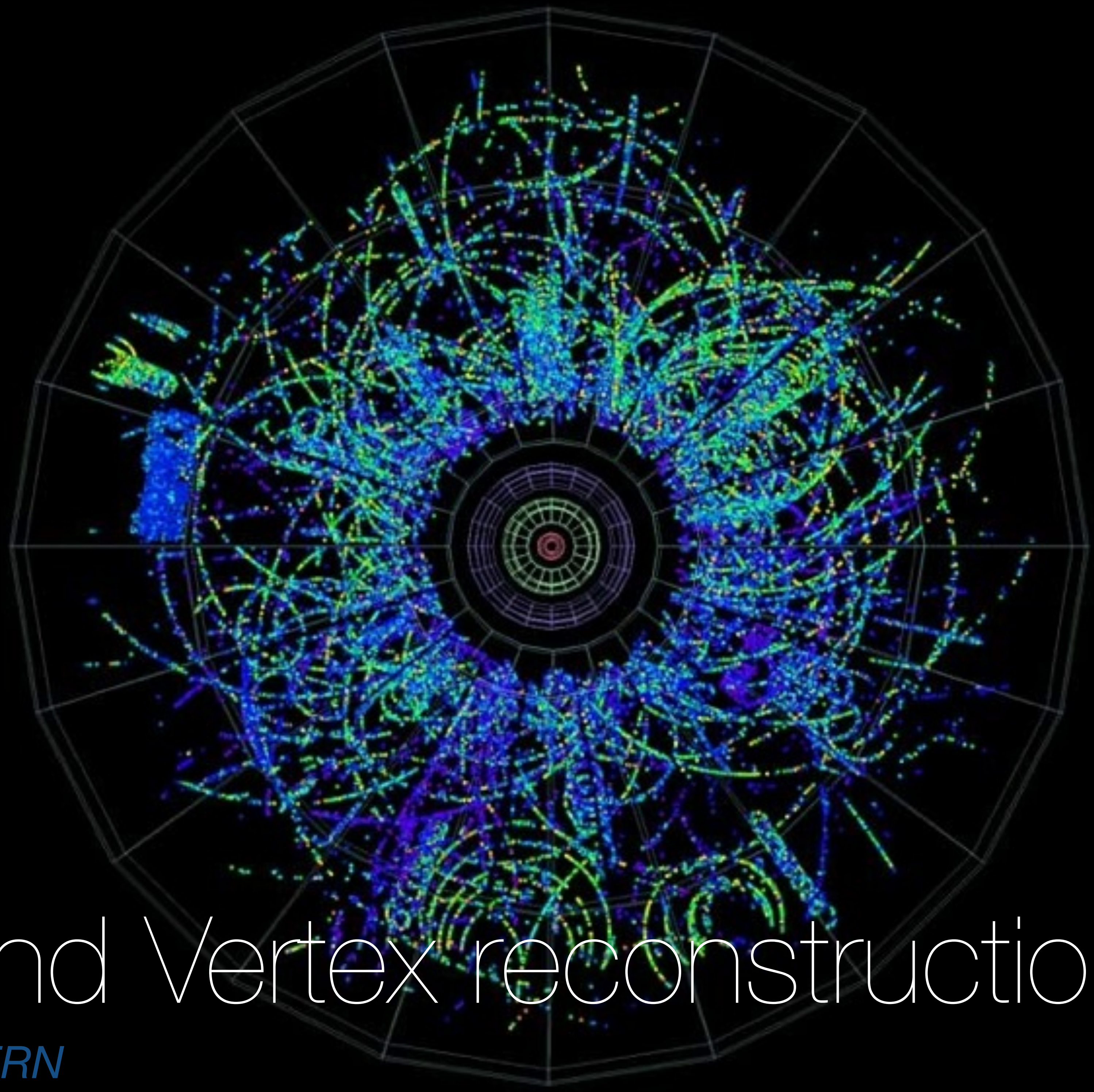
Part 2 - Finding and fitting tracks



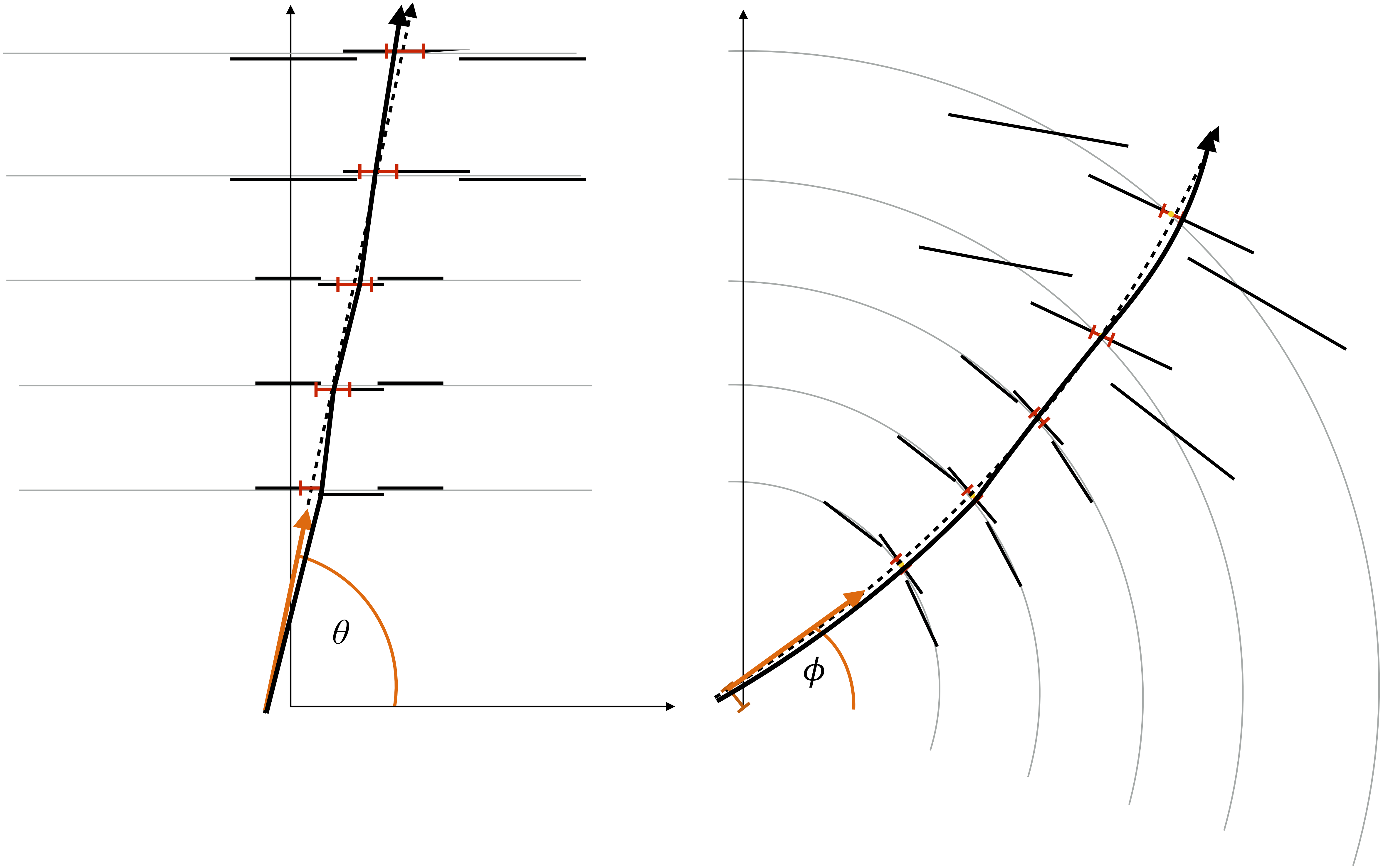
Track and Vertex reconstruction

A. Salzburger, CERN

http://home.web.cern.ch/sites/home.web.cern.ch/files/images/updates/for_scientists/ALICE-proton-lead-event.jpg

Where we stopped yesterday

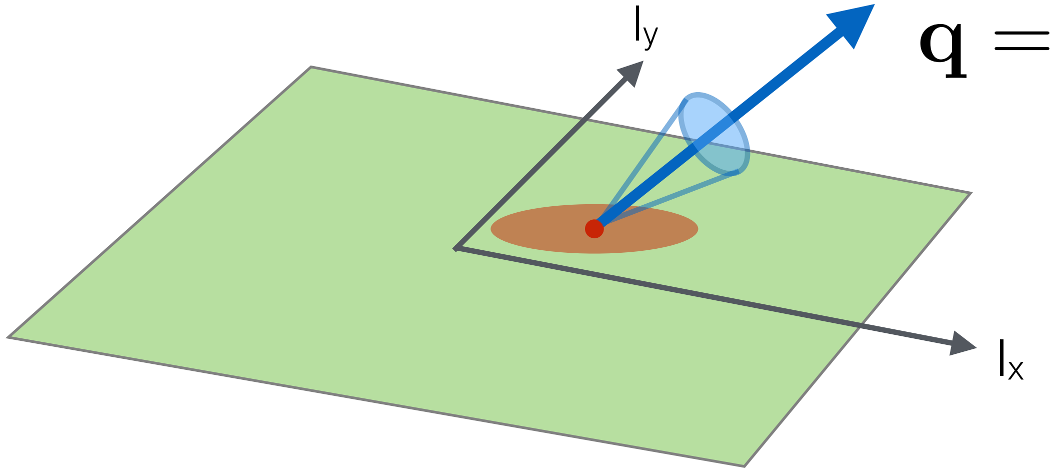
We hadn't found any tracks yet !



Reminder

What we picked up so far …

… and what we will need today.



track parameters
(with covariances)

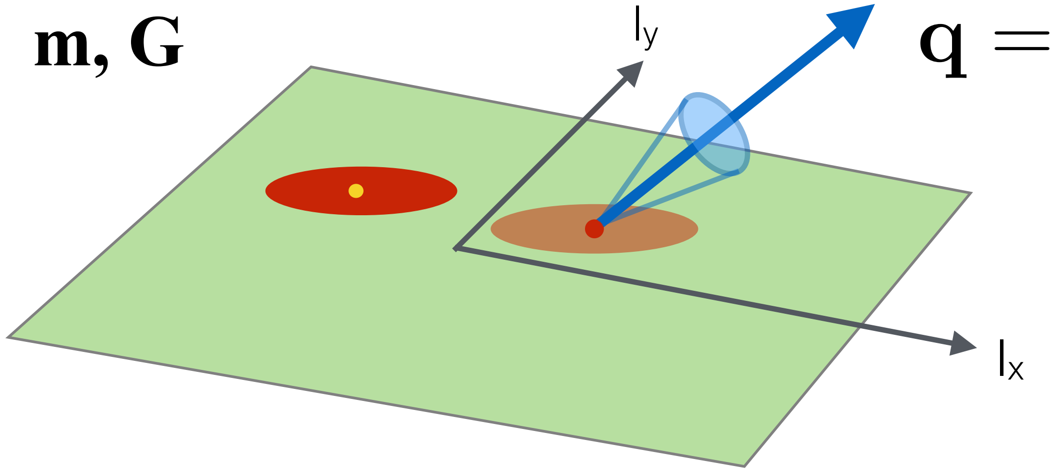
$$\mathbf{q} = (l_1, l_2, \phi, \theta, q/p), \mathbf{C}$$

What we picked up so far …

… and what we will need today.

measurement
(with covariances)
$\mathbf{m}, \mathbf{G}$

track parameters
(with covariances)
$\mathbf{q} = (l_1, l_2, \phi, \theta, q/p), \mathbf{C}$



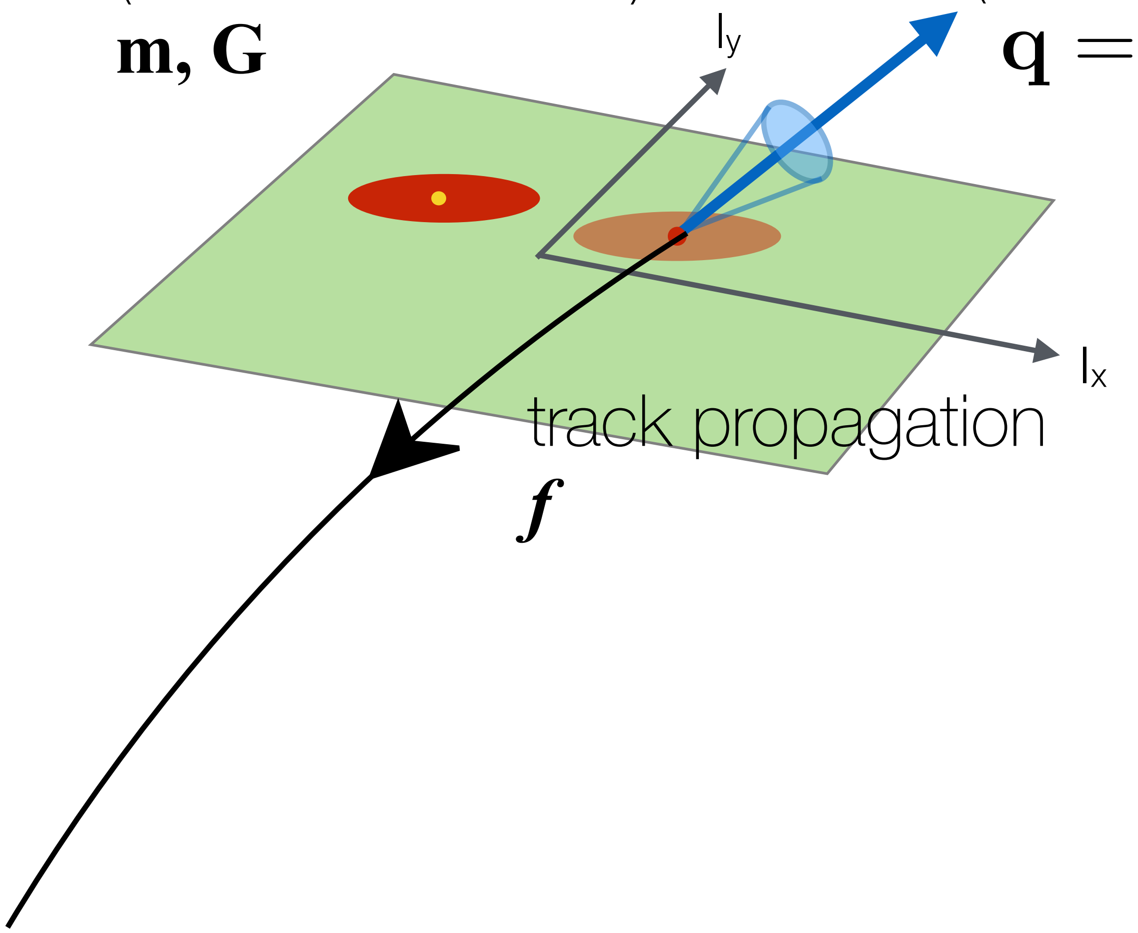
Reminder

What we picked up so far . . .

… and what we will need today.

measurement
(with covariances)
m, G

track parameters
(with covariances)
$\mathbf{q} = (l_1, l_2, \phi, \theta, q/p), \mathbf{C}$

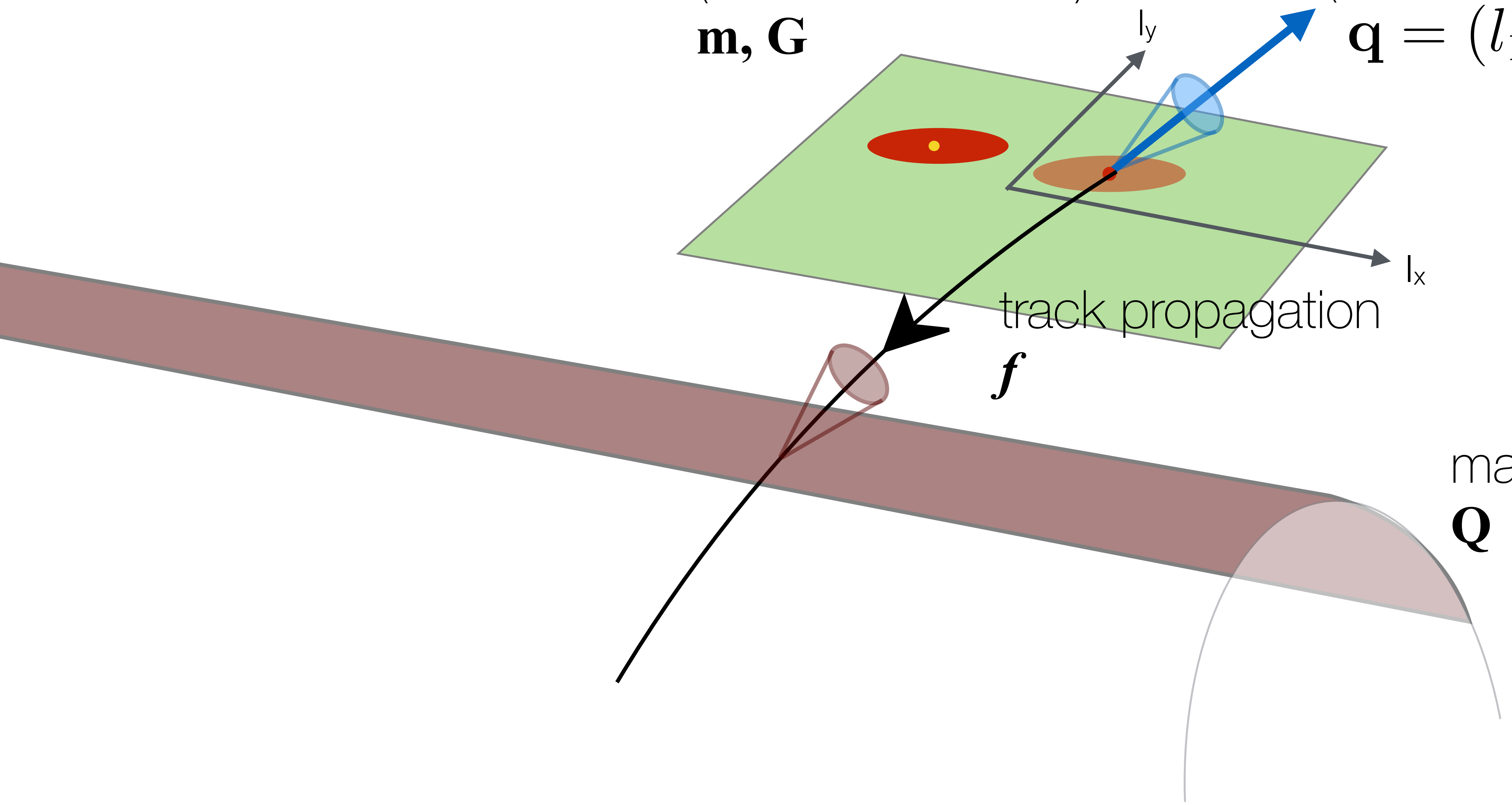


track propagation
$\boldsymbol{f}$

Reminder

What we picked up so far …

… and what we will need today.



measurement
(with covariances)
$\mathbf{m}, \mathbf{G}$

track parameters
(with covariances)
$\mathbf{q} = (l_1, l_2, \phi, \theta, q/p), \mathbf{C}$

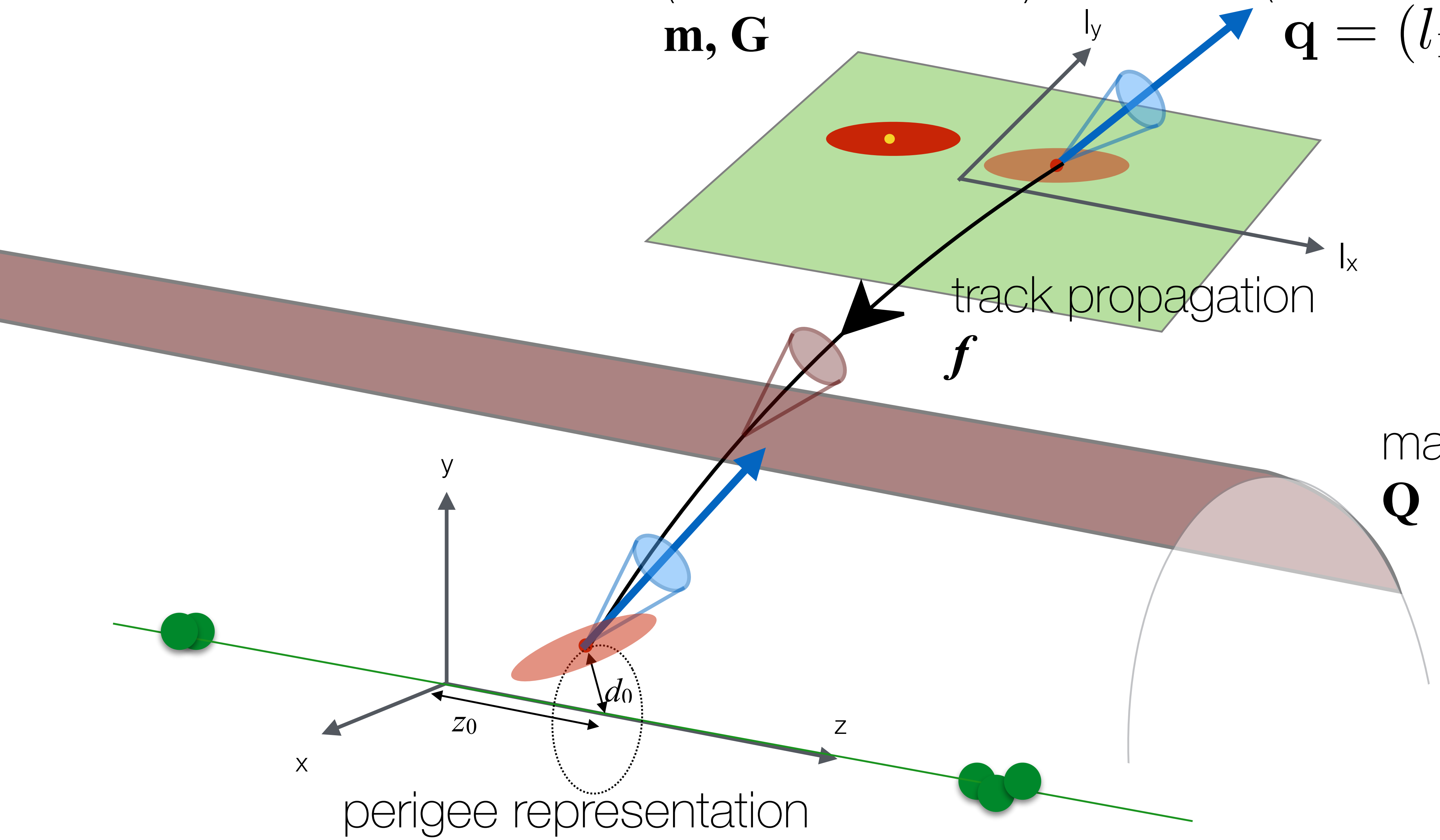
track propagation
$\boldsymbol{f}$

material noise
$\mathbf{Q}$

Reminder

What we picked up so far …

… and what we will need today.



measurement
(with covariances)
$\mathbf{m}, \mathbf{G}$

track parameters
(with covariances)
$\mathbf{q} = (l_1, l_2, \phi, \theta, q/p), \mathbf{C}$

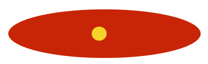
track propagation
$\boldsymbol{f}$

material noise
$\mathbf{Q}$

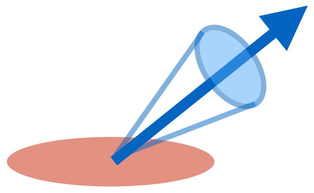
perigee representation
(with covariances)
$\mathbf{q_0} = (d_0, z_0, \phi, \theta, q/p), \ \mathbf{C_0}$

Let's make our toolset complete

measurement
(with covariances)
$\mathbf{m}_k, \mathbf{G}_k$

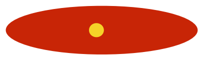


track parameters
(with covariances)
$\mathbf{q}_k = (l_1, l_2, \phi, \theta, q/p), \mathbf{C}_k$

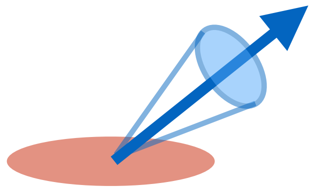


Let's make our toolset complete

measurement
(with covariances)
$\mathbf{m}_k$, $\mathbf{G}_k$



track parameters
(with covariances)
$\mathbf{q}_k = (l_1, l_2, \phi, \theta, q/p)$, $\mathbf{C}_k$

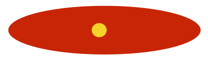


$$[\mathbf{m}_k - h(\mathbf{q}_k)] = \Delta\mathbf{m}_k$$

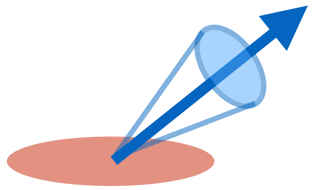
measurement mapping function,
transforms the track parameters
into the measurement frame

Let's make our toolset complete

measurement
(with covariances)
$\mathbf{m}_\mathrm{k}, \mathbf{G}_\mathrm{k}$

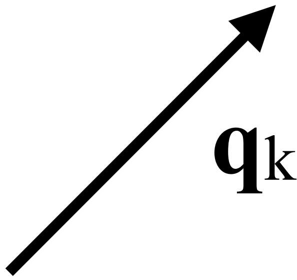


track parameters
(with covariances)
$\mathbf{q}_\mathrm{k} = (l_1, l_2, \phi, \theta, q/p), \mathbf{C}_\mathrm{k}$



$$[\mathbf{m}_\mathrm{k} - \mathrm{h}(\mathbf{q}_\mathrm{k})] = \Delta\mathbf{m}_\mathrm{k}$$

measurement mapping function,
transforms the track parameters
into the measurement frame



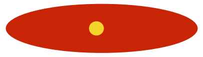
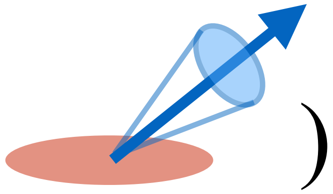
$$\mathbf{q}_\mathrm{k+1} = \boldsymbol{f}_\mathrm{k}^{\mathrm{k+1}}\,(\mathbf{q}_\mathrm{k})$$

transport (or propagation) of track parameters
from reference surface k to k+1

Let's make our toolset complete

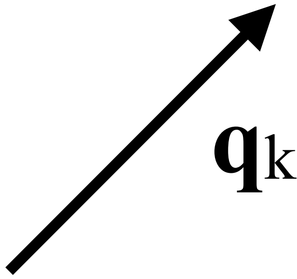
measurement
(with covariances)
$\mathbf{m}_\mathrm{k}$, $\mathbf{G}_\mathrm{k}$

track parameters
(with covariances)
$\mathbf{q}_\mathrm{k} = (l_1, l_2, \phi, \theta, q/p)$, $\mathbf{C}_\mathrm{k}$

( - ) $= \Delta\mathbf{m}_\mathrm{k}$

$$[\mathbf{m}_\mathrm{k} - \mathrm{h}(\mathbf{q}_\mathrm{k})] = \Delta\mathbf{m}_\mathrm{k}$$

measurement mapping function,
transforms the track parameters
into the measurement frame



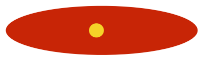
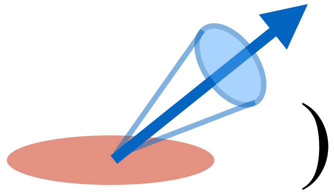
$$\mathbf{q}_{\mathrm{k}+1} = \boldsymbol{f}_\mathrm{k}^{\mathrm{k}+1}(\mathbf{q}_\mathrm{k})$$

transport (or propagation) of track parameters
from reference surface k to k+1

Let's make our toolset complete

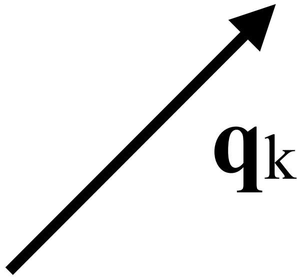
measurement
(with covariances)
$\mathbf{m}_k$, $\mathbf{G}_k$

track parameters
(with covariances)
$\mathbf{q}_k = (l_1, l_2, \phi, \theta, q/p)$, $\mathbf{C}_k$

( - ) $= \Delta\mathbf{m}_k$

$$[\mathbf{m}_k - h(\mathbf{q}_k)] = \Delta\mathbf{m}_k$$

measurement mapping function,
transforms the track parameters
into the measurement frame



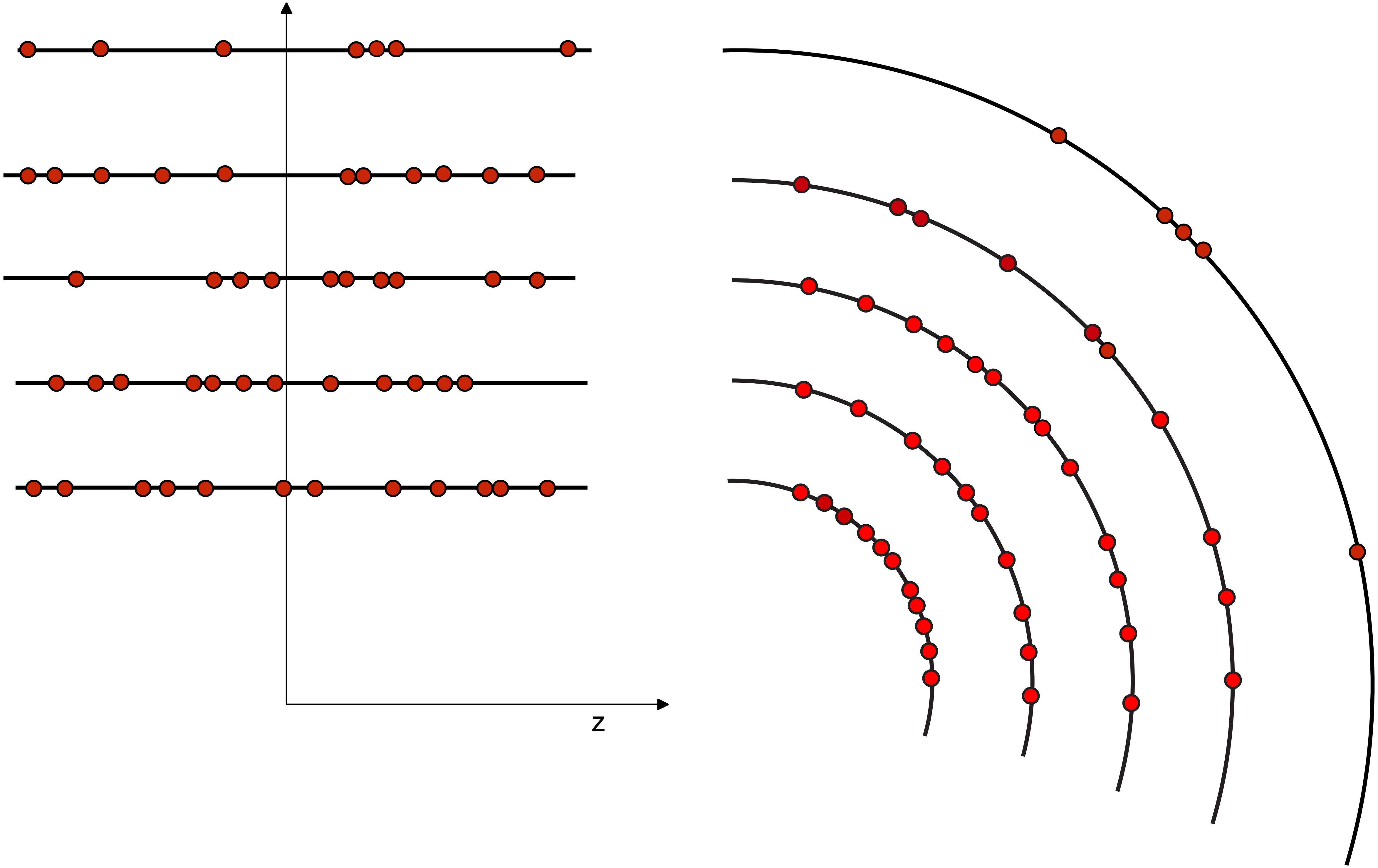
$$\mathbf{q}_{k+1} = \boldsymbol{f}_k^{k+1}(\mathbf{q}_k)$$

transport (or propagation) of track parameters from reference surface k to k+1

$$\mathbf{d}_{k+1} = h_{k+1} \circ \boldsymbol{f}_k^{k+1}$$

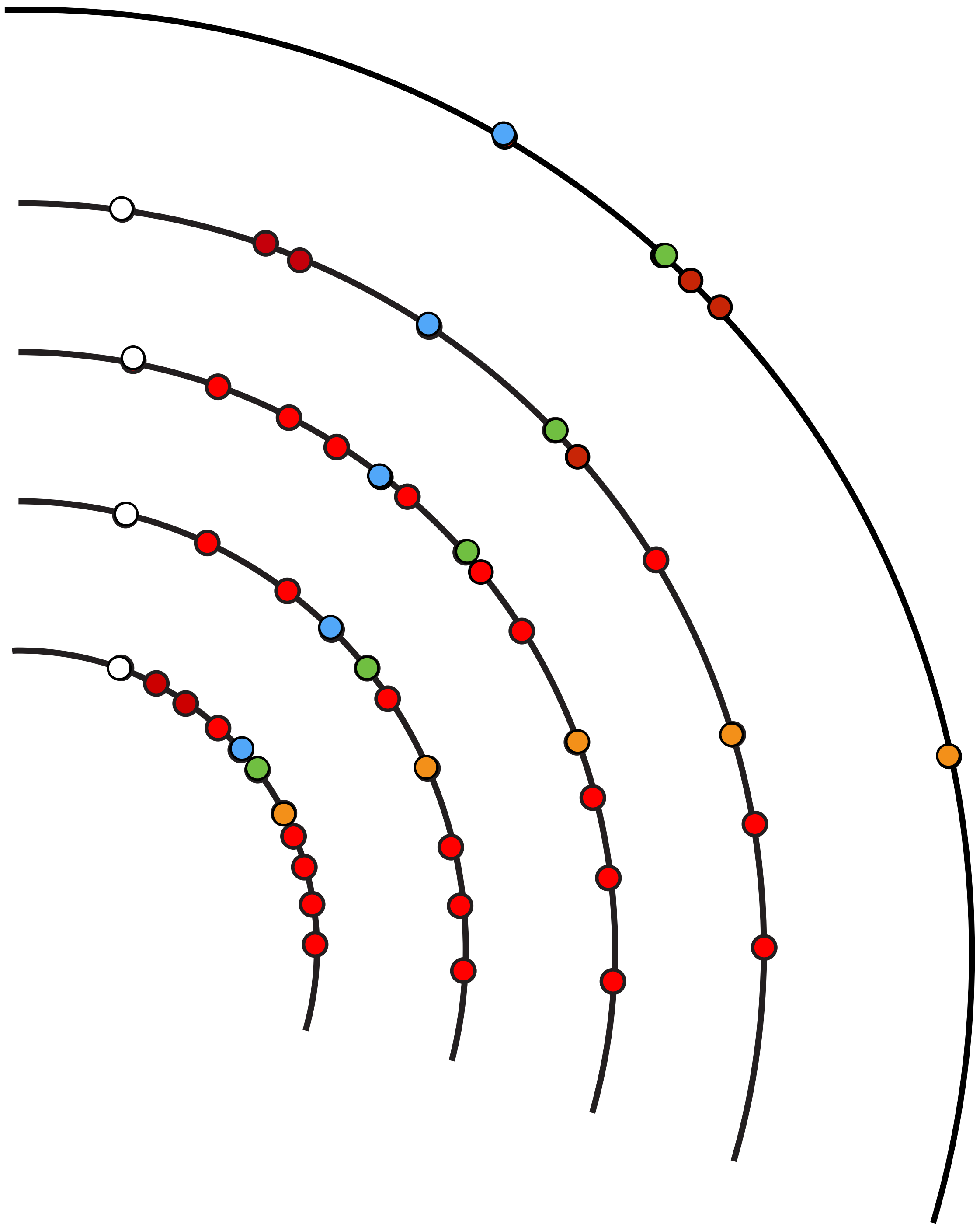
combination of transport and measurement mapping function

