Tracking lecture 3

Pascal Vanlaer (Université Libre de Bruxelles / IIHE)

Belgium-Nederland-Deutschland doctoral school
of particle physics - Spa, 2-13 September 2019

Track finding and fitting

- Track *fitting* is an estimation problem
  - Given measurements (i.e. hits) along the particle track, compute track parameters most compatible with the measurements
    - With best accuracy at particle production vertex
    - With good accuracy at calorimeter or muon chambers for matching
    - Actually, with best accuracy everywhere along the track to handle ambiguous hits or reject noise hits
  - This is a well-defined mathematical problem…
    - …once we know which hits are due to a single particle track ☺
  - Track fitting is typically expensive in CPU
    - Involves extrapolation in detector B-field and material...
      - …but fast, approximative descriptions can be developed
    - Involves 5x5 matrix inversions (~1 per hit)

- Track *finding* is a pattern recognition problem
  - Given many hits, assign them to an *a-priori unknown* number of tracks of unknown (p_T, η, ϕ) and production vertices (v_x, v_y, v_z)
    - This is the tough part
  - What is known a priori:
    - Detector geometry and magnetic field
    - The average trajectory, once a particle (p_T, η, ϕ) and production vertex (v_x, v_y, v_z) are fixed ☺
    - And more, if you think about it:
      - Local detector efficiency and noise occupancy
      - Local hit density, on a per-event basis
      - …

The chicken and the egg…

- From the two previous slides, an iterative approach seems natural
  - Coarse guess -> fit -> refine…
  - What is done actually: initial track segment (seed) -> fit -> extrapolate and look for compatible hits -> fit…
  - Many seeds to try, but tracks quickly refine as iterative tracking proceeds
- Global approaches also exist
  - They try to find the relevant trajectories and all their associated hits in one go
  - Can save CPU if there are less track fits to do

The fun of it…

- No generic solution: creativity welcome !
- Many aspects can (and should) be optimized
  - A good analysis of the concrete case is useful
- What matters in the end is:
  - *Track finding efficiency*
  - *Rate of fake tracks from random hit combinations*
  - *Track parameter resolution*
  - *CPU time (money)*

Track fitting

Kalman filter track fitting

- Track fitting must account for:
  - The resolution of the hit coordinates
  - The magnetic field
  - The average energy loss in the detector material
    - So far, this is a standard fitting problem
  - The random perturbation of the trajectory by multiple scattering
    - This is different: the parameters to fit change between two measurements, due to a random process
- The theory was established in 1985, when the similarity was recognized with a problem of “optimal estimation”, that was developed in the 1960' to guide rockets in the presence of atmospheric perturbations.
  - *R.Frühwirth, NIM A262 (1987) 444-450.*
  - *P.Billoir developed the theory independently at the same time, without recognizing that it was a Kalman filter*

Track parametrization

In a field **B // z:**

5 independent parameters, e.g.:

$$\boldsymbol{p} = (\rho, \theta, \phi_P, d_{xy}, d_z)$$

The ρ parameter represents the signed curvature, defined as:

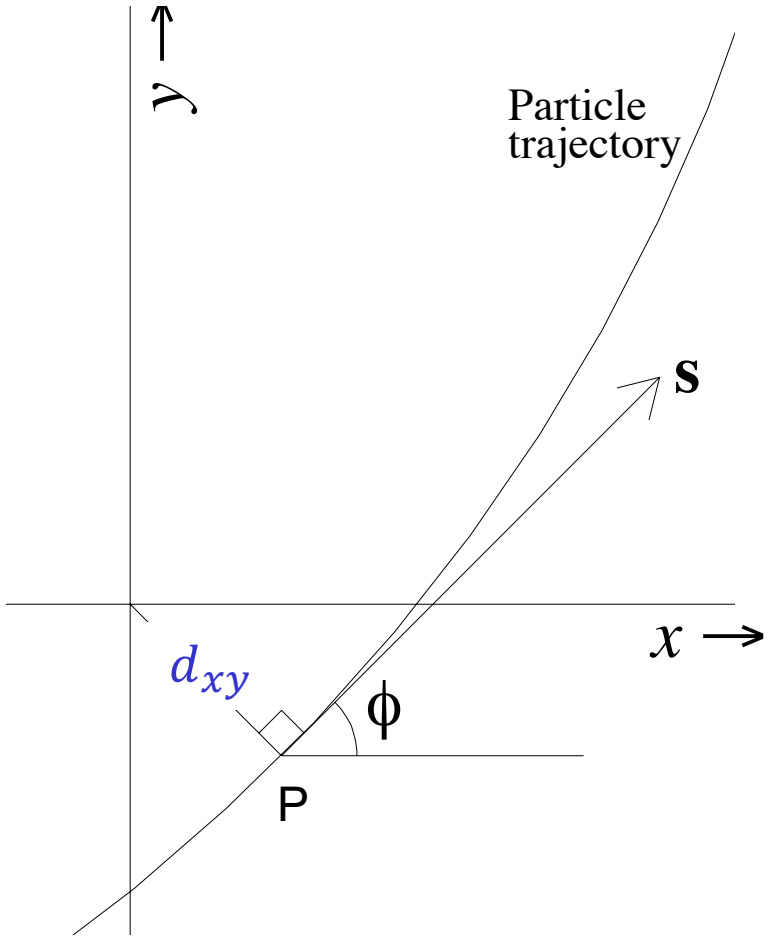
$$\rho = 0.3\frac{qB}{p_T}$$

θ: polar angle wrt **z** axis

ϕ_P: azimuthal angle at the track perigee P, defined as the point of closest approach to (x,y)=(0,0)

d_{xy}: signed transverse impact parameter at the perigee

d_z: z-coordinate of the perigee



Equations of motion

The parameters change along the track according to the equations of motion of the charged particle. In a homogeneous **B** field // to z, the track is a helix with the following equations:

$$\mathbf{p}(s) = \begin{pmatrix} \rho \\ \theta \\ d(s) = d_{xy} + \ldots \\ \phi(s) = \phi_P + s\rho\sin(\theta) \\ z(s) = d_z + s\cos(\theta) \end{pmatrix}$$

s: curvilinear coordinate along helix

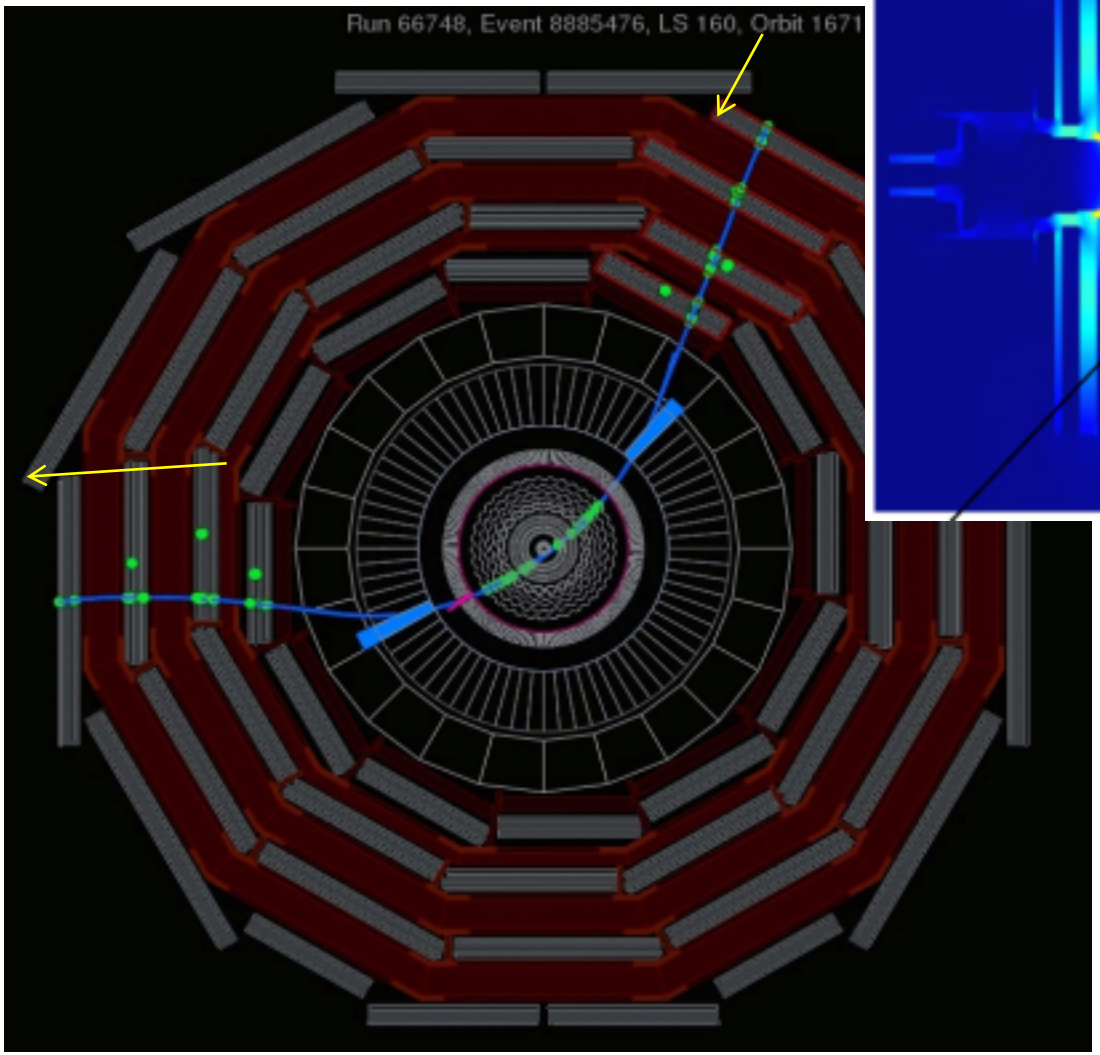
In detector material, the average energy loss must also be accounted for. According to the Bethe-Bloch curve, at the minimum of ionisation energy loss:

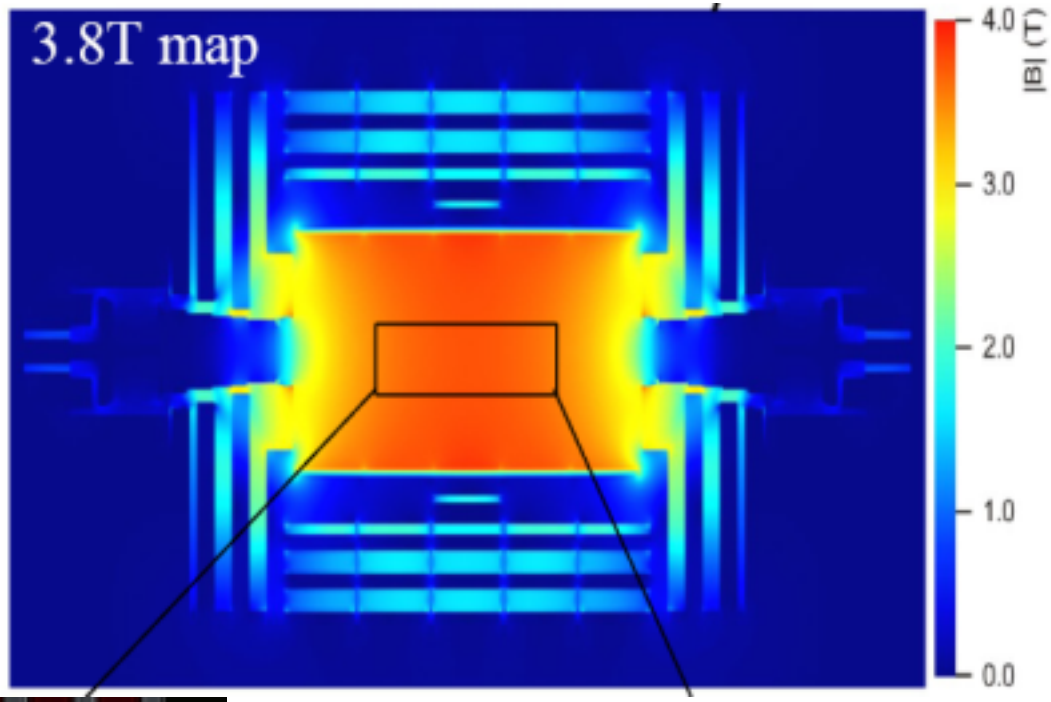
$$\left(-\frac{dE}{dx}\right) \cong 2 \text{ MeV g}^{-1}\text{cm}^2$$

