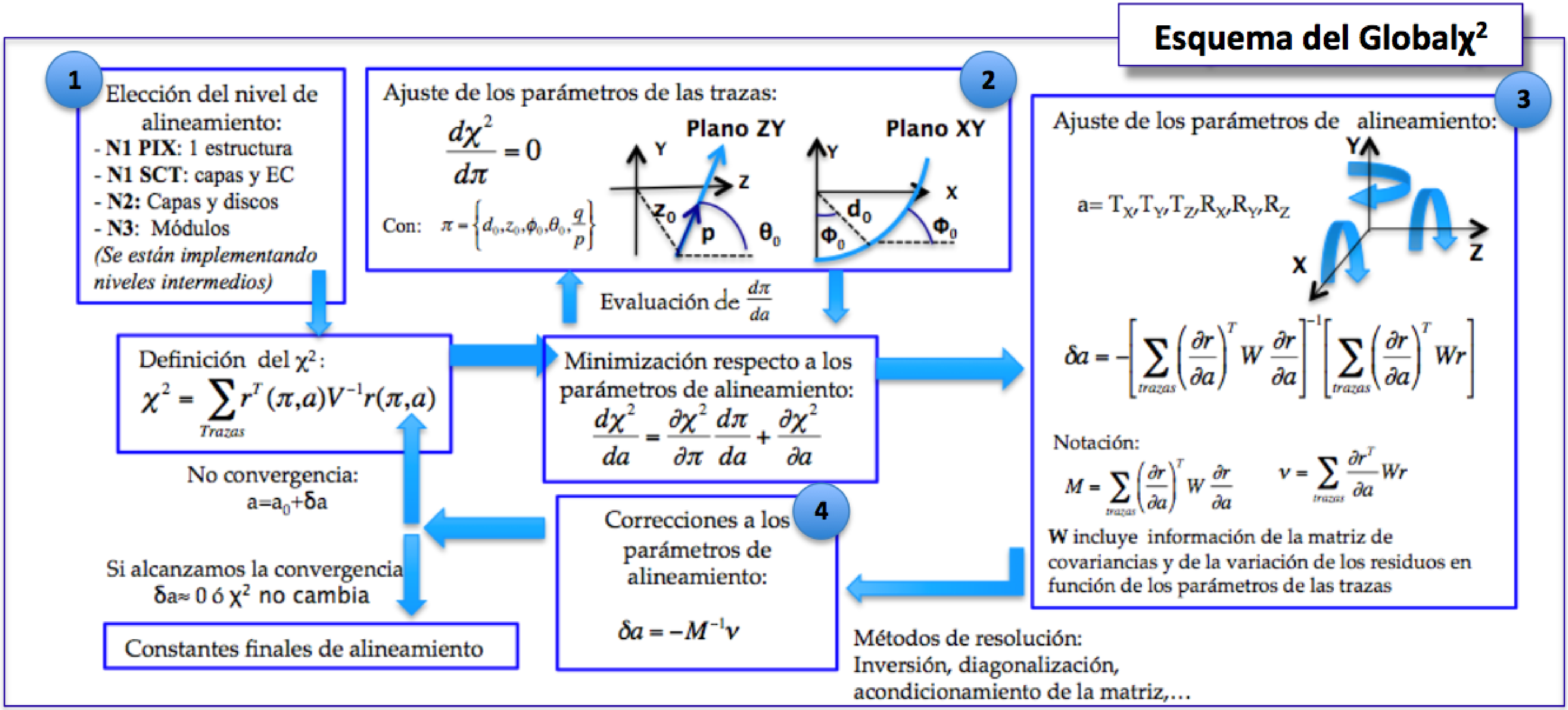
Example of Localχ^2 matrix



- Adding constraints. The alignment χ^2 accepts constraint terms
  - Track parameters: beam spot, invariant masses, E/p for electrons ?
  - Alignment parameters: Assembly survey, online laser survey, soft mode cuts,...