Finding tracks



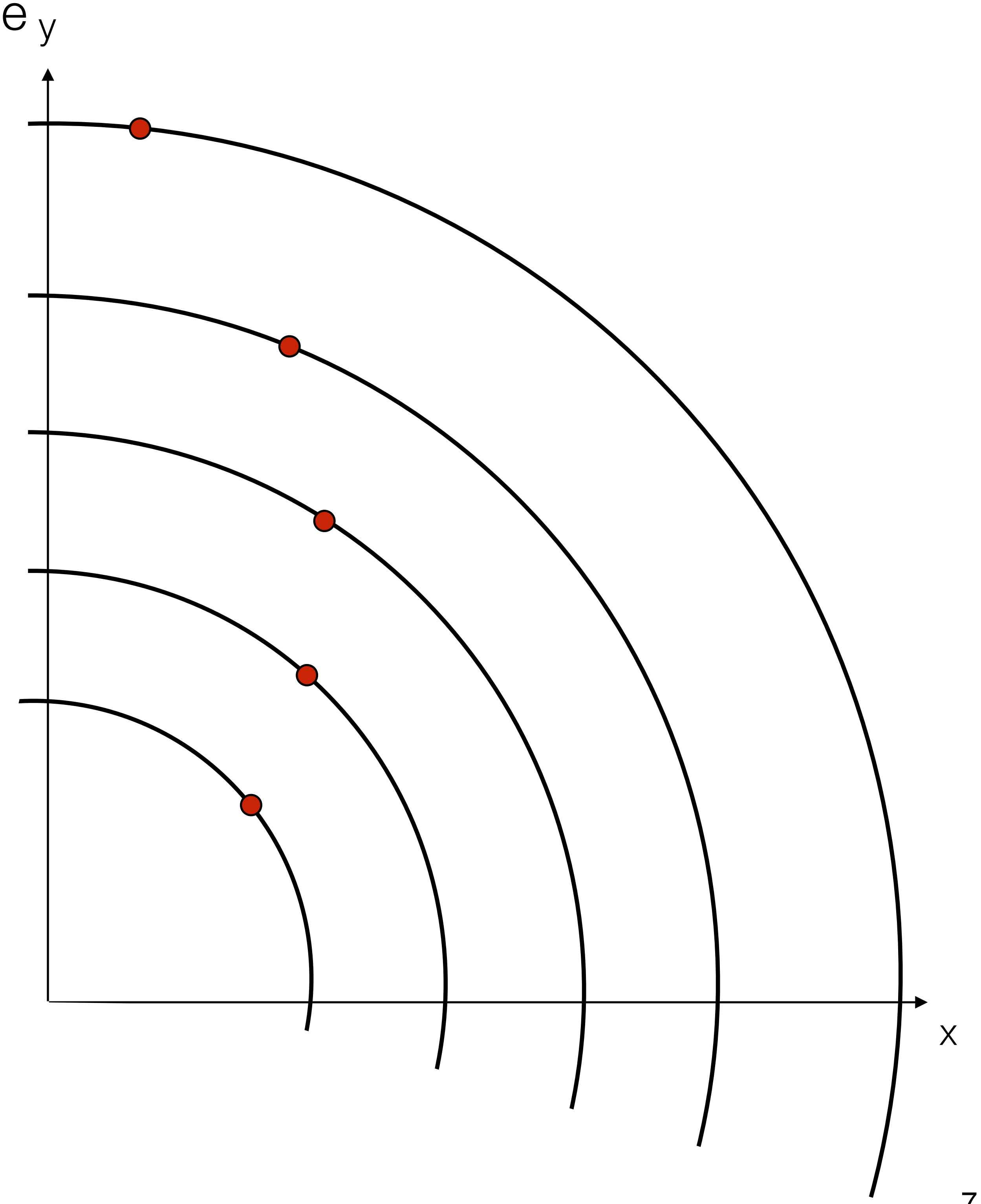
Global pattern recognition strategies

- Conformal mapping
  - the idea of conformal mapping is to transform your hit information into a parameter space, where your groups of hits are visible
  - you need a transformation for it which assumes a track model



Global pattern recognition strategies

- Conformal mapping : Hough transform
  - transform your track hits from the x, y space into a more appropriate space
  - let's assume that particles come from the interaction region + solve in the transverse direction



$$\mathbf{q} = (d_0, z_0, \phi, \theta, q/p)$$

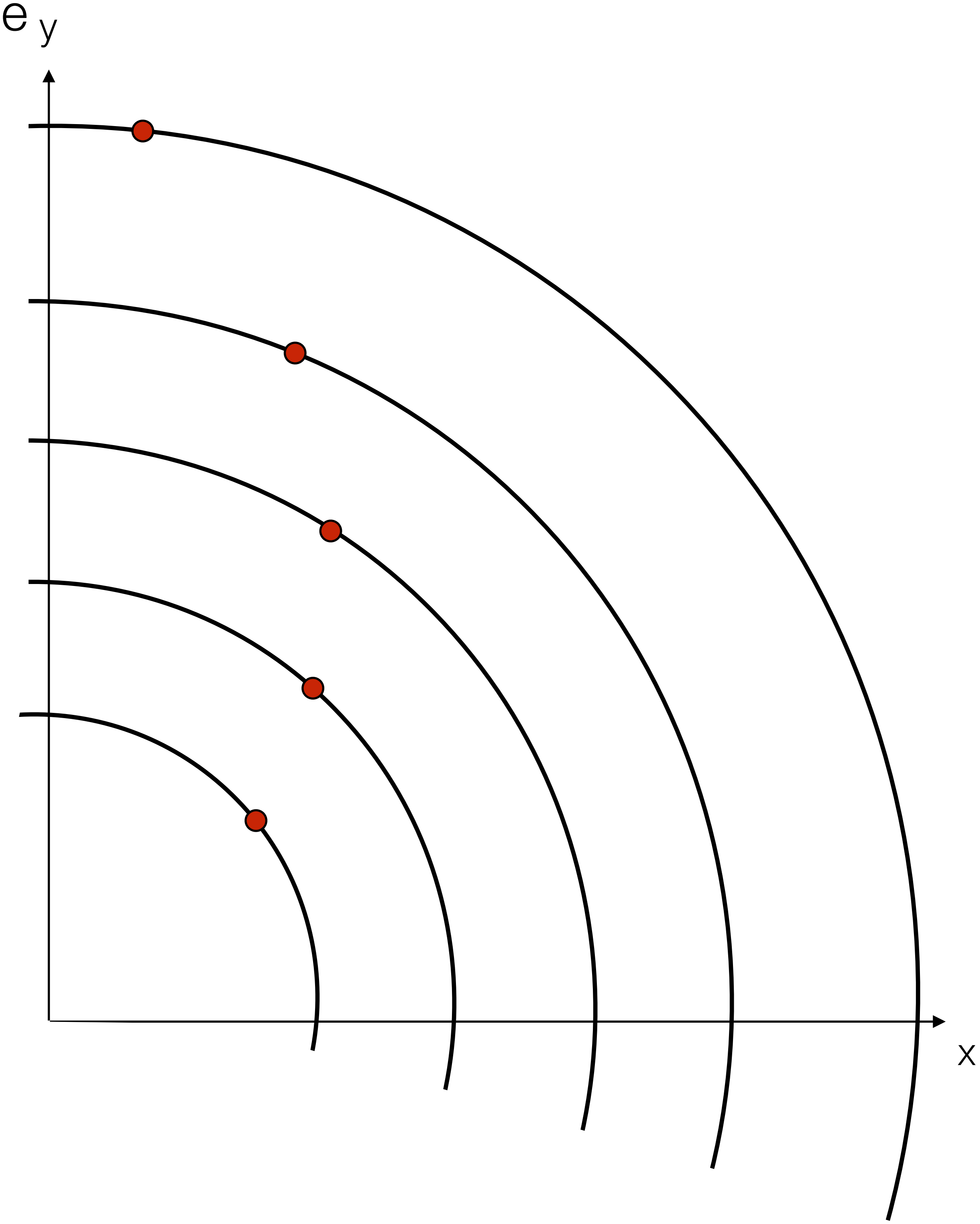
Global pattern recognition strategies

- Conformal mapping : Hough transform
  - transform your track hits from the x, y space into a more appropriate space
  - let’s assume that particles come from the interaction region + solve in the transverse direction

$$\mathbf{q} = (d_0, z_0, \phi, \theta, q/p)$$

$$\downarrow$$

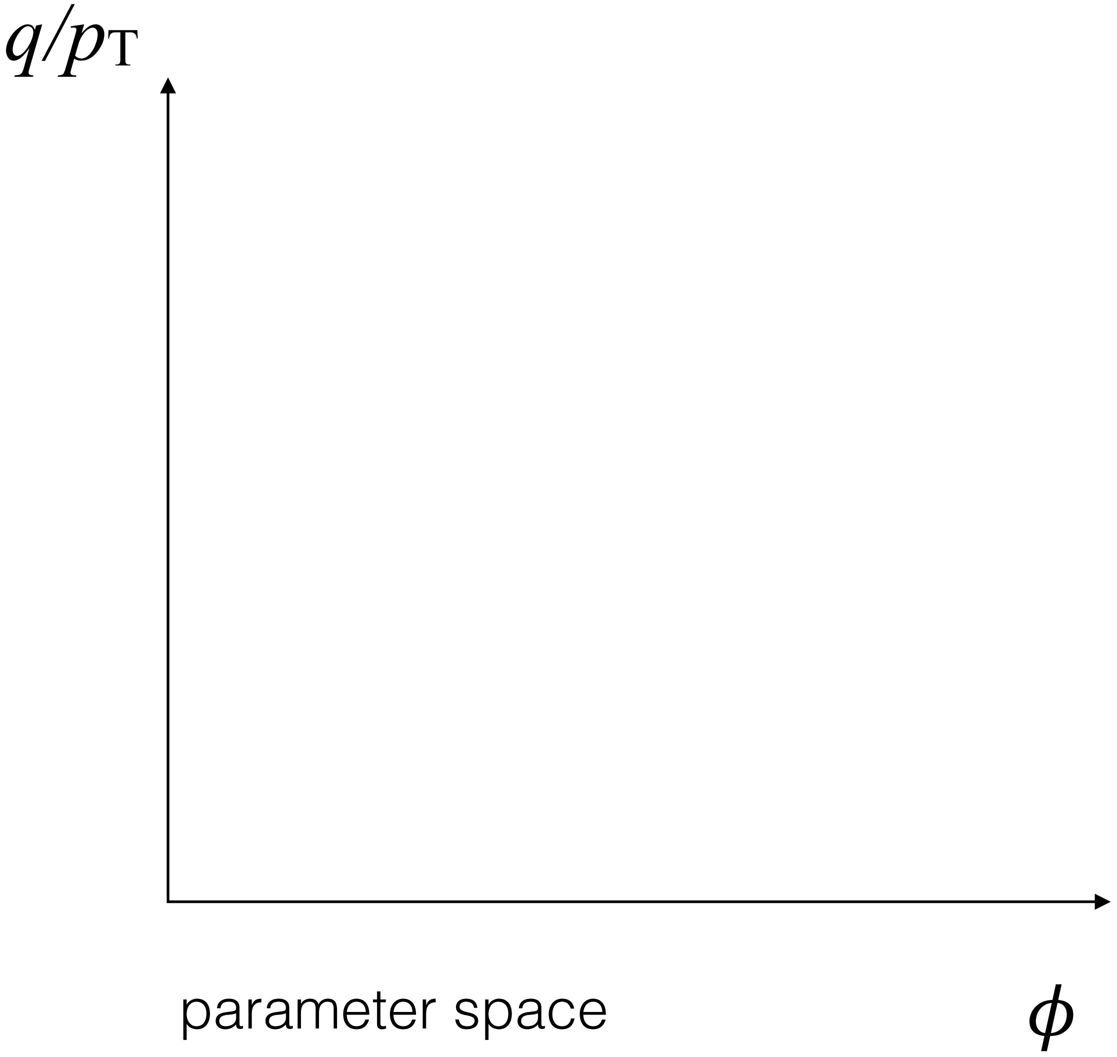
$$\mathbf{q} = (\cancel{d_0}, \cancel{z_0}, \phi, \cancel{\theta}, q/p_T)$$

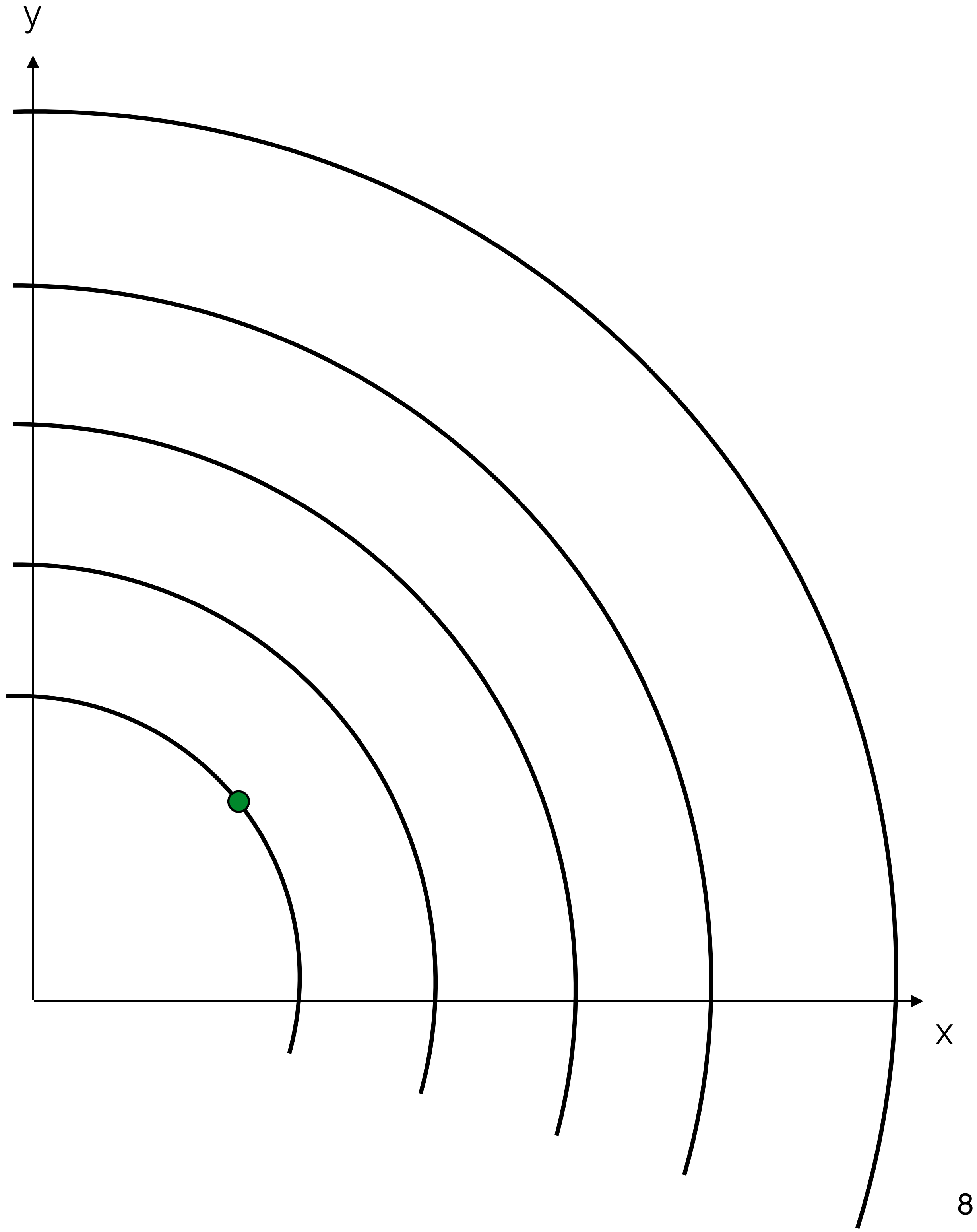


Global pattern recognition strategies

- Conformal mapping : Hough transform
  - transform your track hits in the x, y space

$$\mathbf{q} = (\cancel{d_0}, \cancel{z_0}, \phi, \cancel{\theta}, q/p_T)$$

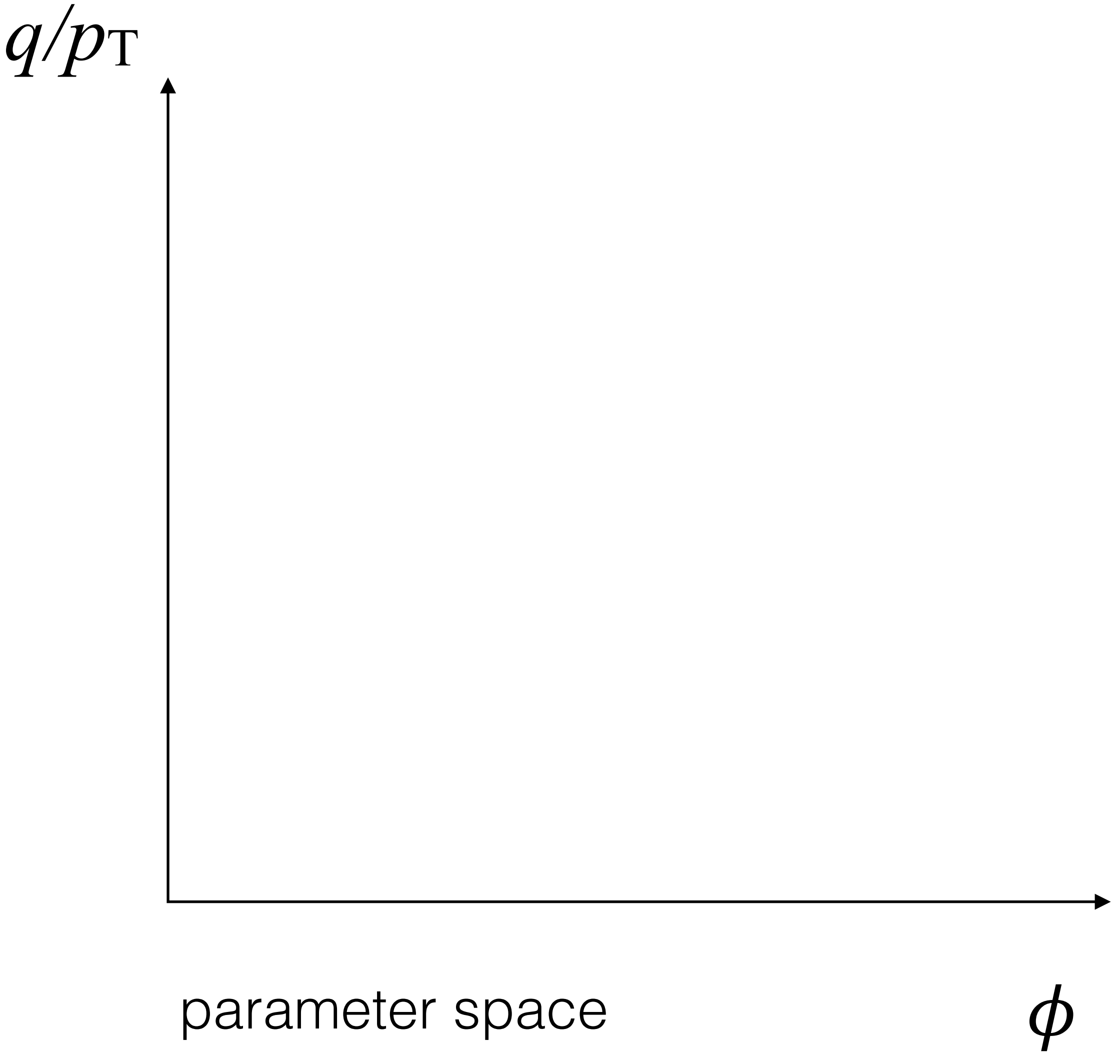


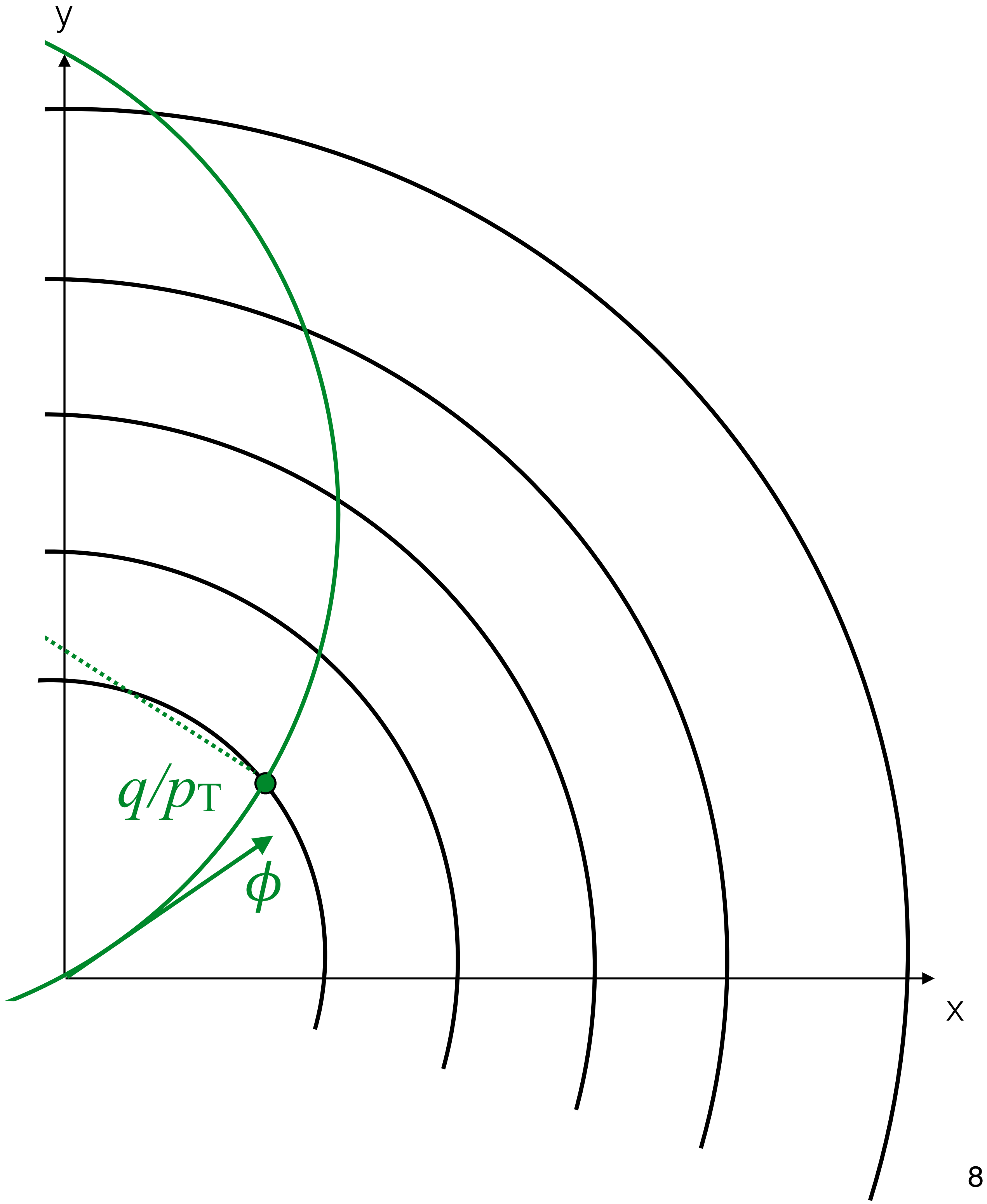


Global pattern recognition strategies

- Conformal mapping : Hough transform
  - transform your track hits in the x, y space

$$\mathbf{q} = (\cancel{d}_0, \cancel{z}_0, \phi, \cancel{\theta}, q/p_{\mathrm{T}})$$

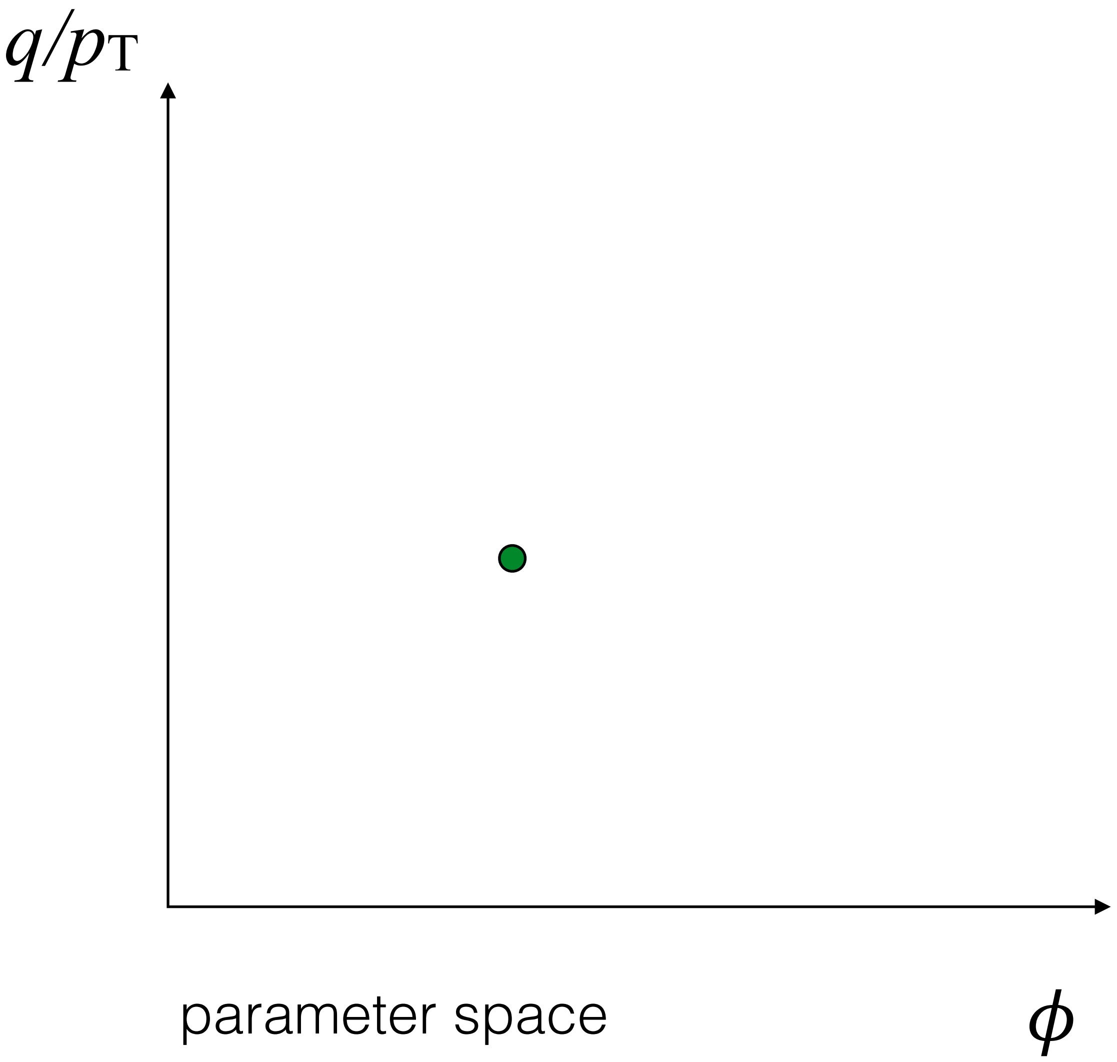


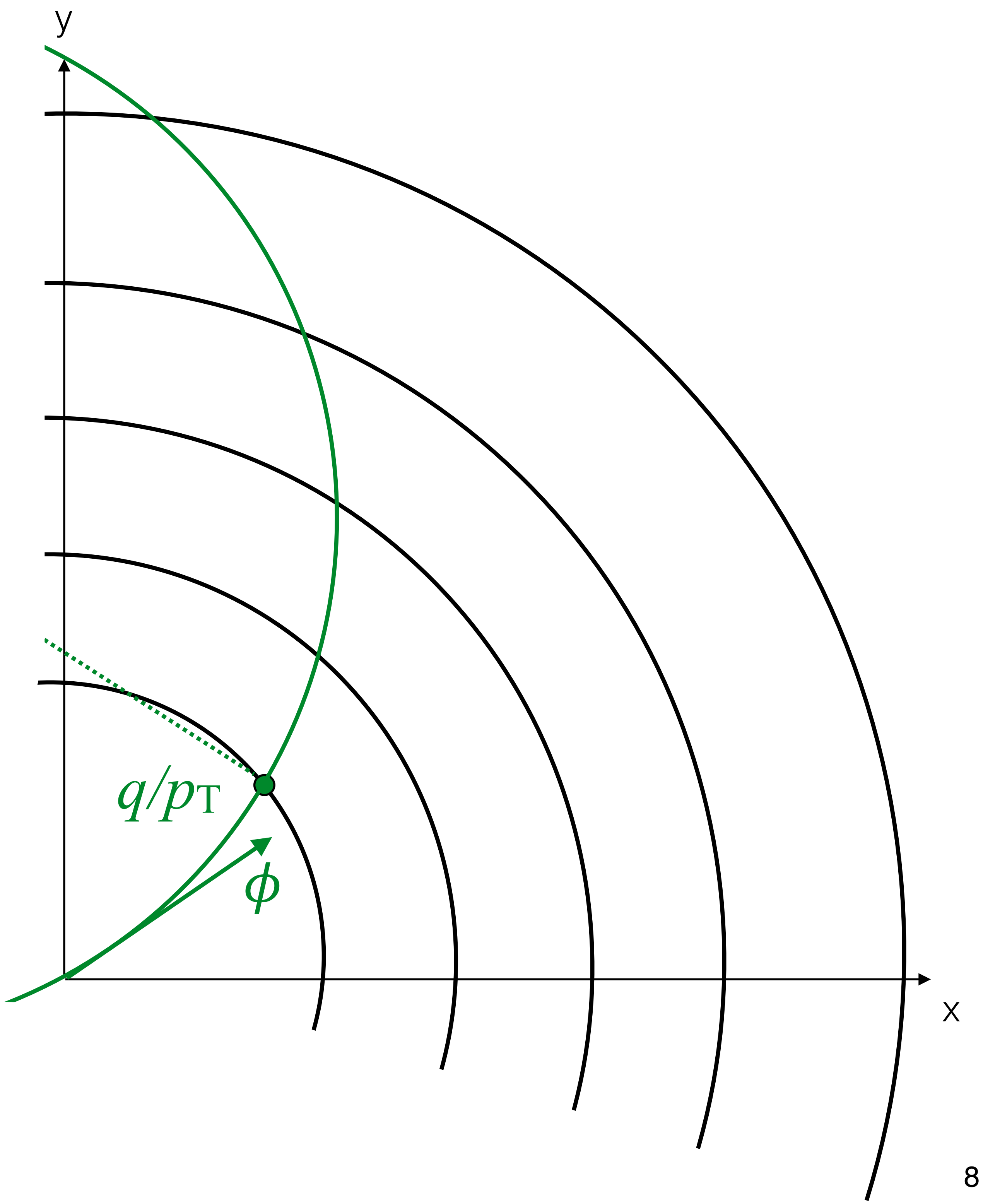


Global pattern recognition strategies

- Conformal mapping : Hough transform
  - transform your track hits in the x, y space

$$\mathbf{q} = (\cancel{d_0}, \cancel{z_0}, \phi, \cancel{\theta}, q/p_{\mathrm{T}})$$

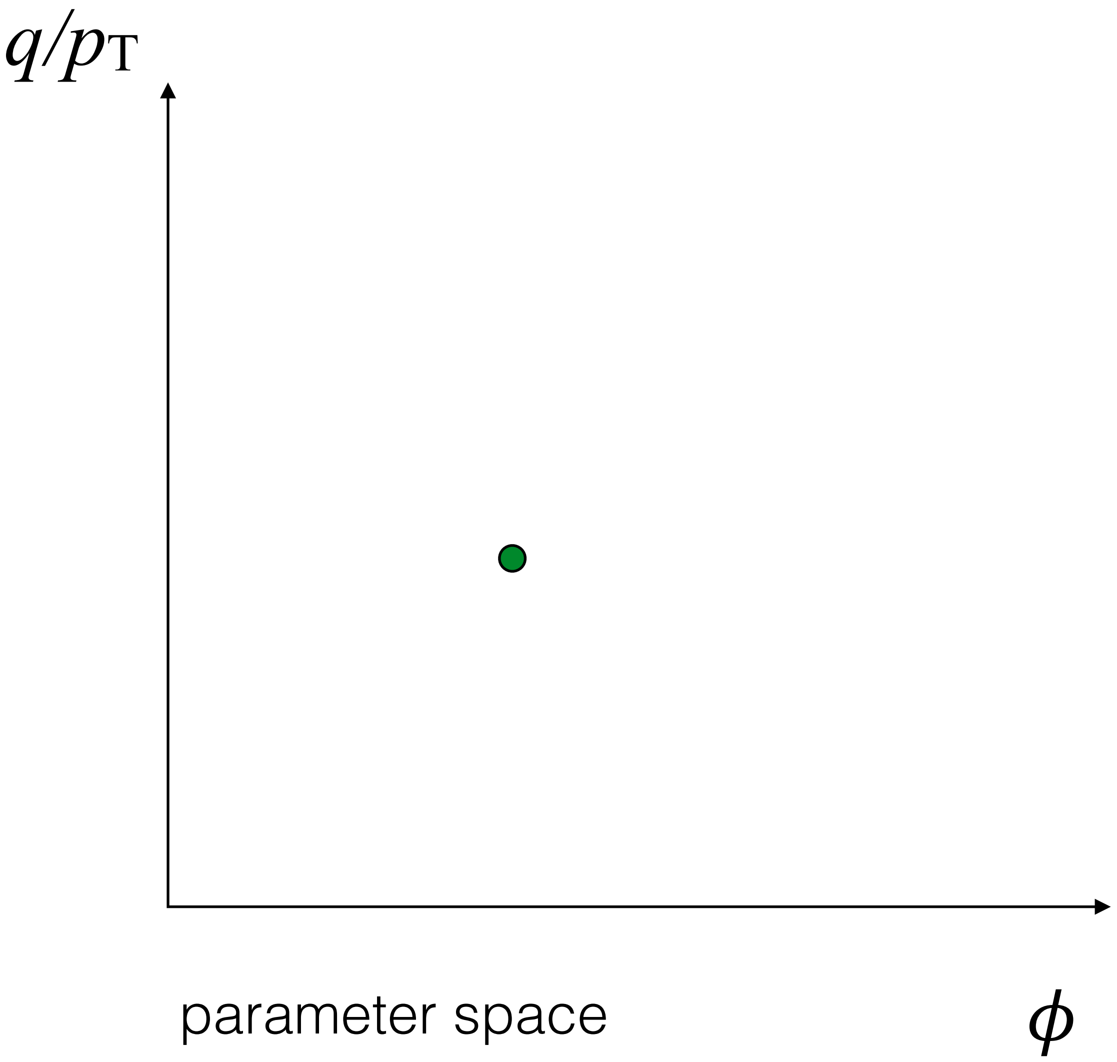


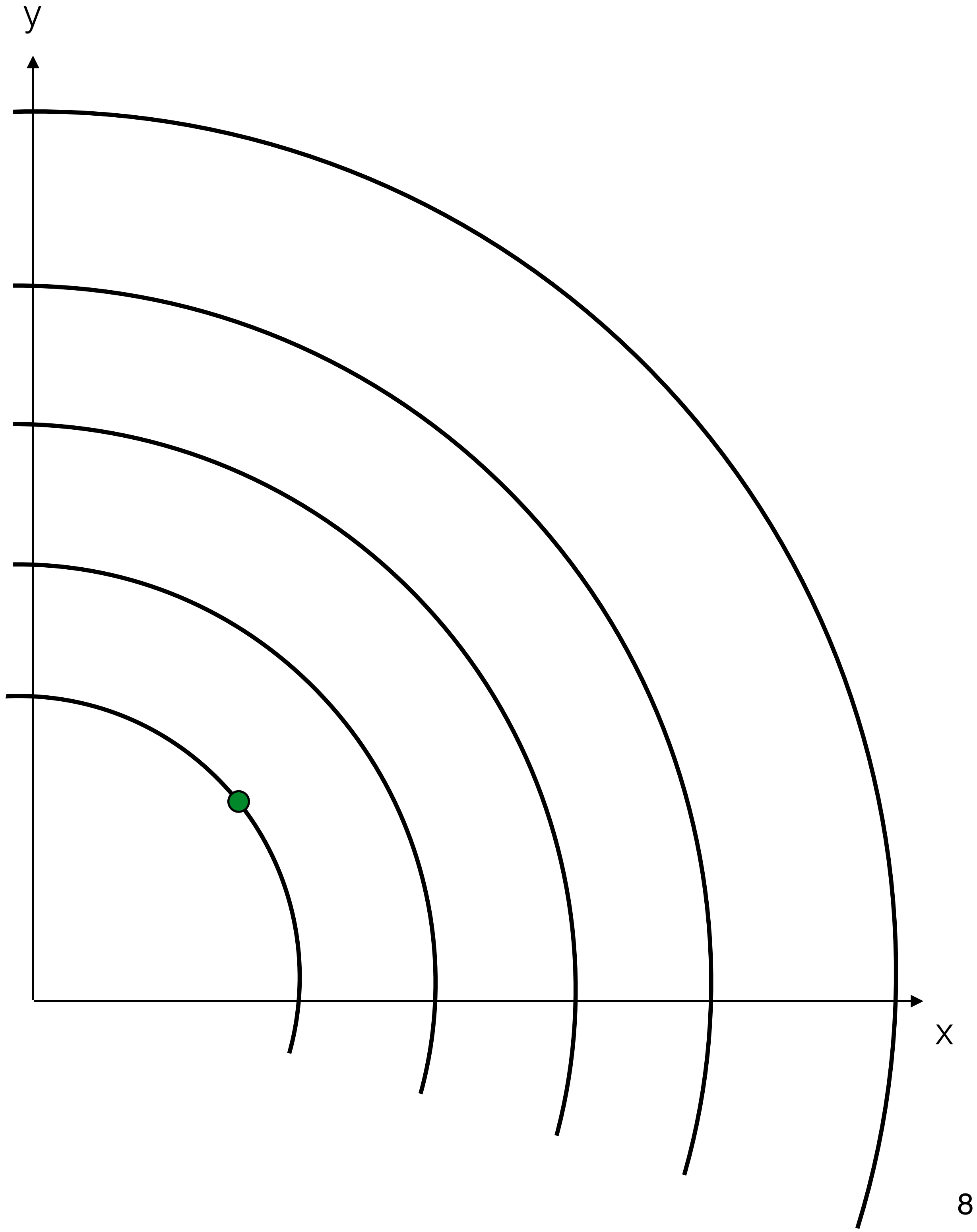


Global pattern recognition strategies

- Conformal mapping : Hough transform
  - transform your track hits in the x, y space

$$\mathbf{q} = (\cancel{d_0}, \cancel{z_0}, \phi, \cancel{\theta}, q/p_{\mathrm{T}})$$