Loss in 1 cm Cu: ≅ 20 MeV, not negligible with respect to the energy of the particles of interest (a few hundred MeV upwards)

Equations of motion

Example: track of a cosmic muon in CMS (July 2008)





B field map from calculations:

- Quite homogeneous inside the tracker, but not in the muon chambers

Note the larger curvature upon exit (energy loss effect)

Multiple Coulomb Scattering

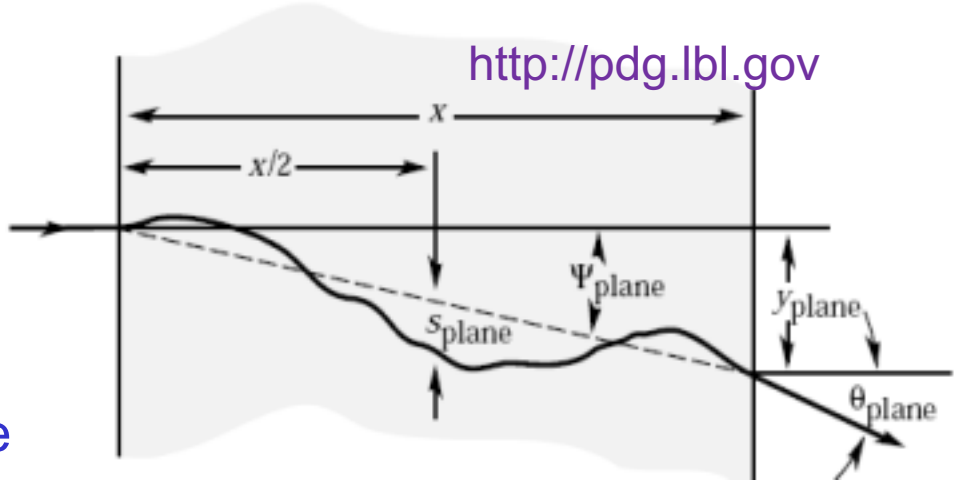
A charged particle with momentum *p* and charge number *z* traversing a thickness *x* of material undergoes multiple random scatters

The scattering angle is approximately Gaussian-distributed. The RMS of the scattering angle projected onto a plane can be estimated as:

$$\theta_0 \approx \frac{13.6MeV}{\beta cp} z\sqrt{x/X_0}$$

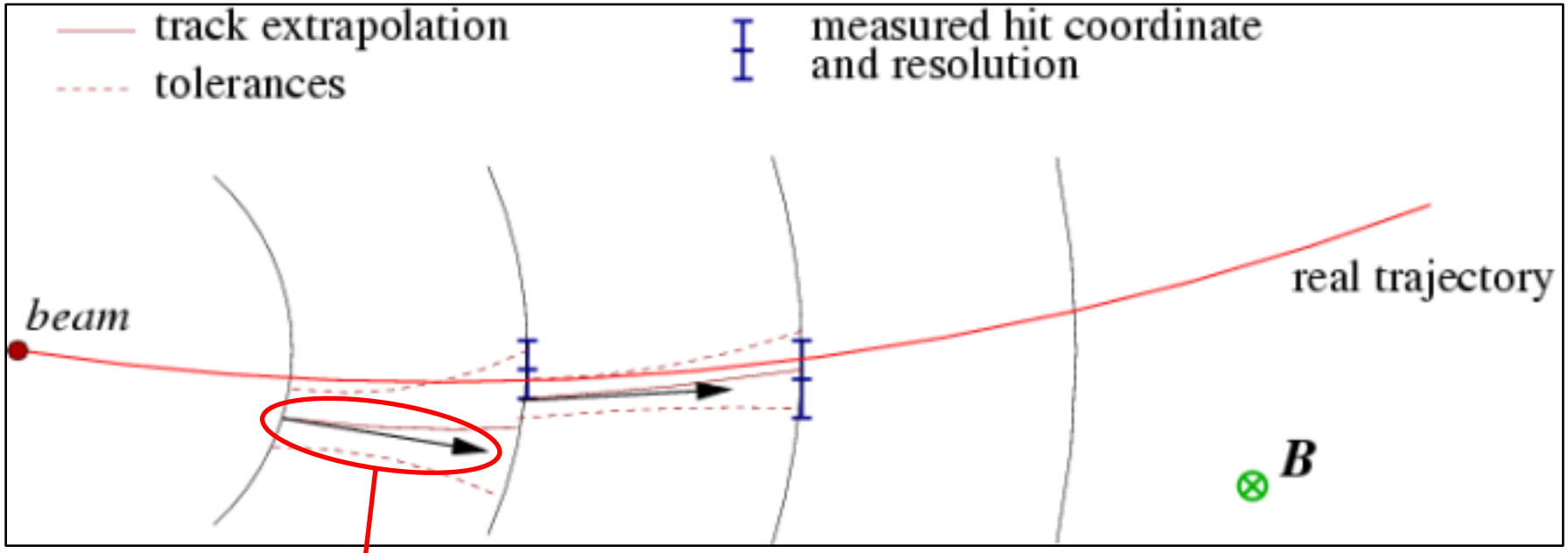
X_0 is the radiation length:

$$X_0 \approx \frac{716.4gcm^{-2}A}{Z(Z+1)\ln(287/\sqrt{Z})}$$



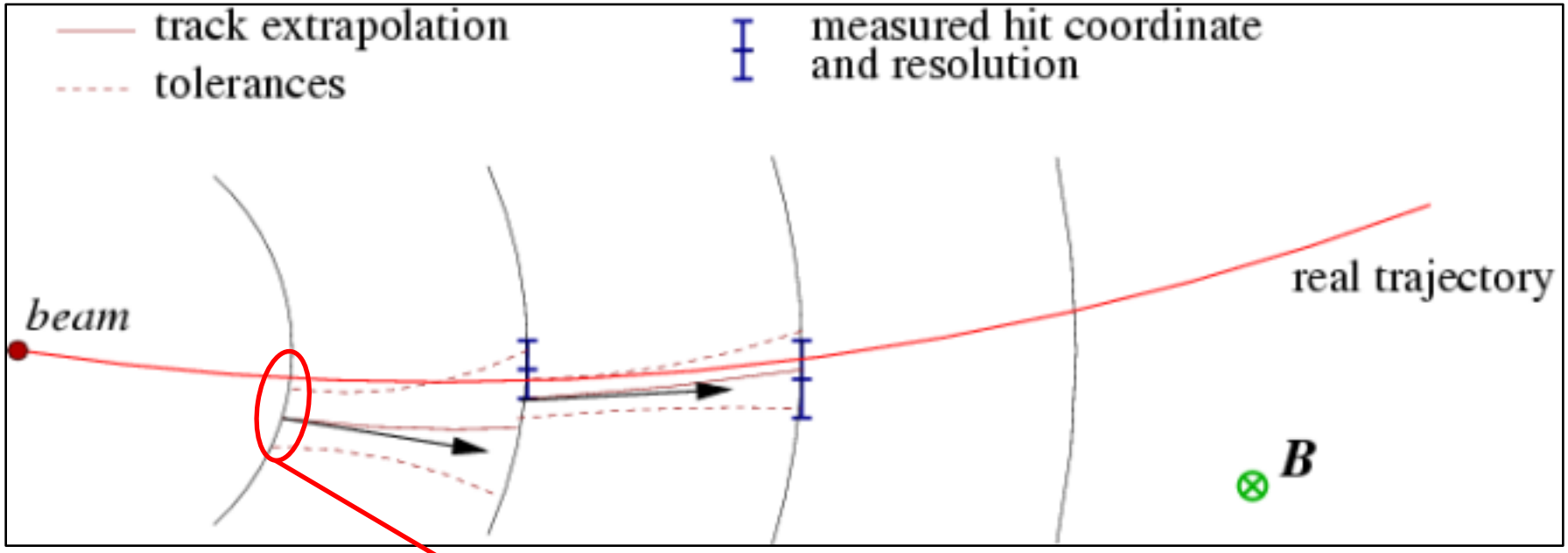
- A muon of 10 GeV traversing 0.1 X_0 of material is deflected by an angle which has an RMS of 4.10^{-4} radians
- If next detection layer is 10 cm further, the deviation from a straight unscattered line is 40μm RMS, comparable to the detector position resolution

Kalman filter in pictures



Initial track parameters
(from fit of track seed)

