Part 4

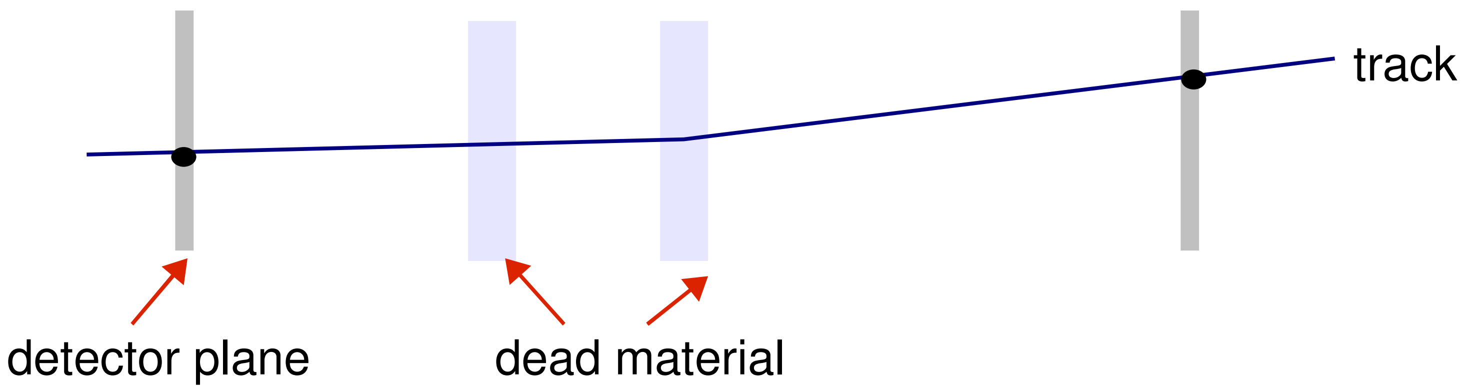
fitting with energy loss and multiple scattering

non-gaussian uncertainties

outliers

material intersections

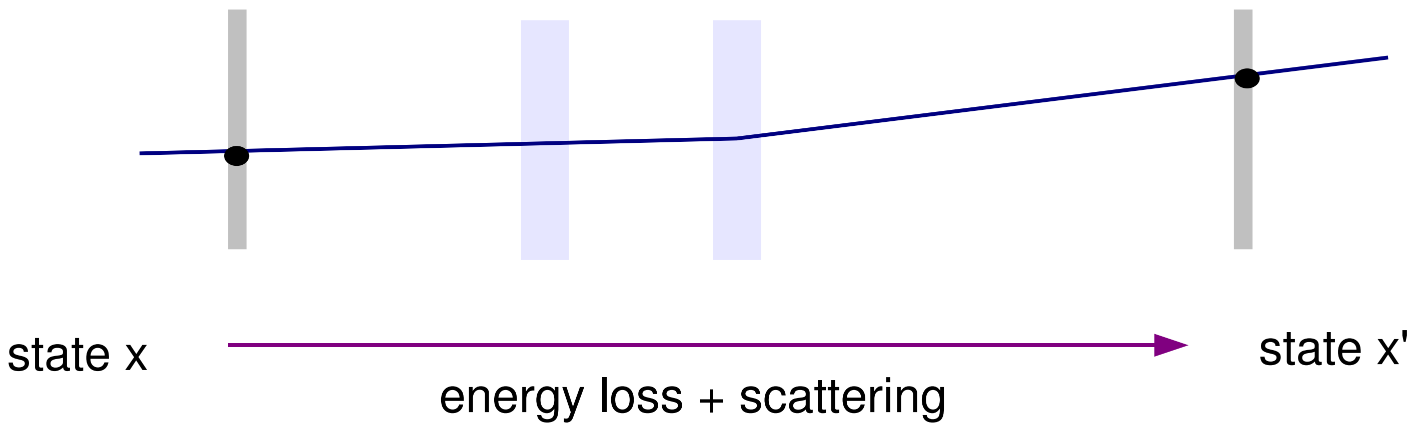
- to treat material effects in track fit, locate material 'intersections' along particle trajectory



- for each intersection use known material properties to obtain
  - average energy loss (from Bethe-Bloch formula)
  - RMS of scattering angle distribution (from Moliere formula)
- to understand how to deal with that, we introduce 'state propagation'

state propagation

- parameter vector of track ('state') changes when traversing material



- ***x'*** is function of ***x***: ***x' = f(x)***
  - function takes into account everything happening in between
  - we call this 'propagation' or 'transportation'
- for example, accounting for energy loss $p' = p - \Delta E$
- note that it matters in which direction the track goes!

state propagation (II)

- function f(x) can eventually be absorbed in measurement model

$$\text{residual} \;=\; m_i - h_i(x)$$

- however, it is more common to redefine track parameters such that they become function of position along track

$$\text{residual} \;=\; m_i - h_i(x_i) \qquad \text{with} \qquad x_i = f_i(x)$$

- e.g. for our toy track fit

$$\begin{pmatrix} x \\ t \\ \omega \end{pmatrix}_i = \begin{pmatrix} x_0 \\ t_0 \\ \omega \end{pmatrix} + z_i \begin{pmatrix} t_0 \\ 2\omega \\ 0 \end{pmatrix}$$

- once you have energy loss corrections, or a non-homogeneous field, this function becomes less trivial

energy loss and multiple scattering in the fit

- energy loss along the trajectory of the particle is usually treated 'deterministically'
  - simply change the momentum by average expected energy loss
  - ignore uncertainty on correction: usually small compared to momentum resolution
  - it's all inside propagation ***f(x)***