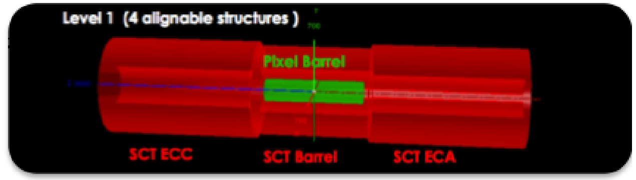
Alignment strategy

- Alignment algorithm is run in an iterative procedure
  - Until convergence is reached
  - Each iteration may take several hours (up to 1 day)

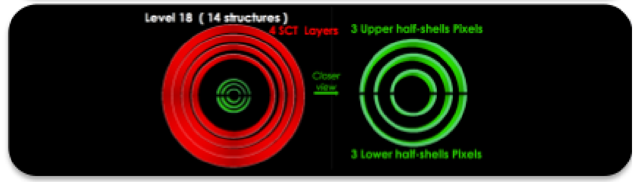


Alignment strategy

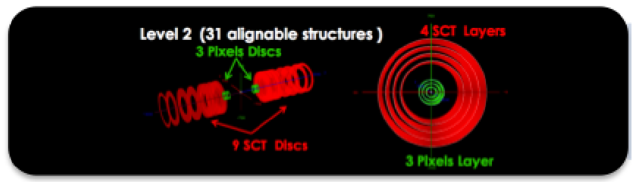
- The alignment procedure mimics the detector assembly structures
  - From large structures
    - PIX, SCT,
    - Barrel, End caps
    - Layers, disks
    - Staves, rings
  - To individual modules
- The size of corrections
  - Large structures
    - mm and mrad
  - Staves
    - 100s microns
  - Modules
    - 10s microns
- Statistics needed:
  - Large structures: O(1000)
  - Staves: O(10,000)
  - Modules: O(1,000,000)



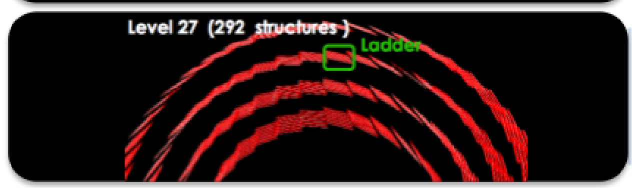
Level 1: 4 struct. → 24 Dofs
PIX: complete detector
SCT: 1 barrel + 2 end caps



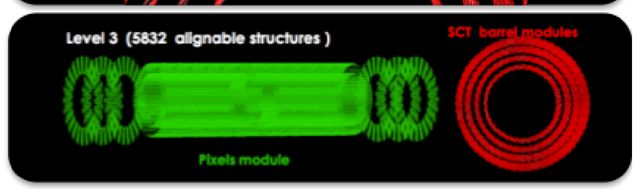
Level 1.8: 14 struct. → 84 Dofs
PIX: (B) 3x2 half layers + 2 EC
SCT: (B) 4 layers + 2 EC



Level 2: 31 struct. → 186 Dofs
PIX: (B) 3 layers + 2x3 EC disks
SCT: (B) 4 layers + 2x9 EC disks