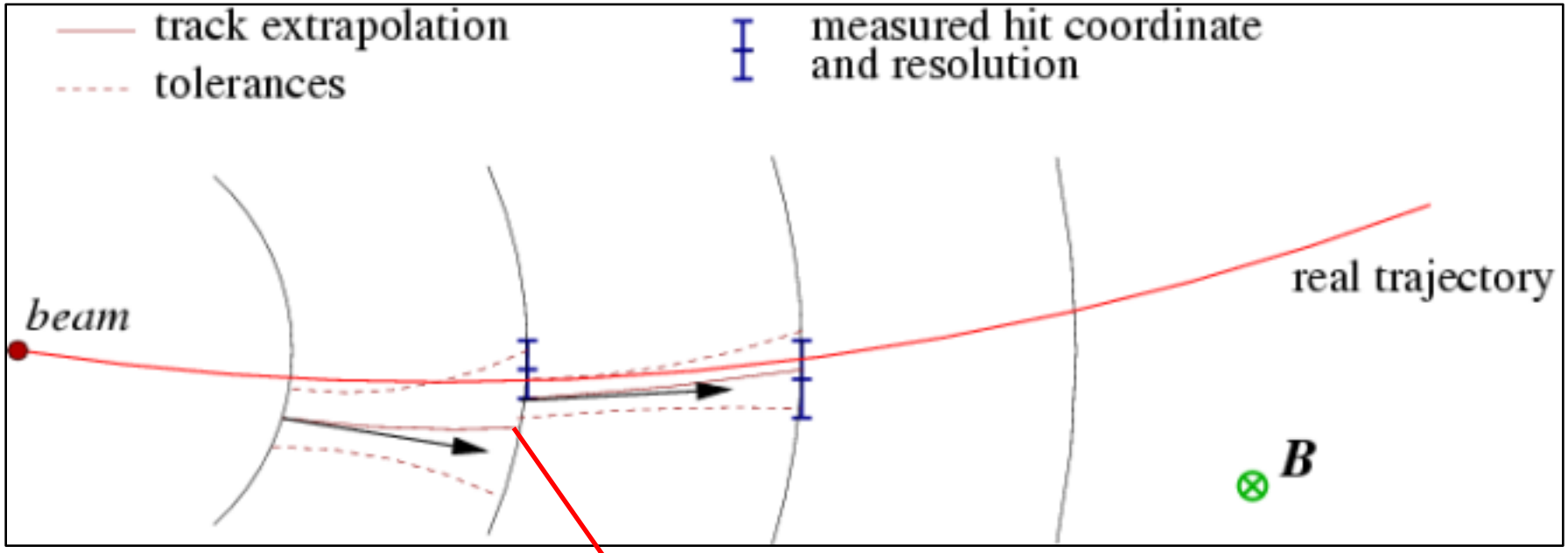
Kalman filter in pictures



Initial track parameter uncertainties

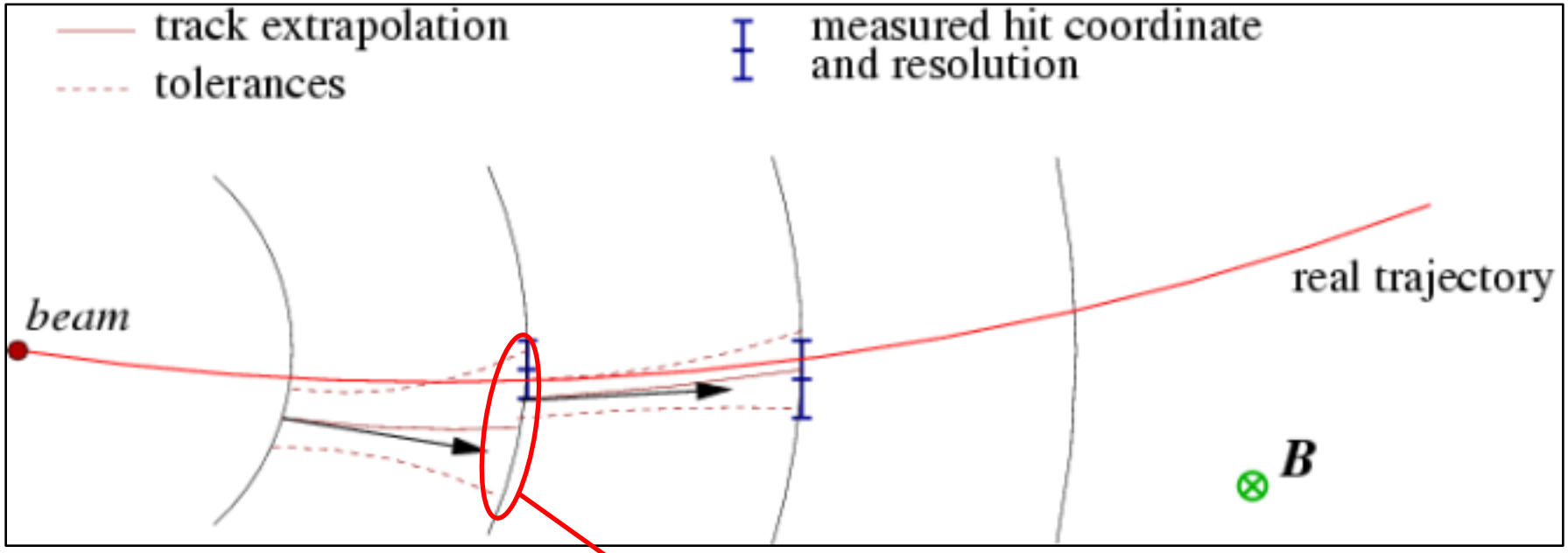
- 5x5 covariance matrix
- from fit of track seed, safely blown up

Kalman filter in pictures



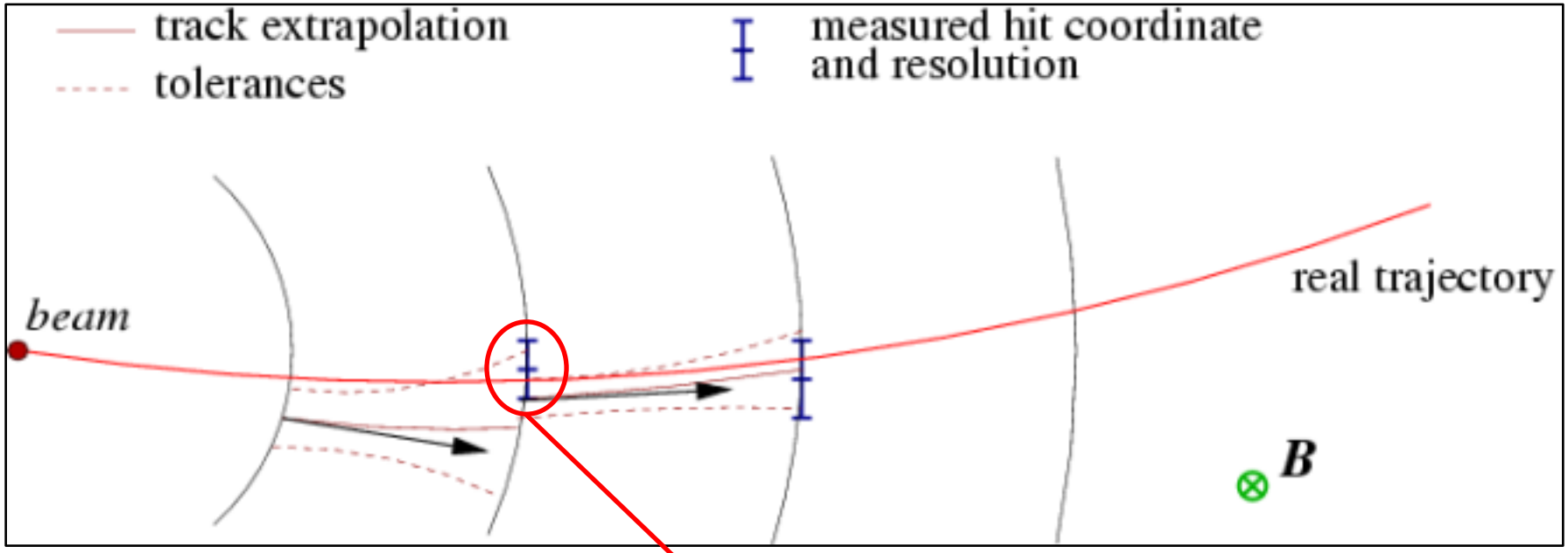
Extrapolation to next detection layer accounting for B-field and energy loss in material

Kalman filter in pictures



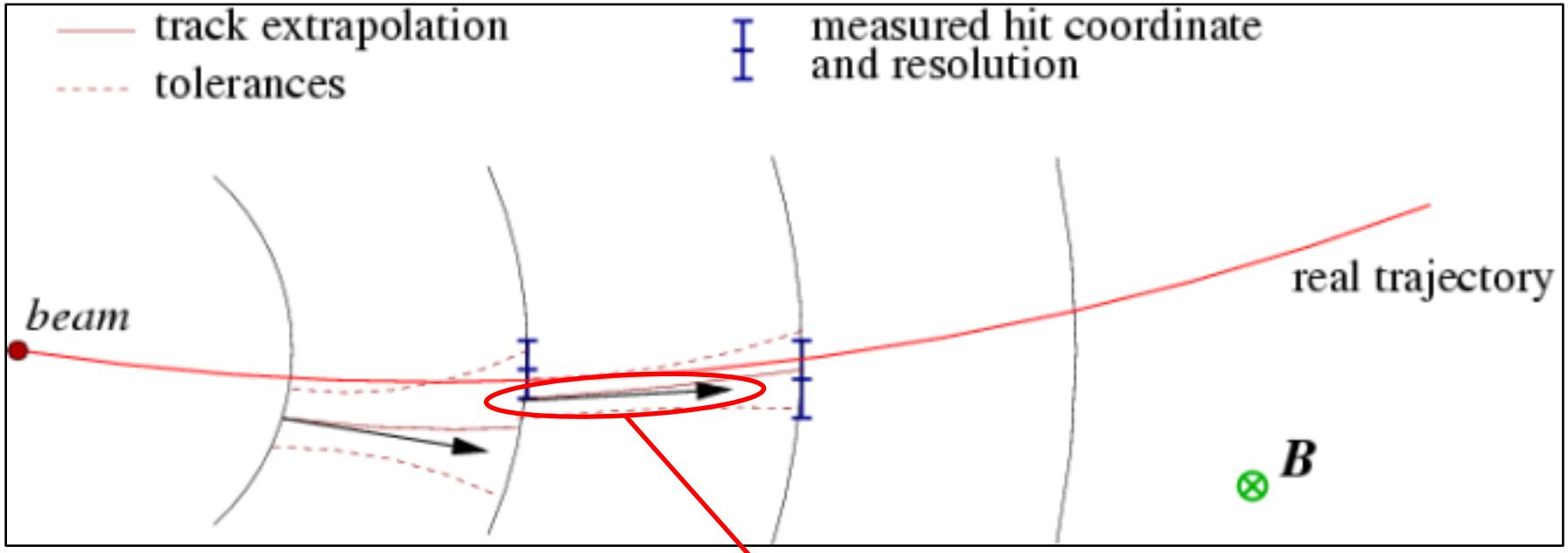
Propagation of uncertainties, adding contribution of Multiple Scattering

Kalman filter in pictures



Check for presence of nearby hit

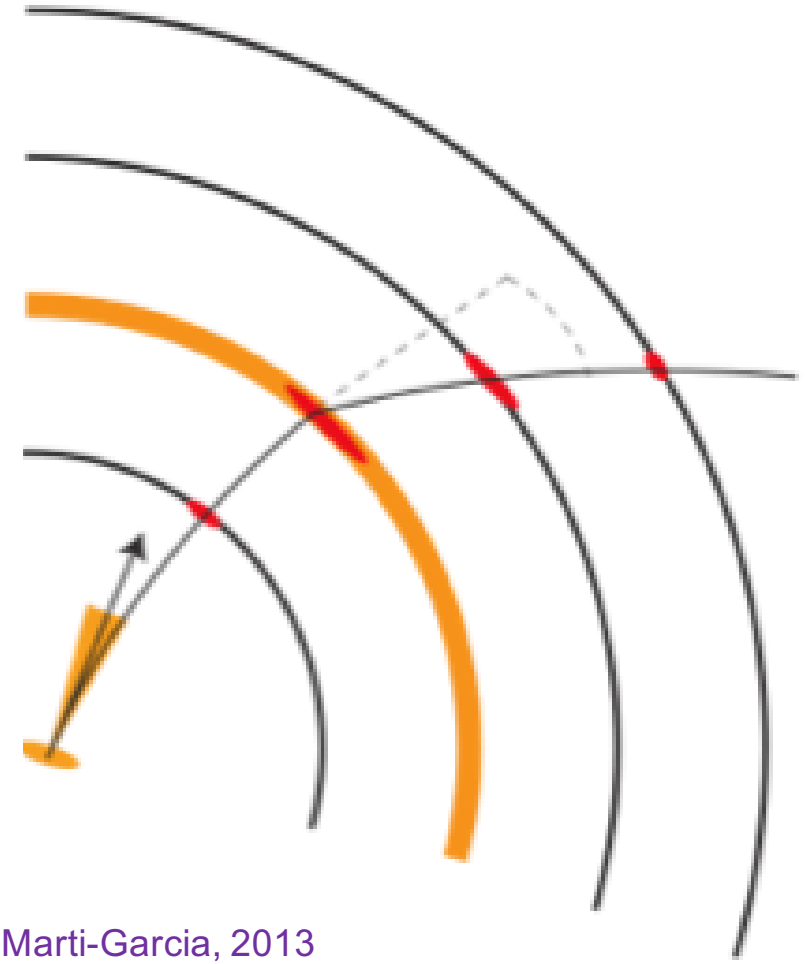
Kalman filter in pictures



Update track parameters and uncertainties by including information from measured hit coordinates