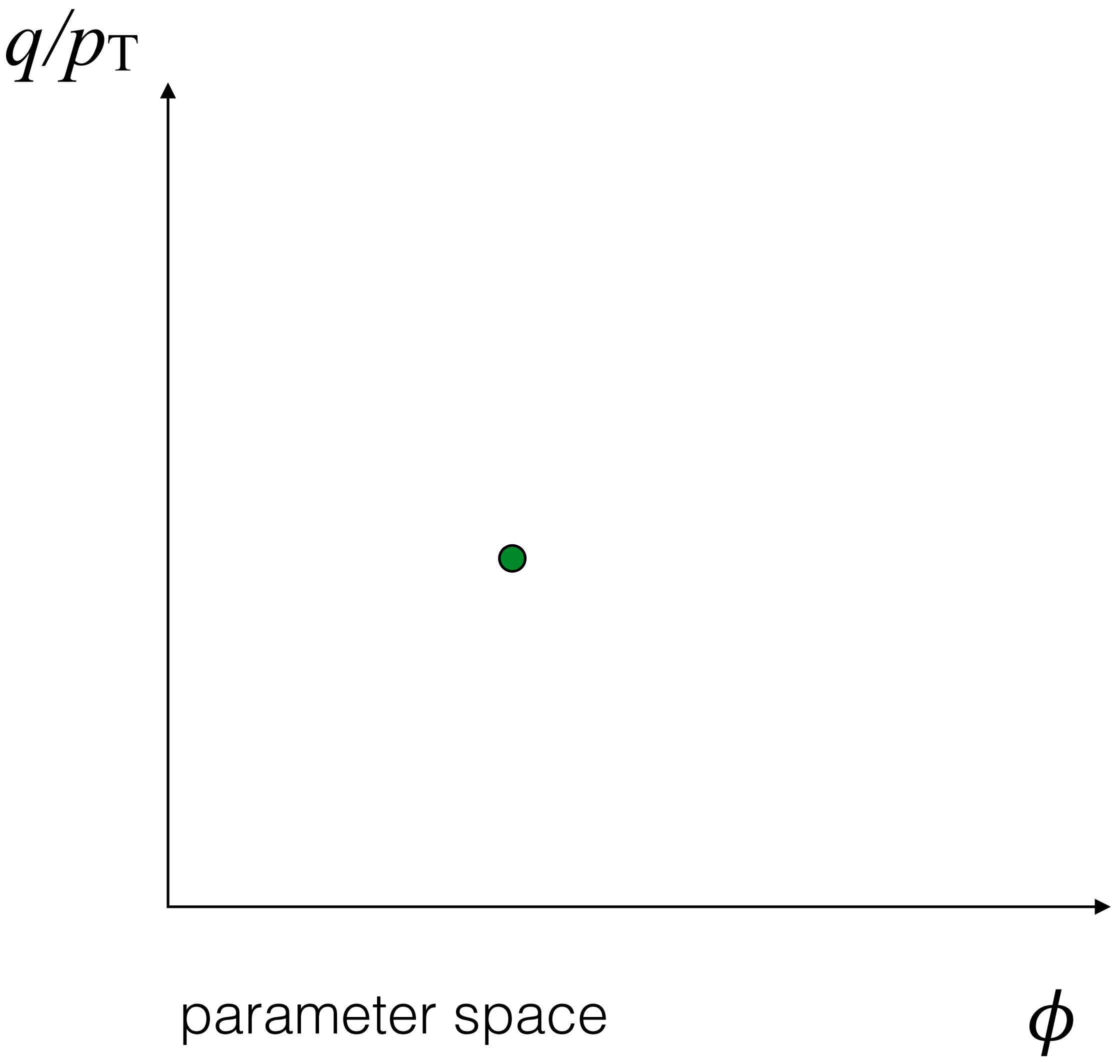


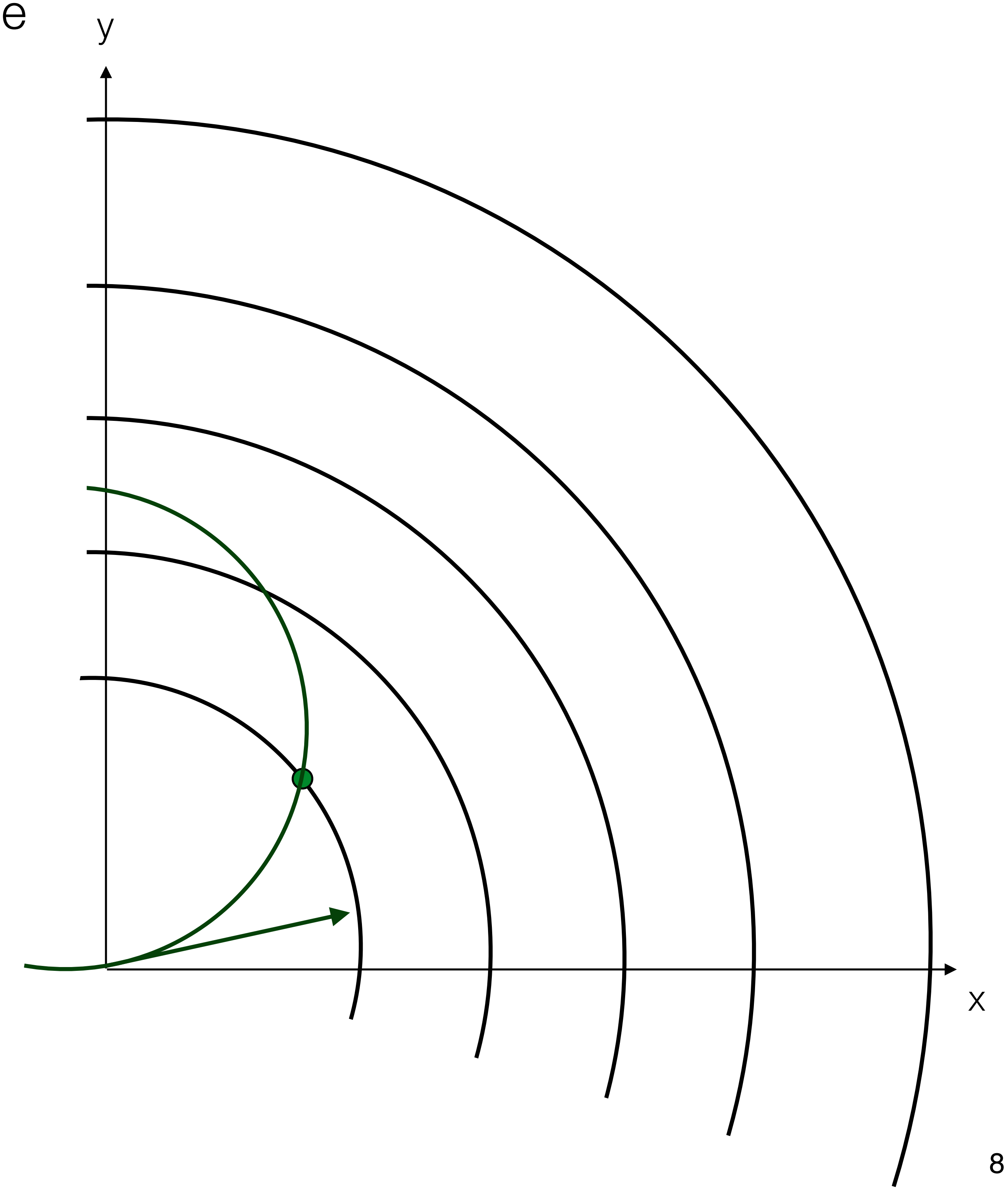


Global pattern recognition strategies

- Conformal mapping : Hough transform
  - transform your track hits in the x, y space

$$\mathbf{q} = (\xcancel{d_0}, \xcancel{z_0}, \phi, \xcancel{\theta}, q/p_{T})$$

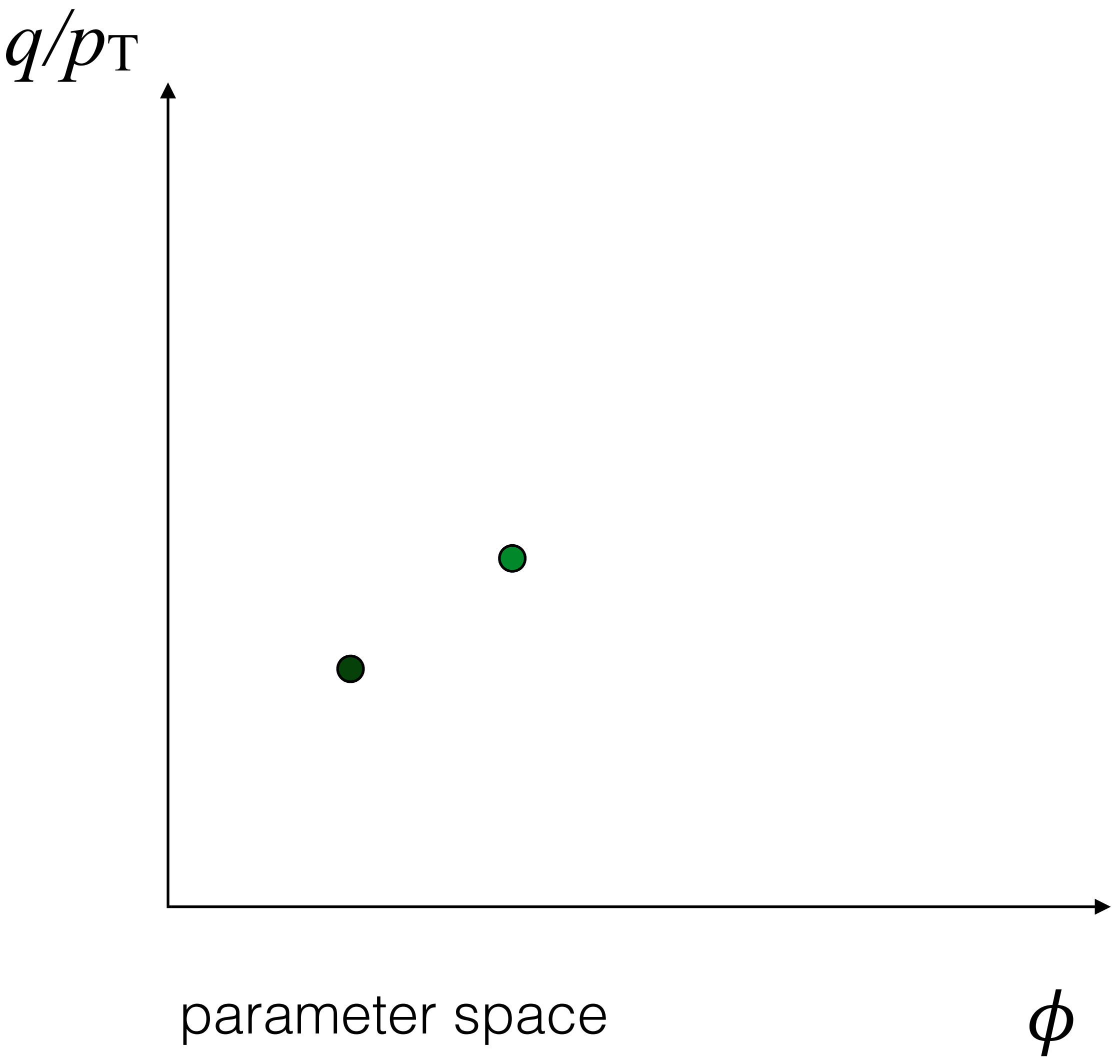


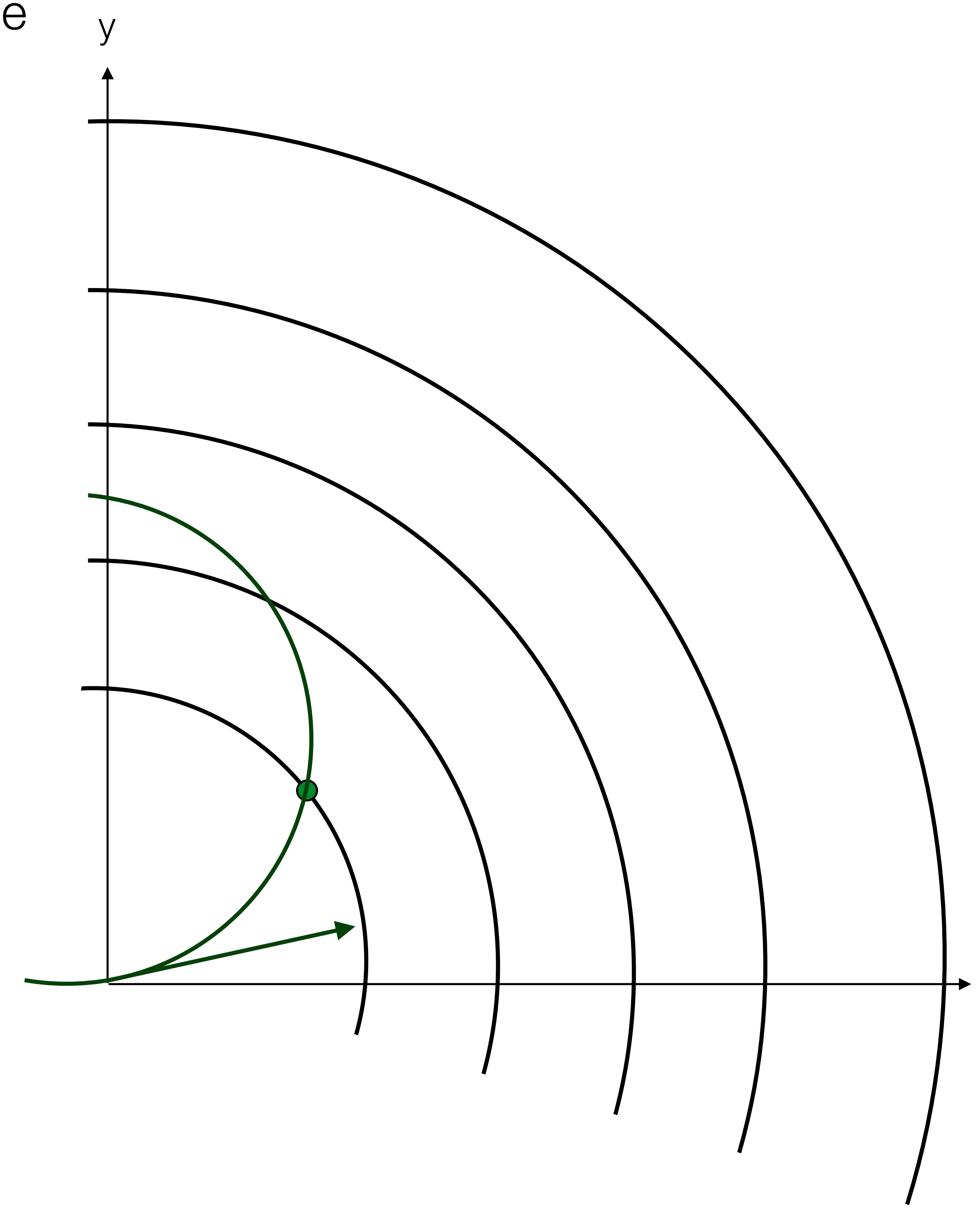


Global pattern recognition strategies

- Conformal mapping : Hough transform
  - transform your track hits in the x, y space

$$\mathbf{q} = (\cancel{d}_0, \cancel{z}_0, \phi, \cancel{\theta}, q/p_{\mathrm{T}})$$

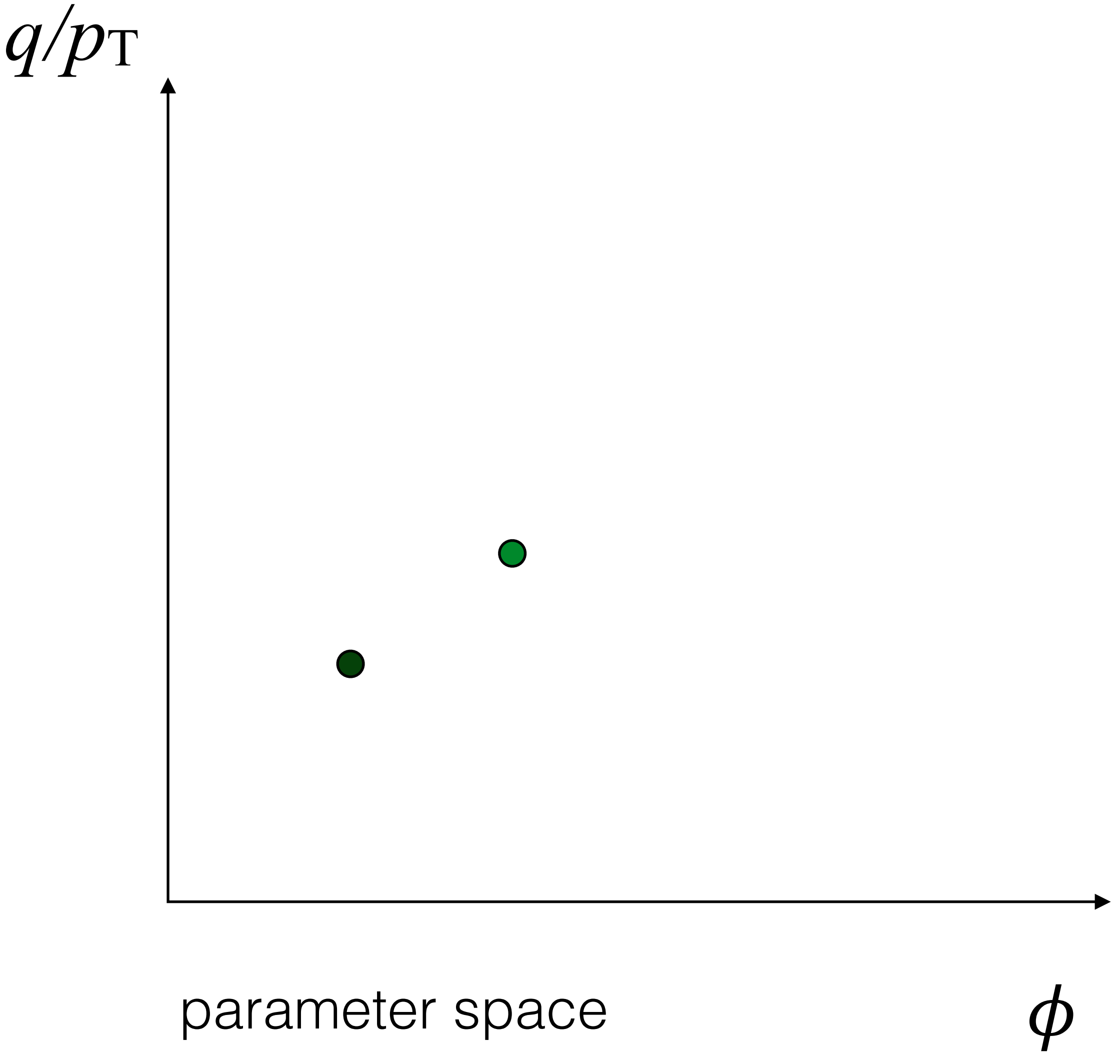


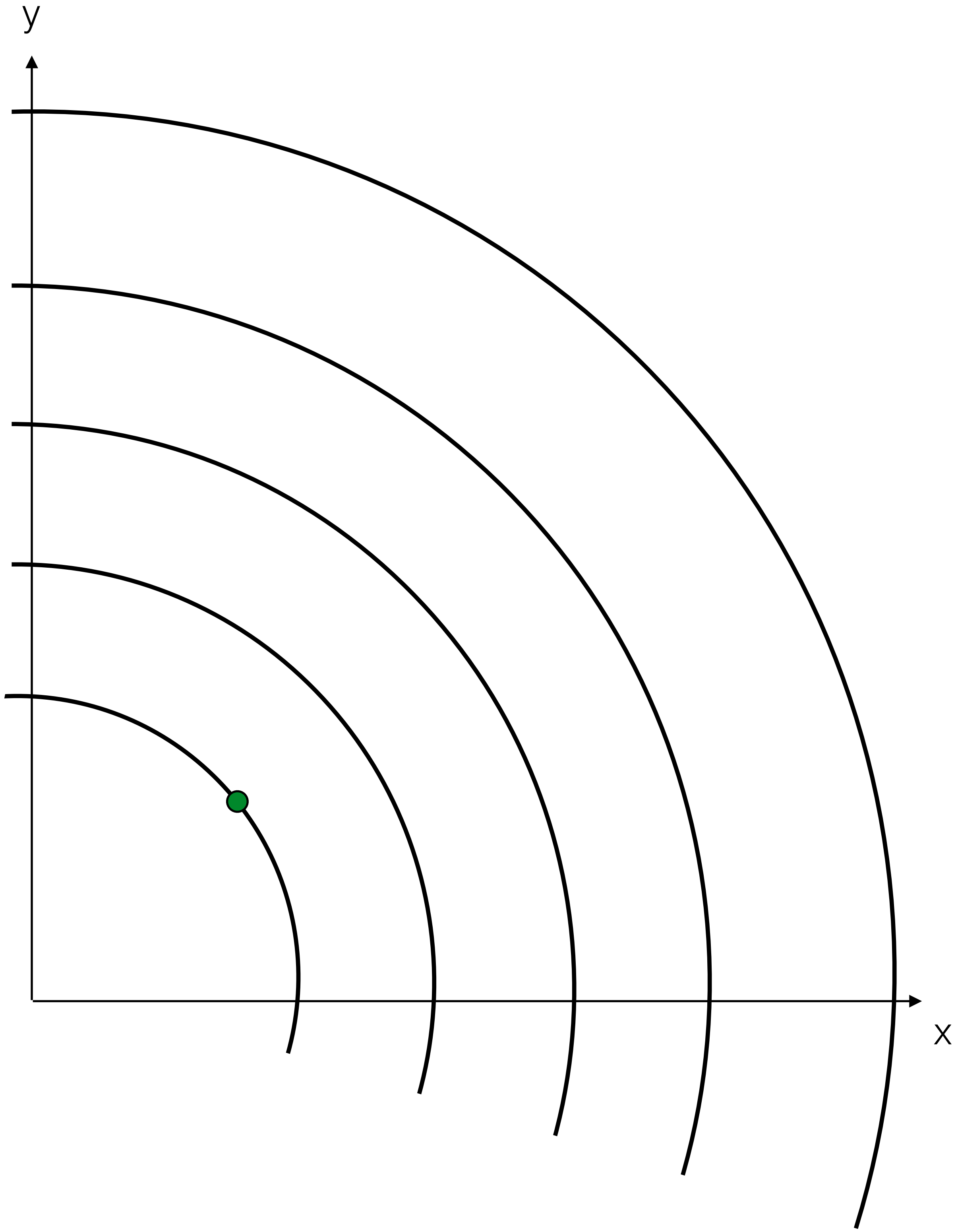


Global pattern recognition strategies

- Conformal mapping : Hough transform
  - transform your track hits in the x, y space

$$\mathbf{q} = (\cancel{d_0}, \cancel{z_0}, \phi, \cancel{\theta}, q/p_{T})$$

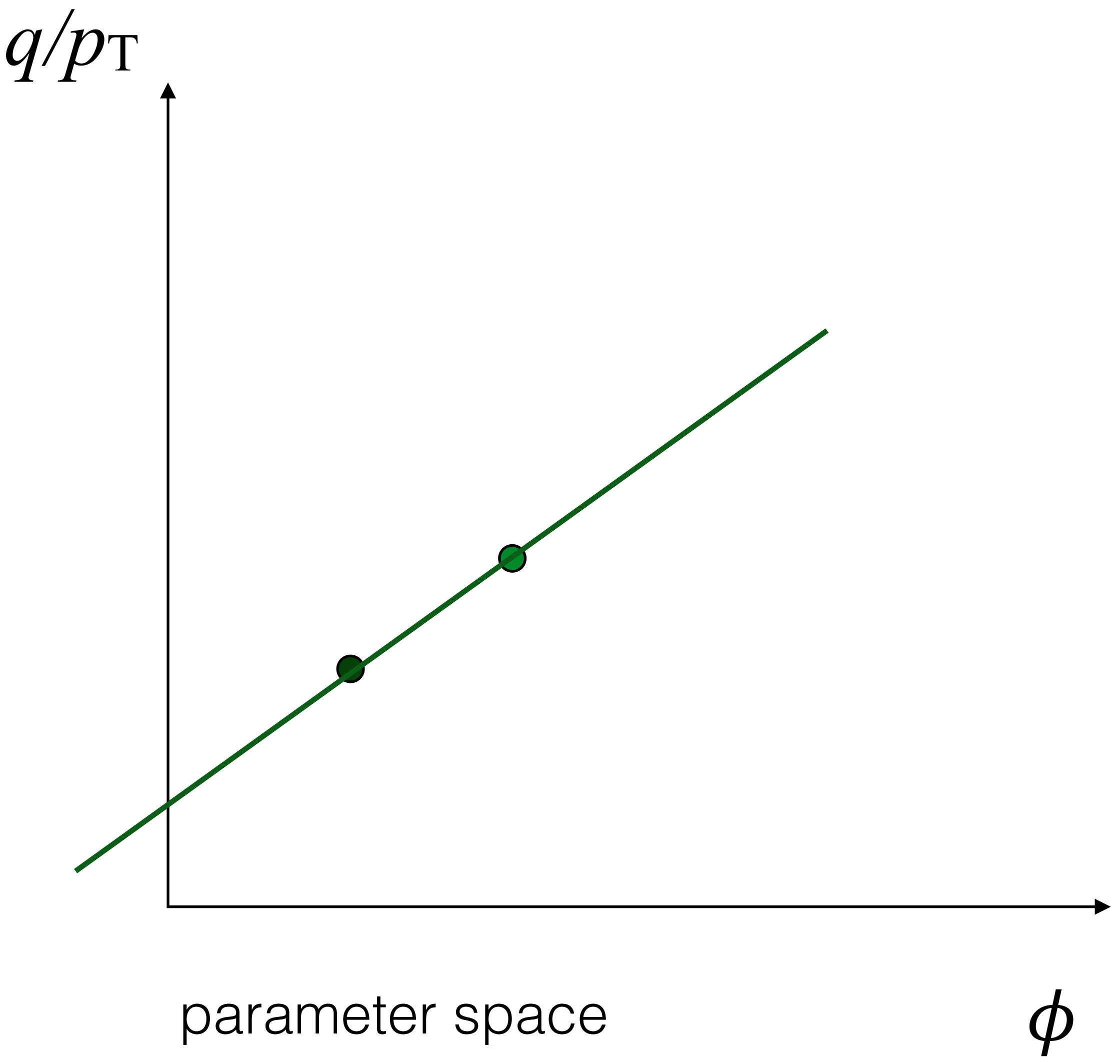


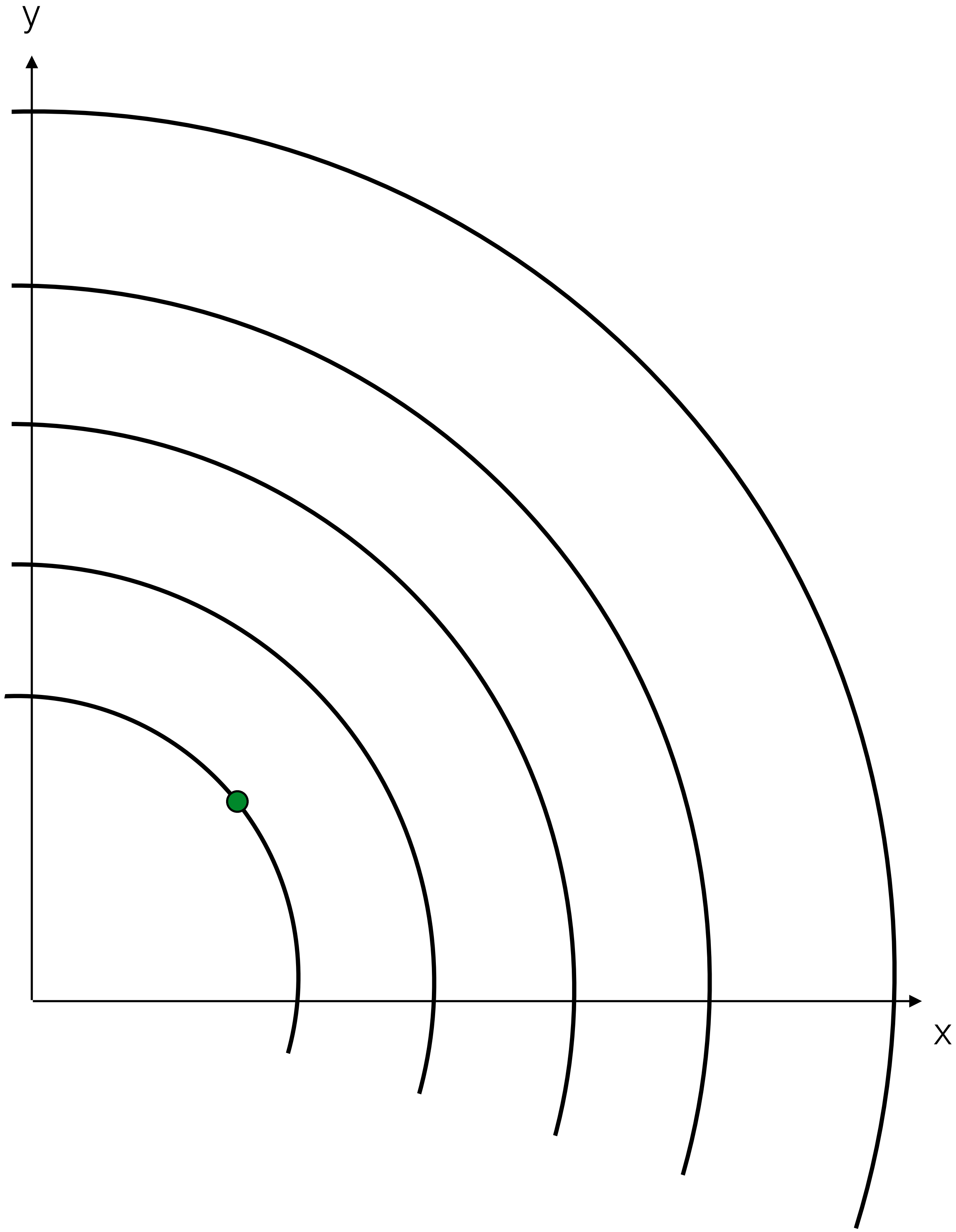


Global pattern recognition strategies

- Conformal mapping : Hough transform
  - transform your track hits in the x, y space

$$\mathbf{q} = (\cancel{d}_0, \cancel{z}_0, \phi, \cancel{\theta}, q/p_{\mathrm{T}})$$