- multiple scattering can only be treated 'stochastically'
  - we do not know the scattering angles, just the variance
  - *scattering angles* become *parameters* in track model
  - it's still inside ***f(x)***, but ***x*** now includes scattering angles

multiple scattering in a global track fit

- for each scattering plane, add *two scattering angles* to track model
- add new contribution to chi-square

normal contribution for measured points

new contribution for scattering angles

$$\chi^2 = \sum_{\text{hits}\, i} \left(\frac{m_i - h_i(x, \theta^{\text{scat}})}{\sigma_i} \right)^2 + \sum_{\text{angles}\, j} \left(\frac{E(\theta^{\text{scat}}) - \theta_j^{\text{scat}}}{\sigma_j^{\text{scat}}} \right)^2$$

new parameters in track model

expectation value for scattering angles **$E(\theta^{\text{scat}}) = 0$**

uncertainty in scattering angles, from Molière formula

- now minimize the chi-square as before, both to $\boldsymbol{x}$ and to $\boldsymbol{\theta}$
- problem: many parameters --> time consuming matrix inversion

multiple scattering in a progressive track fit

- in the Kalman filter we account for multiple scattering by introducing so-called 'process noise'
- state propagation in kalman track fit

$$x_k^{k-1} = f_k^{k-1}(x_{k-1})$$

prediction of state at hit k

function that expresses state at hit k in terms of state after hit k-1

state after processing k-1 hits

- the function ***f*** propagates the track parameter vector from one hit to next
- it takes care of magnetic field integration and energy loss correction
- yesterday I ignored it: $x_k^{k-1} = x_{k-1}$

kalman filter with process noise

- with transportation covariance matrix of prediction is

$$C_k^{k-1} = F_k^{k-1} C_{k-1} {F_k^{k-1}}^T + Q_k$$

derivative of ***f(x)***

variance of $\boldsymbol{x}_{k-1}$

new: process noise

- the matrix Q accounts for 'random noise' in the extrapolation from state 'k-1' to state 'k'
- this is where we insert multiple scattering contributions
- for example, traversing through 300 micron of silicon we would have

$$Q(t_x, t_x) = Q(t_y, t_y) = \left(\frac{0.7 \text{ mrad}}{p/\text{GeV}} \right)^2$$

kalman filter with process noise (II)

- taking into account small correlations, the full formula become

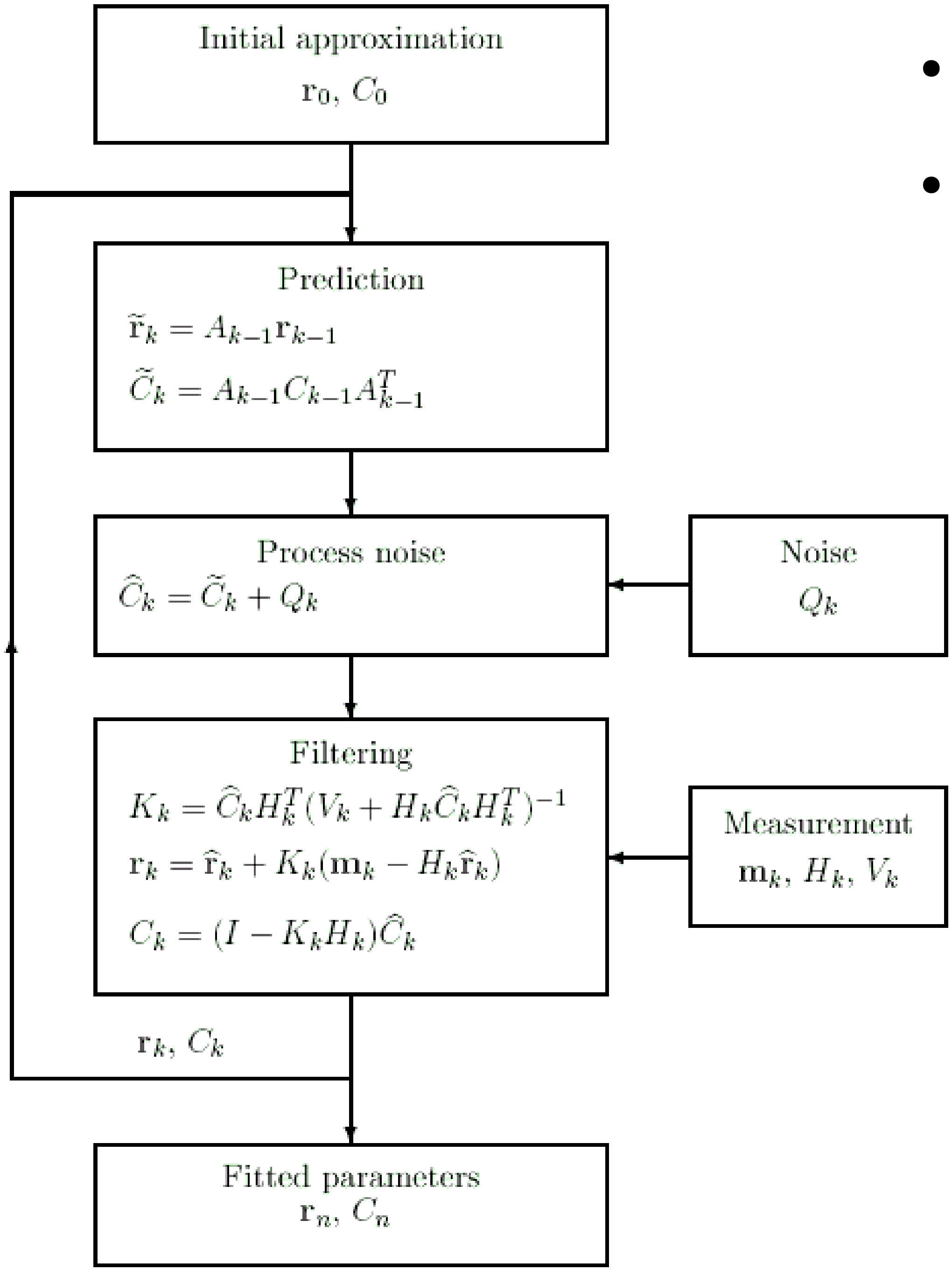
$$\begin{aligned} Q(t_x, t_x) &= \theta_0^2 \left(1 + t_x^2 t_y^2\right) \left(1 + t_x^2\right) \\ Q(t_y, t_y) &= \theta_0^2 \left(1 + t_x^2 t_y^2\right) \left(1 + t_y^2\right) \\ Q(t_x, t_y) &= \theta_0^2 \left(1 + t_x^2 t_y^2\right) t_x t_y \end{aligned}$$