- By magic of correlations and matrix algebra, a 1D or 2D hit improves a 5D track parameter vector
  - See e.g. E.G.Wolin and L.L.Ho, NIM A329 (1993) 493
- Can reject hit if χ^2 is large

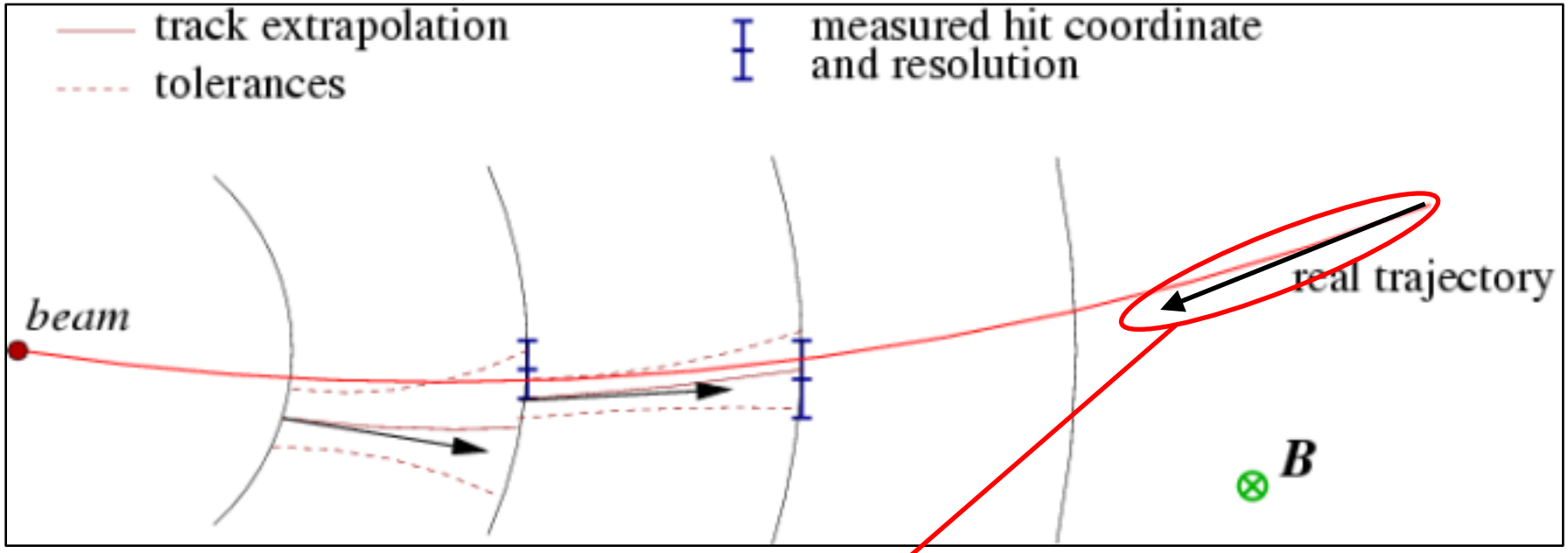
Kalman filter in pictures



S.Marti-Garcia, 2013

- By including the MS contribution into the track covariance matrix, a hit consistent with a track kink can be included in the fit
- The track parameters are updated accordingly
- Without the KF algorithm, one would have to treat the track kink angles as additional parameters in the fit

Kalman filter in pictures



Smoothing: run backwards to improve precision at the beginning of the track

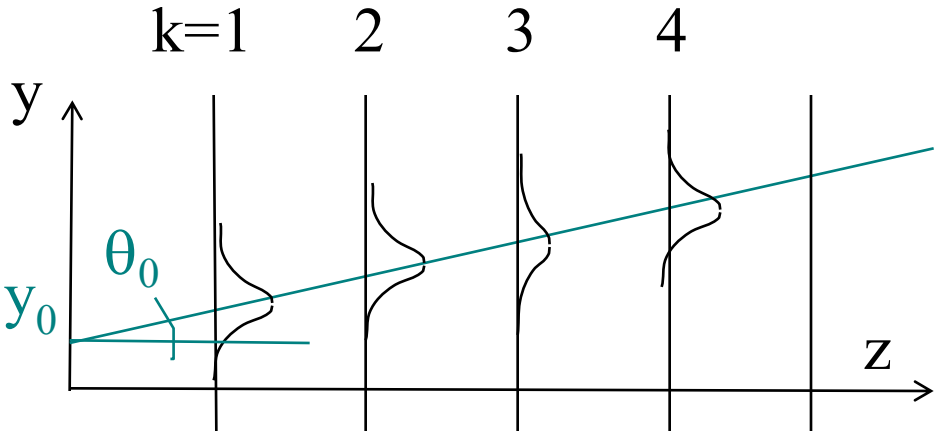
- Remove bad hits included at an early stage
- Get optimal precision at vertex (and all along the track)

Kalman filter properties

- The KF is an iterative, least-squares estimator
- Least-squares estimators are unbiased and efficient, if:
  - The measurements depend linearly on the parameters to fit
    - Not a big issue: tracks are “almost straight”; non-linearity bias is small
  - The measurements have Gaussian uncertainties
    - Mmm, not always the case
  - The track parameters are disturbed by a Gaussian process
    - OK for Multiple Scattering
    - Not true for electrons, they can loose a lot of energy by Bremsstrahlung ⇒ **Gaussian-Sum Filter (GSF)**
- Least-squares estimators are not robust against noise
  - Largely helped by rejecting hits during iterative tracking

Precision

- Due to Multiple Scattering, the precision *does not* keep on improving when including more measurements
- You can experience this with this simple setup:



- 10 measurement layers, each layer is 0.1 X_0 thick
- Spacing between layers = 10 cm
- Gaussian detector resolution, RMS = 20μm

- Simulate muons of 1, 10 and 100 GeV/c with various y_0 and θ_0
- Apply KF to fit (y, θ) at each layer, and check the RMS of the fitted parameters wrt. simulated ones
- At what layer does the precision level off ? How does this change with momentum ?

Track finding

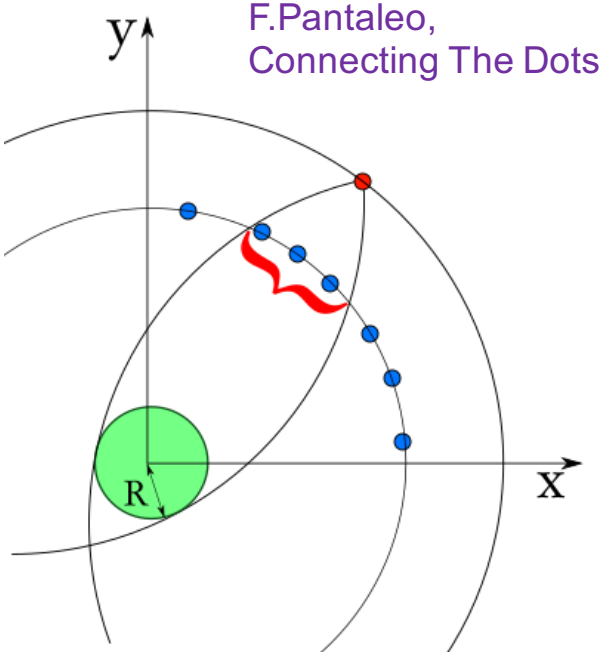
The Combinatorial Kalman Filter

- Since the KF is iterative and can reject incompatible hits, it does a good deal of track finding as well

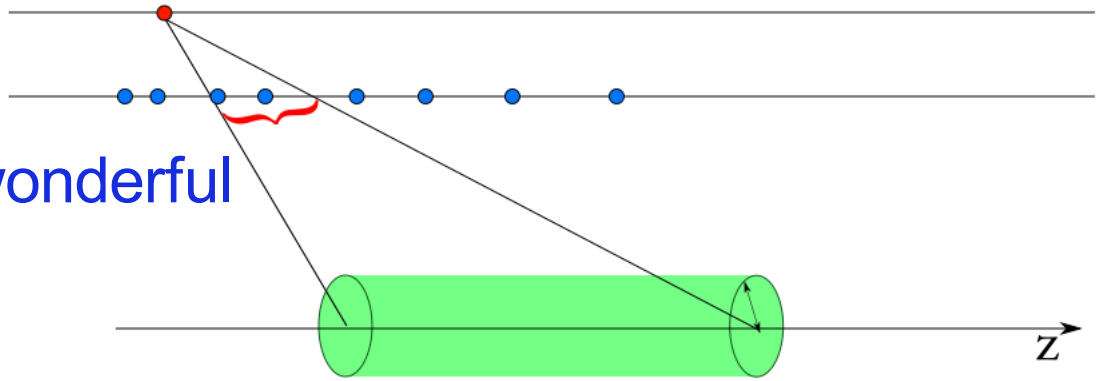
- The KF uses, at every iteration, *all* the information available up to that iteration. One *cannot* be more selective, unless:
  - *The track seed is wrong* (a random hit combination)
    - **Philosophy:** test as few seeds as needed, and abandon the wrong ones as soon as possible
  - *Noise is very high*, and many hits do not belong to the track being followed
    - **Global track finding might help here, together with a sophisticated track fit** e.g. with soft assignment of hits to tracks

Track seeds

F.Pantaleo,
Connecting The Dots 2019



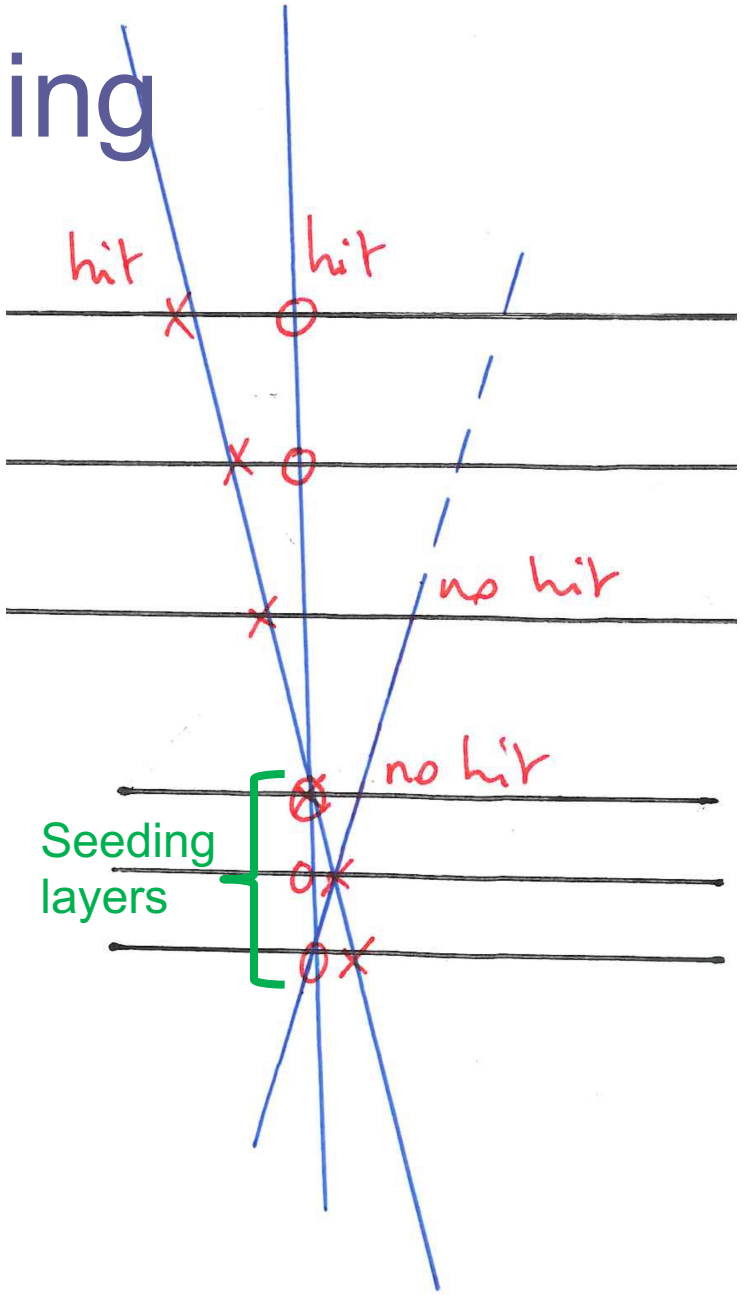
- Seed = hit pair or triplet consistent with the physics you want to do
  - Minimal p_T
  - Maximal impact parameter wrt. collision region
  - …
- Simple hit matching in ϕ and z to save CPU