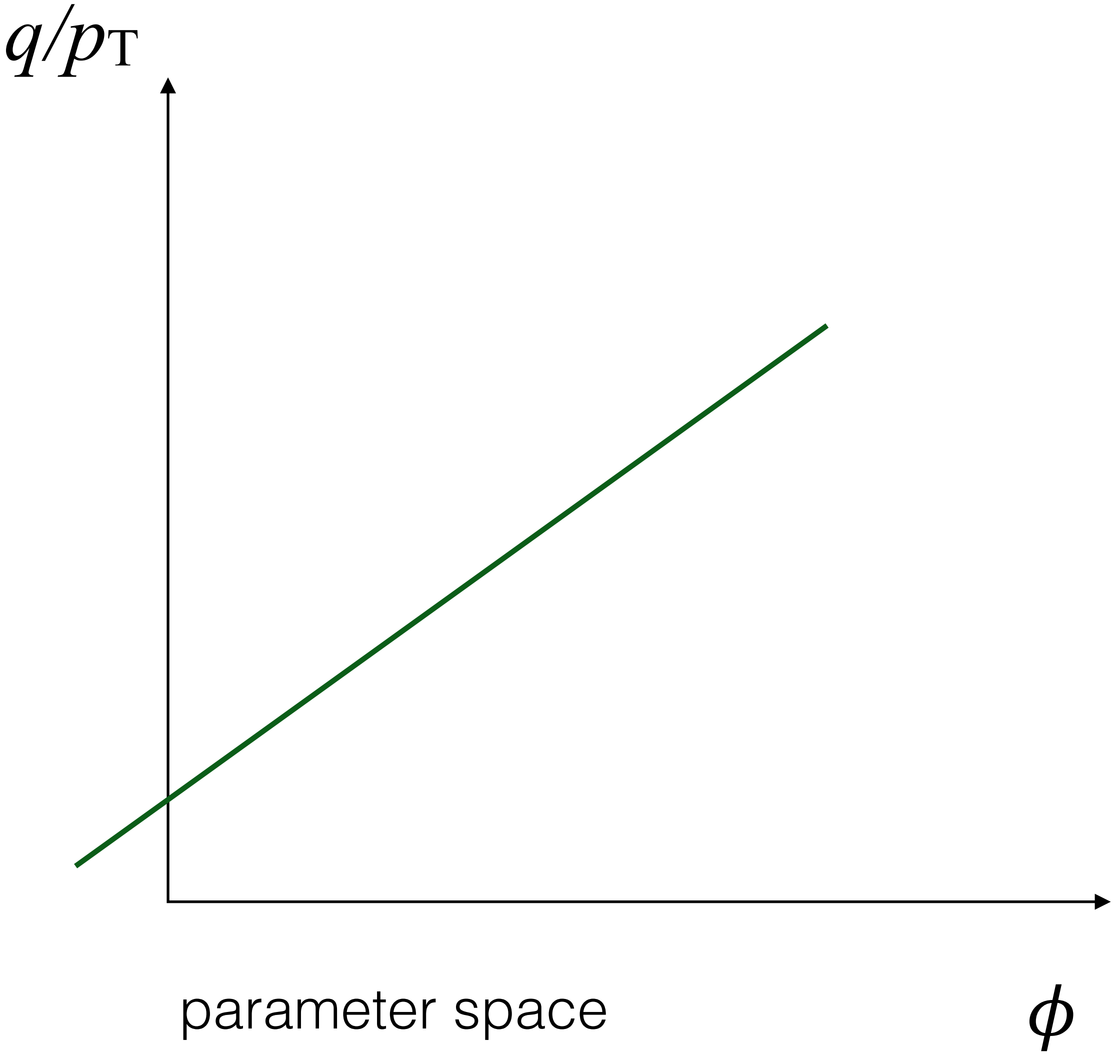


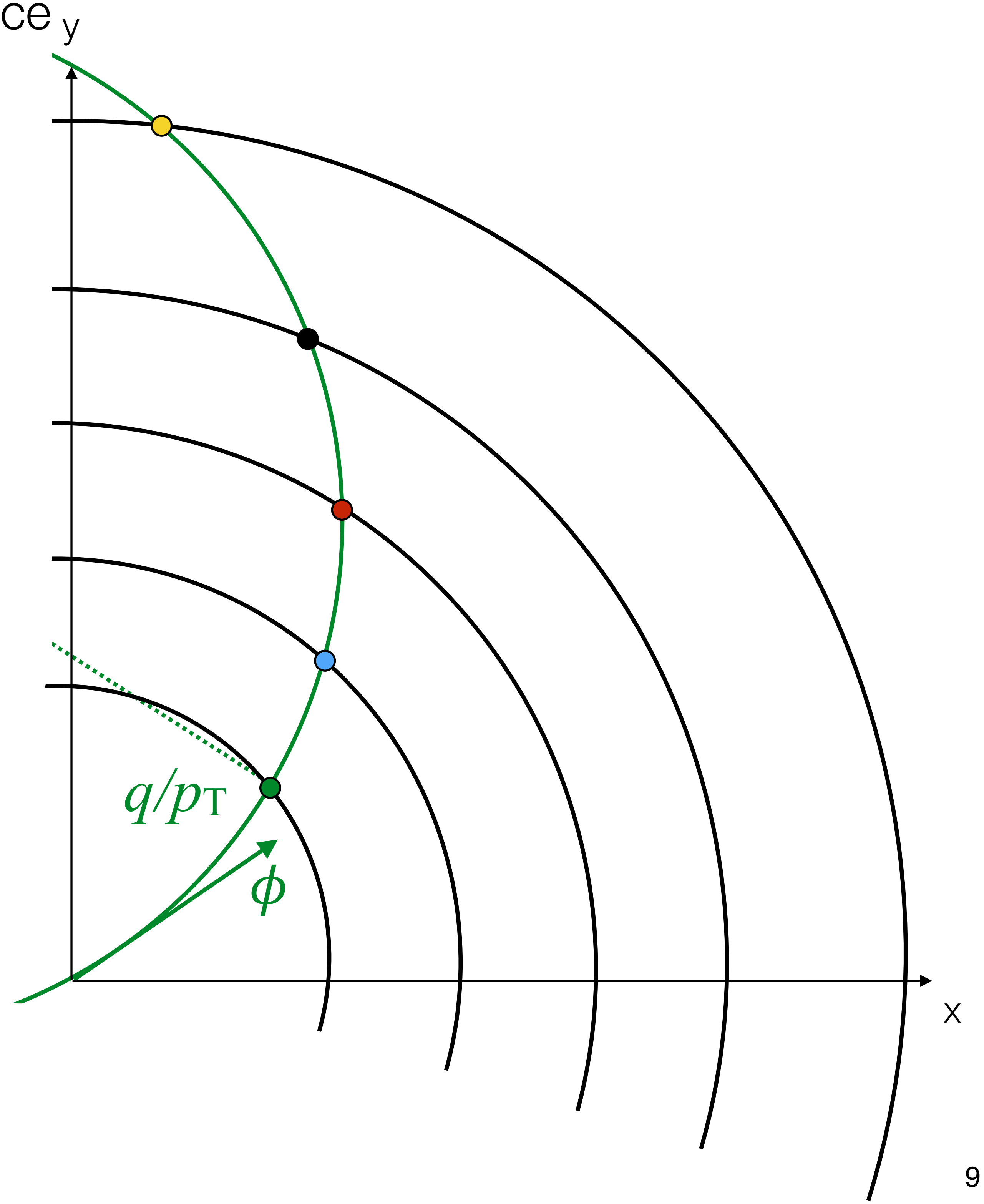


Hough transform

- Conformal mapping
  - transform your track hits from the x, y space

$$\mathbf{q} = (\cancel{d_0}, \cancel{z_0}, \phi, \cancel{\theta}, q/p_{\mathrm{T}})$$

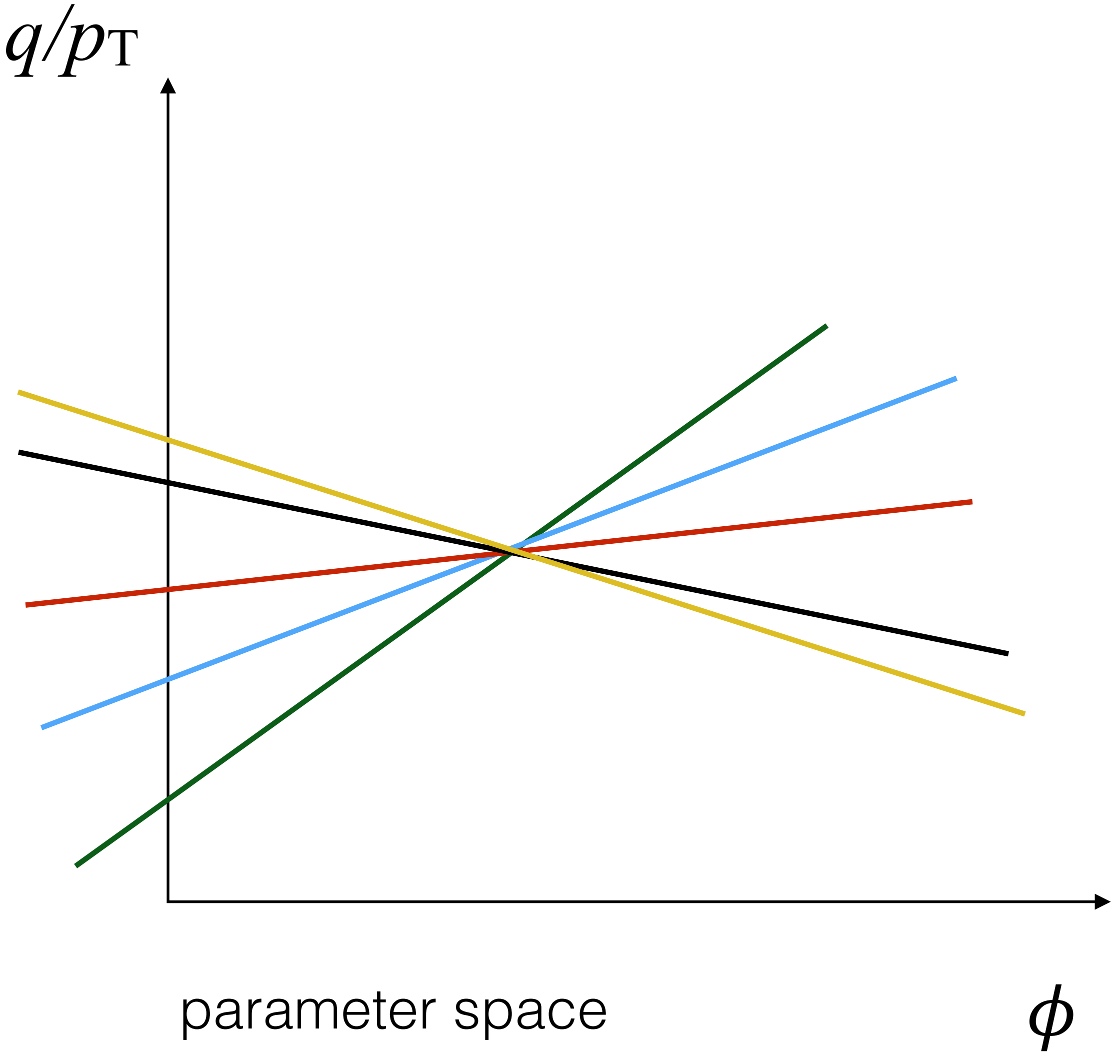


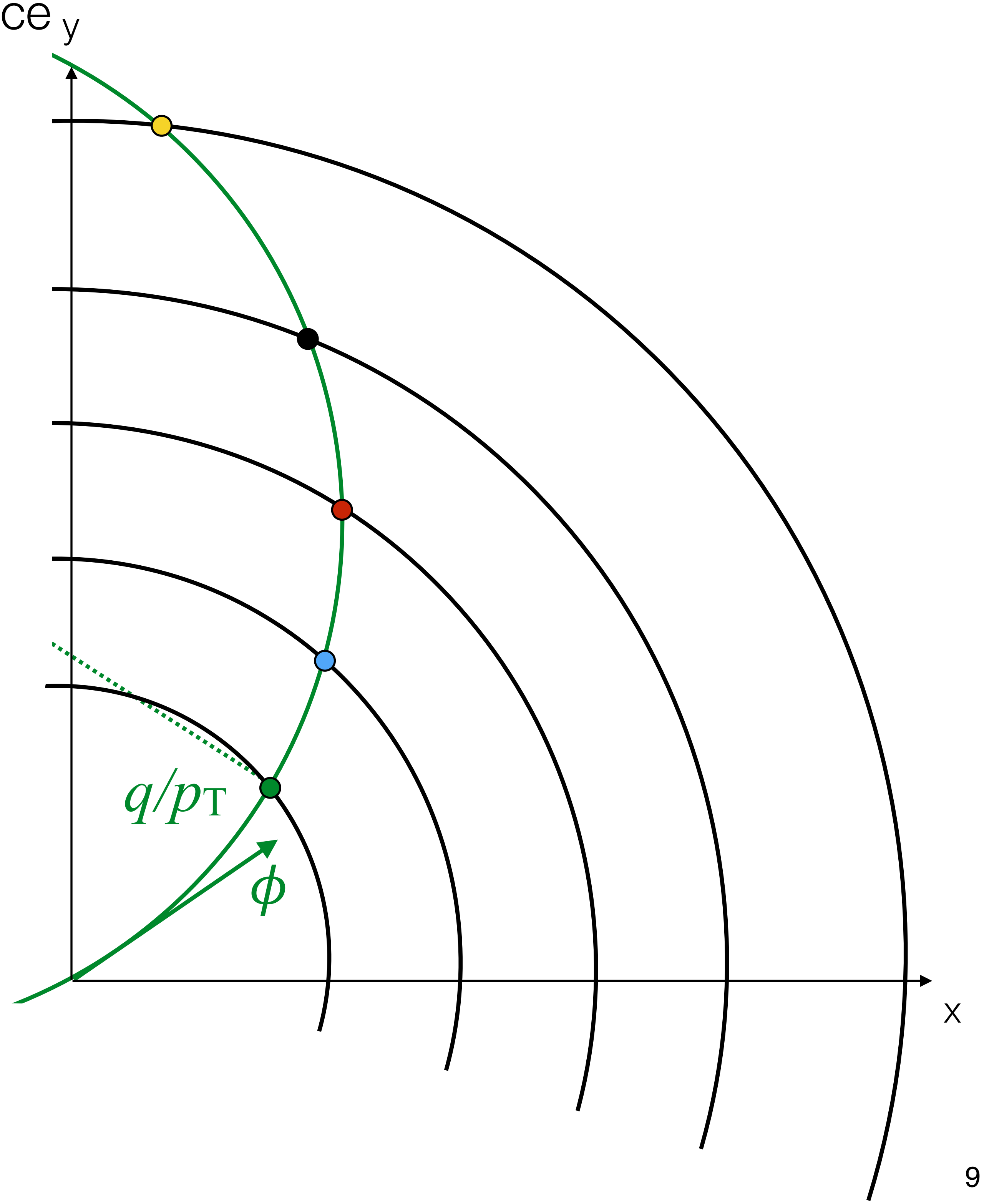


Hough transform

- Conformal mapping
  - transform your track hits from the x, y space

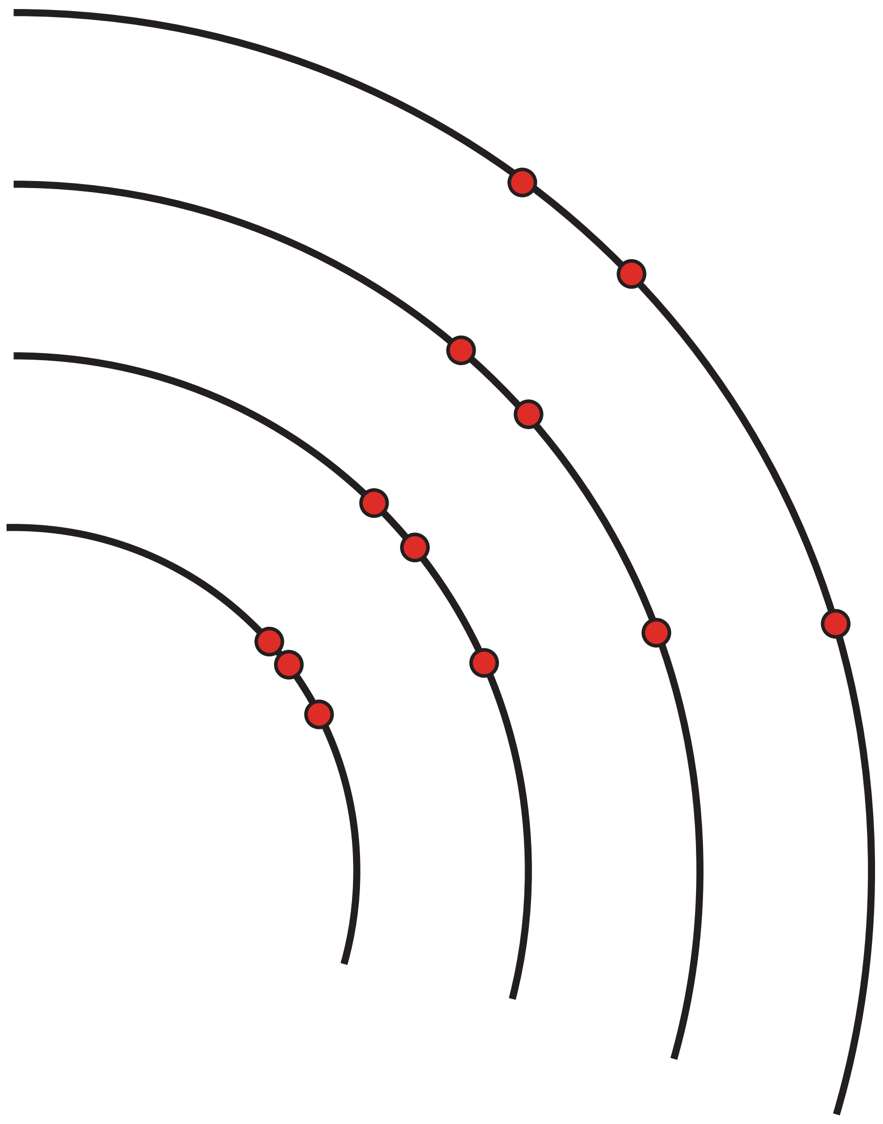
$$\mathbf{q} = (\cancel{d}_0, \cancel{z}_0, \phi, \cancel{\theta}, q/p_{\mathrm{T}})$$





A different approach: seeding & following

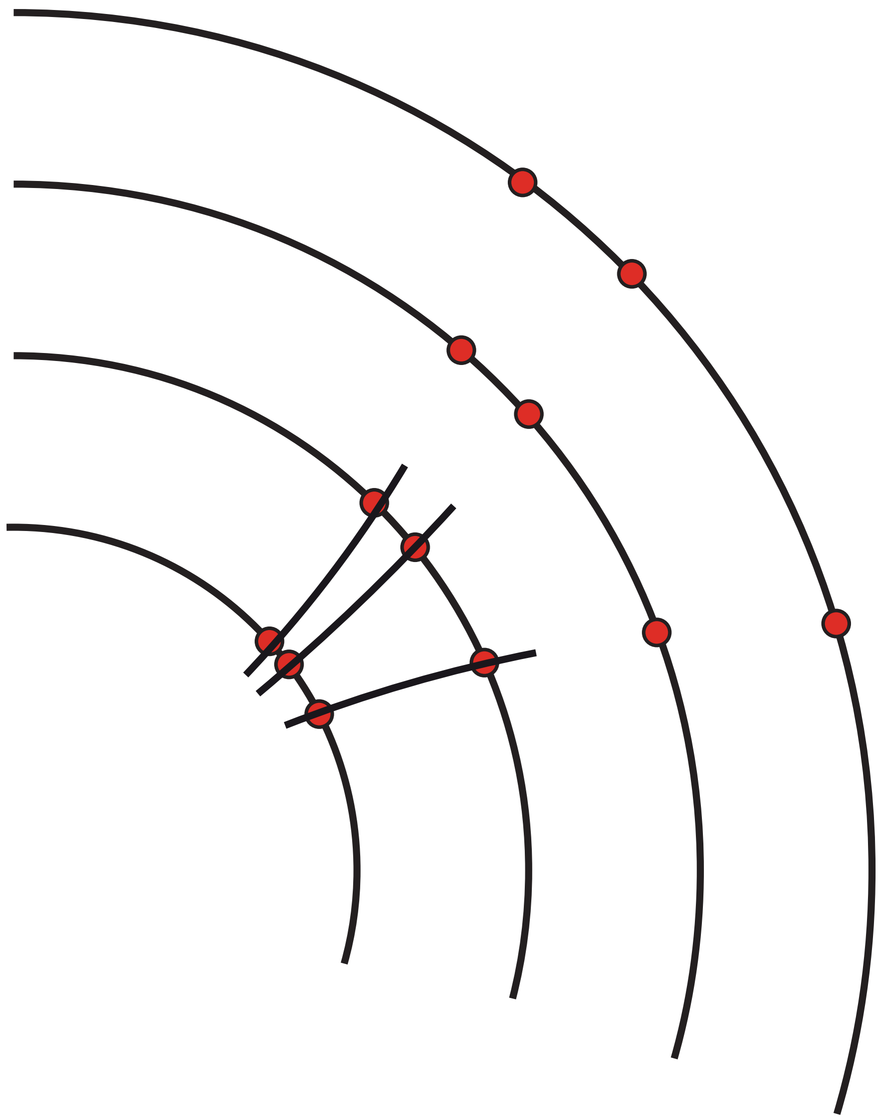
- Start of many track finding algorithms is the building of track seeds
  - groups of 2 or 3 measurements that are compatible with a crude track hypothesis
  - seeds are used to build roads to find track candidates



A. Salzburger - Track and Vertex Reconstruction - Part 2 - HCPSS Aug 11-22, 2014

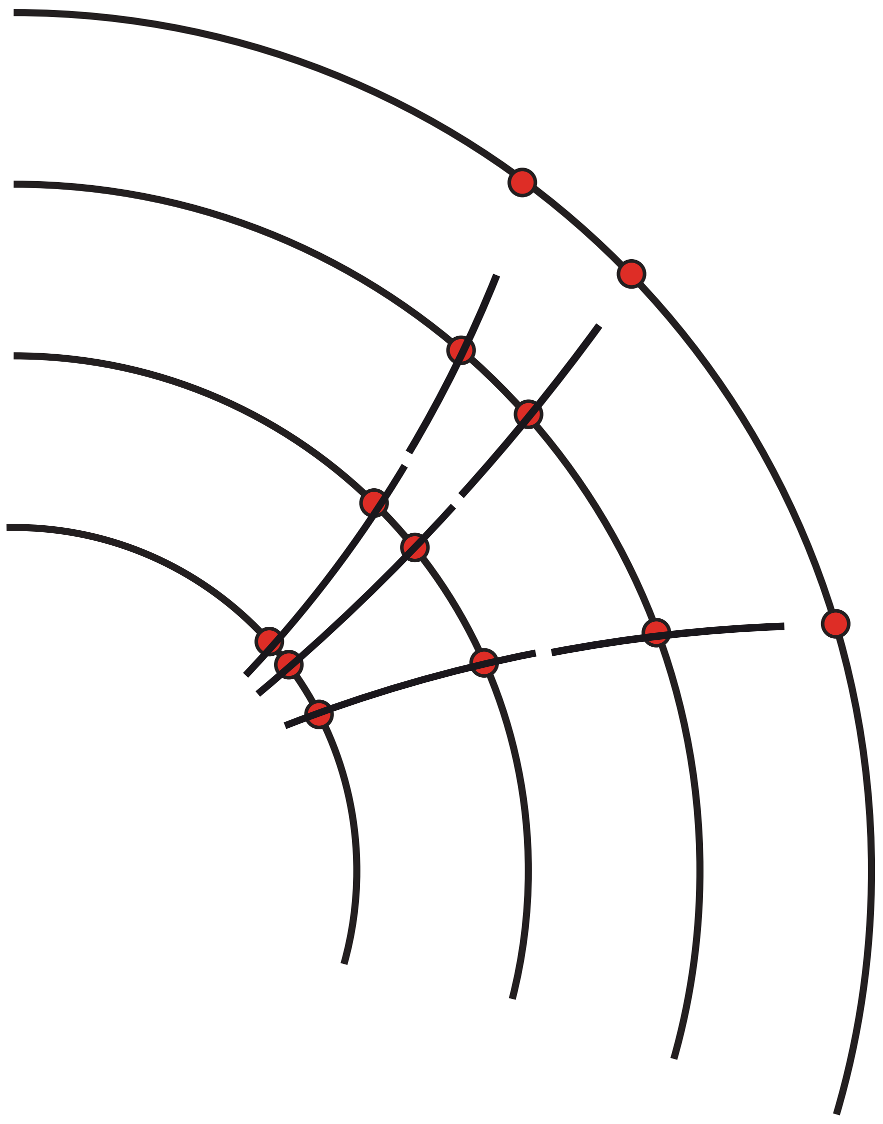
A different approach: seeding & following

- Start of many track finding algorithms is the building of track seeds
  - groups of 2 or 3 measurements that are compatible with a crude track hypothesis
  - seeds are used to build roads to find track candidates



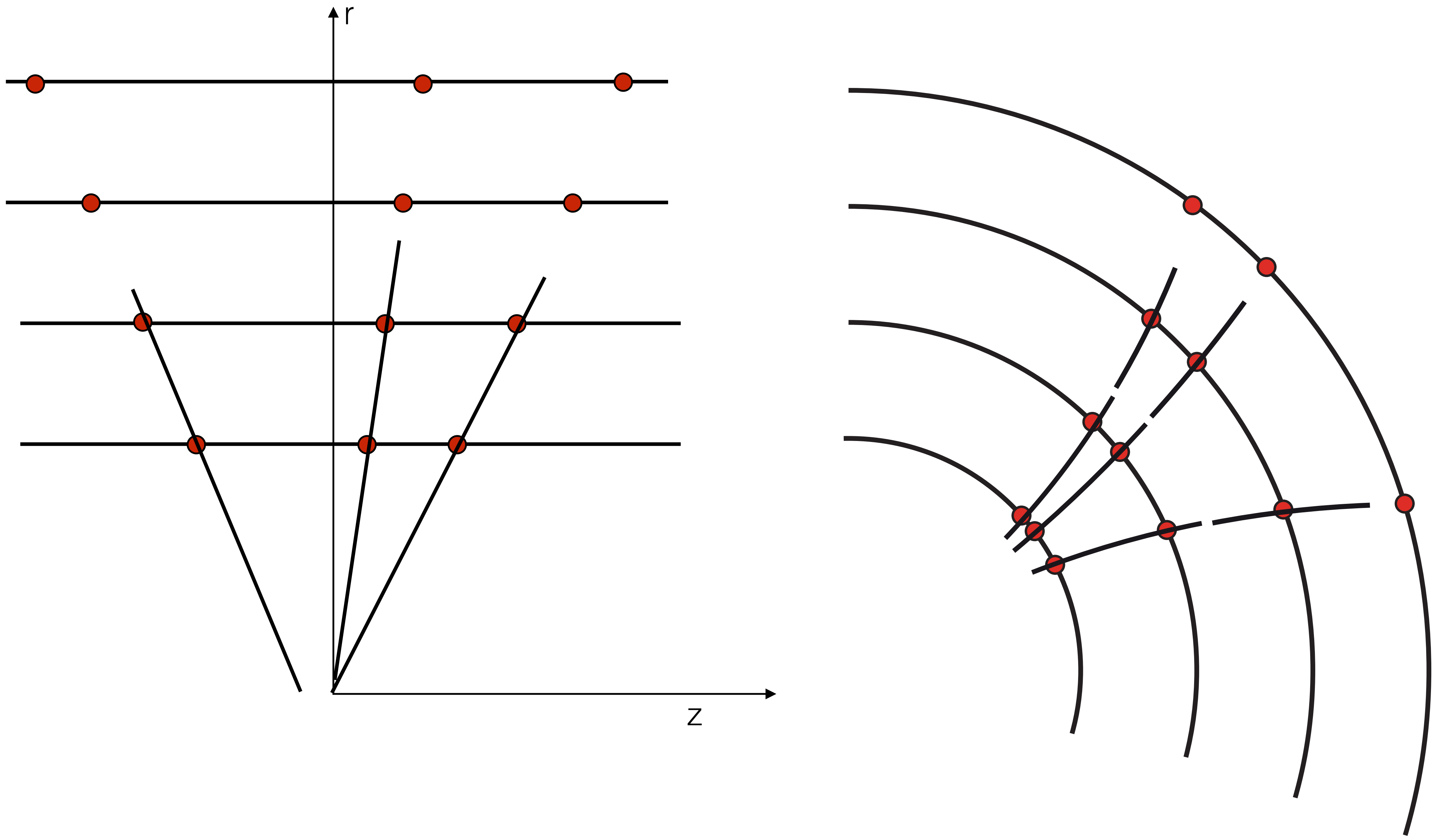
A different approach: seeding & following

- Start of many track finding algorithms is the building of track seeds
  - groups of 2 or 3 measurements that are compatible with a crude track hypothesis
  - seeds are used to build roads to find track candidates



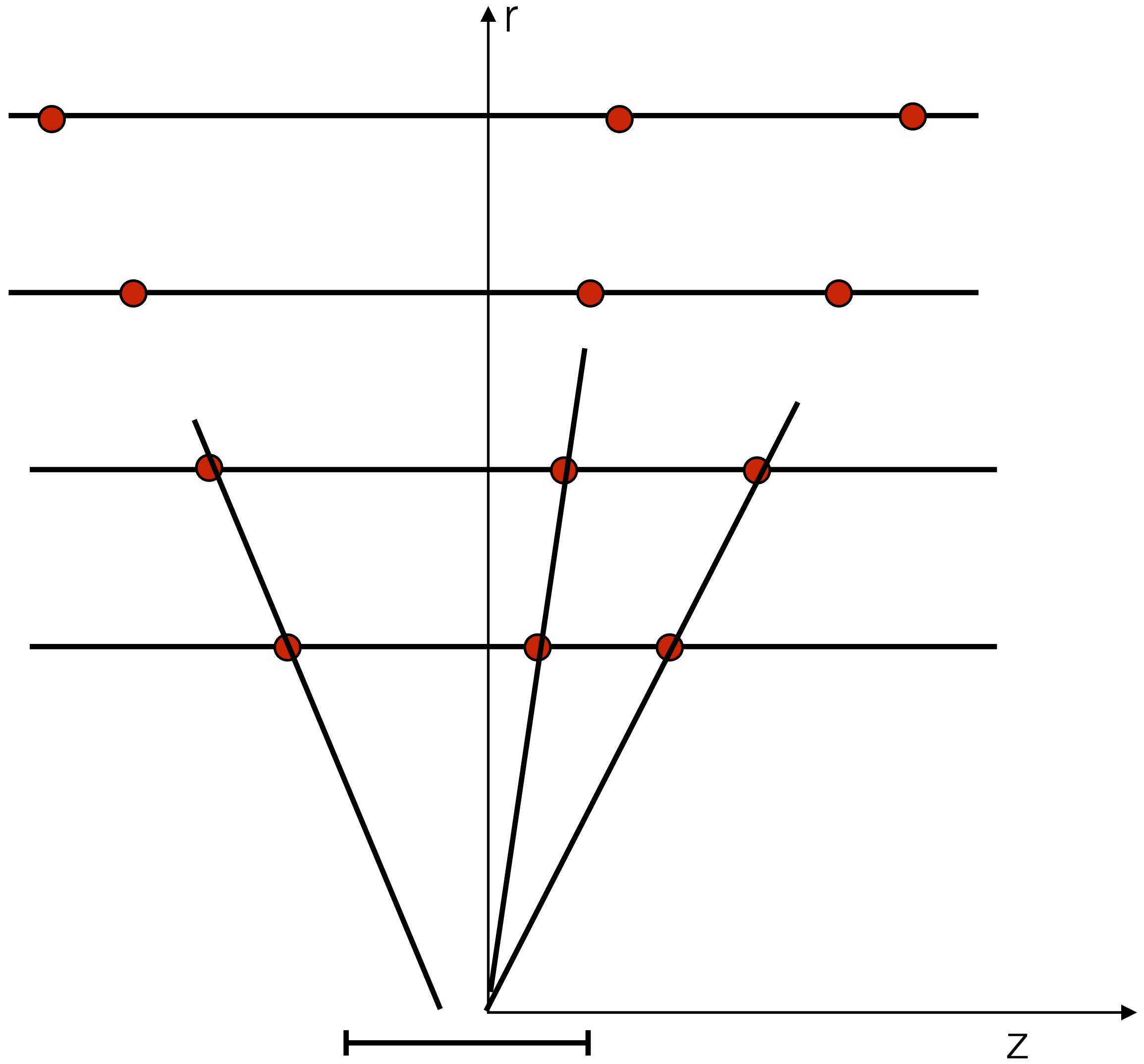
A different approach: seeding & following

- Start of many track finding algorithms is the building of track seeds
  - groups of 2 or 3 measurements that are compatible with a crude track hypothesis
  - seeds are used to build roads to find track candidates

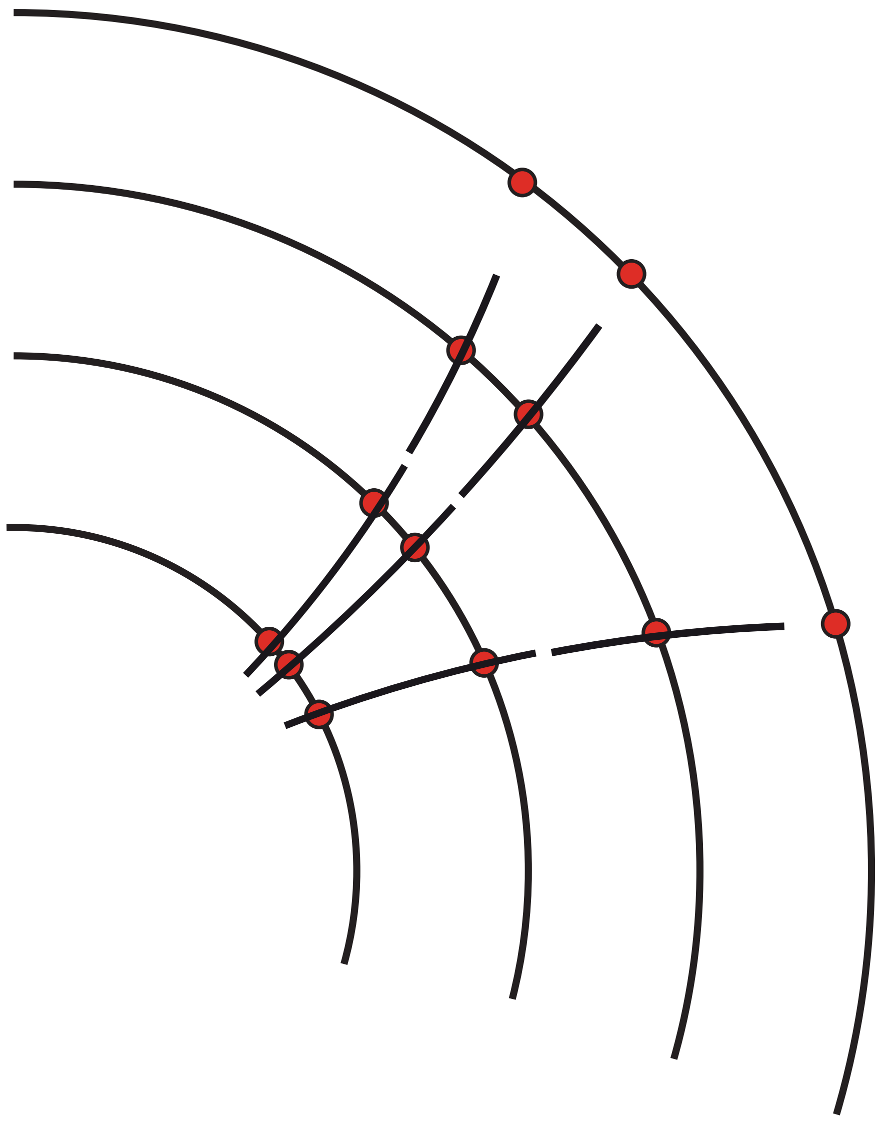


A different approach: seeding & following

- Start of many track finding algorithms is the building of track seeds
  - groups of 2 or 3 measurements that are compatible with a crude track hypothesis
  - seeds are used to build roads to find track candidates