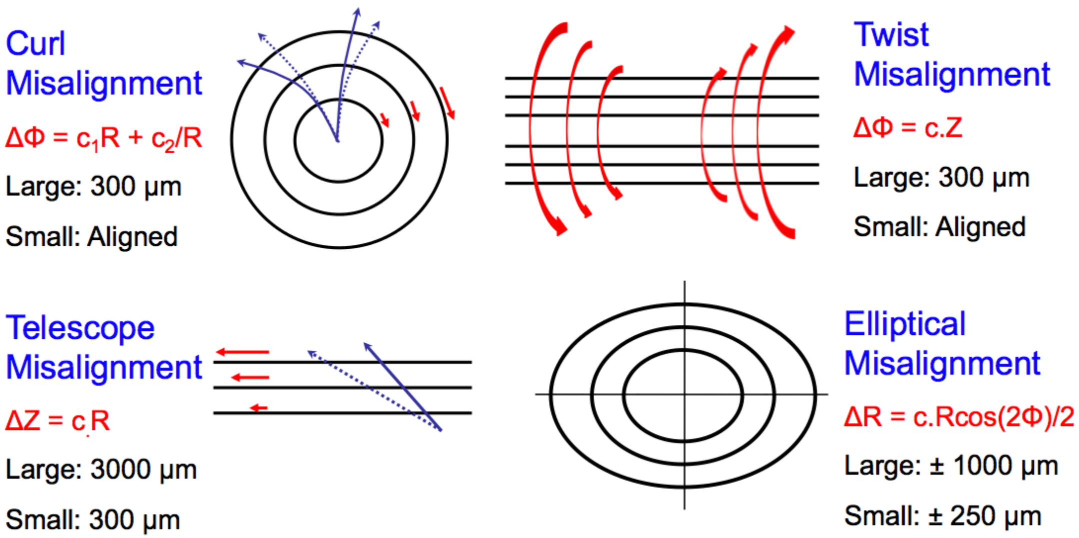
Level 2.7: 292 struct → 1752 Dofs
PIX: (B) 112 staves + 2 EC
SCT: (B) 176 staves + 2 EC



Level 3: 5832 struct → 34992 Dofs
PIX: (B) 1456 + (EC) 2x144
SCT: (B) 2112 + (EC) 2x988

Alignment systematics

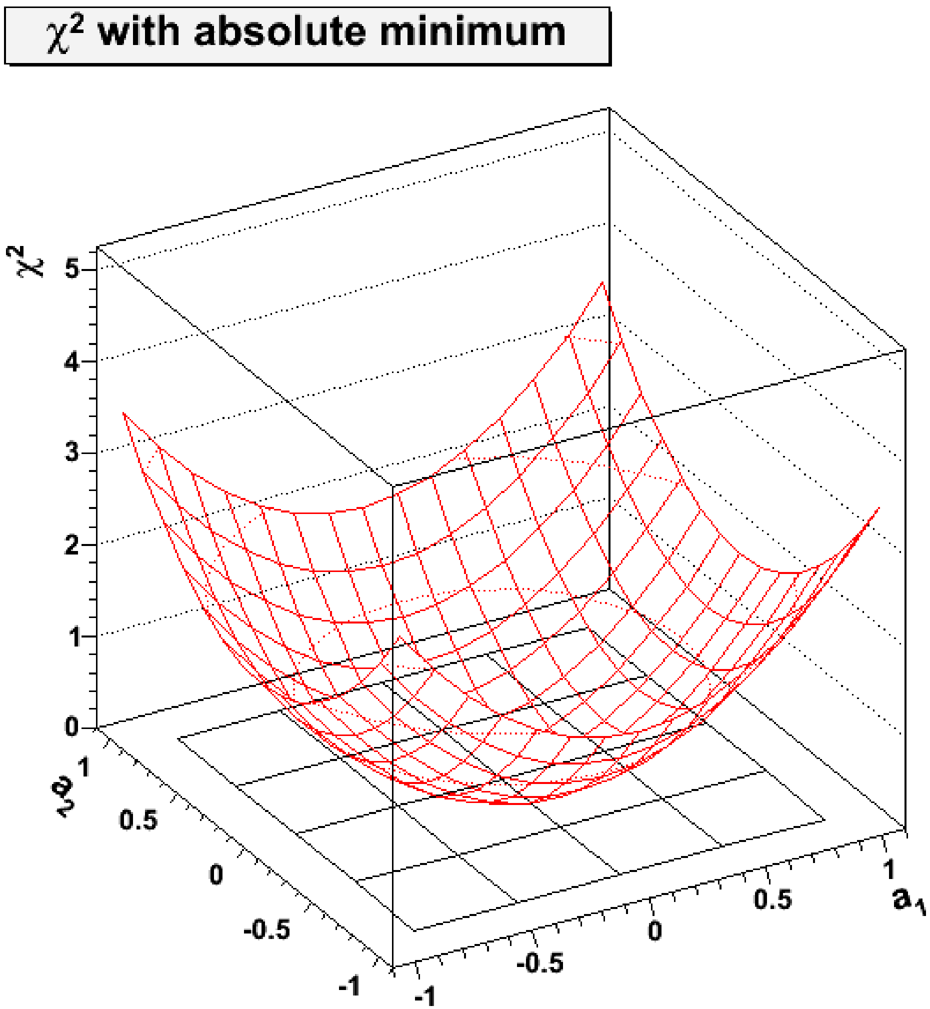
- Weak modes: these are solutions of the alignment that do not correspond with real movements, but that preserve the helicoidal path of the tracks, leaving the track χ^2 almost unchanged
- Examples of weak modes:

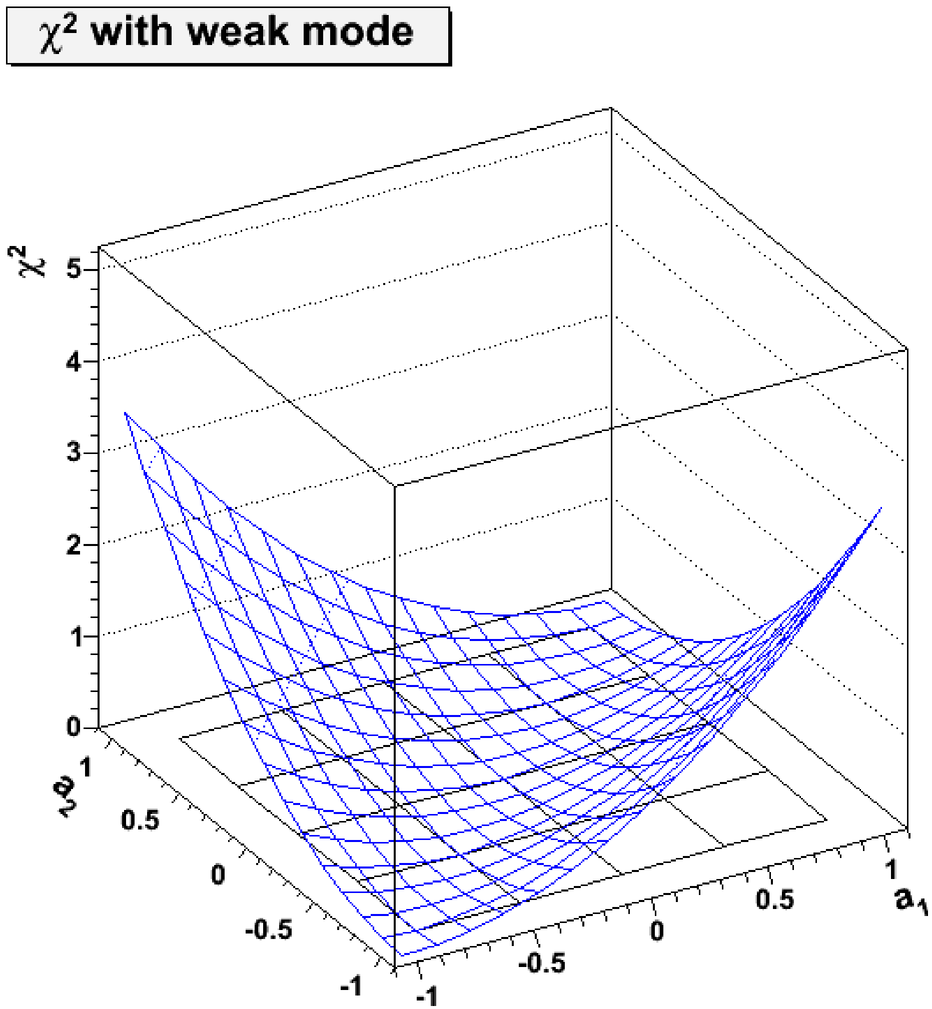


- Material effects:
  - In order to achieve a resolution of the alignment corrections down to 1 micron one needs to consider closely the material effects in the track reconstruction.
  - The material description must be accurate and all operational conditions under control
  - Detector deformation: out of plane twisting and bending (planar silicon devices), wire sag (gas systems)