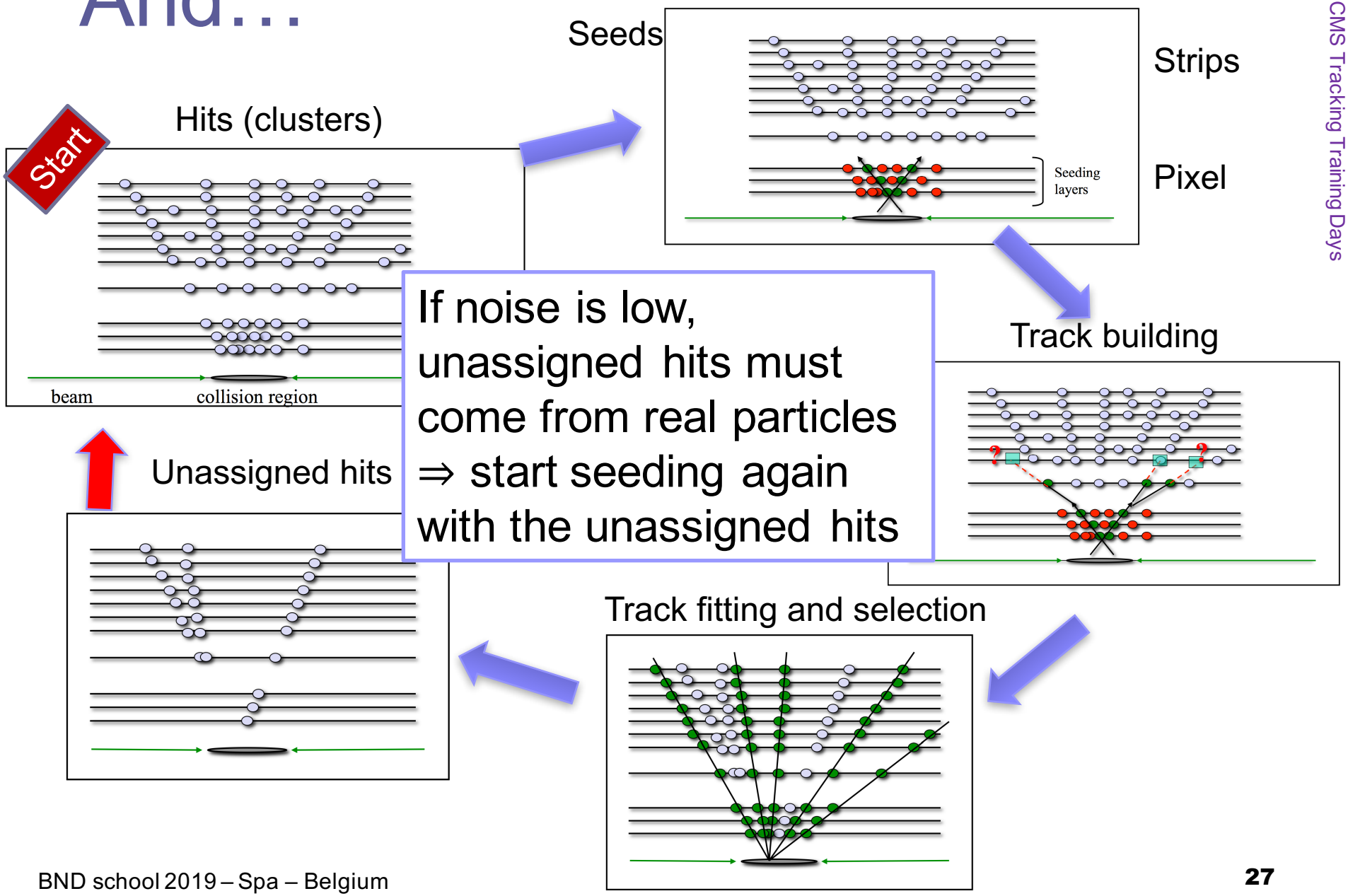
- Pixel detectors are wonderful track seeders
  - Noise-free
  - Precise 3D hits

Trajectory tree building

- At every layer create:
  - 1 trajectory per compatible hit
  - + 1 trajectory where this layer is skipped (even if there is a hit)
    - Avoids bias in case of inefficiency
- i.e., at least 2 trajectories per layer…
  - Must control the growth…
- Stopping conditions
  - In detectors with 95% efficiency, probability of 2 consecutive missing hits is 0.25%
    - Stop
  - If you need to save CPU, you could stop as soon as the asymptotic precision is reached



And…

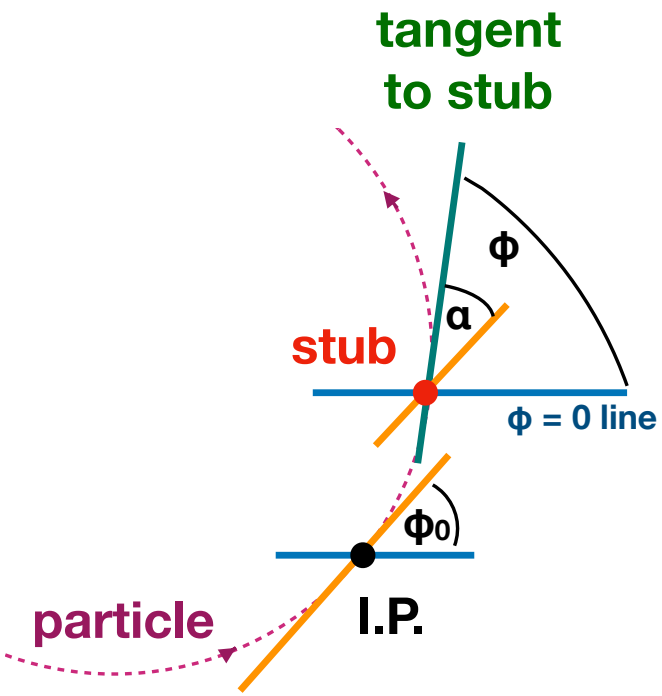


If noise is low, unassigned hits must come from real particles ⇒ start seeding again with the unassigned hits

Slide from CMS Tracking Training Days

Hough transform

- Global method to find parametrically described curves
- For circular tracks originating from the interaction point I.P. in the transverse plane:



T.James, PhD thesis [6]

Every point (r, ϕ) along the circle obeys $\frac{r}{2R} = \sin(\phi - \phi_0)$, where *R* is radius of trajectory.

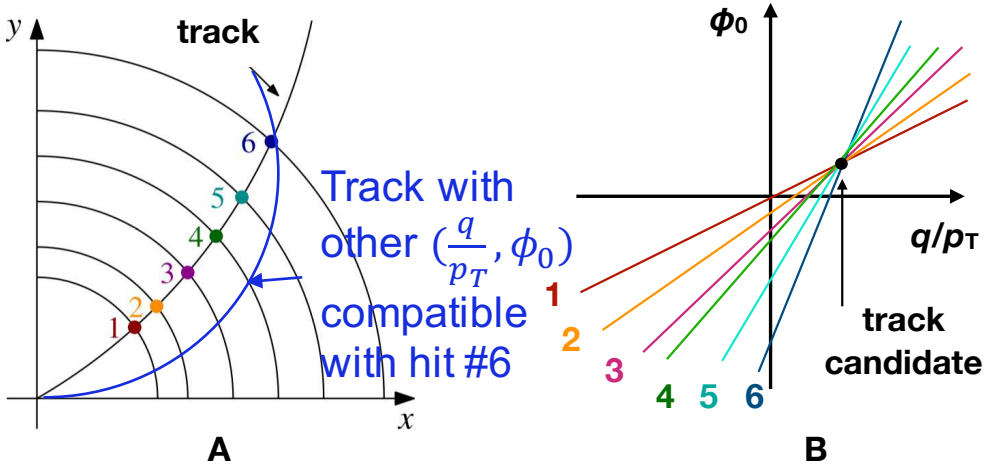
For small curvature: $\frac{r}{2R} \cong \phi - \phi_0$, or:

$$\phi_0 = \phi - \left(\frac{Bcr}{2}\right)\frac{q}{p_T}$$

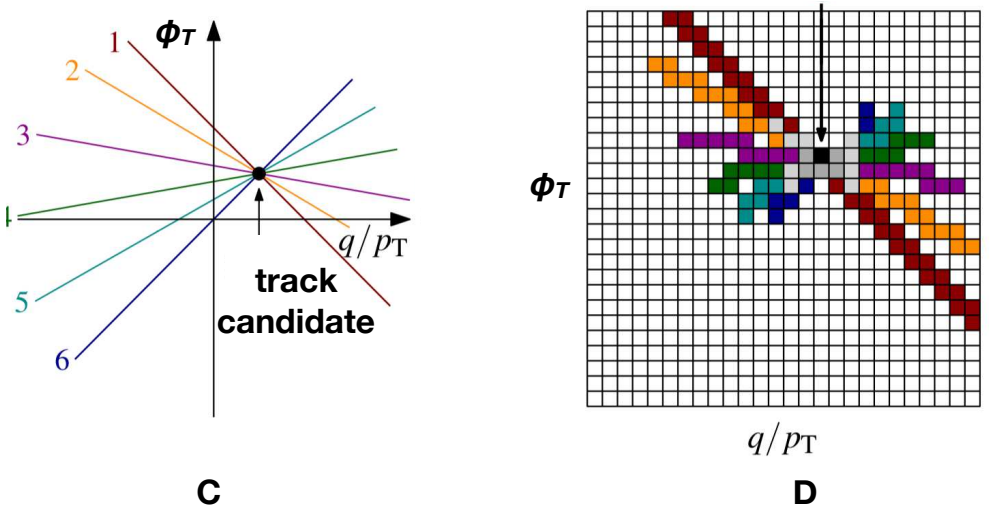
This relation shows that a point (r, ϕ) is consistent with all the trajectories with parameters $\frac{q}{p_T}$ and ϕ_0 obeying that relation:

a hit defines a line in $(\frac{q}{p_T}, \phi_0)$ space.