- if the finite thickness of the scatterer cannot be neglected, there are also contributions to the position coordinates
- in the helix parameterization this looks of course different, but the ingredients are the same
- in terms of these new 'prediction' variables, the Kalman is

$$\begin{aligned} K_k &= C_k^{k-1} H_k^T (V_k + H_k C_k^{k-1} H_k^T)^{-1} \\ x_k &= x_k^{k-1} + K_k \left(m_k - h_k(x_k^{k-1}) \right) \\ C_k &= (1 - K_k H_k)\, C_k^{k-1} \end{aligned}$$

(same as before, just substitute new symbols)

graphical representation of Kalman filter

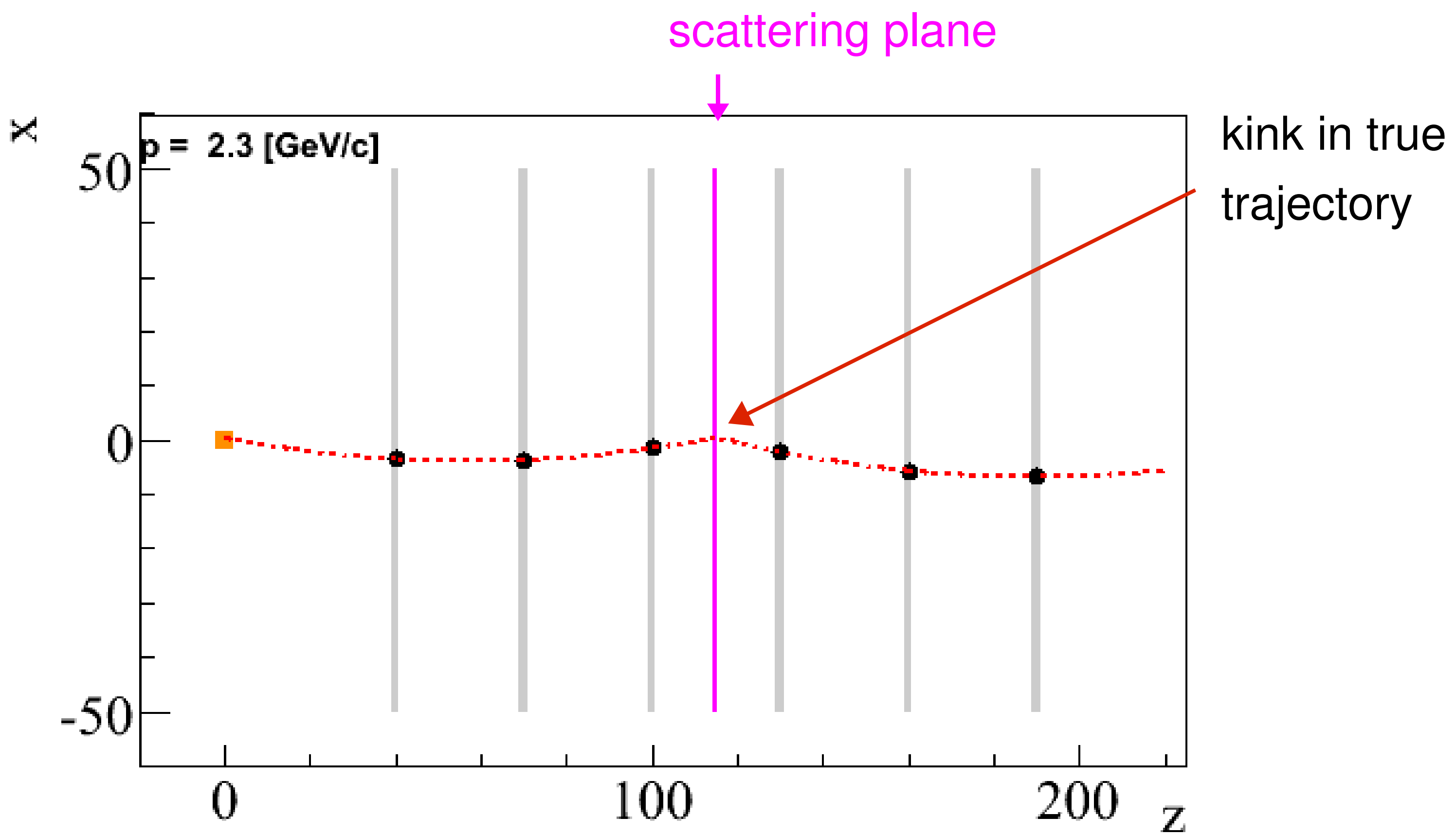


- borrowed from CBM-SOFT-note-2007-001
- note: slightly different notation
  - r → x
  - A → F

- we call the points at which states are evaluated 'nodes' (labeled by ***k***)
- sometimes it is useful to split 'measurement' and 'noise' in separate nodes
- I'll do that in the toy track fit