Weak modes

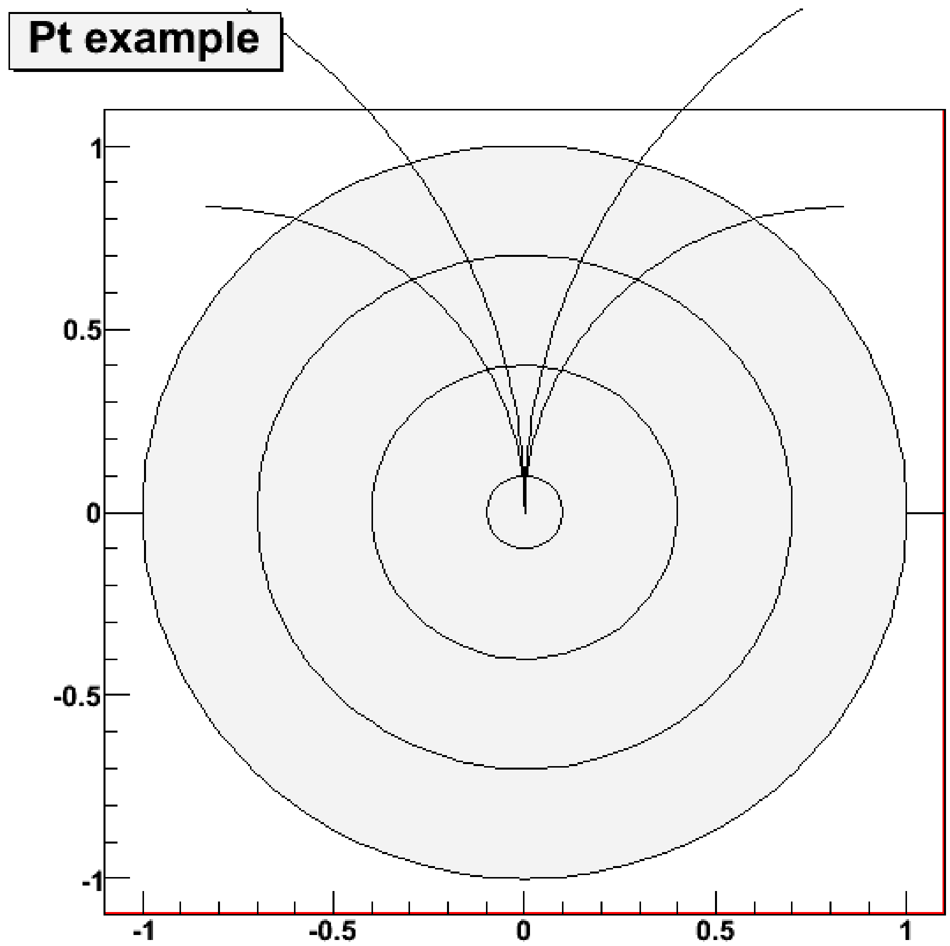
- Example of χ^2 distributions with weak modes
  - Alignment parameter space (just 2 dimensions: a_1 and a_2)