Hough transform



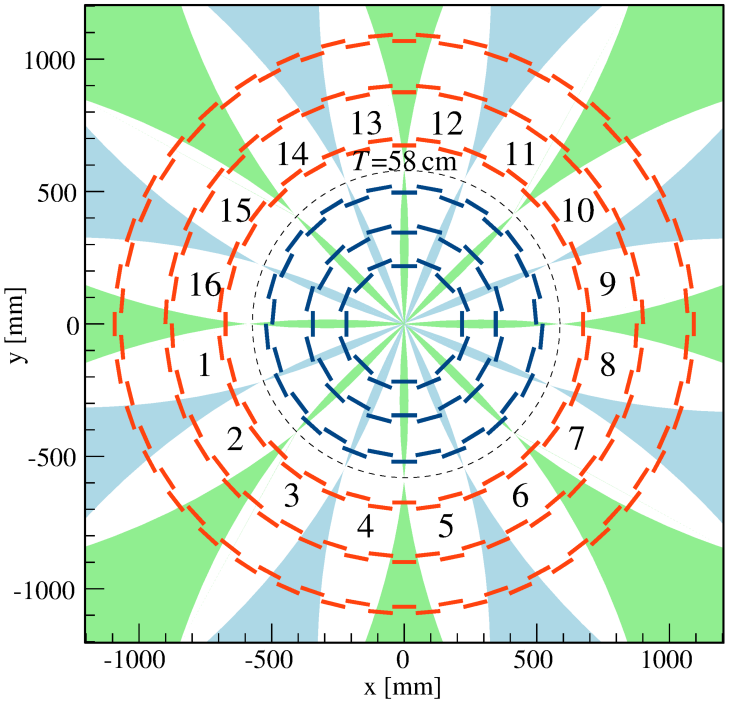
T.James, PhD thesis [6]



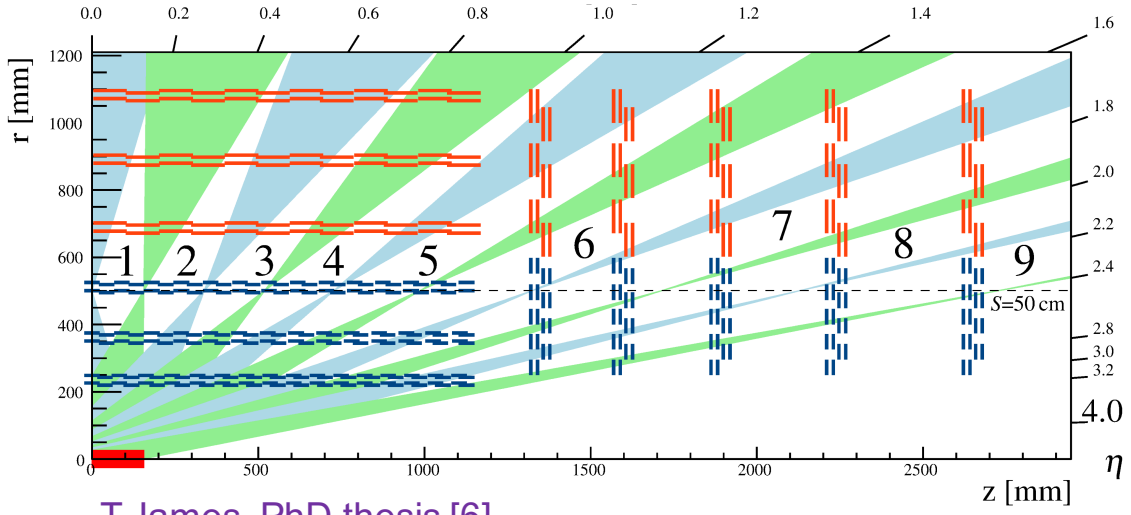
- By crossing the lines from several hits, one gets track candidates
- Track finding boils down to finding peaks in a suitably-binned $(\frac{q}{p_T}, \phi_0)$ space
  - Neat trick: define *r* and ϕ_T with respect to the middle of the tracker radius, so that lines spread wider in $(\frac{q}{p_T}, \phi_T)$ space
- For tracks not originating from (0,0) a simple-minded Hough transform might not be affordable
- Kinked tracks may get lost

Limited time ? Parallelize !!

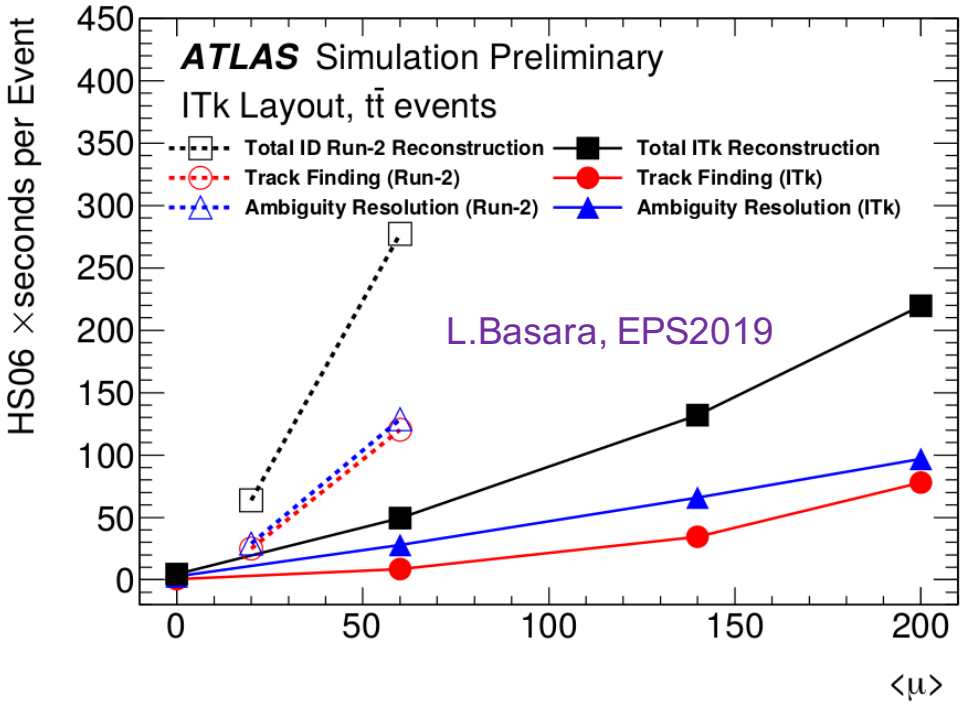
- The detector can "easily" be sliced in sectors in ϕ and η
  - Not so easy if this is done in hardware, but doable
- To avoid inefficiencies at sector boundaries, the sectors must overlap
- There will thus be track duplicates to clean up later (OK)



White areas with numbers show the non-overlapping parts of the sectors; the colored ones are the overlapping parts



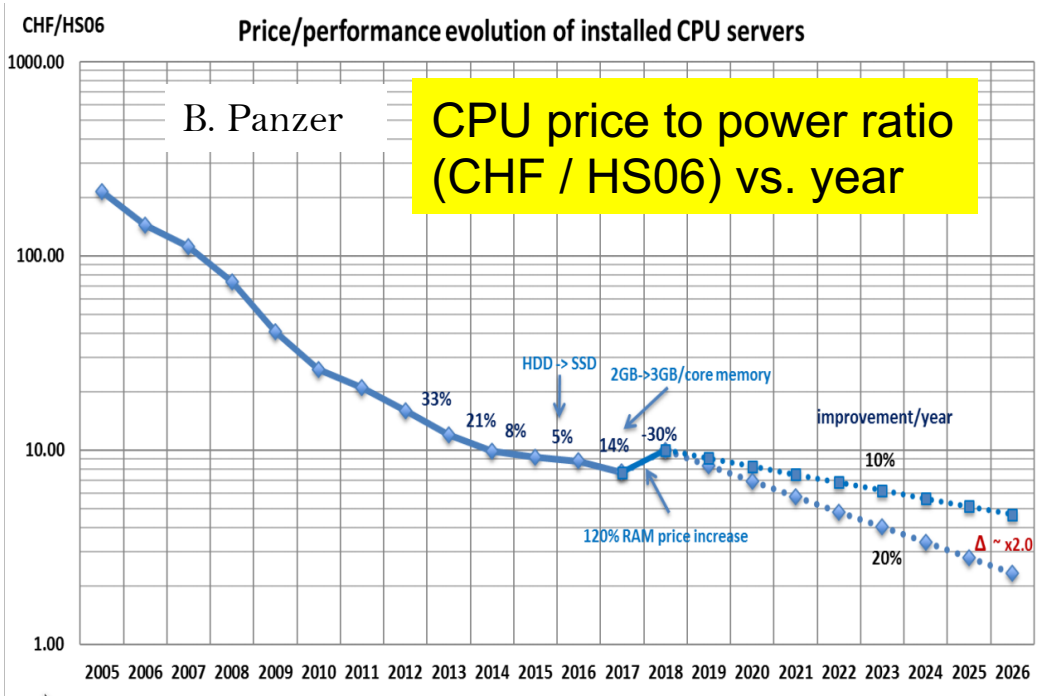
T.James, PhD thesis [6]



CPU crisis

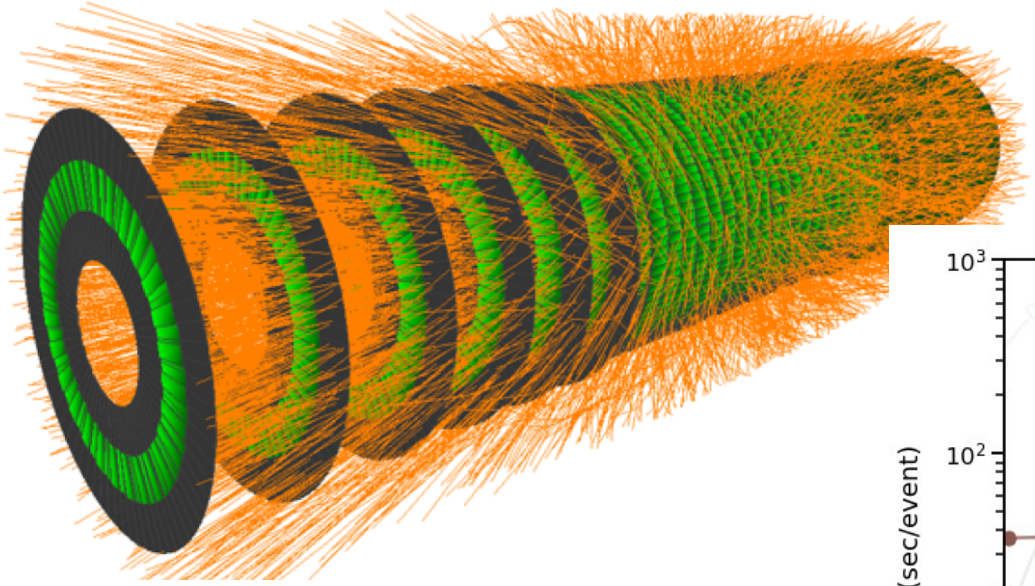
Factor 30 increase in CPU needs at the start of HL-LHC compared to now

- 10-20% improvement per year
- Factor 2-5 increase in performance per € spent ☹
- Try to benefit from growing GPU market…
  - CMS "Patatrack" hackatons (F.Pantaleo et al.)
- Must rewrite and parallelize the reconstruction code