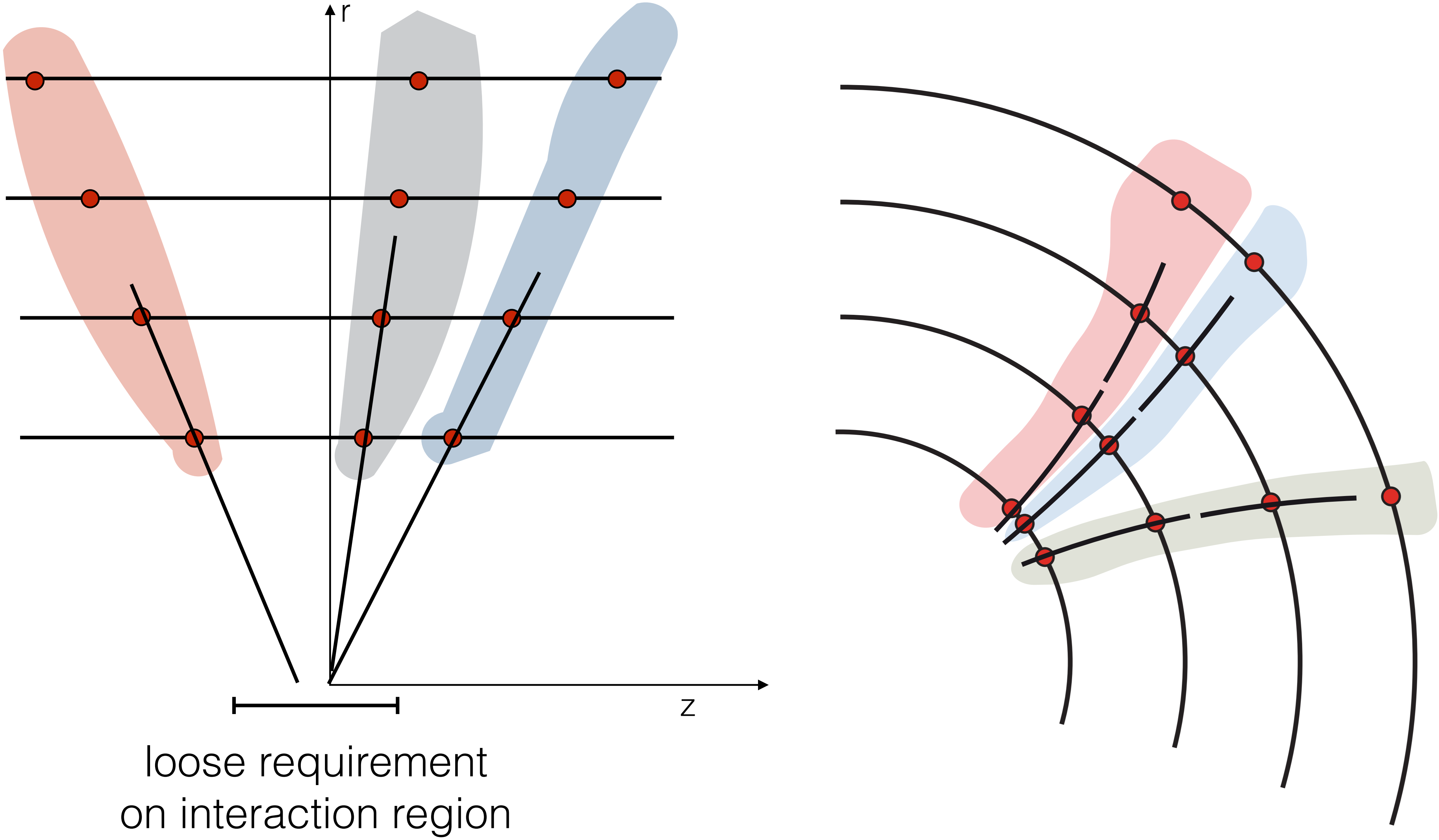


loose requirement
on interaction region



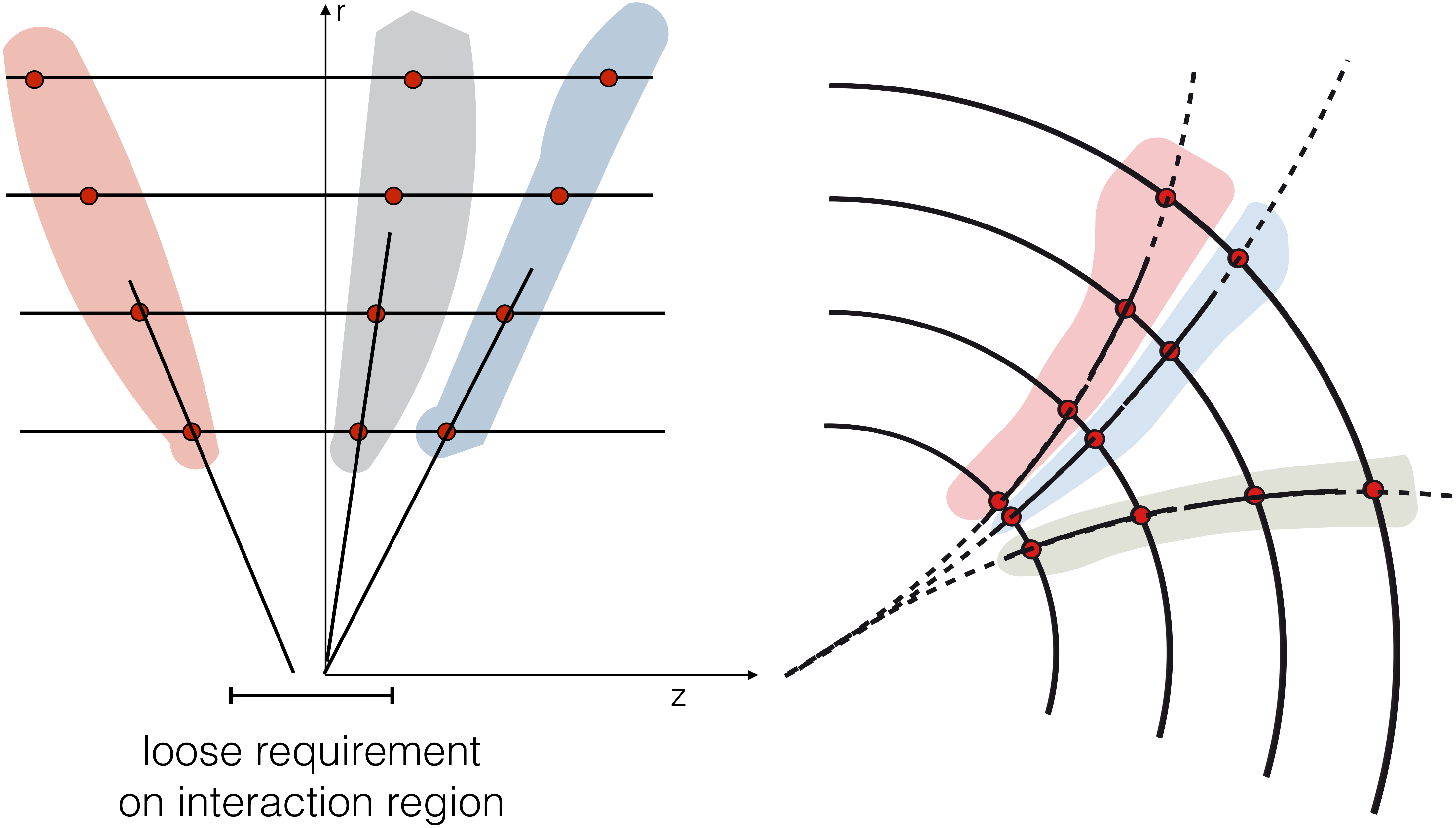
A different approach: seeding & following

- Start of many track finding algorithms is the building of track seeds
  - groups of 2 or 3 measurements that are compatible with a crude track hypothesis
  - seeds are used to build roads to find track candidates



A different approach: seeding & following

- Start of many track finding algorithms is the building of track seeds
  - groups of 2 or 3 measurements that are compatible with a crude track hypothesis
  - seeds are used to build roads to find track candidates



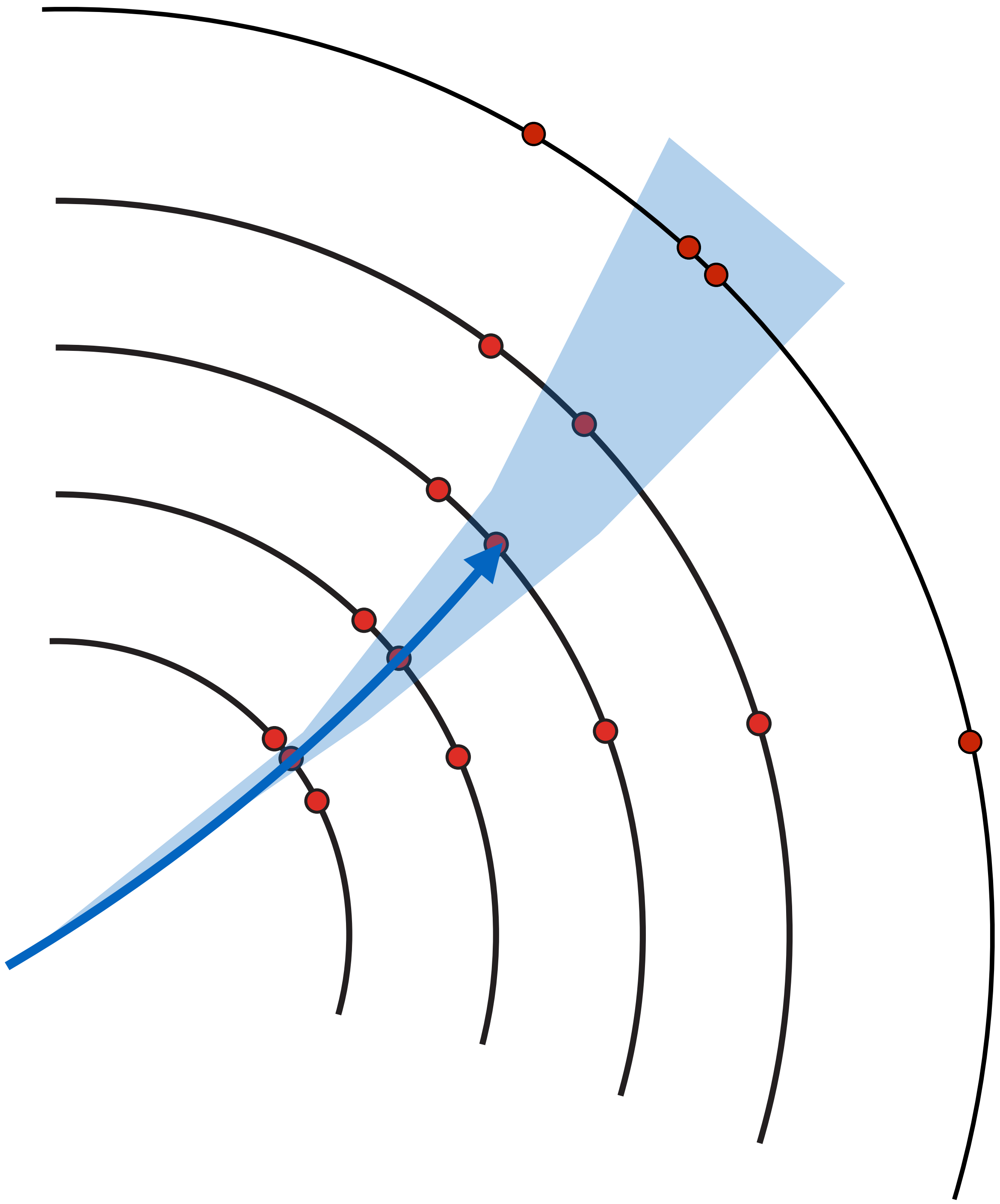
From seeds to track candidates

- The progressive filter
  - roads are built from track seeds and define a search window
  - **following** the road direction to find hits that are ***compatible*** with the track

    needs a measure to define compatibility
  - a found hit used to ***update*** the track to follow to the next measurement layer

    needs a mechanism to update a track hypothesis
  - multiple hypothesis can be tested for one layer
  - only one track hypothesis is followed further

    needs a measure which candidate is better



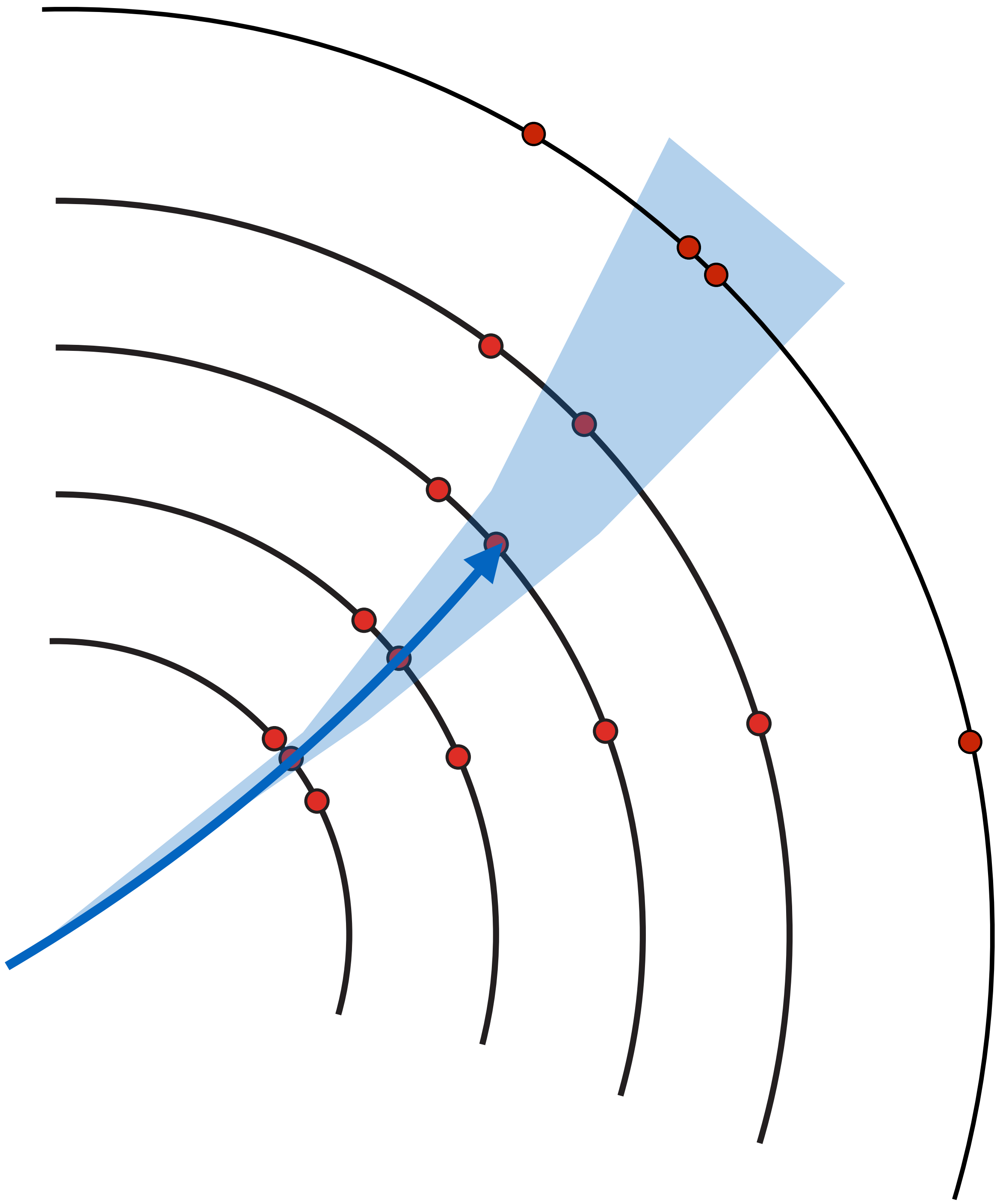
From seeds to track candidates

- The progressive filter
  - roads are built from track seeds and define a search window
  - **following** the road direction to find hits that are ***compatible*** with the track

    needs a measure to define compatibility
  - a found hit used to ***update*** the track to follow to the next measurement layer

    needs a mechanism to update a track hypothesis
  - multiple hypothesis can be tested for one layer
  - only one track hypothesis is followed further

    needs a measure which candidate is better



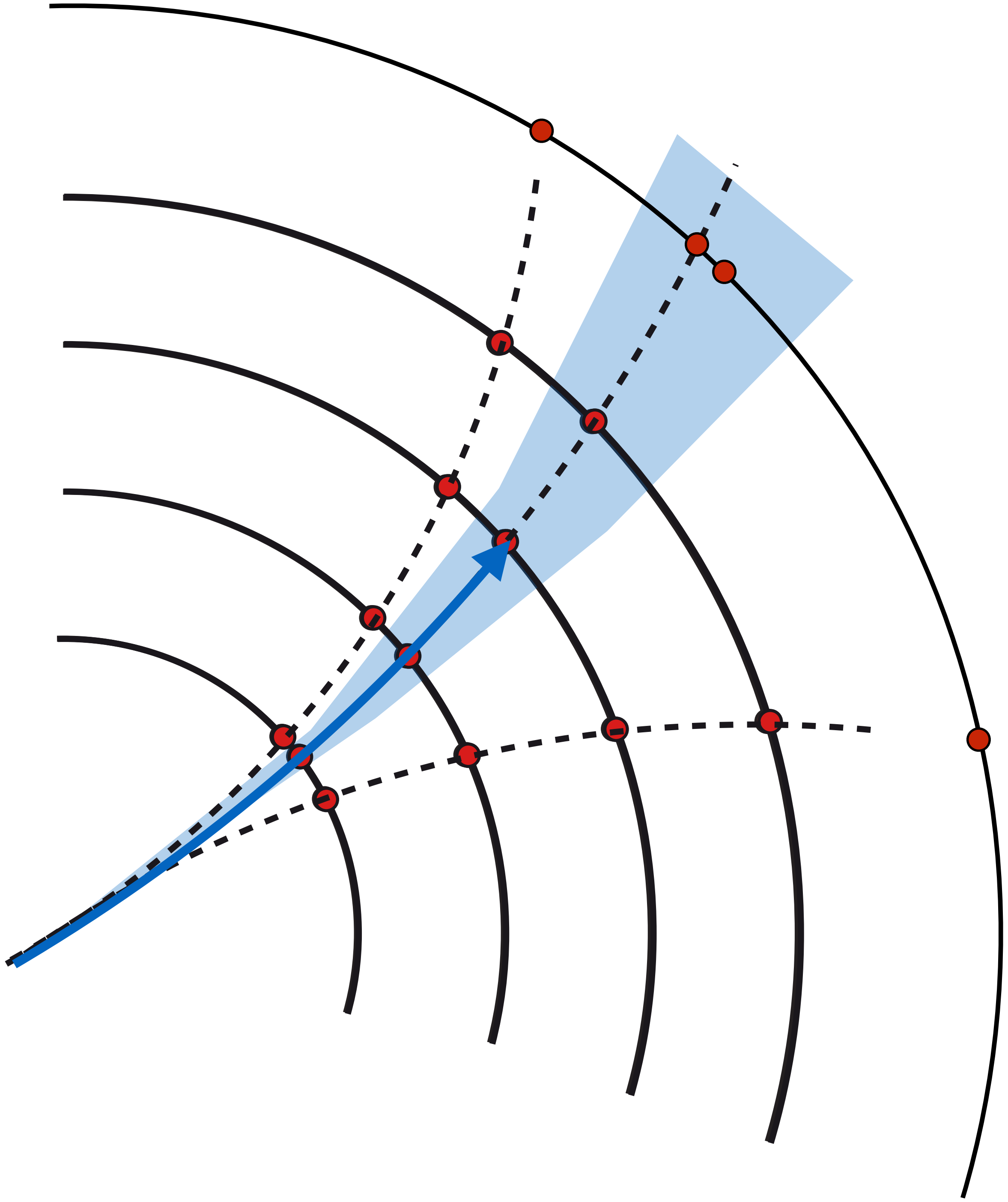
From seeds to track candidates

- The progressive filter
  - roads are built from track seeds and define a search window
  - **following** the road direction to find hits that are ***compatible*** with the track

    needs a measure to define compatibility
  - a found hit used to ***update*** the track to follow to the next measurement layer

    needs a mechanism to update a track hypothesis
  - multiple hypothesis can be tested for one layer
  - only one track hypothesis is followed further

    needs a measure which candidate is better



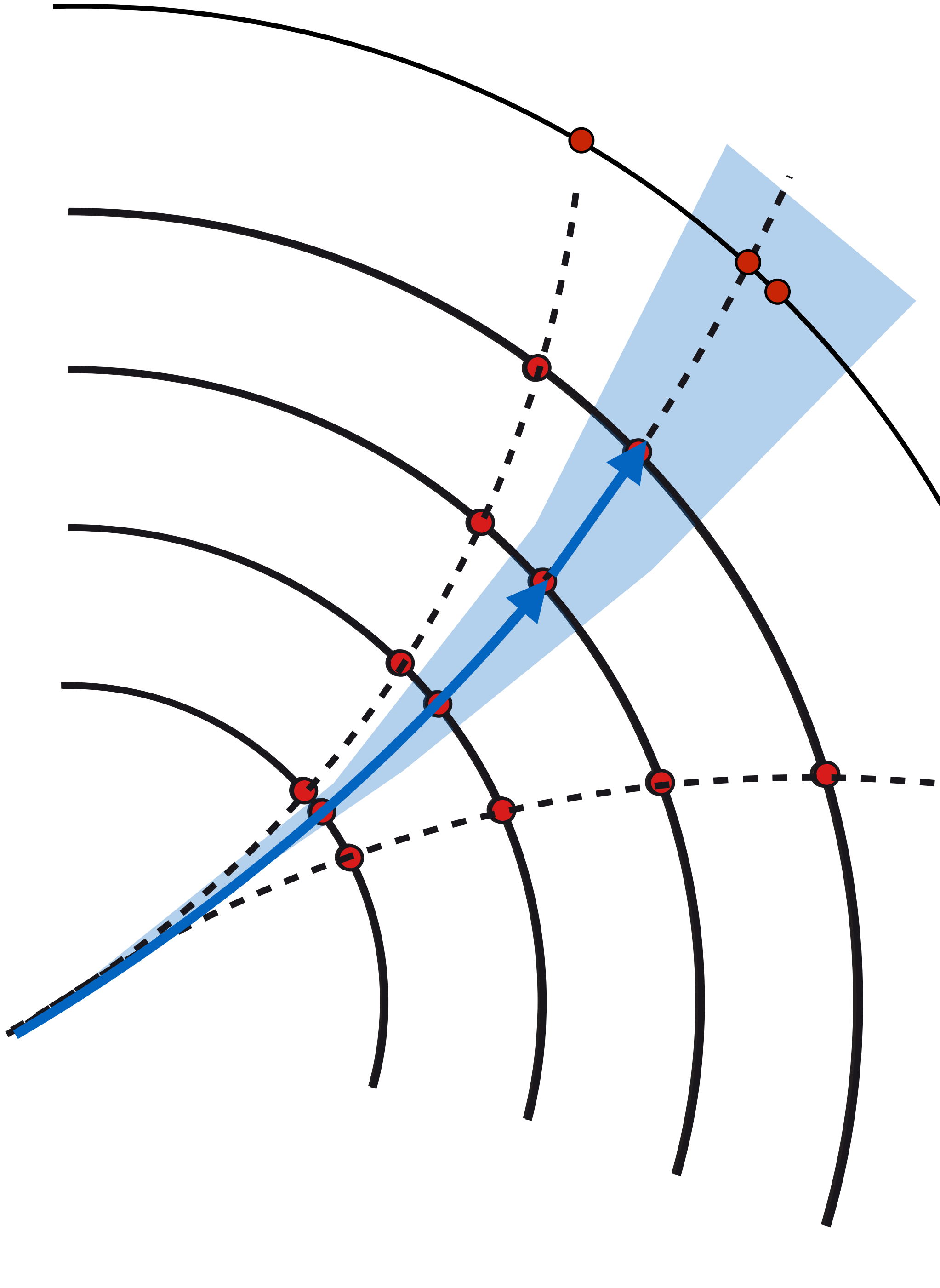
From seeds to track candidates

- The progressive filter
  - roads are built from track seeds and define a search window
  - **following** the road direction to find hits that are ***compatible*** with the track

    needs a measure to define compatibility
  - a found hit used to ***update*** the track to follow to the next measurement layer

    needs a mechanism to update a track hypothesis
  - multiple hypothesis can be tested for one layer
  - only one track hypothesis is followed further

    needs a measure which candidate is better



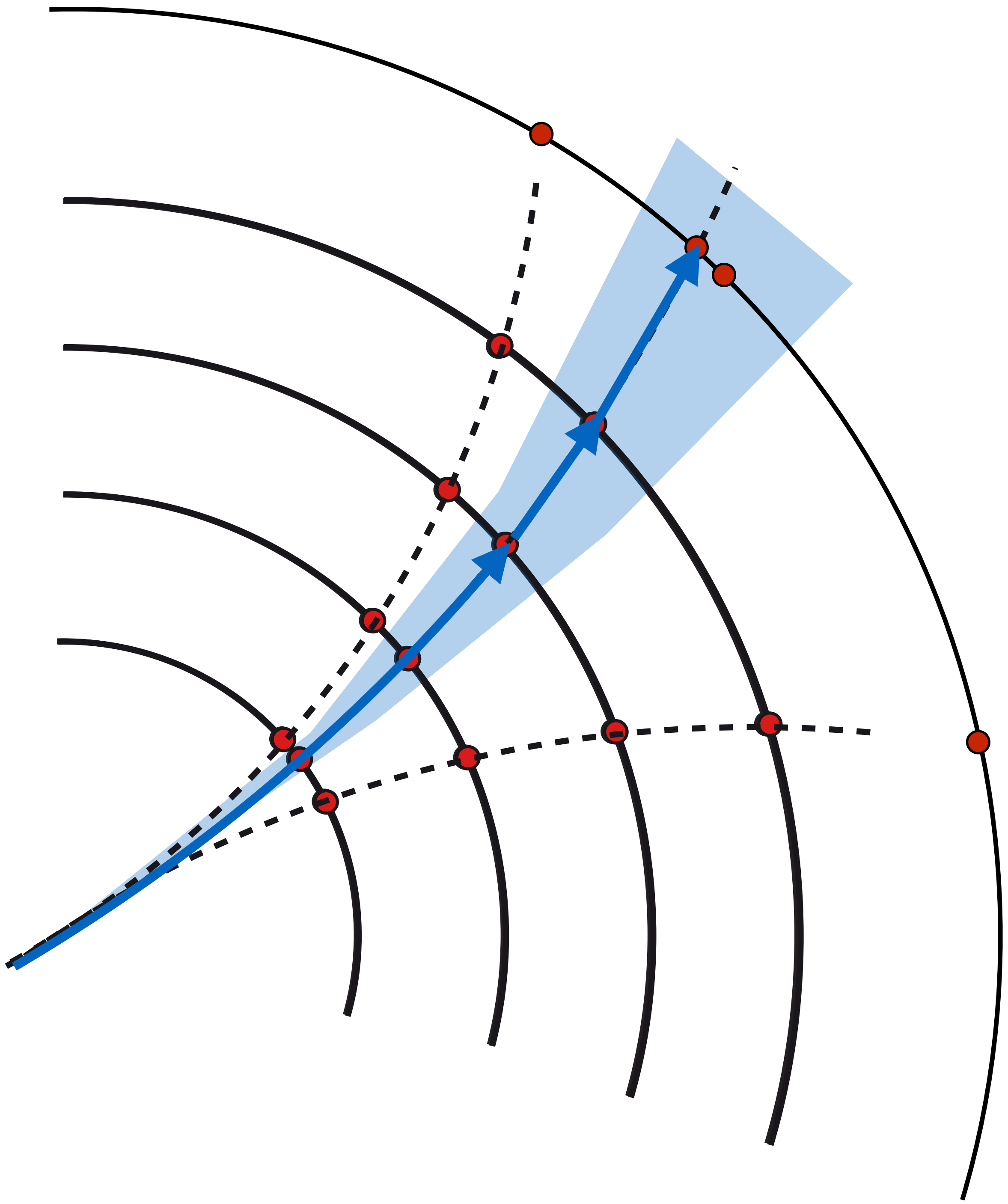
From seeds to track candidates

- The progressive filter
  - roads are built from track seeds and define a search window
  - **following** the road direction to find hits that are ***compatible*** with the track

    needs a measure to define compatibility
  - a found hit used to ***update*** the track to follow to the next measurement layer

    needs a mechanism to update a track hypothesis
  - multiple hypothesis can be tested for one layer
  - only one track hypothesis is followed further

    needs a measure which candidate is better



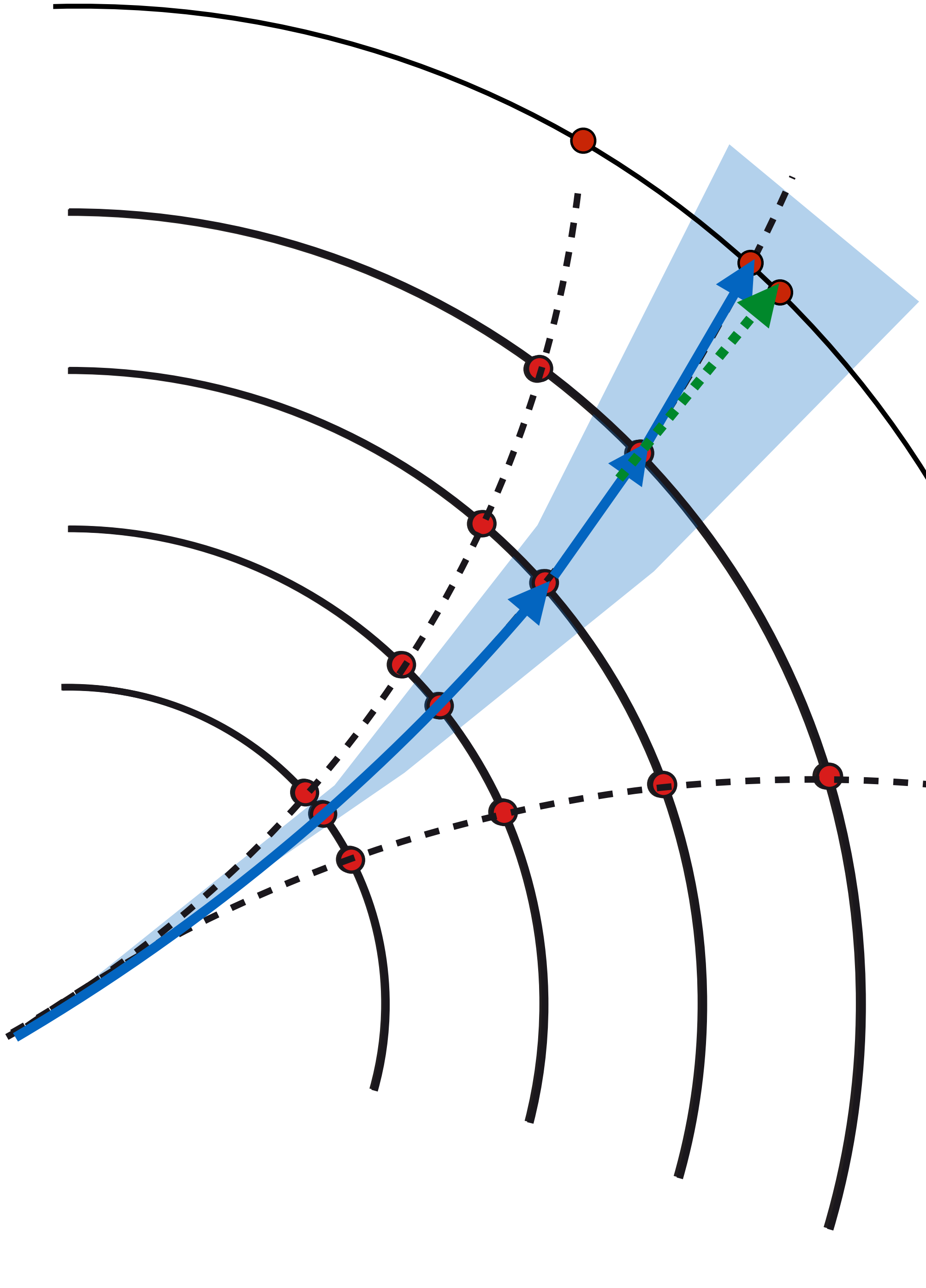
From seeds to track candidates

- The progressive filter
  - roads are built from track seeds and define a search window
  - **following** the road direction to find hits that are ***compatible*** with the track

    needs a measure to define compatibility
  - a found hit used to ***update*** the track to follow to the next measurement layer

    needs a mechanism to update a track hypothesis
  - multiple hypothesis can be tested for one layer
  - only one track hypothesis is followed further

    needs a measure which candidate is better



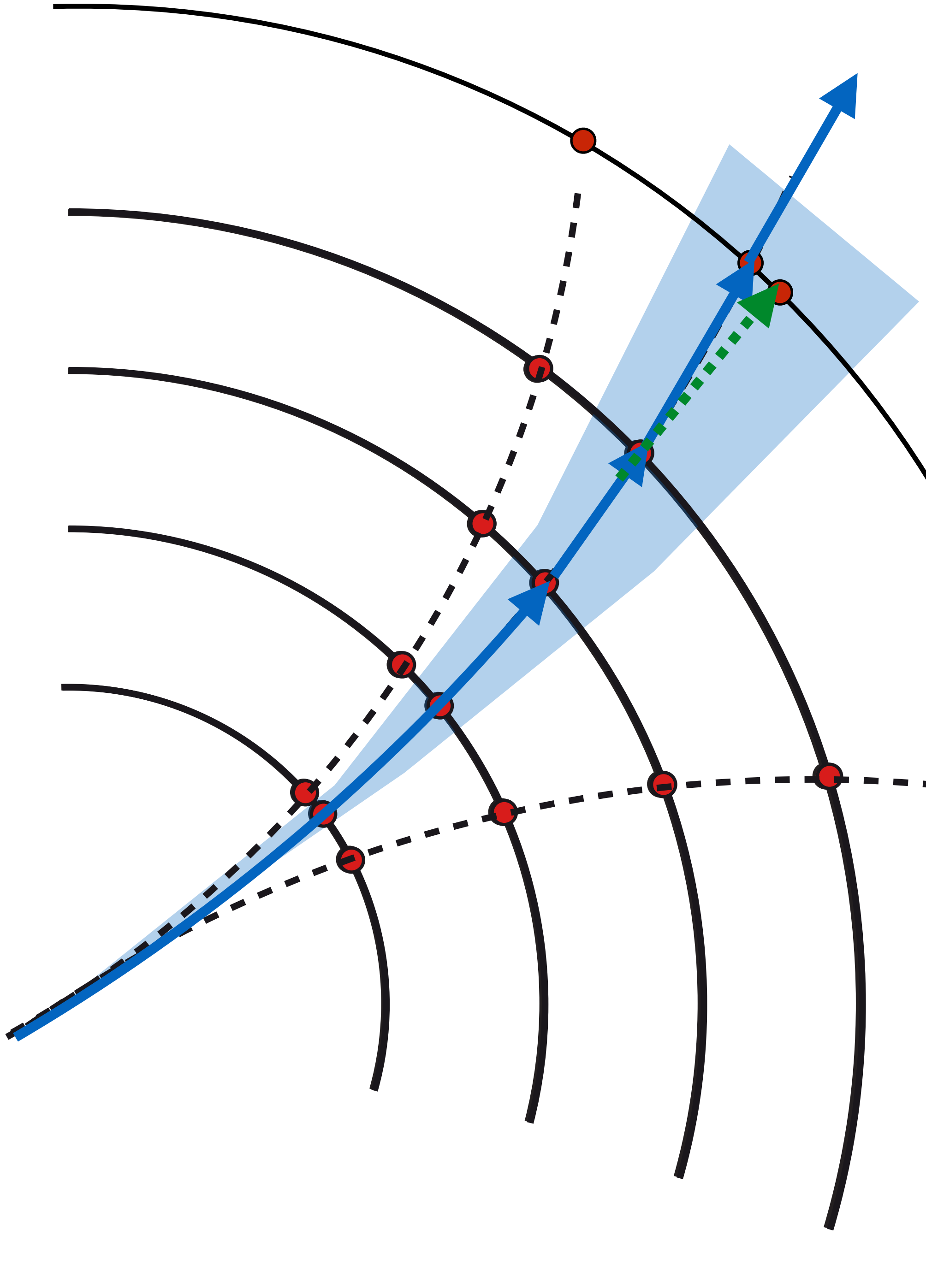
From seeds to track candidates

- The progressive filter
  - roads are built from track seeds and define a search window
  - **following** the road direction to find hits that are ***compatible*** with the track

    needs a measure to define compatibility
  - a found hit used to ***update*** the track to follow to the next measurement layer