adding multiple scattering in our toy track fit

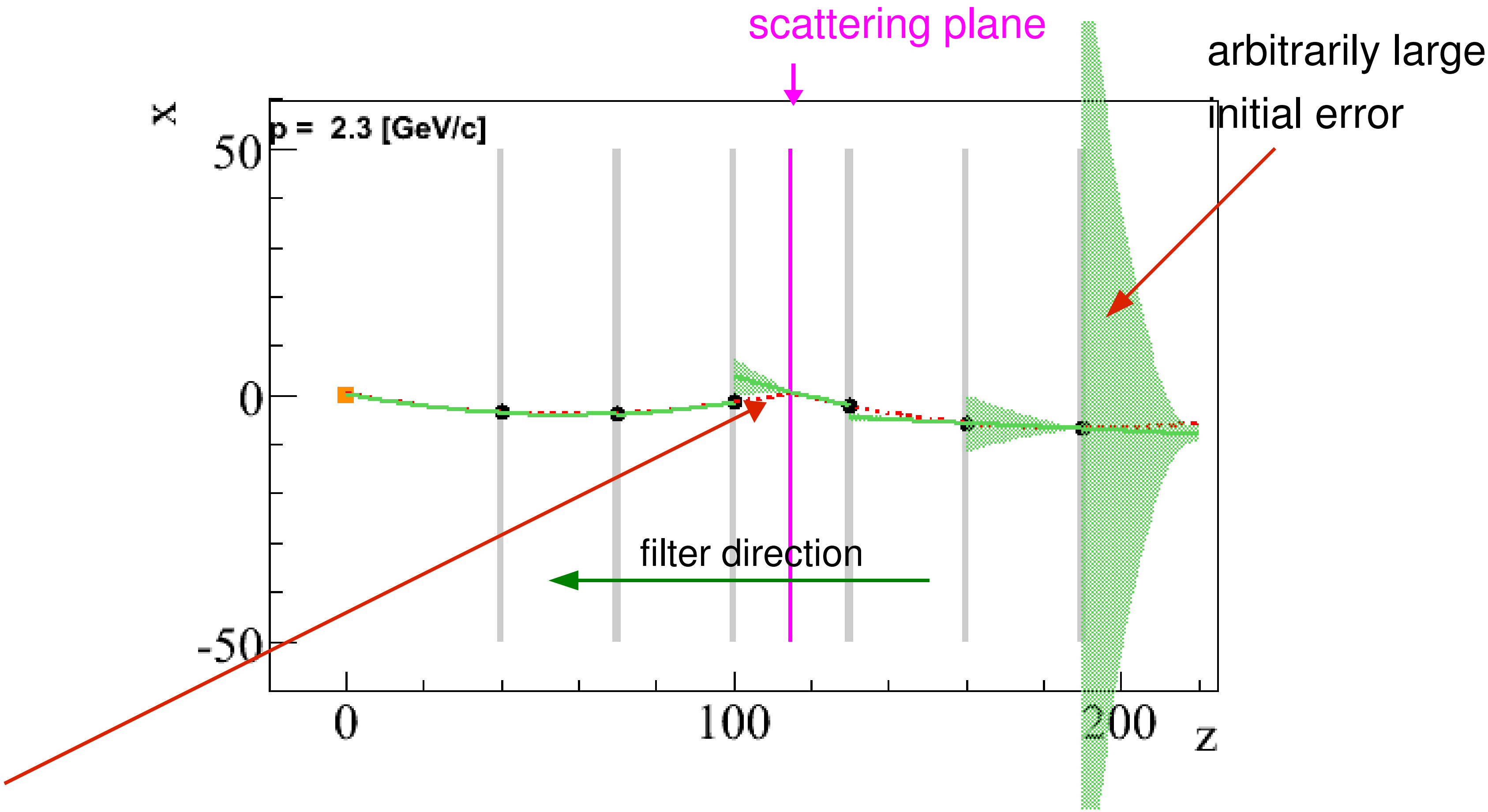
- introduce a single scattering plane



- very thick plane: equivalent to 1cm iron ($x/X_0 = 0.6$)
- to emphasize its importance, reduced detector resolution to 1mm

adding multiple scattering in our toy track fit

- evolution of track state in the Kalman filter



- note how uncertainty in predicted state blows up exactly at the scattering plane: that is the noise contribution
- as a result, measurement before scattering plane do not really contribute

comparison of two 'scattering' approaches

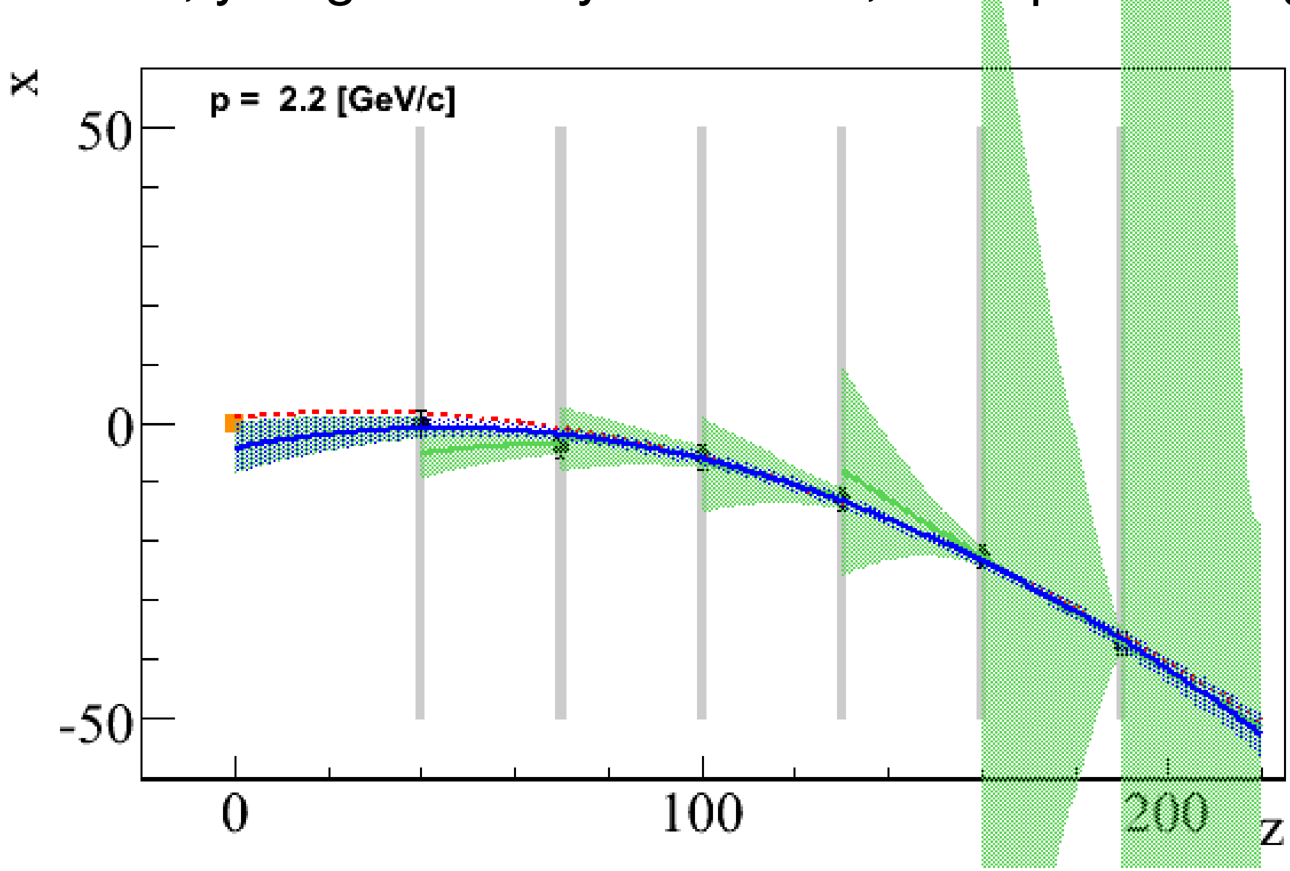
- we could do the same in the global fit, but that's too much work now
- once more, difference between two approaches
  - global fit: explicit parameterization in terms of angles. matrix to be inverted has size “5 + 2 x number of planes”
  - kalman fit: implicit parameterization by allowing state vector to change along track. only 1-dimensional matrix inversions.