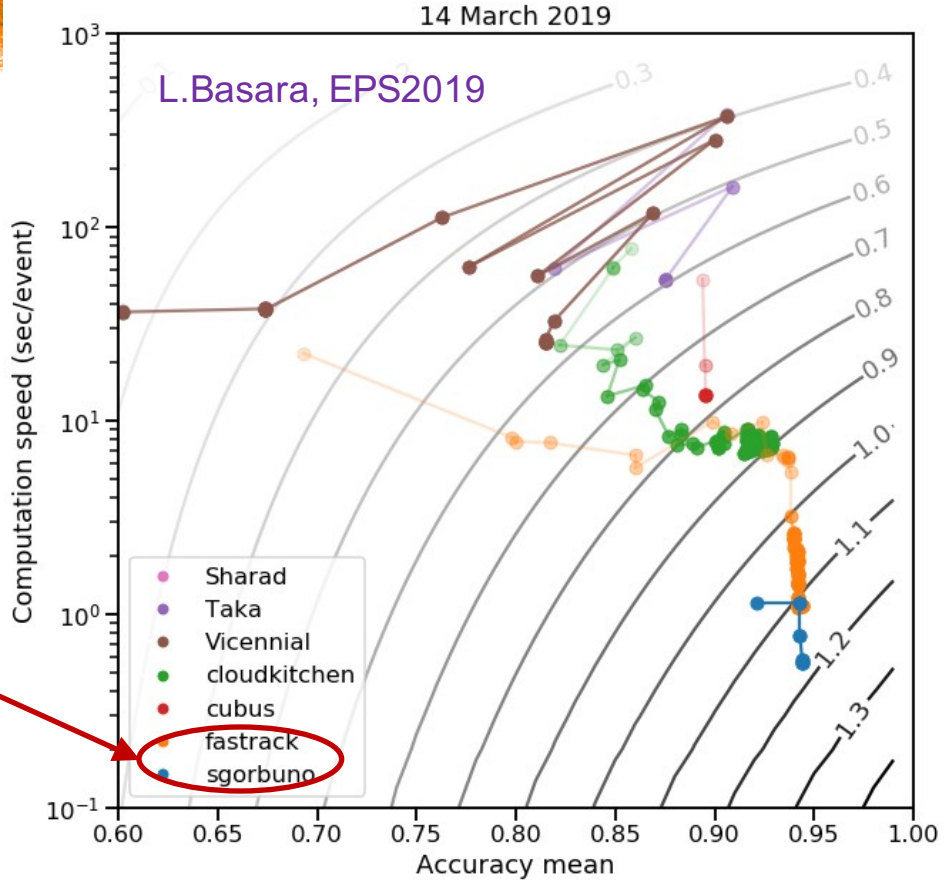


The trackML challenge

- Organized by HEP community to reach out to Machine Learning experts



- complicated detector geometry
- non-uniform magnetic field
- multiple scattering
- high dense events



- Performance metrics: accuracy (hit association efficiency) and CPU per event
- Leaders are: HEP tracking experts, by far !
- Winning approach not even ML but some CKF (gold is in details)

Vertex finding and fitting

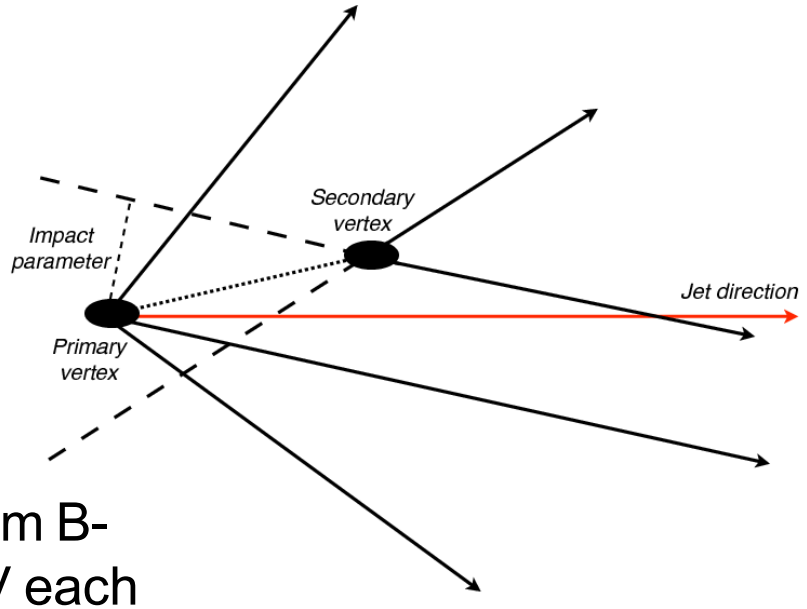
Physics cases

- Primary vertex finding
  - Issue at LHC: separate the hard proton-proton interaction from pile-up interactions
  - Transverse beam spot is very small (~15μm) while collisions are spread in z (~few cm RMS)
    - 1D clustering problem along z
  - Already with pile-up densities of Run 1, a simple gap-finding algorithm with 2 mm gap lead to unacceptable rate of merged primary vertices
- High Level Trigger
  - A fast and coarse primary vertex finding using only the pixel detector may help a lot, in selecting the seeds of tracks coming from the primary vertex

Physics cases

❑ Secondary vertex finding

B-jet tagging



- ❖ b-quark fragmentation is hard:
  - in average 5-6 secondary tracks from B-hadron decay chain, of several GeV each
  - A few additional primary tracks in b-jet
- ❖ There can be a tertiary vertex
- ❖ At some decays vertices, a single charged particle is produced

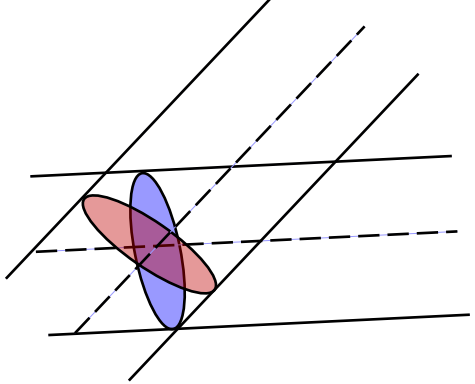
Tagging of boosted H(->bb) boson

- ❖ several secondary vertices to look for in a wide jet

Physics cases

- The association of tracks to vertices is often ambiguous
  - There are only a few tracks per vertex, not well separated from the neighboring vertices
  - All the information available should be used optimally to make the association
    - This is possible by using vertex fitting at the core of vertex finding

Least-squares vertex fit



- The tracks are 2D constraints
- The vertex coordinates can be found by minimizing the sum of squared distances between vertex and tracks:

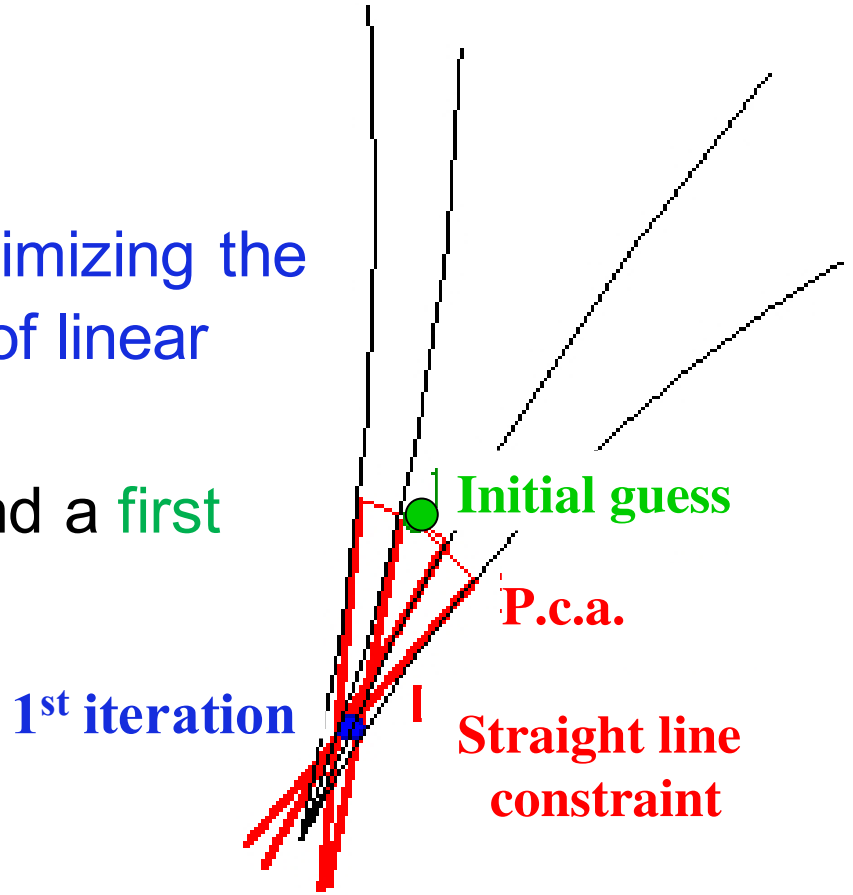
$$\chi^2 = \sum_i^{Ntk} \chi_i^2 = \sum_i^{Ntk} \vec{r}_i^T V_i^{-1} \vec{r}_i$$

where $\vec{r}_i$ is the vector of closest approach between track i and vertex with coordinates (v_x, v_y, v_z), and V_i is the covariance matrix of $\vec{r}_i$. One minimizes for (v_x, v_y, v_z).

- In its full implementation, the fit also improves the track parameters by imposing a common vertex to all tracks
  - see P.Billoir and S.Qian [7]
- Efficient computation relies on linearization

Linearization

- By linearizing $\vec{r}_i(v_x, v_y, v_z)$, minimizing the χ^2 becomes solving a system of linear equations
- The tracks are linearized around a first guess of the vertex position
- The matrix algebra brings the vertex candidate where the straight lines "meet"



- Iterate if needed: re-linearize around the fitted vertex, etc.
- A good first guess can save a lot of CPU
  - Can be found without full calculation of vertex covariance etc.
  - E.g. compute analytically the coordinates of points of closest approach of several track pairs modelled as perfect helices, then find a mode in these (or a good robust average)

Adaptive vertex fit – see Refs [3,4]

- Reweighted least-squares fit: one minimizes iteratively:

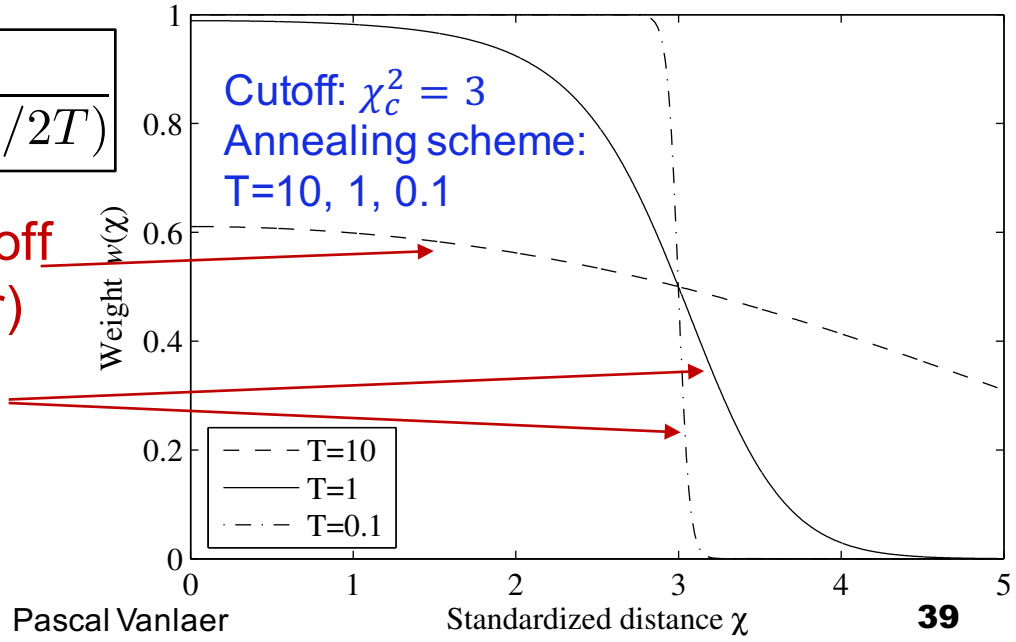
$$\chi^2 = \sum_i^{Ntk} w_i(\chi_i^2).\chi_i^2$$

where the weighting by $w_i(\chi_i^2)$ down-weights distant tracks

- This is called soft assignment: the weight can be viewed as a probability of having track i originating from the vertex
- To avoid local minima, the weights are "annealed":

$$w_i\left(\chi_i^2\right) = \frac{\exp(-\chi_i^2/2T)}{\exp(-\chi_i^2/2T) + \exp(-\chi_c^2/2T)}$$