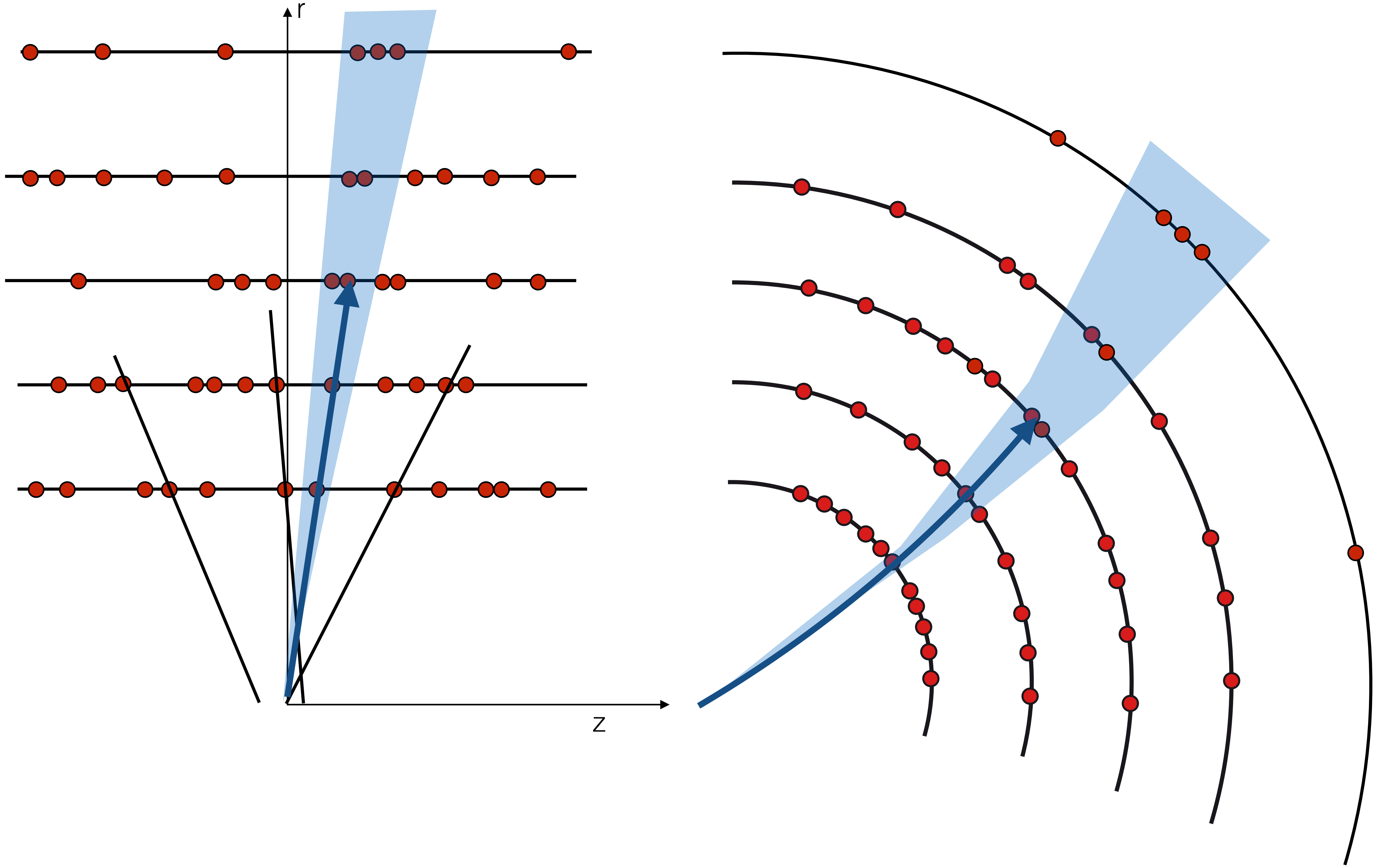
    needs a mechanism to update a track hypothesis
  - multiple hypothesis can be tested for one layer
  - only one track hypothesis is followed further

    needs a measure which candidate is better



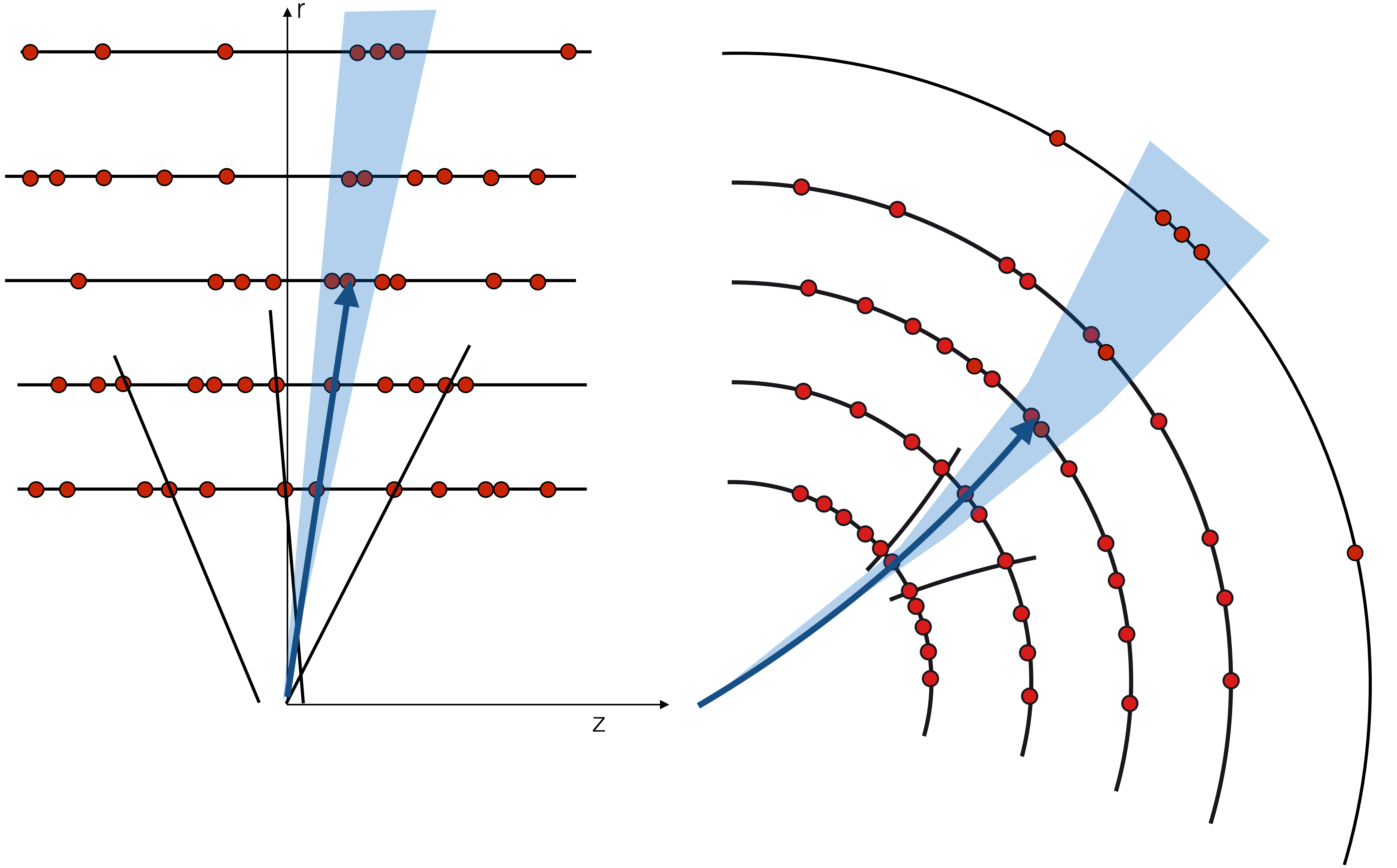
From seeds to track candidates

- Dense environments create problems for the progressive filter
  - there may not always be one obvious path to be followed: The combinatorial filter



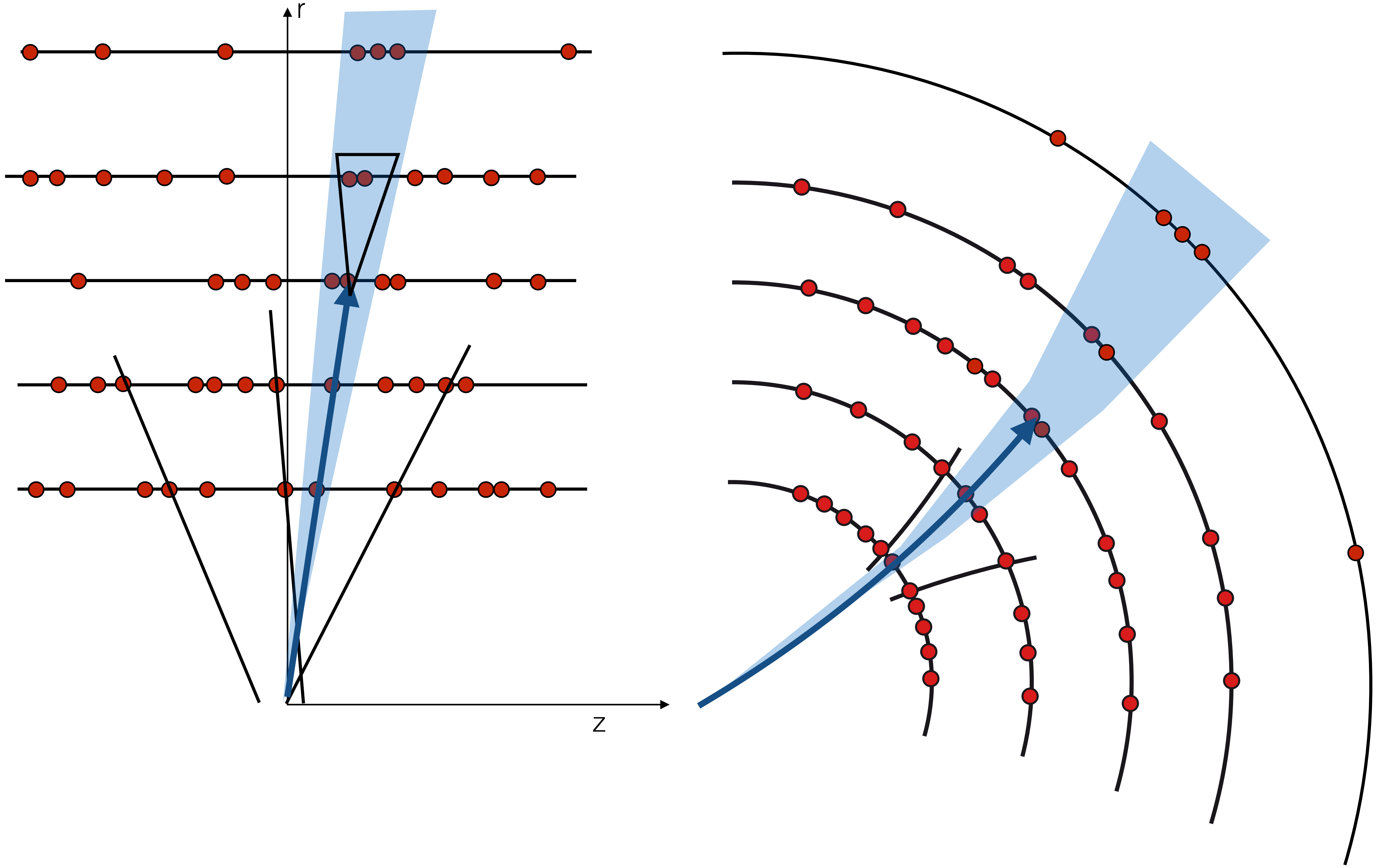
From seeds to track candidates

- Dense environments create problems for the progressive filter
  - there may not always be one obvious path to be followed: The combinatorial filter



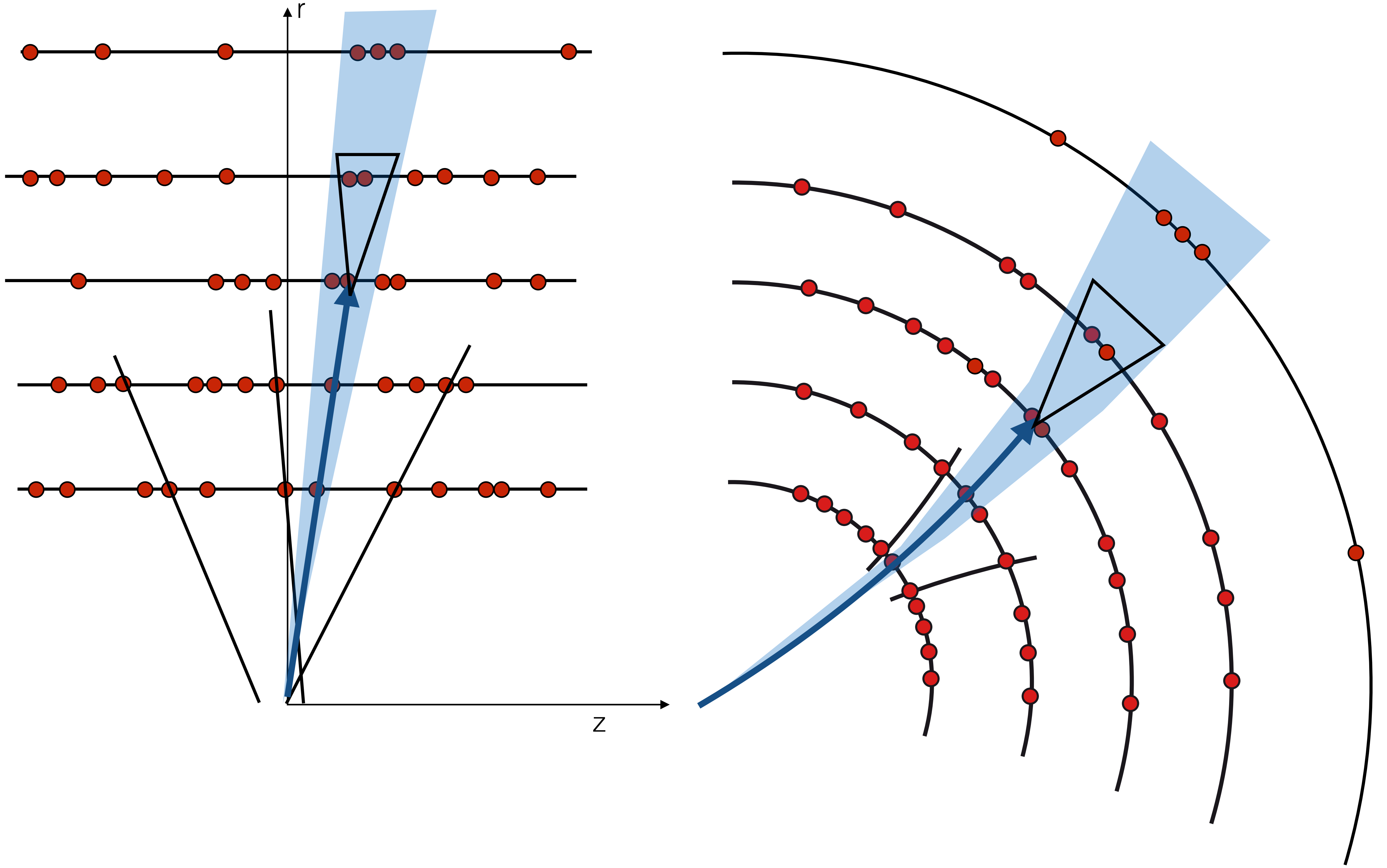
From seeds to track candidates

- Dense environments create problems for the progressive filter
  - there may not always be one obvious path to be followed: The combinatorial filter



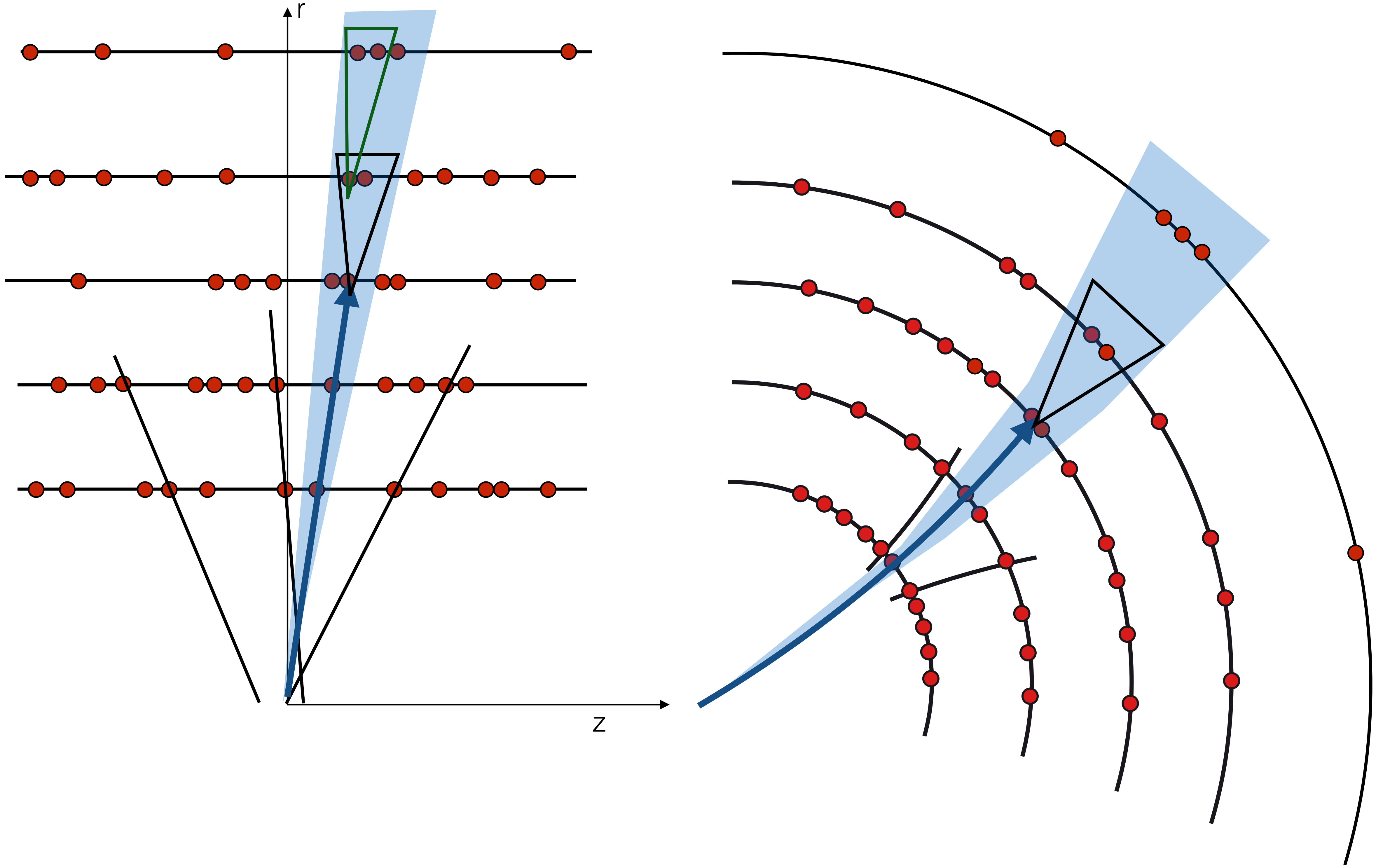
From seeds to track candidates

- Dense environments create problems for the progressive filter
  - there may not always be one obvious path to be followed: The combinatorial filter



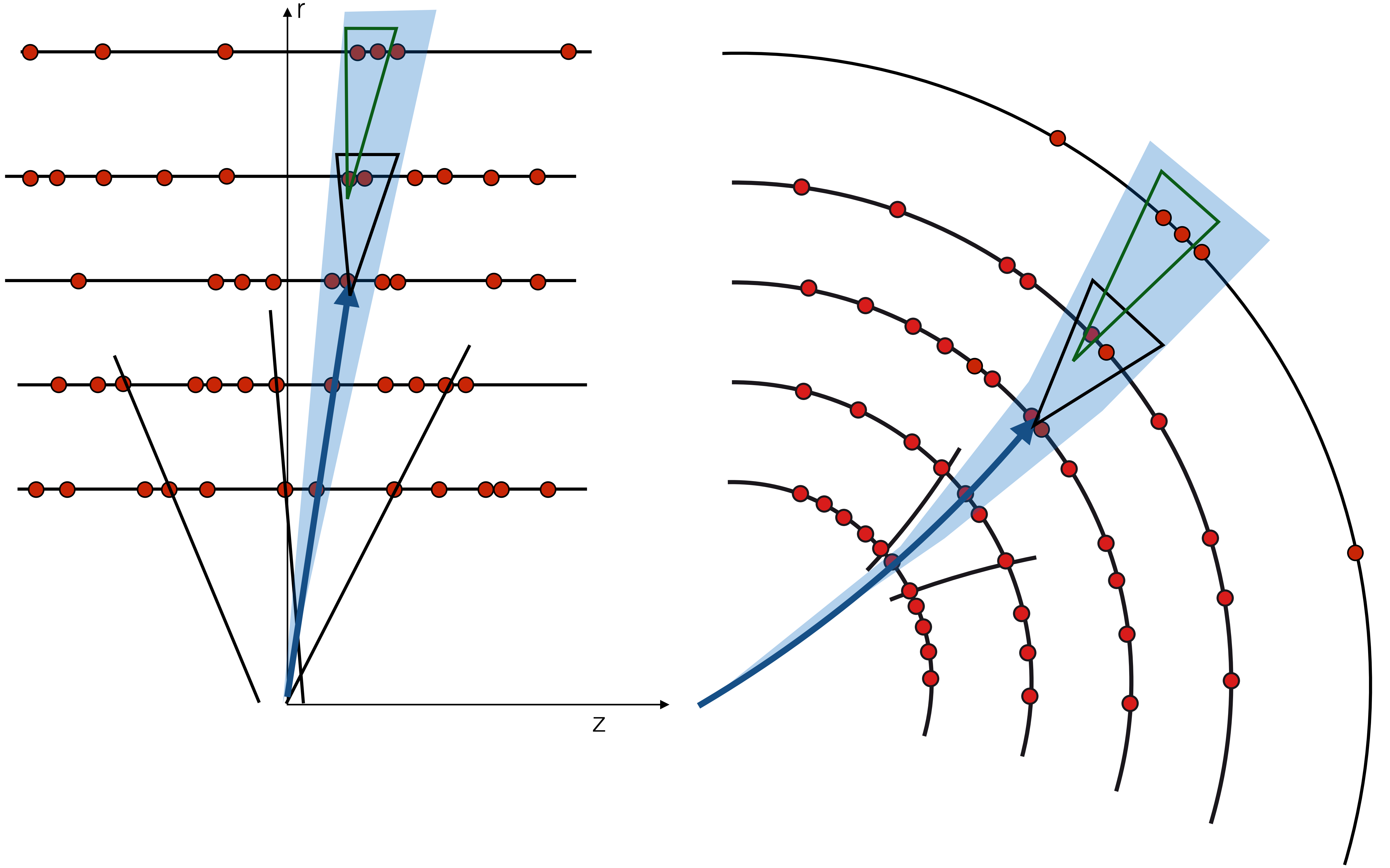
From seeds to track candidates

- Dense environments create problems for the progressive filter
  - there may not always be one obvious path to be followed: The combinatorial filter



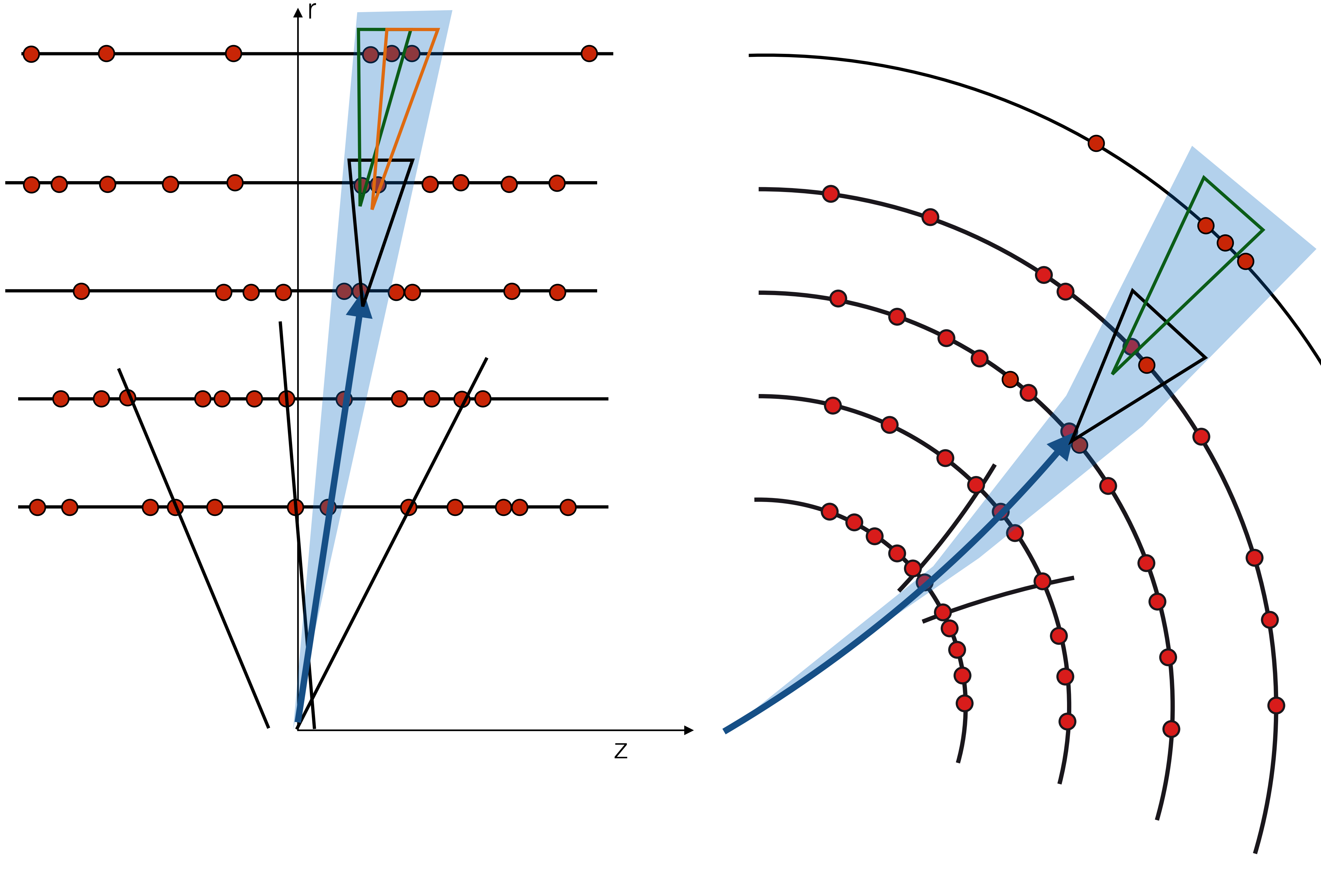
From seeds to track candidates

- Dense environments create problems for the progressive filter
  - there may not always be one obvious path to be followed: The combinatorial filter



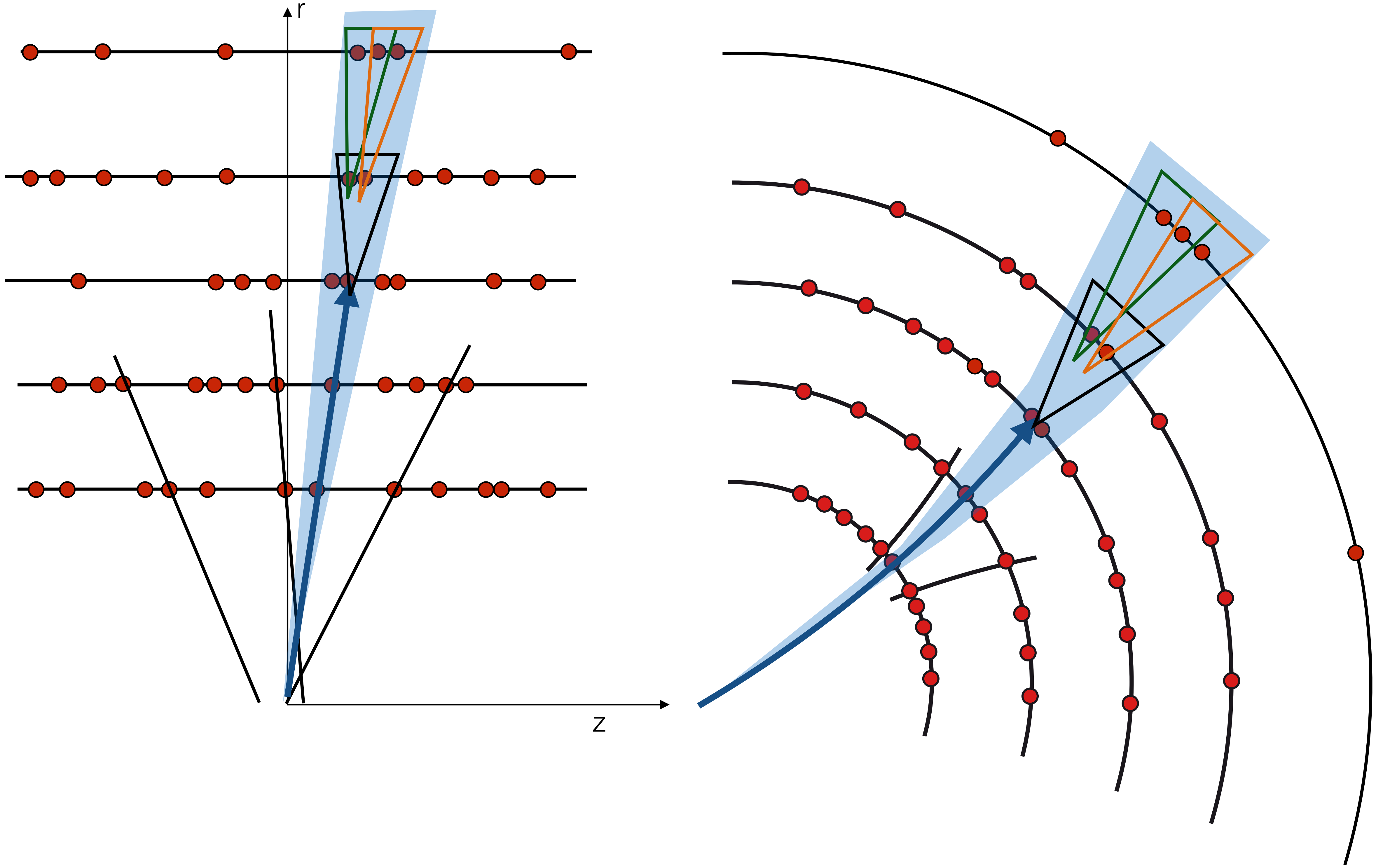
From seeds to track candidates

- Dense environments create problems for the progressive filter
  - there may not always be one obvious path to be followed: The combinatorial filter



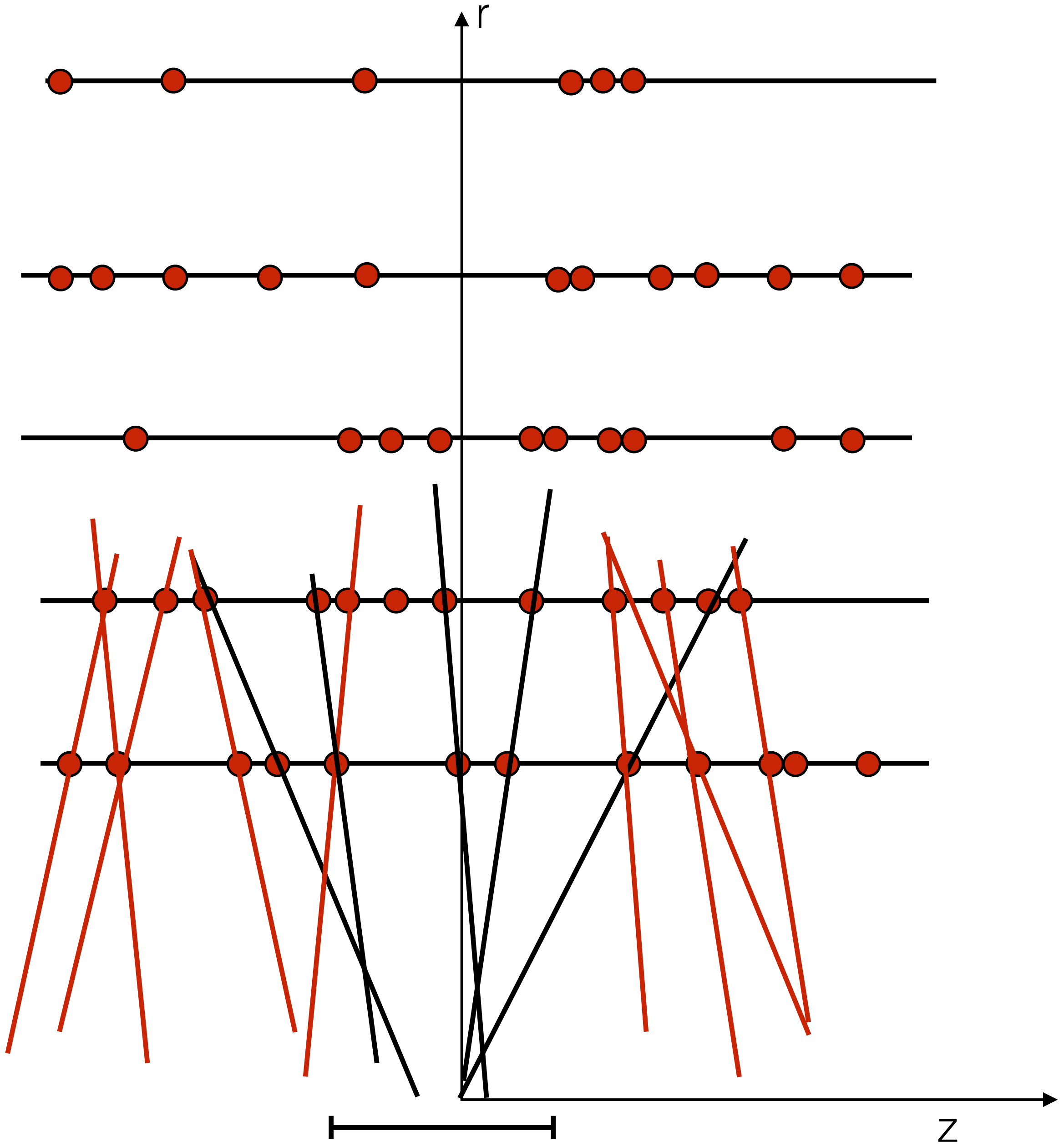
From seeds to track candidates

- Dense environments create problems for the progressive filter
  - there may not always be one obvious path to be followed: The combinatorial filter