treatment of material effects in KF is easier and faster

- contrary to common belief, *track model* does not necessarily differ in the two cases
  - if you would 'draw' the track, you would get the same result
  - scattering angles in the kalman fit do not appear explicitly, but can be calculated from the difference of states on neighbouring nodes

back propagation

- global fit leads to single set of parameters that describe track state everywhere along trajectory
- in the Kalman fit, you get exactly that state, after processing all hits



kalman filter

global fit

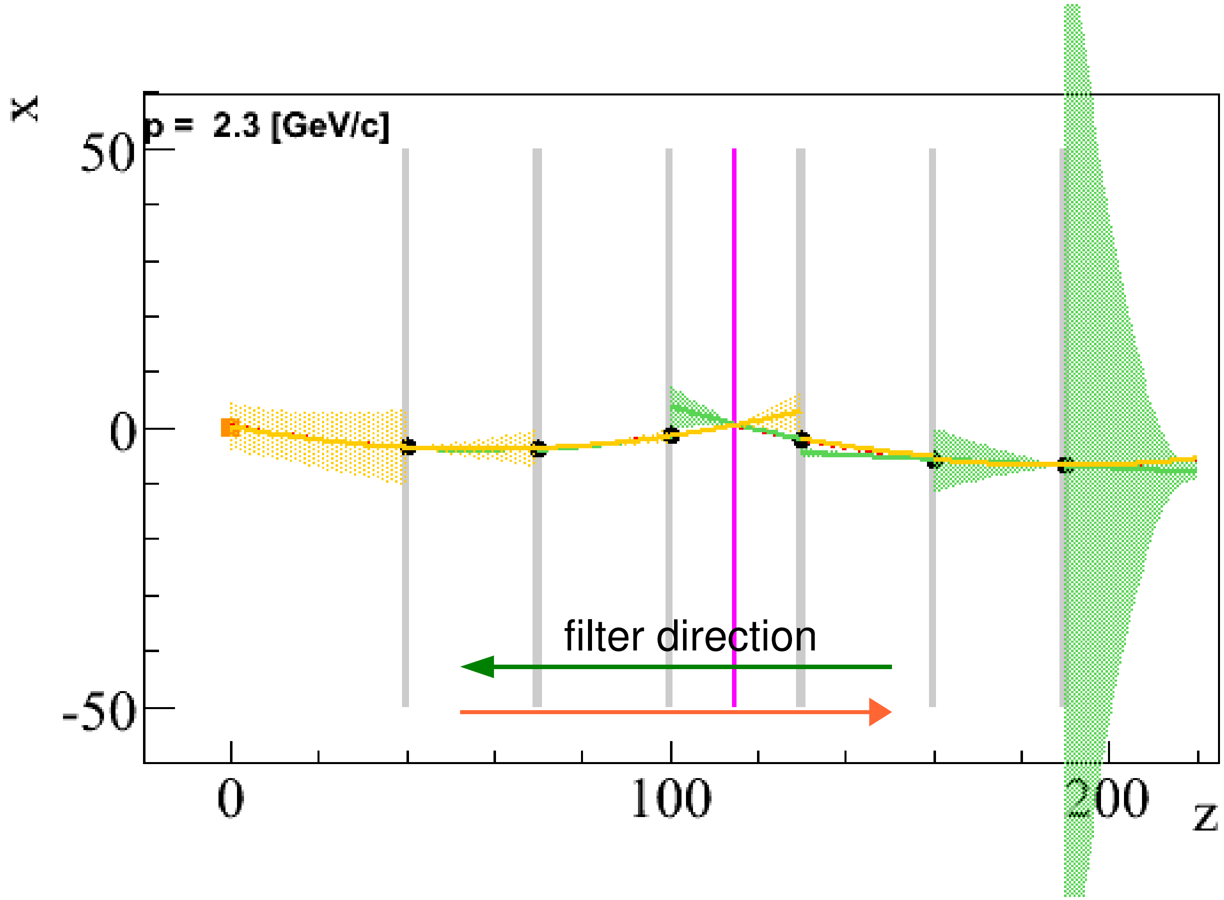
- if there is no scattering, you can propagate the 'final' state back to everywhere along track

track state smoothing

- in the global fit scattering angles explicit: still know position everywhere
- in Kalman fit, scattering information is not kept
  - you need to do extra work to propagate final state 'backward'
  - this is called track state 'smoothing'
- there are two common approaches
  - use so-called 'smoothing-matrix' formalism (see Fruhwirth)
  - run another Kalman filter in opposite direction and perform weighted average on each node
- latter procedure is more popular now, because
  - math is simpler (not a very good reason)
  - it is more stable (a good reason)
  - it is more economic if you don't need smoothed state on every node