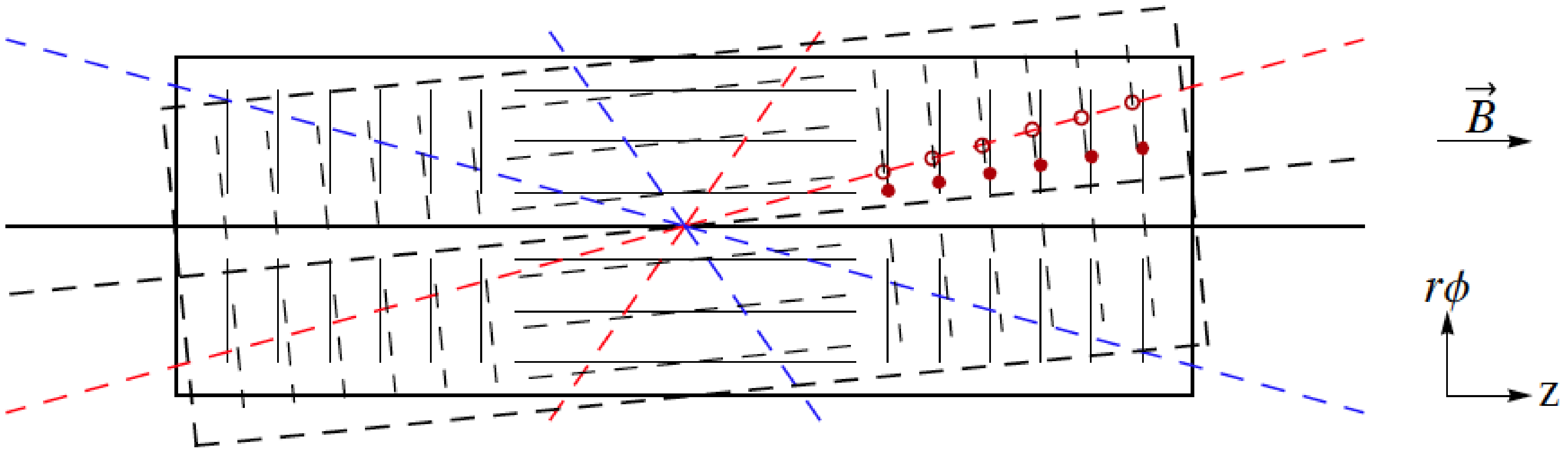


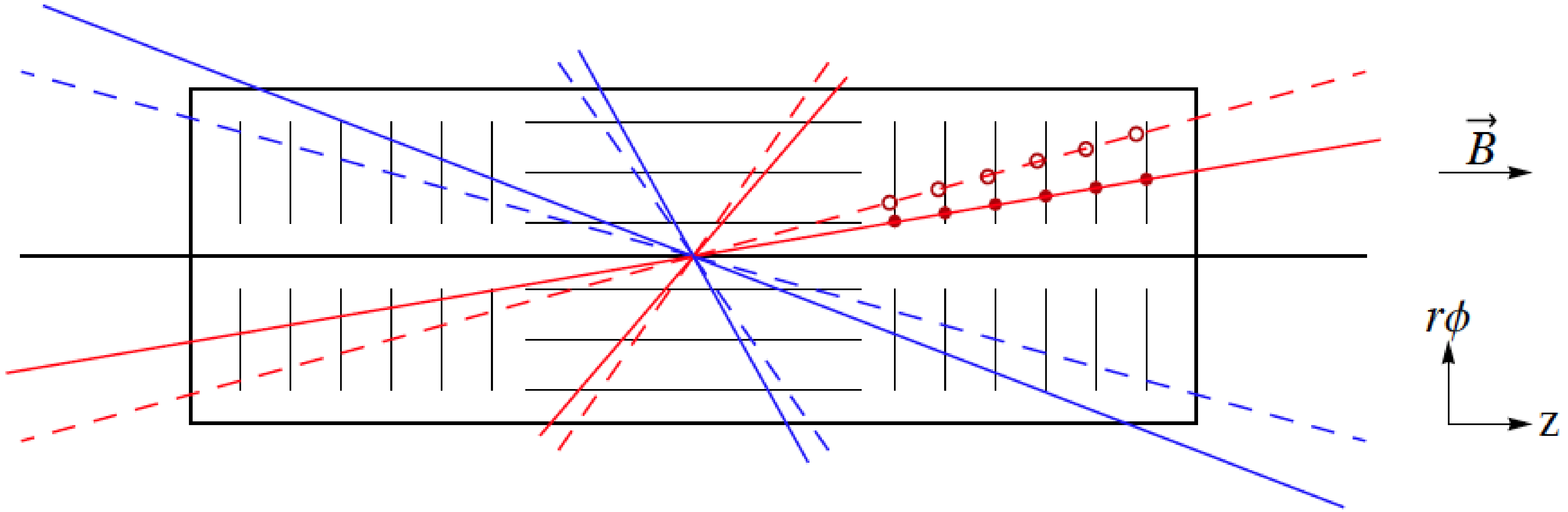
Weak modes: example of curl

- The curl of the different detector layers may bias the transverse momentum reconstruction
- Under a curl: tracks preserve their helicoidal path



Altre tiupus de problemes





Global X2 vs Local X2

- In the GX2, both track and alignment parameters are refitted
- In the LX2, tracks are frozen and only the alignment parameters are fitted

$$GX2 \rightarrow \boldsymbol{r} = \boldsymbol{r}_0 + \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{t}} \delta \boldsymbol{t} + \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{a}} \delta \boldsymbol{a}$$