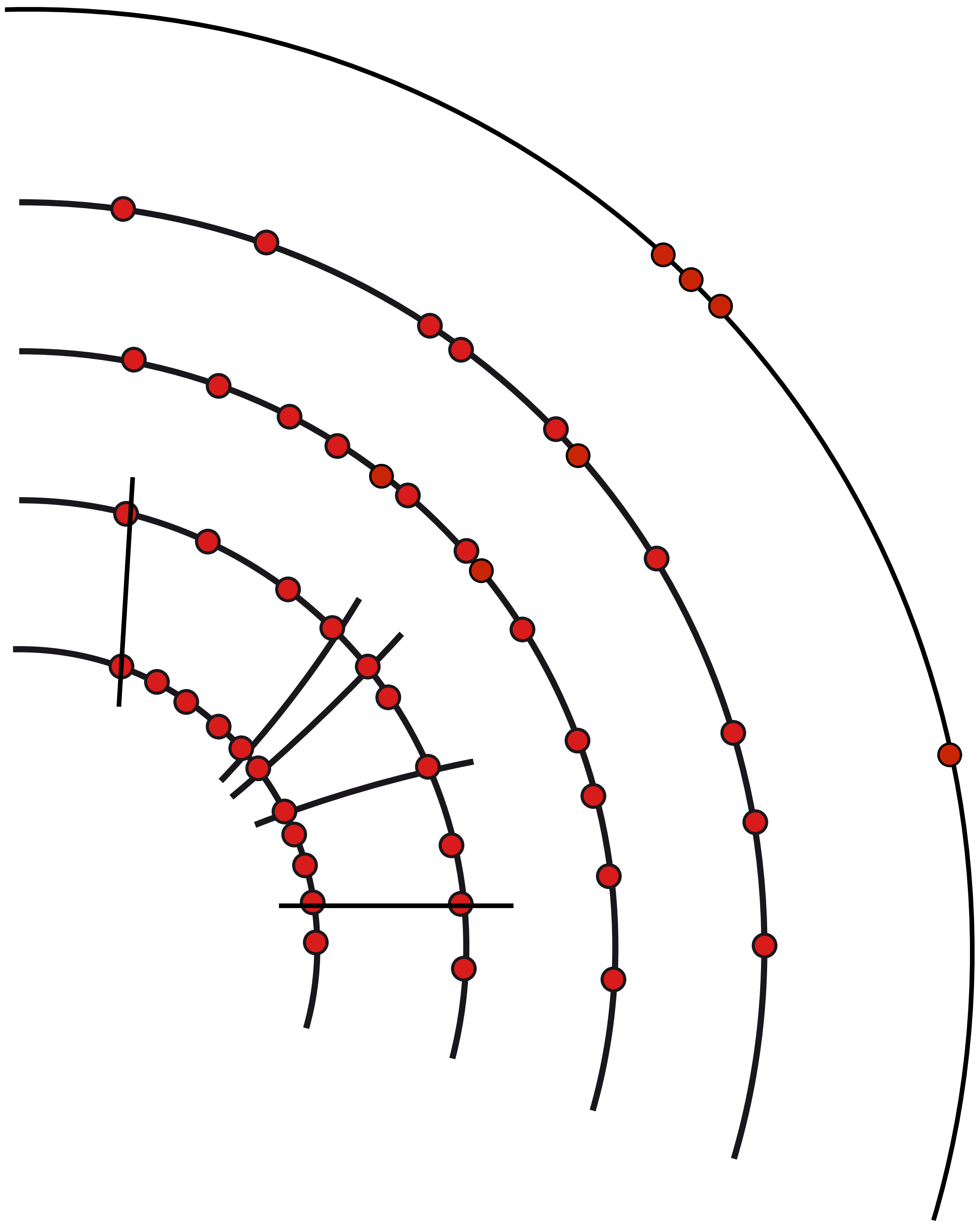


Enemy No. 2: ghosts

- avoid ghosts, i.e. fake combinations from simply combinatorial grouping
  - start off with high quality seeds (clearly 2 hit seeds are not very stringent)

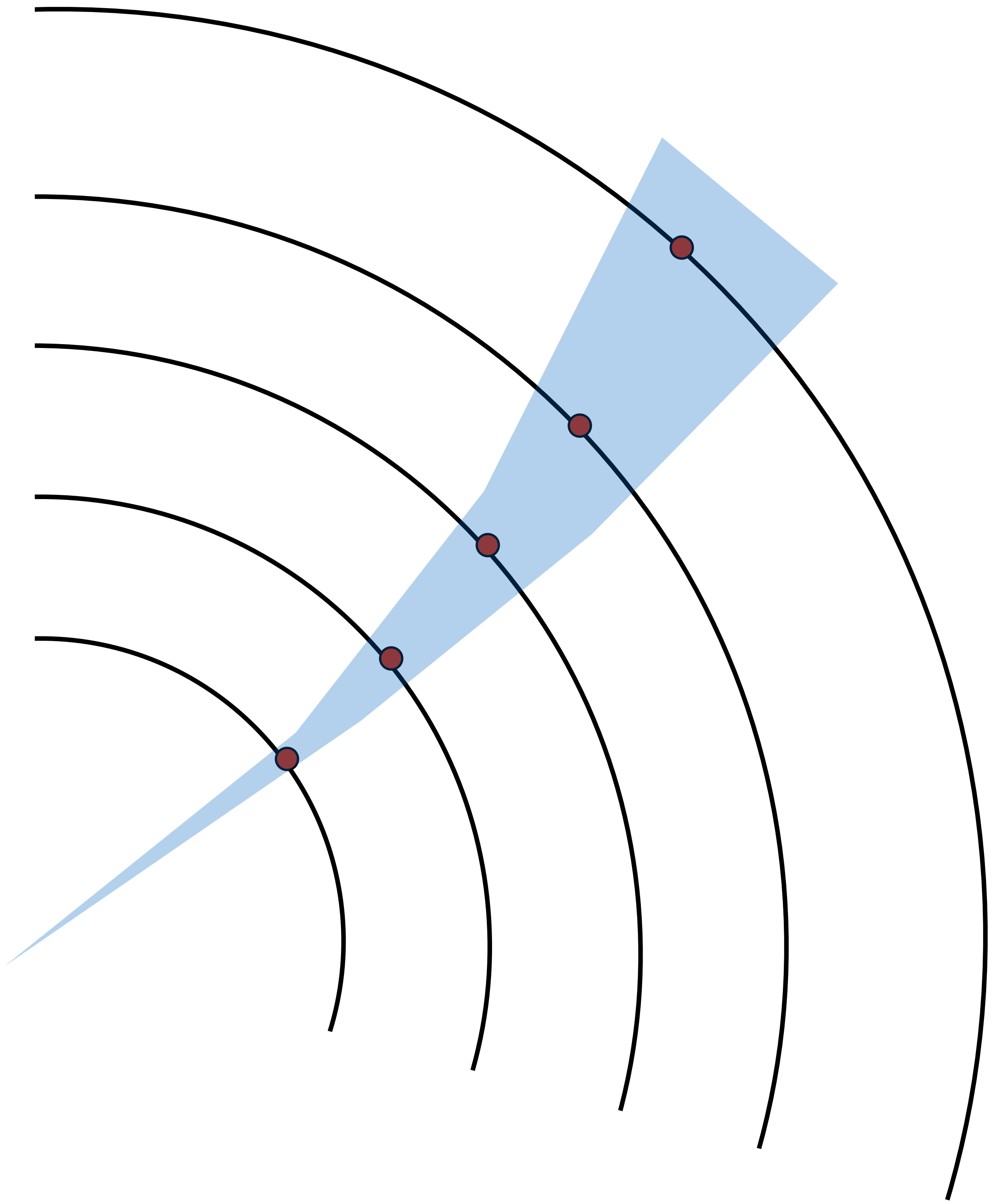


loose requirement
on interaction region



Track fitting

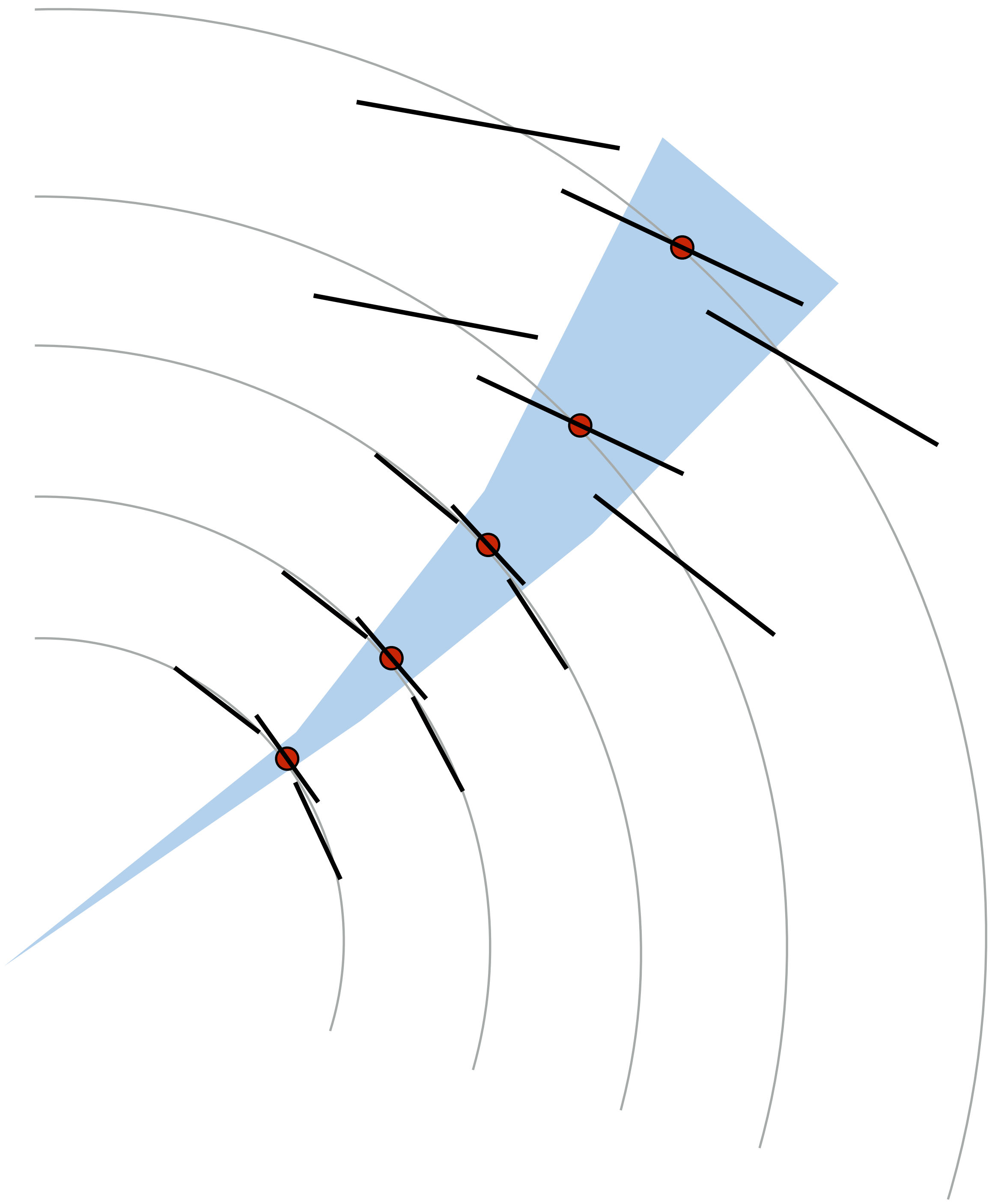
- pattern recognition provides a set of measurements
  - are the measurements compatible with a track hypothesis ?
  - what are the track parameters closest to the interaction region (e.g. as perigee)
  - how *well* is the track measured ?



- we need to perform a track fit
  - track fits are mostly based on least square estimators
  - this implies a gaussian error assumption (how close to the truth is this ?)

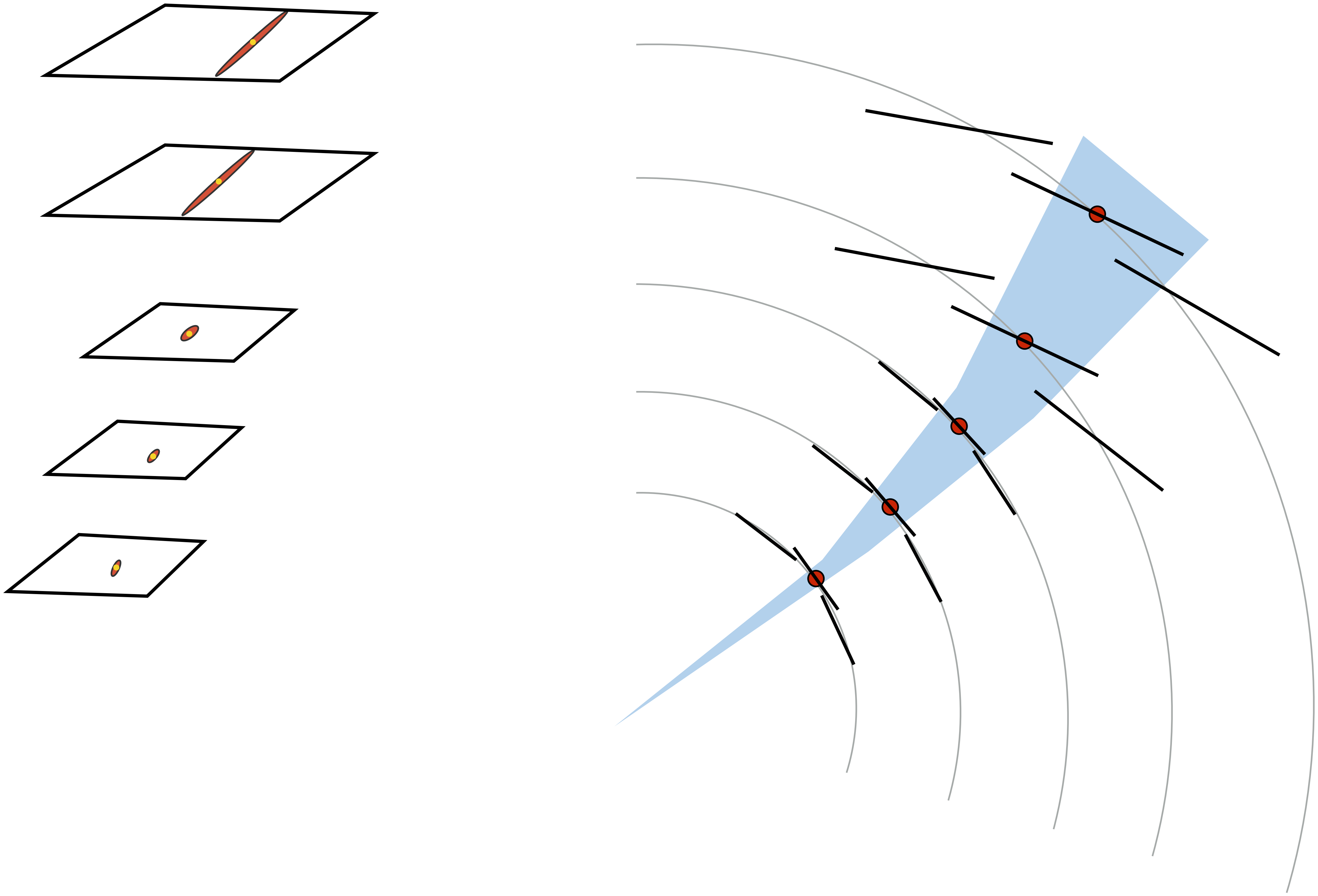
Track fitting

- a more detailed look onto our toy detector



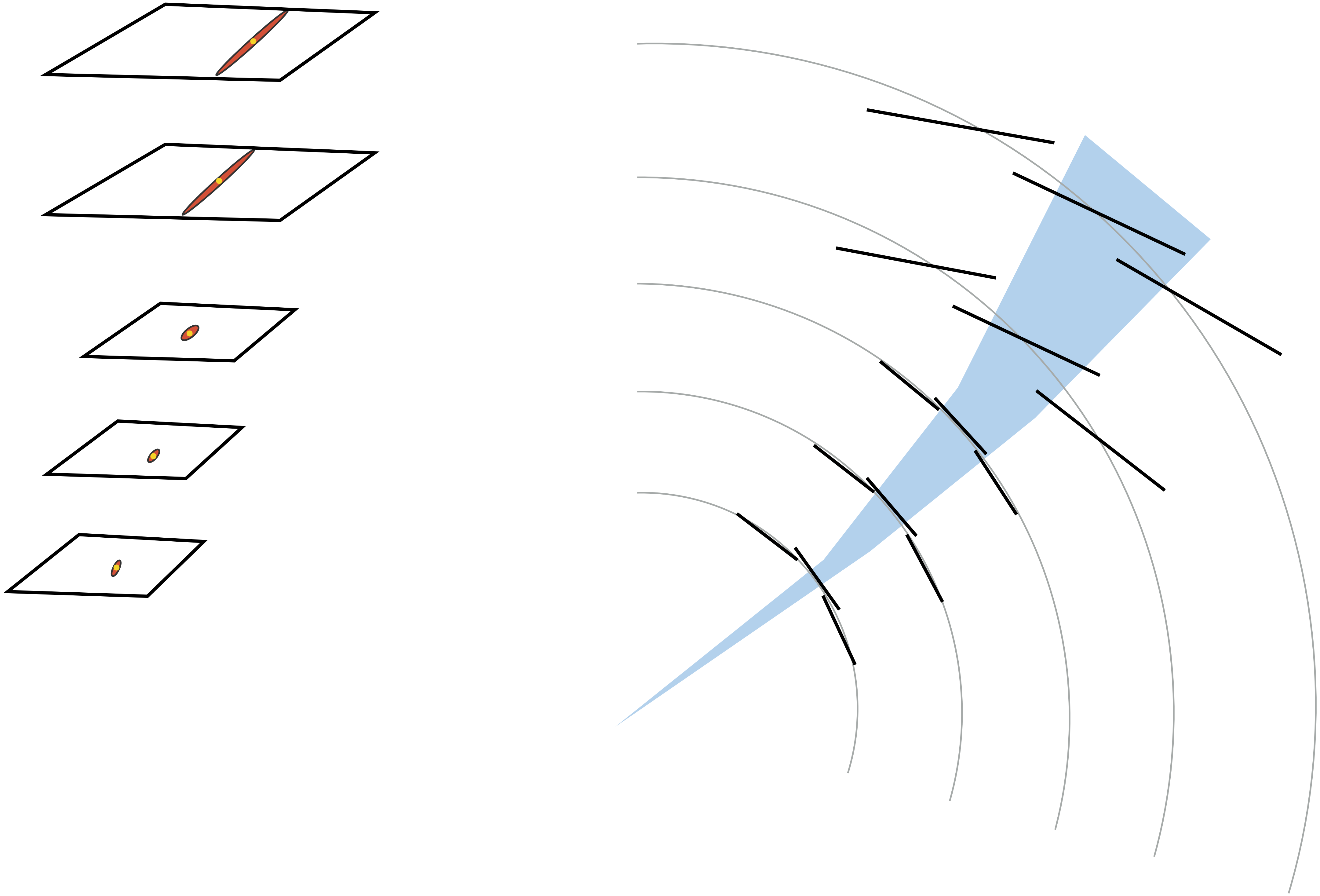
Track fitting

- a more detailed look onto our toy detector



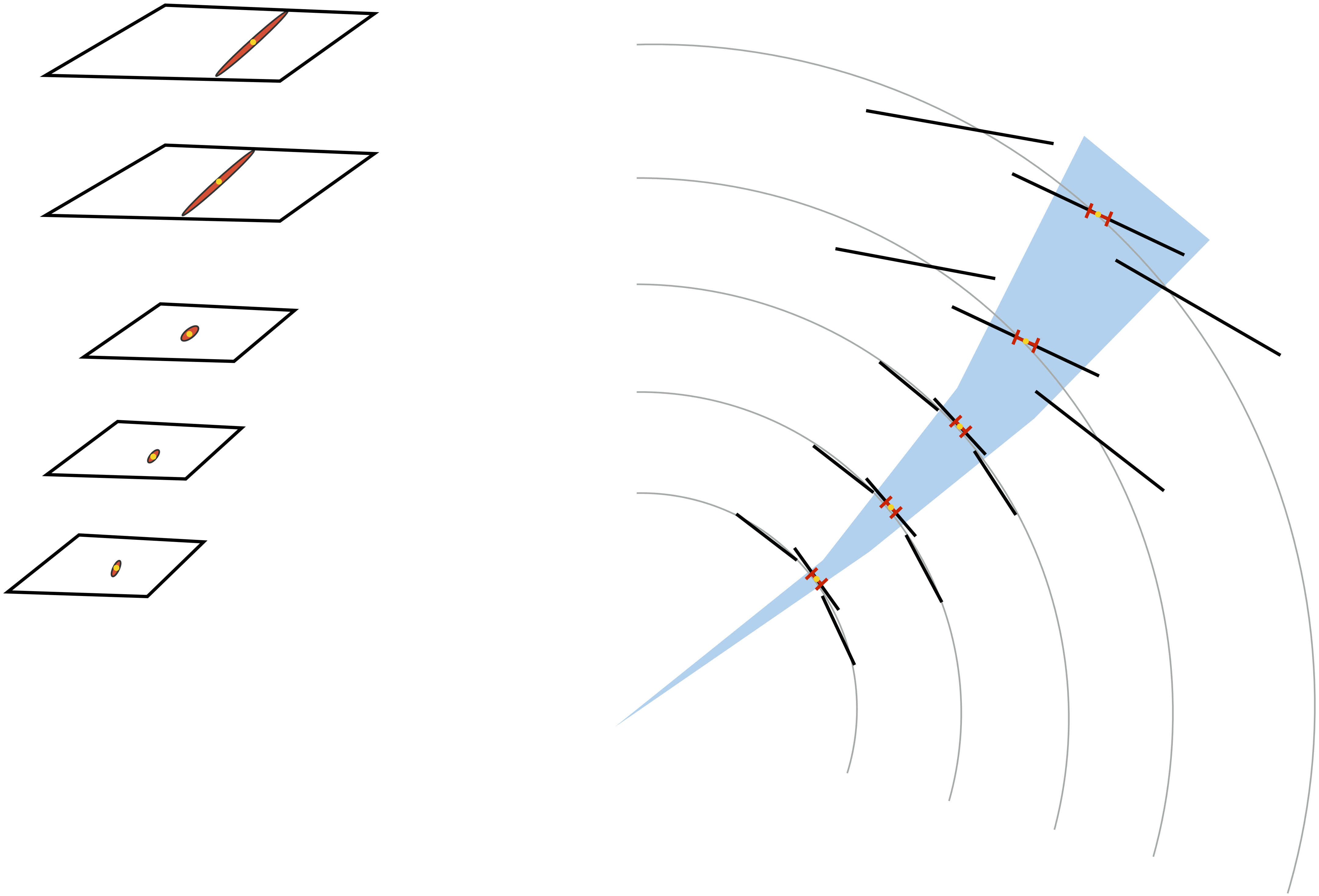
Track fitting

- a more detailed look onto our toy detector



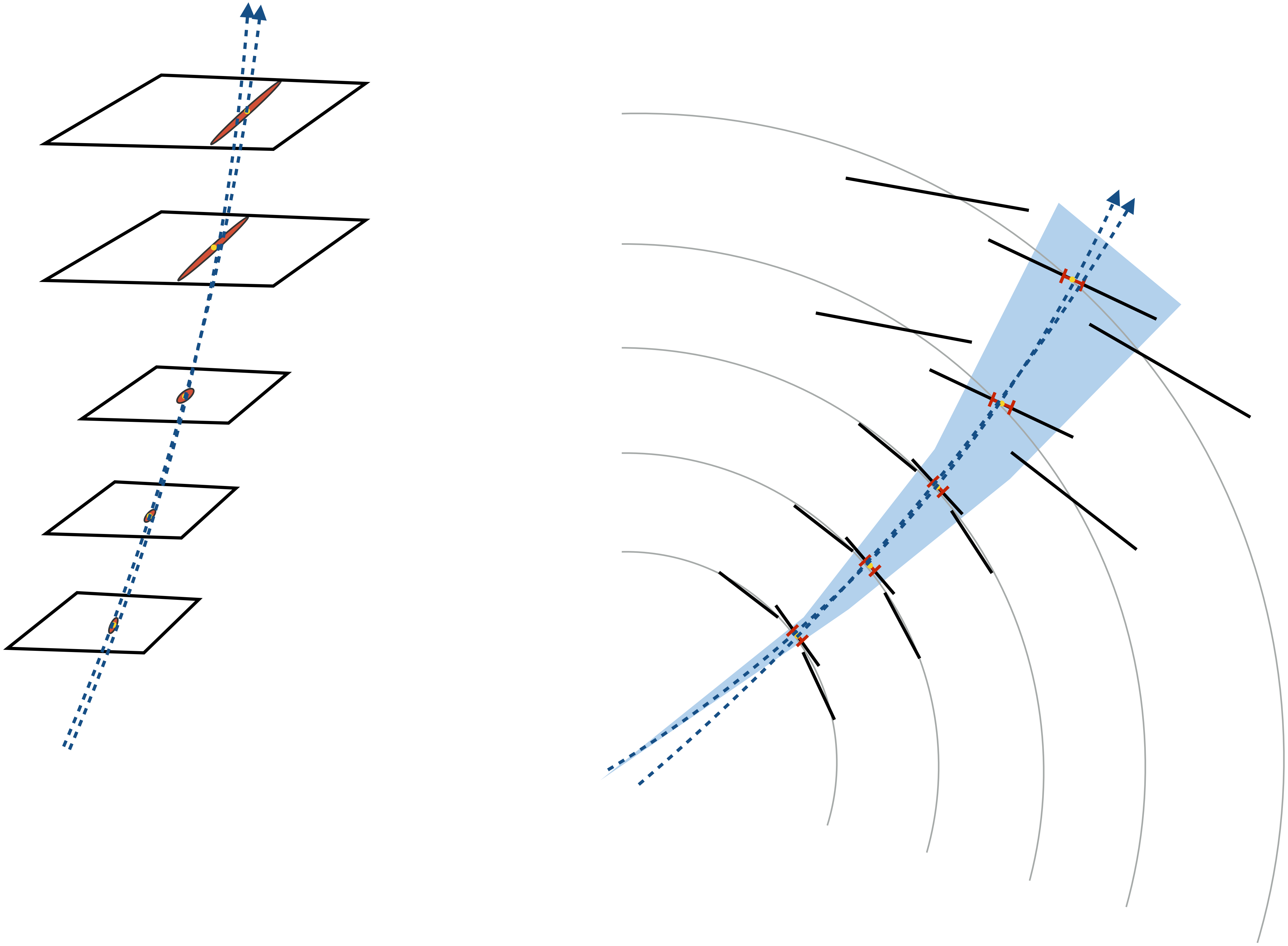
Track fitting

- a more detailed look onto our toy detector



Track fitting

- a more detailed look onto our toy detector



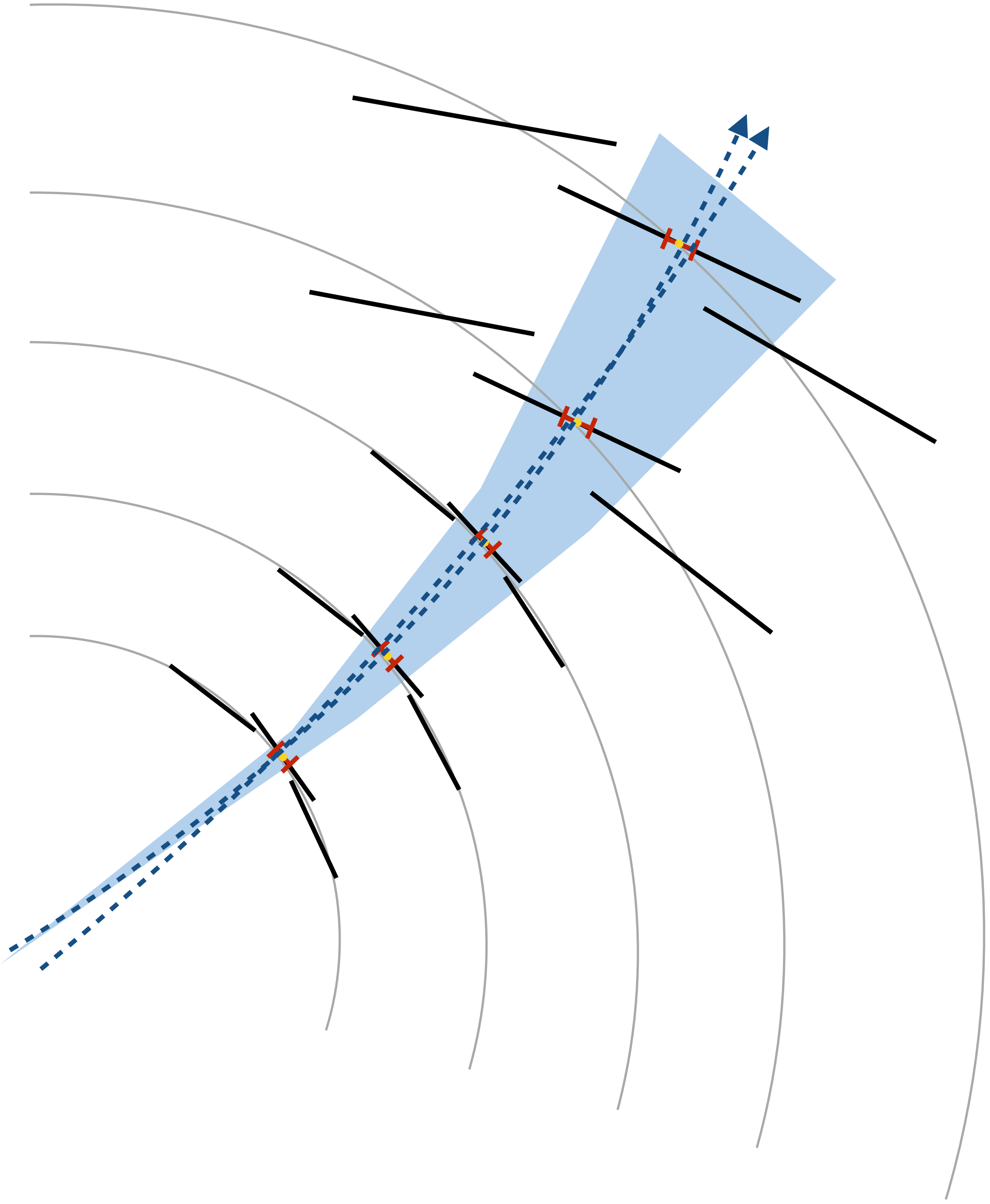
Global χ^2 minimisation

- a classical least squares estimator problem !

$$\chi^2 = \sum_k \Delta m_k^T G_K^{-1} \Delta m_k \quad \text{with} \quad \Delta m_k = m_k - d_k(\mathbf{q}) \quad \text{and } \boldsymbol{G_k} \text{ the covariance of measurement } \mathbf{m}_k$$

d_k including transport of $\mathbf{q}$ to measurement layer k and measurement mapping function

$$\boldsymbol{d}_k = \boxed{\boldsymbol{h}_k} \boxed{\circ \boldsymbol{f}_{k|k-1} \circ \cdots \circ \boldsymbol{f}_{2|1} \circ \boldsymbol{f}_{1|0}}$$



Global χ^2 minimisation

- a classical least squares estimator problem !

$$\chi^2 = \sum_k \Delta m_k^T G_K^{-1} \Delta m_k$$ with $$\Delta m_k = m_k - d_k(\mathbf{q})$$ and $\boldsymbol{G_k}$ the covariance of measurement $\mathbf{m}_k$

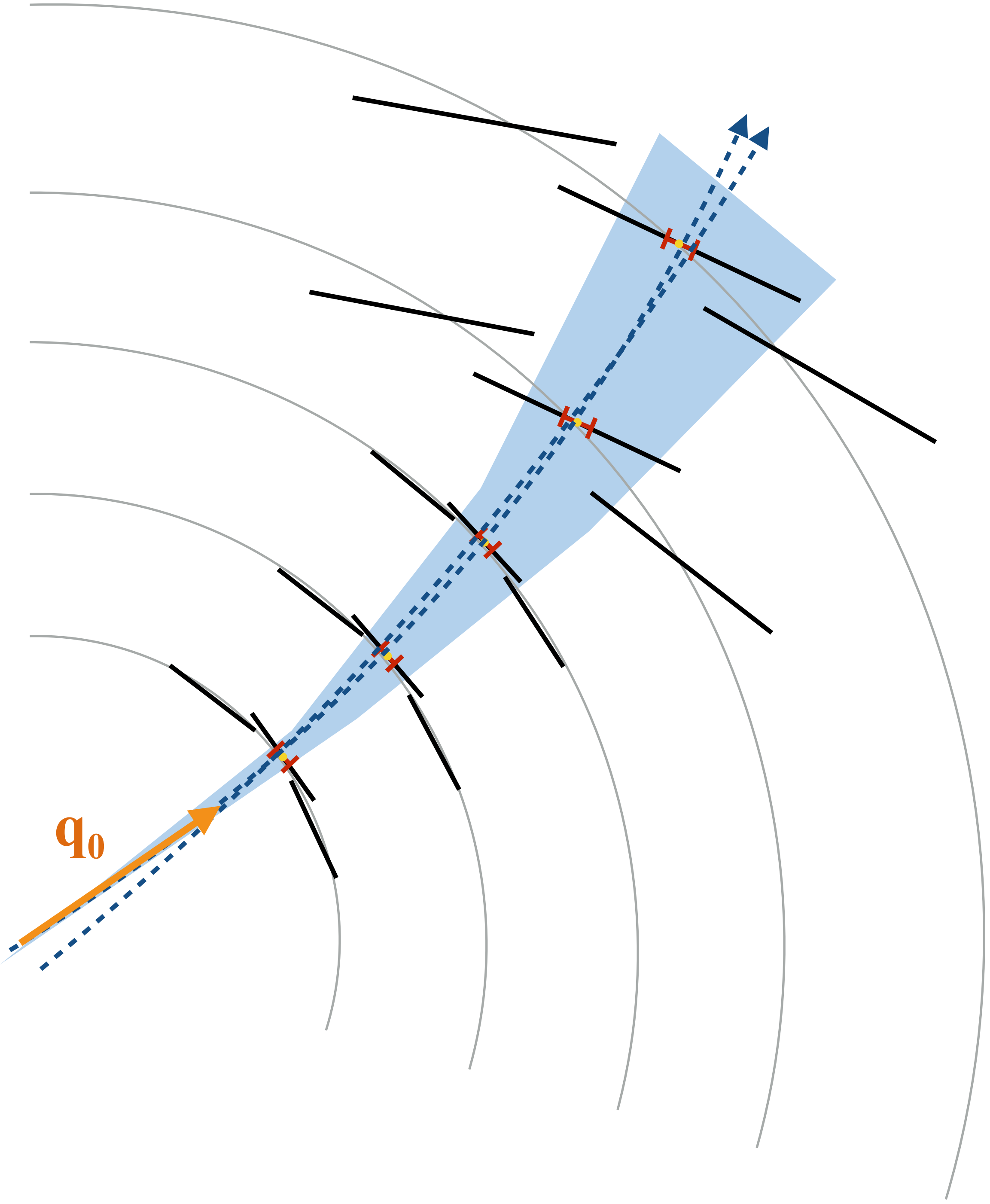
d_k including transport of $\mathbf{q}$ to measurement layer k and measurement mapping function

$$\boldsymbol{d}_k = \boxed{\boldsymbol{h}_k} \boxed{\circ \boldsymbol{f}_{k|k-1} \circ \cdots \circ \boldsymbol{f}_{2|1} \circ \boldsymbol{f}_{1|0}}$$

linearise the problem, starting from an initial state $\mathbf{q_0}$

$$d_k(\mathbf{q_0} + \delta \mathbf{q}) \cong d_k(\mathbf{q_0}) + D_k \cdot \delta \mathbf{q}$$

with Jacobian $$\boldsymbol{D}_k = \boxed{\boldsymbol{H}_k} \boxed{\boldsymbol{F}_{k|k-1} \cdots \boldsymbol{F}_{2|1} \boldsymbol{F}_{1|0}}$$



Global χ^2 minimisation

- a classical least squares estimator problem !