reverse filter: Kalman filter in opposite direction

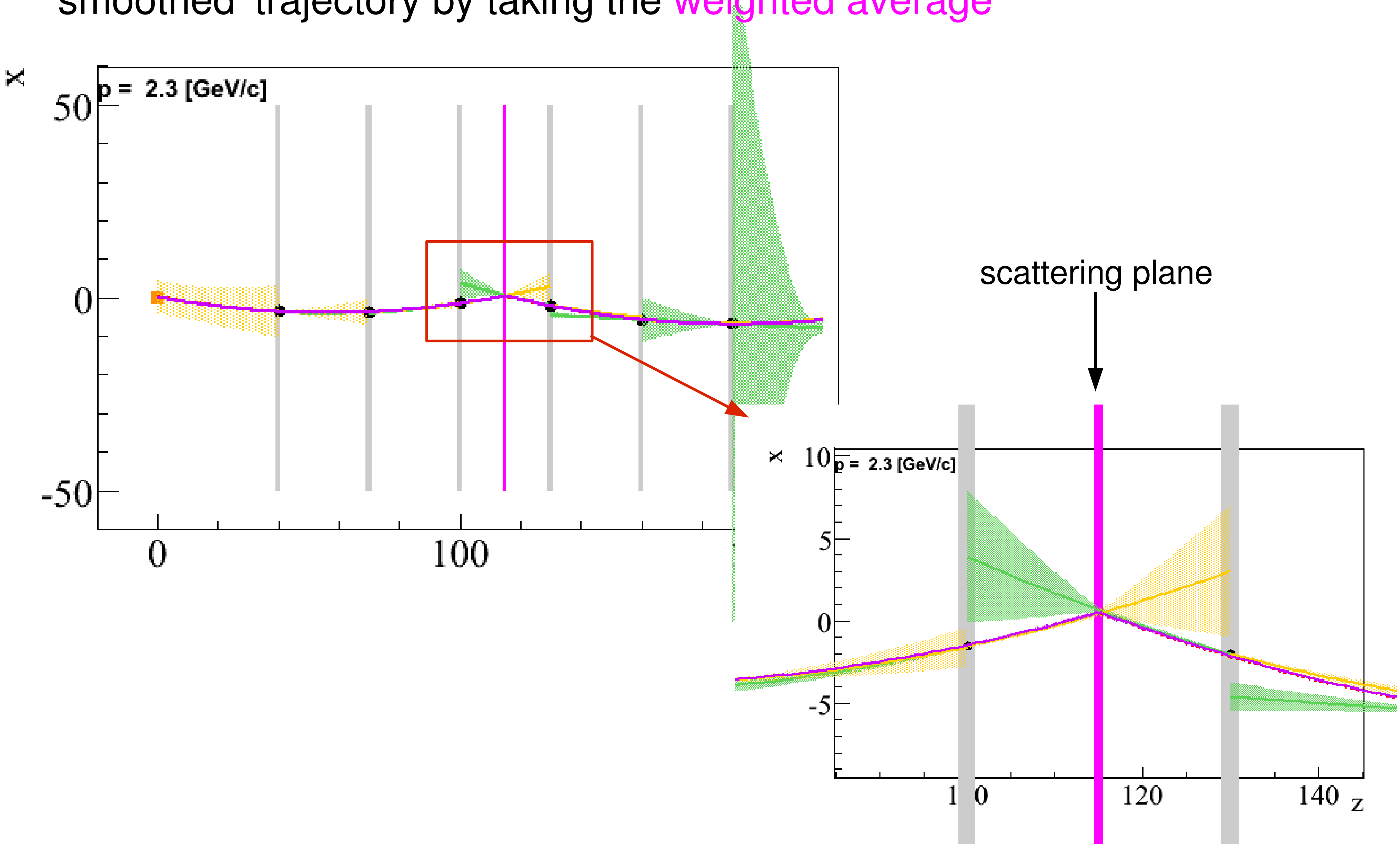
- in orange the result from the reverse filter (which runs from left-to-right)



- to initialize the reverse filter, we can use the result from the forward fit
- need to blow up the uncertainty by a large factor, for example 1000

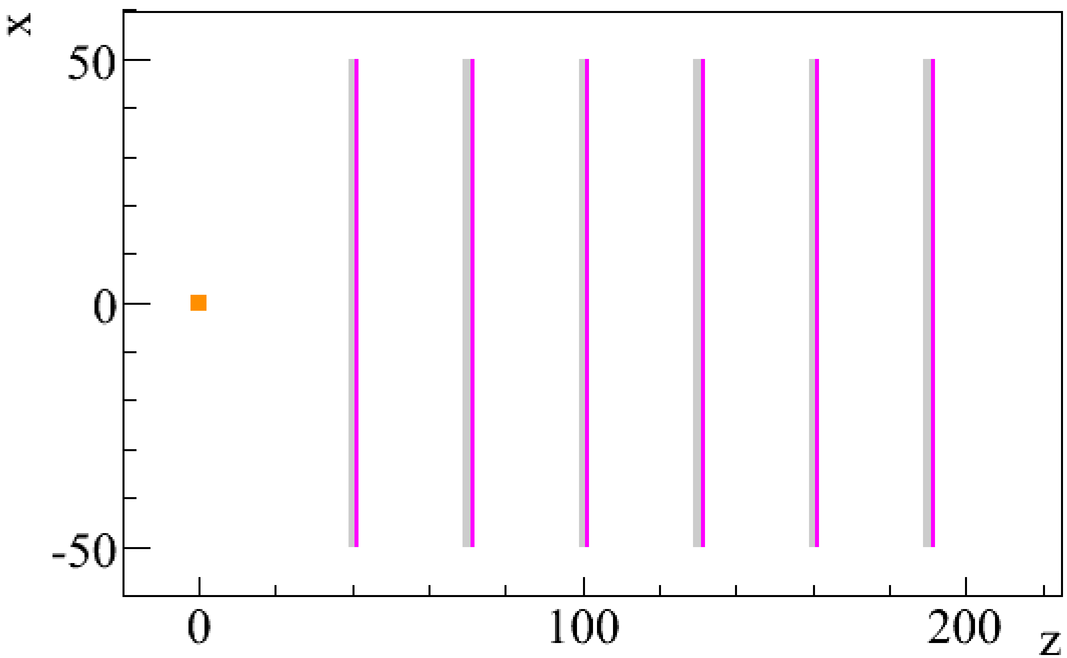
smoothing with a weighted mean

- once we have the result in both directions, we can compute the 'smoothed' trajectory by taking the weighted average