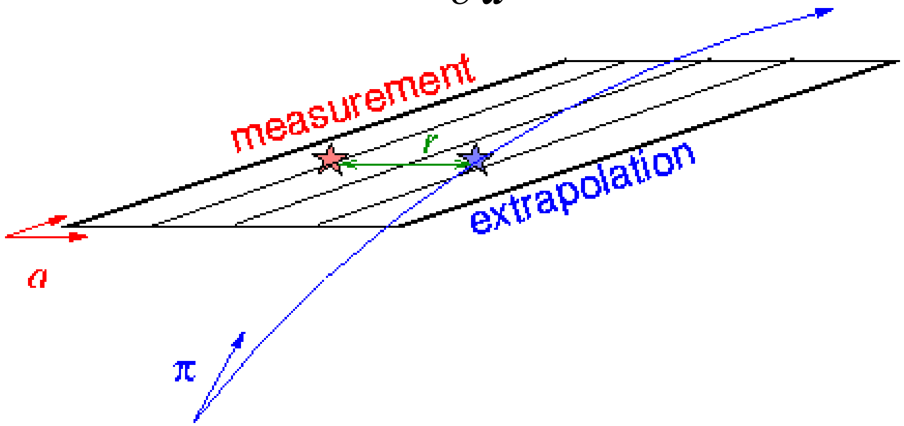
$$LX2 \rightarrow \boldsymbol{r} = \boldsymbol{r}_0 + \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{a}} \delta \boldsymbol{a}$$

$$\chi^2 = \sum \left[\boldsymbol{r}^{\boldsymbol{T}}(t,a) V^{-1} \boldsymbol{r}(t,a) \right]$$

$$\boldsymbol{t} = (d_0, z0, \phi_0, \theta, q/p)$$

$$\boldsymbol{a} = (T_x, T_y, T_z, R_x, R_y, R_z)$$

$$V = \begin{pmatrix} \sigma_{hit} & 0 \\ 0 & \sigma'_{hit} \end{pmatrix}$$



$$\frac{d\chi^2}{da} = 0 \quad \rightarrow \quad \sum \left[\boldsymbol{r}^{\boldsymbol{T}} V^{-1} \left(\frac{\partial \boldsymbol{r}}{\partial t} \frac{dt}{da} + \frac{\partial \boldsymbol{r}}{\partial a} \right) \right] = 0$$