- First start with a smooth cutoff (high Temperature parameter)
- Then gradually make the cutoff sharper by lowering T ("freeze in")



Adaptive vertex fit – see Refs [3,4]

Illustration: convergence to global minimum

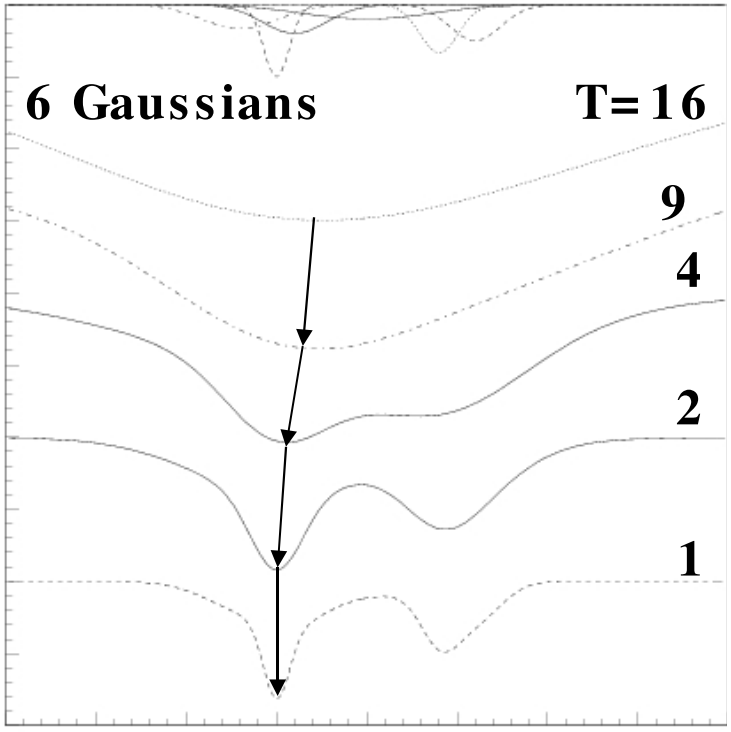
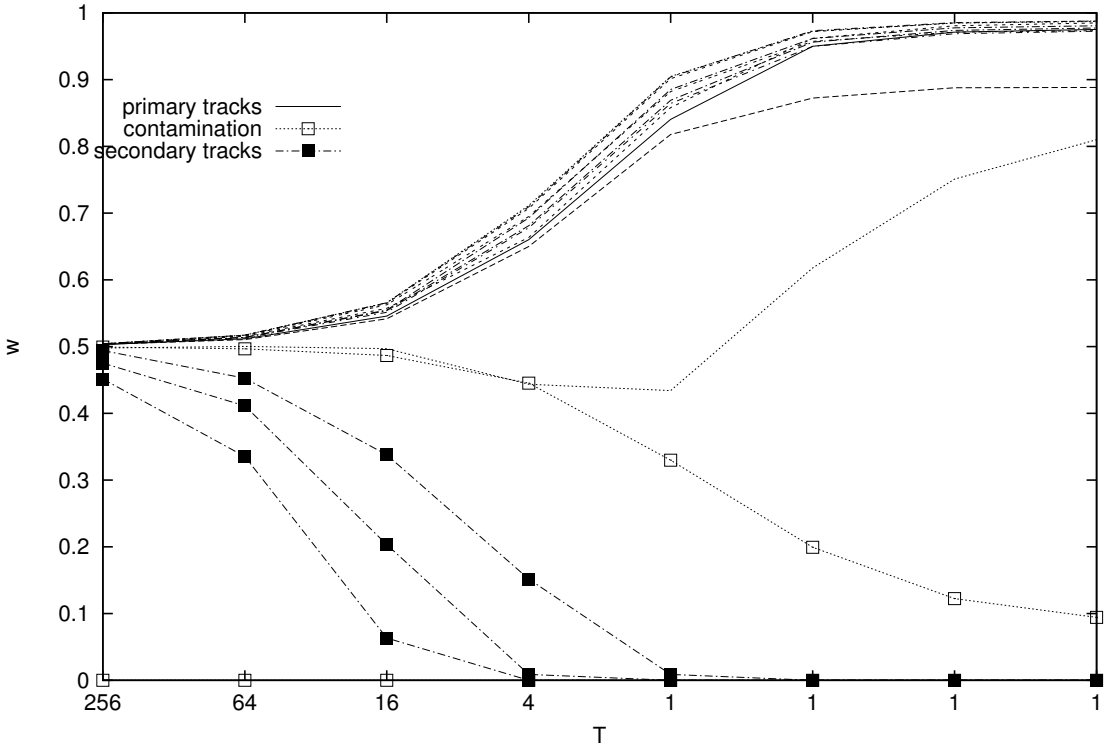


Illustration: weights vs. T
fit of primary vertex in c-jet events
Lines with squares: secondary tracks



Vertex finding – what's special ?

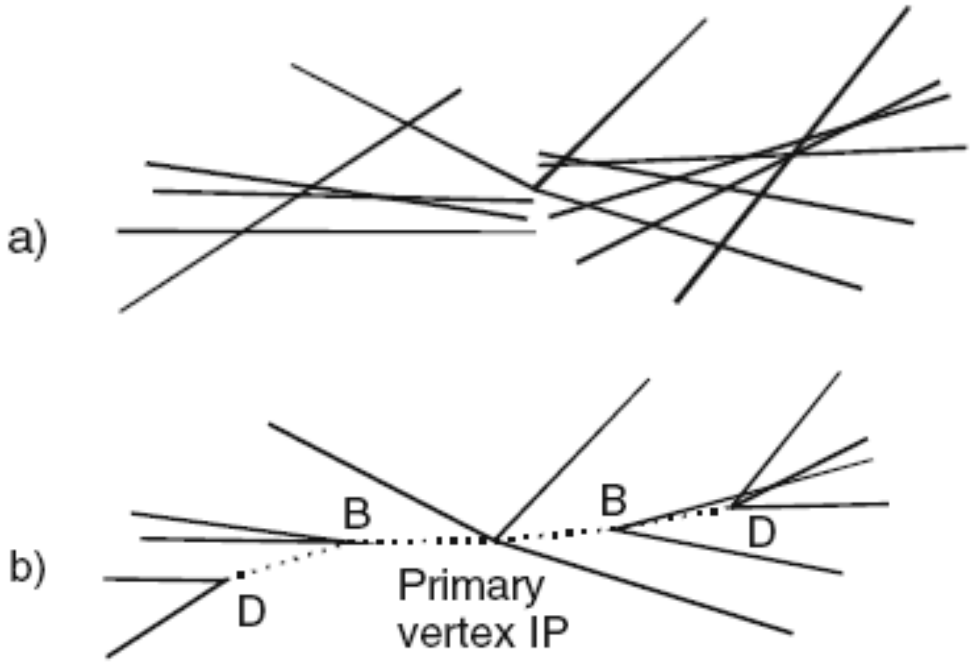
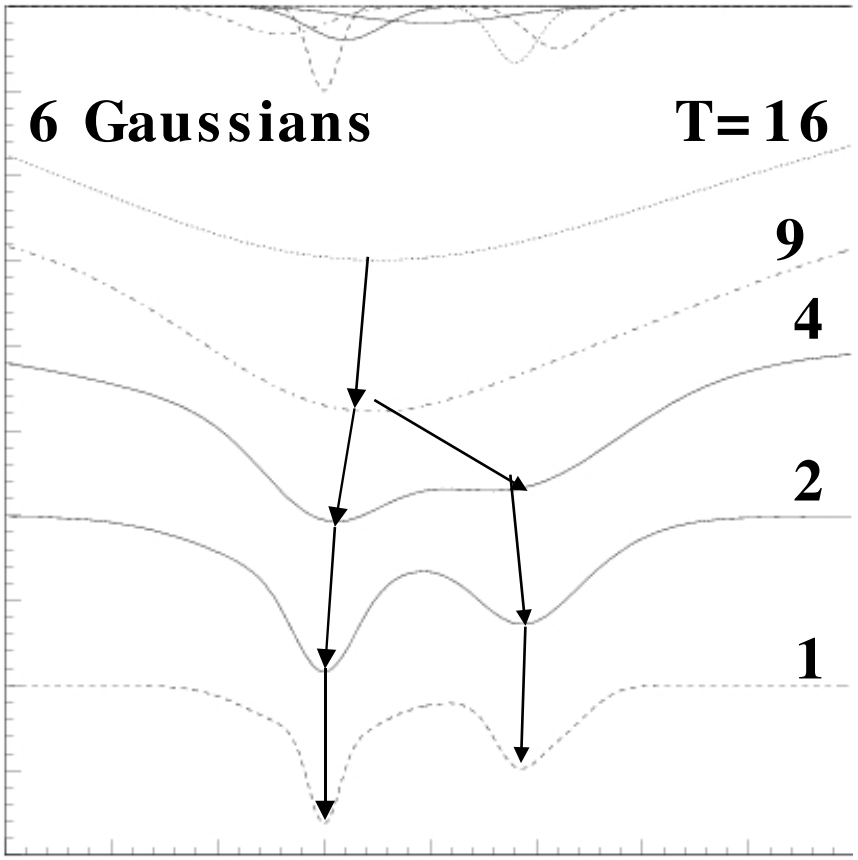


Fig. 3.15 A $b\bar{b}$ event, tracks and physics reconstruction. Two B mes
vertex are each decaying into a D meson, which shortly after decay

- The tracks are extended objects
  - track A close to track B
  - track B close to track C
  - $\not\Rightarrow$ A close to C!!
- If the track multiplicity is not too high, you can always try all the possible combinations…

Deterministic annealing - see [4]

- …or you can exchange this (discrete) combinatorial problem for a (continuous) minimization problem: the Deterministic Annealing (DA) vertex finder
- In this algorithm, a track is allowed to contribute to several vertex candidates
- DA implements vertex splitting during annealing
- When a saddle point appears in χ^2 function, two prototypes are generated, with a small separation, and they move apart as T cools down



Tracking and vertexing

References:

[1] R.Fruhwirth, Nucl. Instr. and Meth. A262 (1987) 444

[2] R.Fruhwirth, M.Regler, R.K.Bock, H.Grote, D.Notz, “Data Analysis Techniques for High-Energy Physics”, Cambridge University Press, 2000

[3] R.Frühwirth, P.Vanlaer, W.Waltenberger, J.Phys. G34 (2007) N343

[4] The CMS Collaboration, “Description and performance of track and primary-vertex reconstruction with the CMS tracker”, 2014 *JINST* 9 P10009

[5] Connecting The Dots (CTD) workshops

[6] T.James, “A hardware track-trigger for CMS at the High Luminosity LHC”, PhD thesis, Imperial College London, 2018.

[7] P.Billoir and S.Quian, “Fast vertex fitting with a local parametrization of tracks”, Nucl. Instrum. Meth. A319(1992)139.

Thank you for your attention !

backup