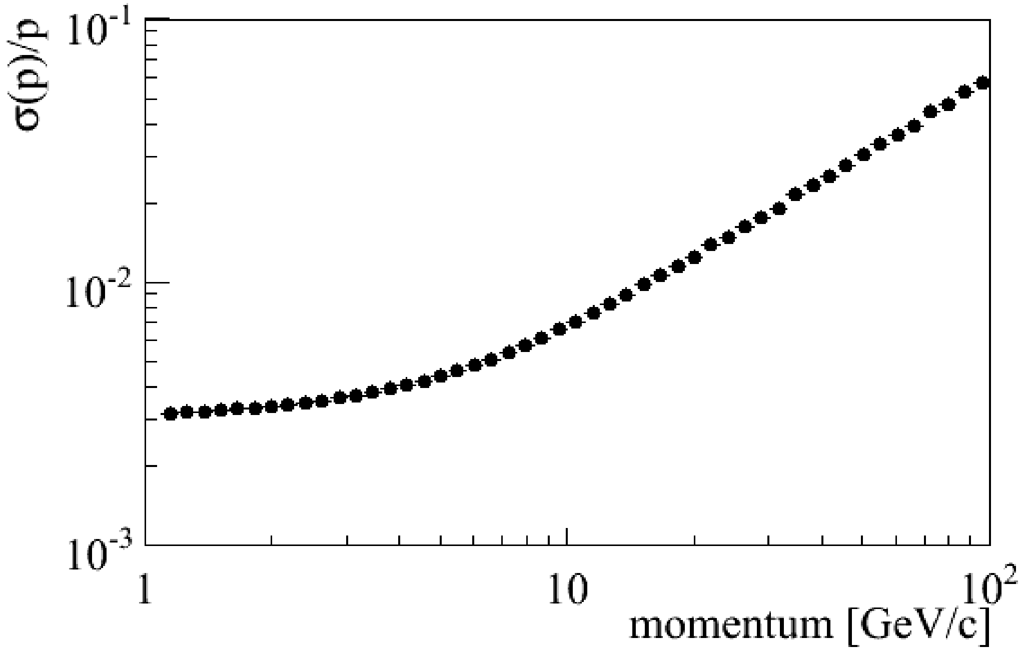


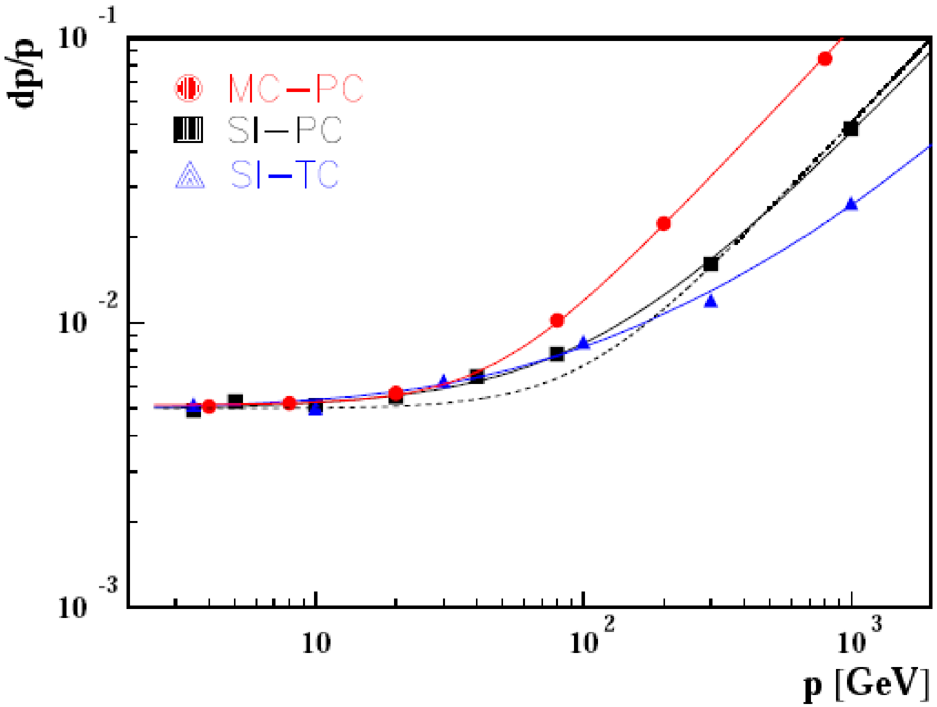
momentum resolution with multiple scattering

- we have seen: without multiple scattering $\sigma(p)/p \sim p$
- to see what multiple scattering does, we use the toy track fit
- we add just a bit more realism
  - hit resolution 100 micron (an excellent drift chamber)
  - scattering x/X_0 = 0.01 per layer (not untypical for forward detectors)



momentum resolution with multiple scattering





- in the low momentum limit resolution is scattering dominated: $\boldsymbol{\sigma(p)/p \sim}$ ***constant***
- in the high momentum limit resolution is hit-resolution dominated: $\boldsymbol{\sigma(p)/p \sim p}$
- this is the same plot for the hera-B spectrometer, which has
  - more field integral
  - a much longer arm

$\Rightarrow$ better resolution at high momentum

toward a real track fit

- we now have almost all ingredients to state-of-the art track fitting
  - track models for two types of detectors
  - two track fitting procedures, both with 'material' corrections
- the missing ingredients are really detector specific
  - measurement model h(x): depends on geometry, 'strips' or wires, etc.
  - the field integration
- to finish this, we'll briefly touch on two related subjects
  - how to deal with non-Gaussian errors
  - how to deal with outliers