Genuine GX2 term

$$GX2 \rightarrow \sum \left[\left(\frac{d\boldsymbol{r}}{d\boldsymbol{a}} \right)^T V^{-1} \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{a}} \right) \right] \delta \boldsymbol{a} + \sum \left(\frac{d\boldsymbol{r}}{d\boldsymbol{a}} \right)^T V^{-1} \boldsymbol{r} = 0$$

N_{DoF} x N_{DoF} matrix

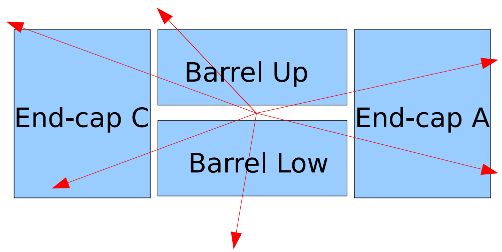
N_{DoF} vector

$$LX2 \rightarrow \sum \left[\left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{a}} \right)^T V^{-1} \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{a}} \right) \right] \delta \boldsymbol{a} + \sum \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{a}} \right)^T V^{-1} \boldsymbol{r} = 0$$

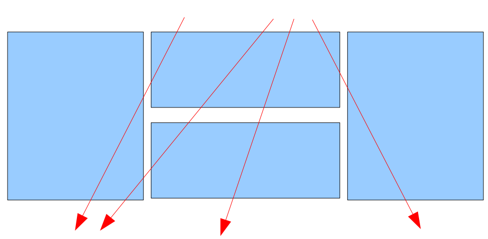
Global X2 vs Local X2

- In the GX2, the alignment matrix may become dense
  - Inversion may be an issue for large matrices
  - Introduction of global degrees of freedom
- In the LX2, the alignment matrix is block diagonal
- Example:

collision



cosmics



GX2 matrices

Barrel Up, Barrel Low, End-cap A, End-cap C

$$\begin{pmatrix} x & x & 0 & 0 & x & x & x & x \\ x & x & 0 & 0 & x & x & x & x \\ 0 & 0 & x & x & x & x & x & x \\ 0 & 0 & x & x & x & x & x & x \\ x & x & x & x & x & x & 0 & 0 \\ x & x & x & x & x & x & 0 & 0 \\ x & x & x & x & 0 & 0 & x & x \\ x & x & x & x & 0 & 0 & x & x \end{pmatrix}$$

$$\begin{pmatrix} x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & 0 & 0 \\ x & x & x & x & x & x & 0 & 0 \\ x & x & x & x & 0 & 0 & x & x \\ x & x & x & x & 0 & 0 & x & x \end{pmatrix}$$

LX2 matrices

$$\begin{pmatrix} x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & x & x & 0 & 0 & 0 & 0 \\ 0 & 0 & x & x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x & x & 0 & 0 \\ 0 & 0 & 0 & 0 & x & x & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & x & x \end{pmatrix}$$

Example: 4 structures, 2 DoF/structure

Alignment summary

- The goal of the ID alignment is to determine the position of the tracking modules with enough precision for the physics analysis
  - This requires precision below 10 microns (ultimate goal 1 micron)
  - Determination of almost 40K ATLAS & 100K CMS degrees of freedom
    - 6 per module (Tx, Ty, Tz, Rx, Ry, and Rz)
- Track based alignment algorithms can reach good precision
  - Combination almost mandatory with survey constraints
  - Track parameters constraints
- Study of random and systematic deformations is difficult to tackle
- ATLAS & CMS alignment of tracking systems ready for first LHC collisions

Thanks to: Carlos Escobar, Vicente Lacuesta and Regina Moles

Bibliography

- "Data Analysis Techniques for High-Energy Physics"
  - R. Früwirth et al.
- "Statistical Methods in Experimental Physics"
  - F. James
- "Introduction to experimental particle physics"
  - R. Fernow
- "Techniques for Nuclear and Particle Physics Experiments"
  - W.R. Leo
- "Inner Detector Reconstruction: tracking"
  - A. Salzburger
  - Artemis school, 15-19 September 2008, MPI Munich, Germany