non-gaussian error PDFs

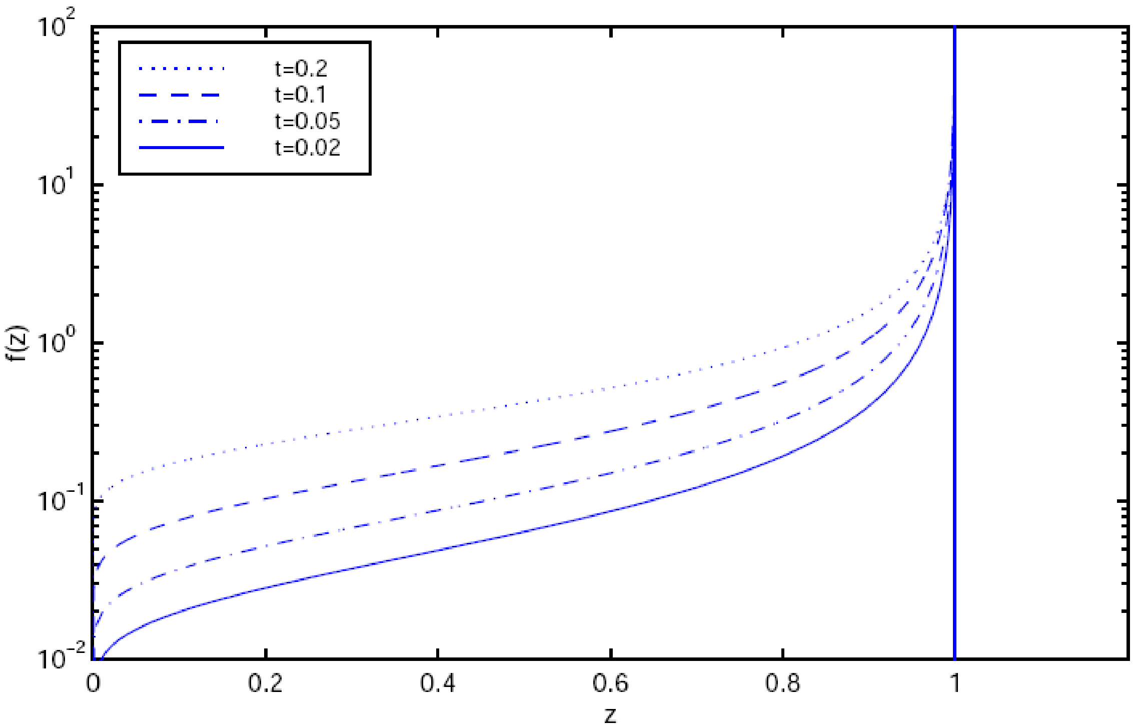
- as we have seen, for linear models and data with Gaussian errors, extracted parameters have Gaussian errors
- in real life, things are not entirely Gaussian
  - hit errors might have tails due to noise, overlapping events, ...
  - Moliere scattering has larger tails than Gauss
  - ionization energy loss follows Landau distribution
  - electron energy loss (bremstrahlung) follows Bethe-Heitler distribution
- the last effect is particularly important for tracking electrons in 'heavy' detectors like ATLAS and CMS
- we'll use it as an example

energy loss of electrons

- 'fractional' energy loss of electrons described by Bethe-Heitler pdf

$$\mathcal{P}(z)\,\mathrm{d}z = \frac{(-\ln z)^{t/\ln 2}}{\Gamma(t/\ln 2)}\,\mathrm{d}z \qquad z \equiv \frac{E_{\rm after}}{E_{\rm before}}$$

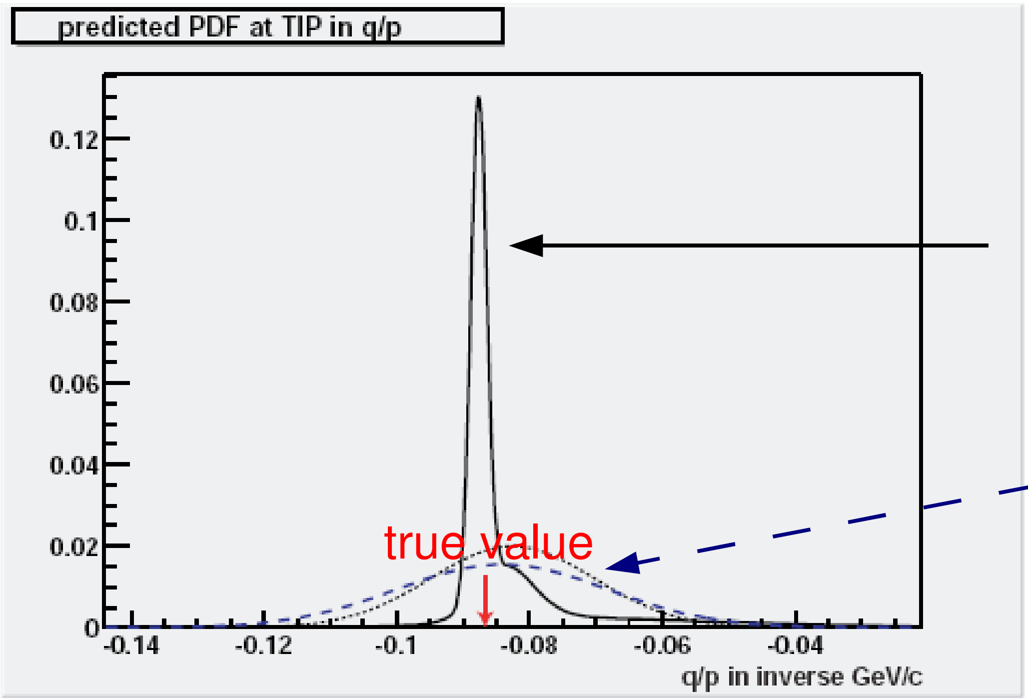
where 't = x/X0' is the radiation thickness of the obstacle



- very non-gaussian. how could you deal with this in track fit?

Gaussian Sum Filter

- idea: describe P(z) as a weighted sum of several Gaussian distributions
- split fit in different component, one for each Gauss. add up the final results.



From Adam, Fruhwirth,Strandlie,Todorov. This plot shows the result from a single fit.

PDF of sum of components from Gaussian Sum Filter

PDF of result from normal Kalman Filter, describing e-loss with single gaus

- but
  - number of components can become very large ... must run many fits to fit a single track
  - what do you do with the final pdf? reuse pdf components in vertexing?

outliers

- up to now we have assumed that we know which hits belong to the track
- but what if we have made a mistake?
  - hits from other tracks
  - noise hits
- this becomes especially important in LHC era
  - many tracks per event
  - some tracks very close (e.g. in jets)
  - overlapping events, 'spill-over' events (from previous bunch crossing)
- how do we deal with this in tracking?

outlier rejection

- most simple idea
  - reject hits with large contribution to the chisquare
  - refit the track
  - repeat if necessary

  this is sometimes called 'trimmed' fitting

- more advanced idea: treat also 'competition between tracks'
  - assign weights to hit-track combinations, depending on chi-square contribution of hit
  - eventually, combine this with an 'annealing' scheme in which weights also depend on a 'temperature'

  example: Deterministic Annealing Filter

- Marijke is jarig vandaag en trackteert taart
- daarom zingen we, in het Nederlands natuurlijk

Lang zal ze leven
Lang zal ze leven
Lang zal ze leven in de gloriiiaa
In de gloriiiiaaa
In de gloriiiiaaa
Hieperdepiep ... hoera
Hieperdepiep ... hoera
Hieperdepiep ... hoera