$$\chi^2 = \sum_k \Delta m_k^T G_K^{-1} \Delta m_k \quad \text{with} \quad \Delta m_k = m_k - d_k(\mathbf{q}) \quad \text{and } \boldsymbol{G_k} \text{ the covariance of measurement } \mathbf{m}_k$$

d_k including transport of $\mathbf{q}$ to measurement layer k and measurement mapping function

$$\boldsymbol{d}_k = \boxed{\boldsymbol{h}_k} \boxed{\circ \boldsymbol{f}_{k|k-1} \circ \cdots \circ \boldsymbol{f}_{2|1} \circ \boldsymbol{f}_{1|0}}$$

linearise the problem, starting from an initial state $\mathbf{q_0}$

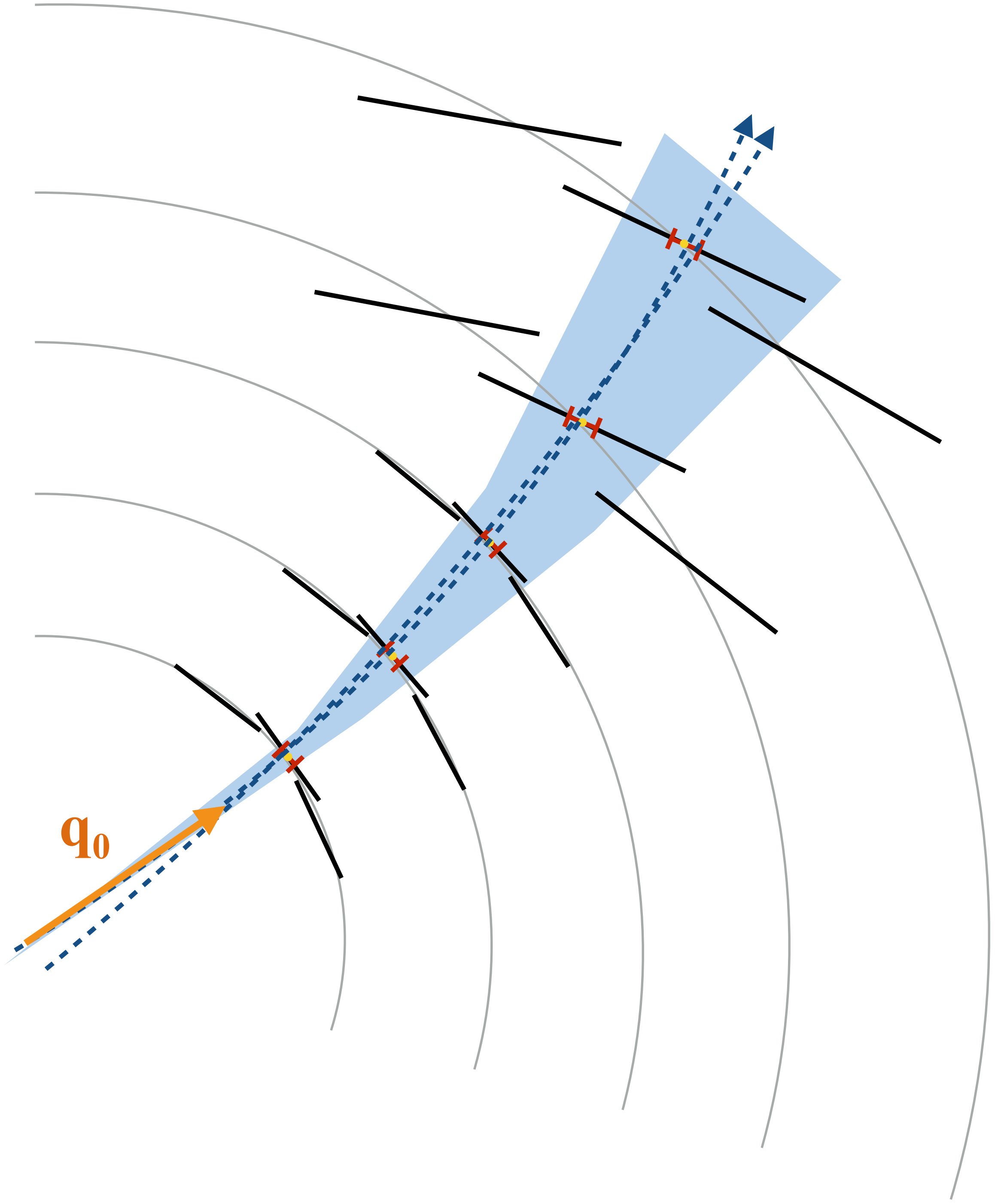
$$d_k\left(\mathbf{q_0} + \delta\mathbf{q}\right) \cong d_k\left(\mathbf{q_0}\right) + D_k \cdot \delta\mathbf{q}$$

with Jacobian $\boldsymbol{D}_k = \boxed{\boldsymbol{H}_k} \boxed{\boldsymbol{F}_{k|k-1} \cdots \boldsymbol{F}_{2|1} \boldsymbol{F}_{1|0}}$

find the global minimum: $\dfrac{\partial \chi^2}{\partial \mathbf{q}} \overset{!}{=} 0$

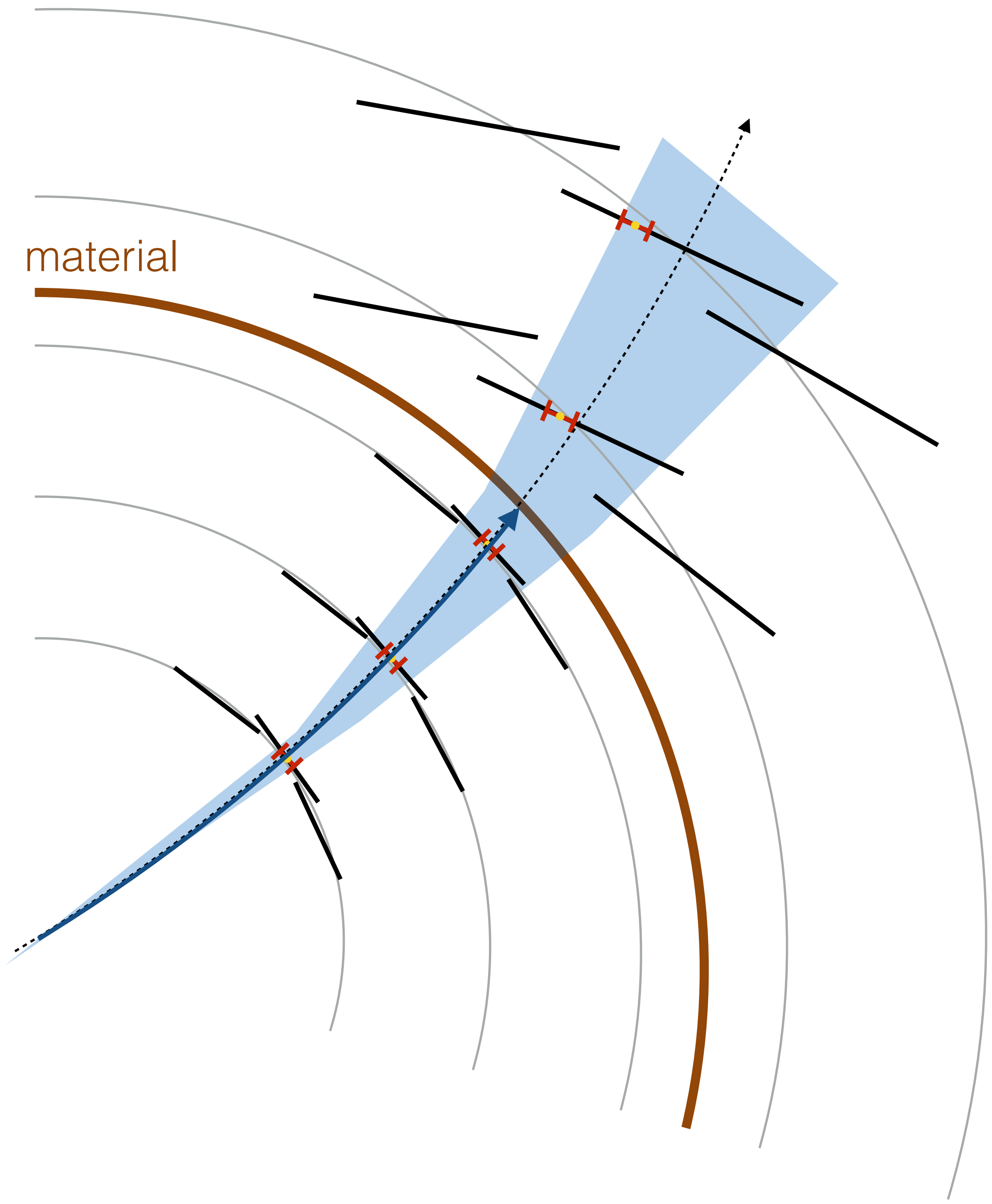
$$\partial\mathbf{q} = \left(\sum_k D_k^T G_k^{-1} D_k\right)^{-1} \sum_k D_k^T G_k^{-1} \left(m_k - d_k(\mathbf{q_0})\right)$$

$$C = \left(\sum_k D_k^T G_k^{-1} D_k\right)^{-1}$$



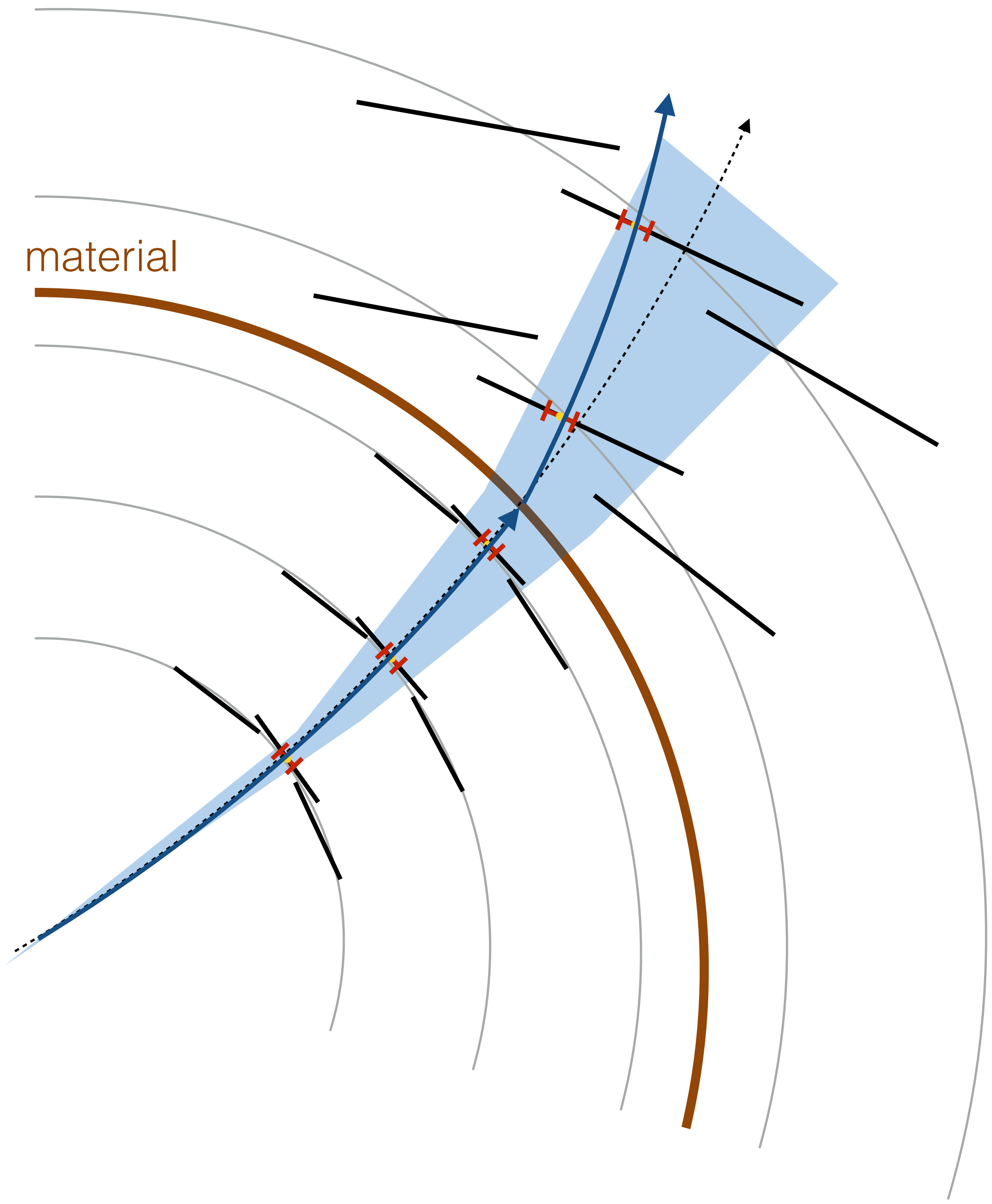
Global χ^2 fit with material

- in reality the particle gets deflected by **material**
  - multiple coulomb scattering



Global χ^2 fit with material

- in reality the particle gets deflected by **material**
    - multiple coulomb scattering



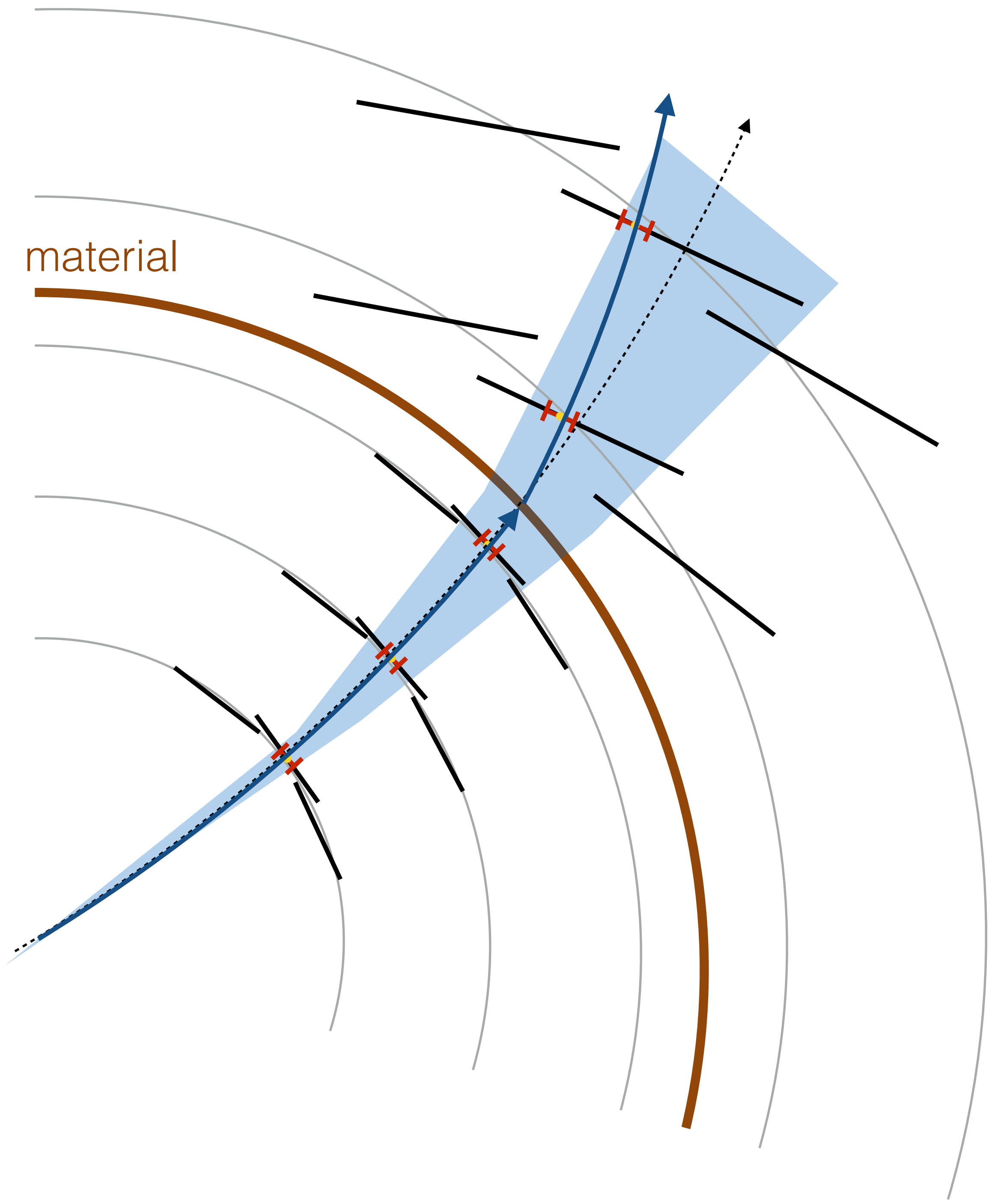
Global χ^2 fit with material

- in reality the particle gets deflected by **material**
  - multiple coulomb scattering

- modification of the χ^2 function

$$\chi^2 = \sum_k \Delta m_k^T G_K^{-1} \Delta m_k + \sum_i \delta\theta_i^T Q_i^{-1} \delta\theta_i$$

with: $\Delta m_k = m_k - d_k\left(\mathbf{q}, \delta\theta_i\right)$



Global χ^2 fit with material

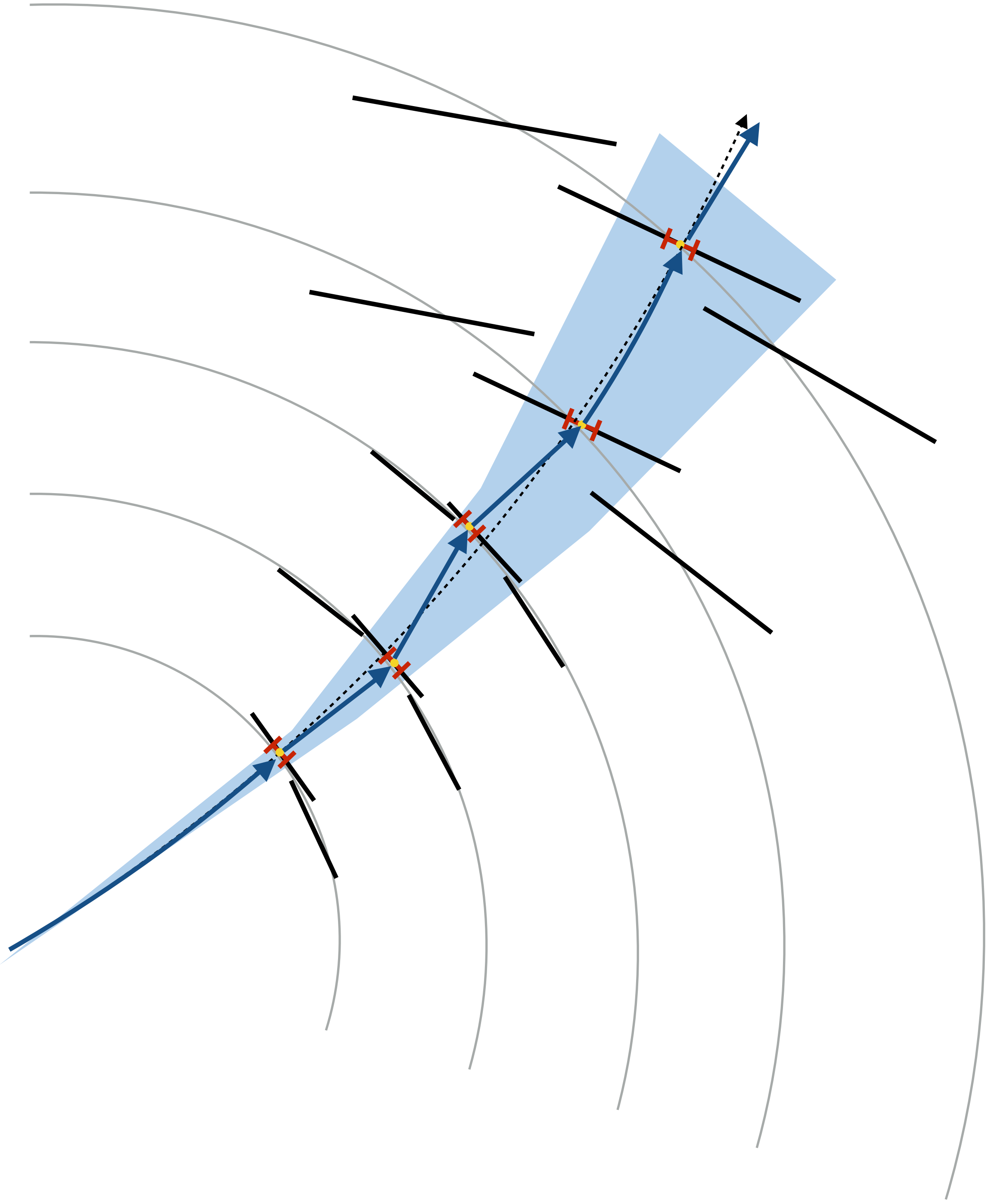
- in reality the particle gets deflected by **material**
  - multiple coulomb scattering

- modification of the χ^2 function

$$\chi^2 = \sum_k \Delta m_k^T G_K^{-1} \Delta m_k + \sum_i \delta\theta_i^T Q_i^{-1} \delta\theta_i$$

with: $\Delta m_k = m_k - d_k\left(\mathbf{q}, \delta\theta_i\right)$

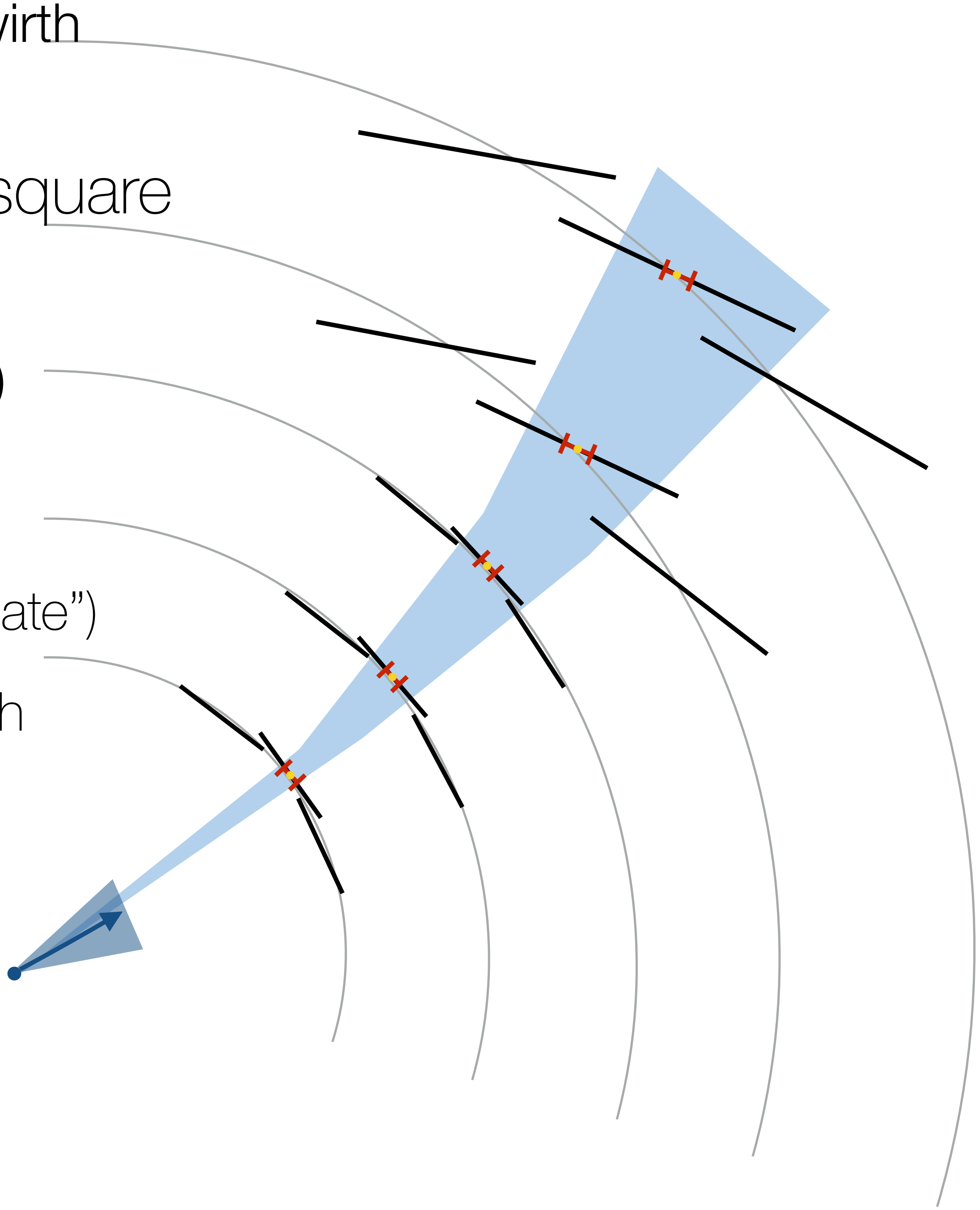
- every layer is a **material layer**
  - creates a computational problem: matrix inversion of huge matrix to find the χ^2 minimum



The Kalman Filter

- offers an alternative solution to the large matrix inversion
  - initially developed by I. Kalman to track missiles
  - for HEP pioneered by Billoir and R. Fruehwirth

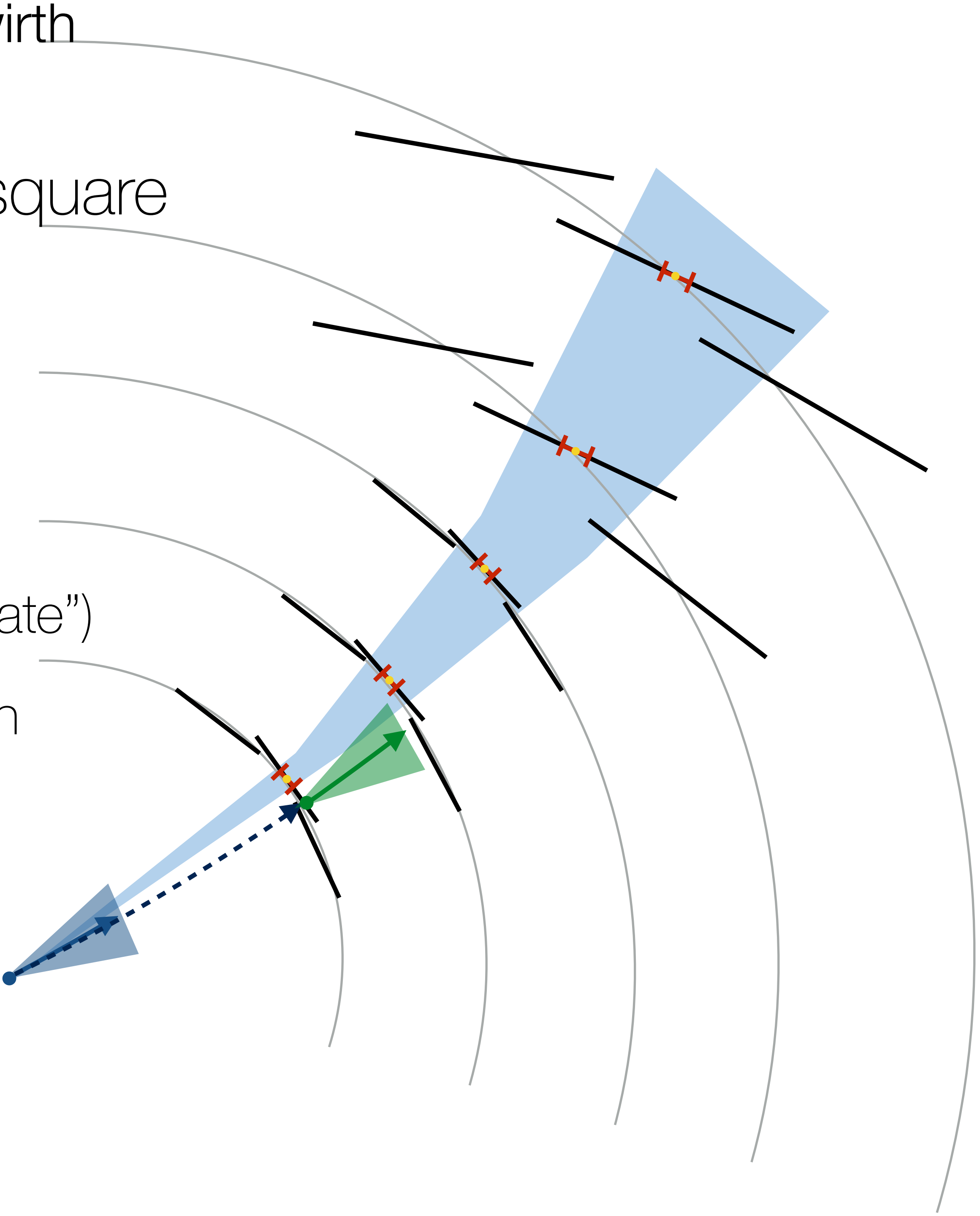
- performs a progressive way of least square estimation
  - equivalent to a χ^2 fit (if run with a smoother)
  - start with transport of track parameters (and covariances) to measurement surface, create predicted parameters ("predicted state")
  - combine/update predicted parameters with measurement to updated parameters ("filtered state")



The Kalman Filter

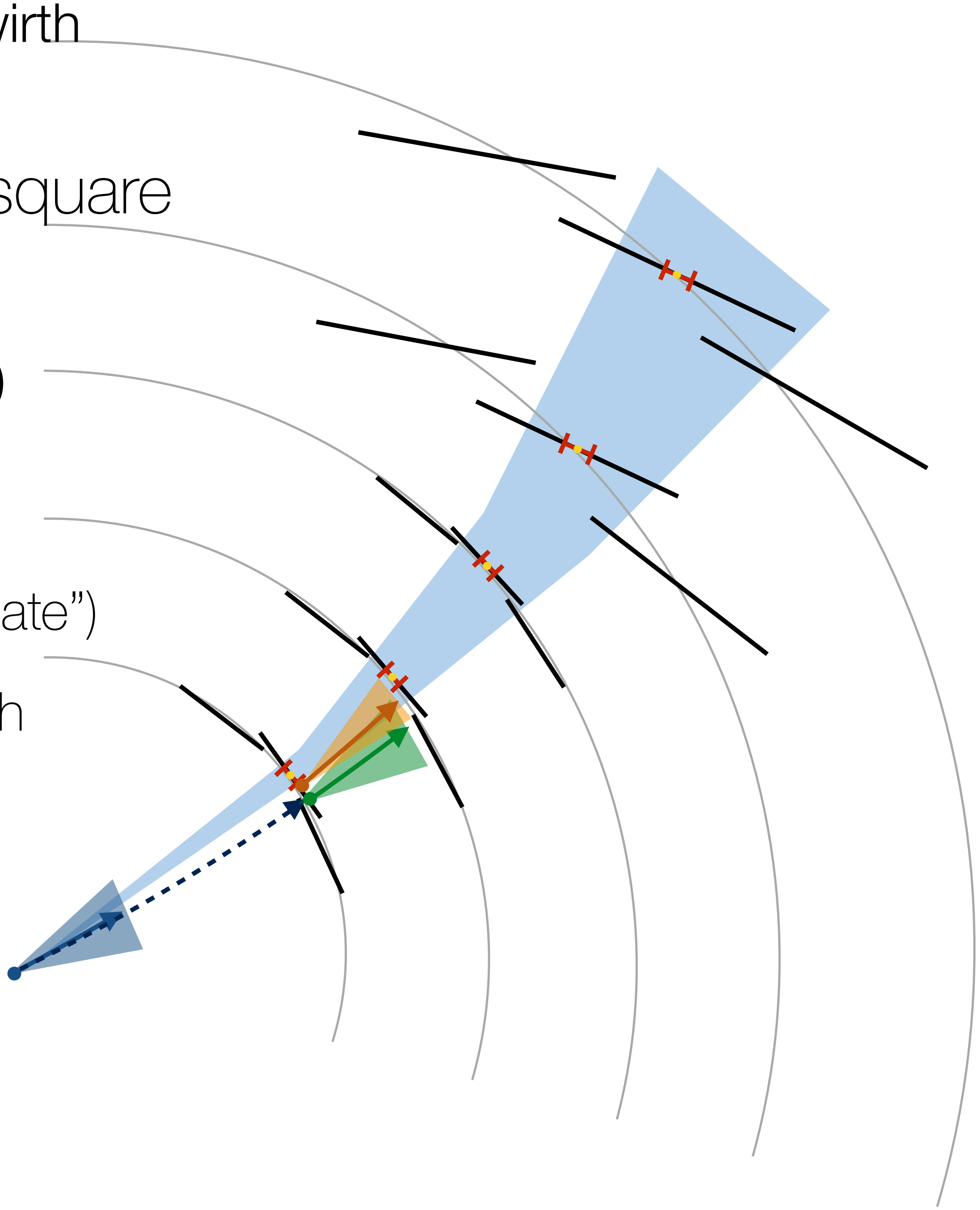
- offers an alternative solution to the large matrix inversion
  - initially developed by I. Kalman to track missiles
  - for HEP pioneered by Billoir and R. Fruehwirth

- performs a progressive way of least square estimation
  - equivalent to a χ^2 fit (if run with a smoother)
  - start with transport of track parameters (and covariances) to measurement surface, create predicted parameters ("predicted state")
  - combine/update predicted parameters with measurement to updated parameters ("filtered state")



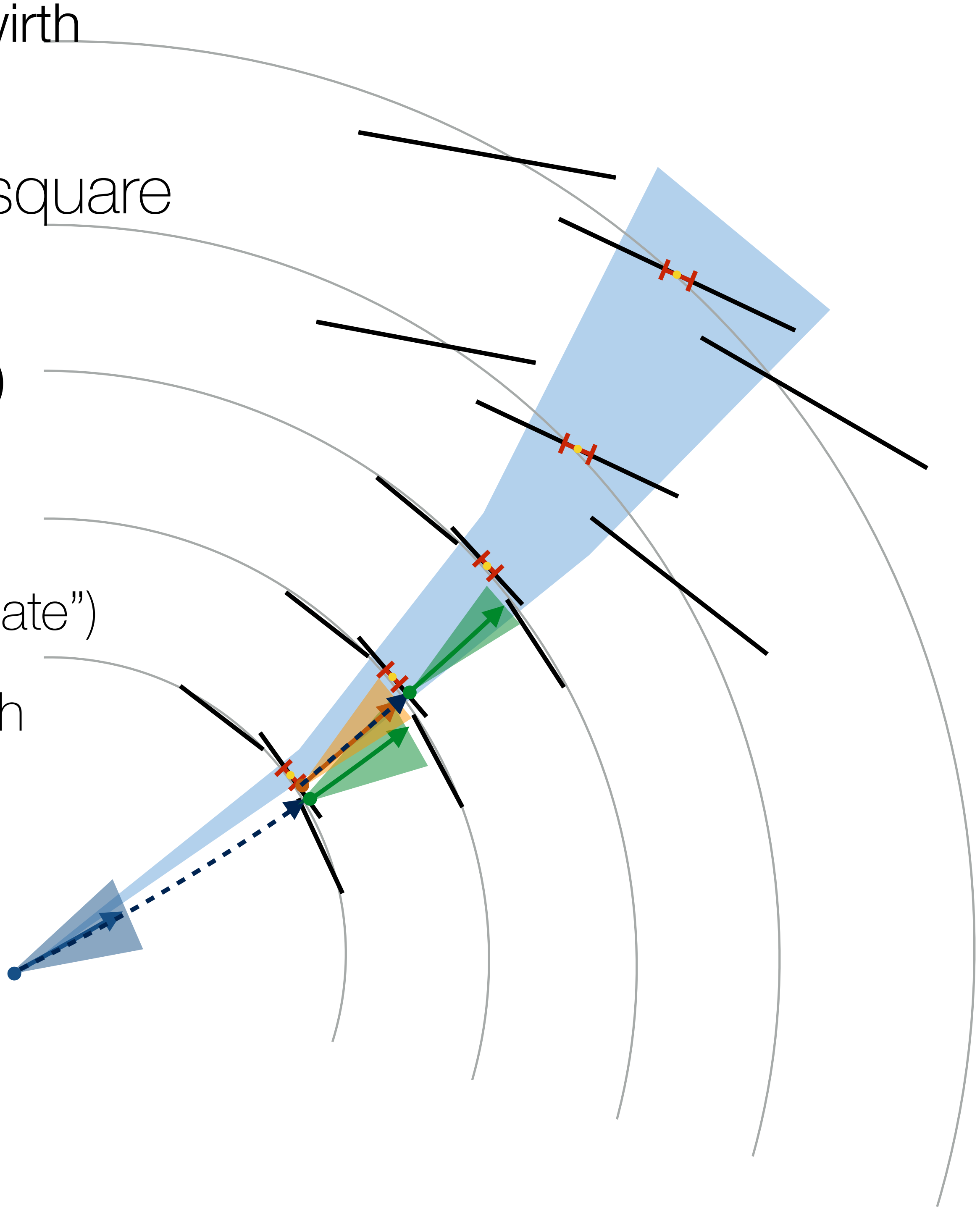
The Kalman Filter

- offers an alternative solution to the large matrix inversion
  - initially developed by I. Kalman to track missiles
  - for HEP pioneered by Billoir and R. Fruehwirth
- performs a progressive way of least square estimation
  - equivalent to a χ^2 fit (if run with a smoother)
  - start with transport of track parameters (and covariances) to measurement surface, create predicted parameters ("predicted state")
  - combine/update predicted parameters with measurement to updated parameters ("filtered state")



The Kalman Filter

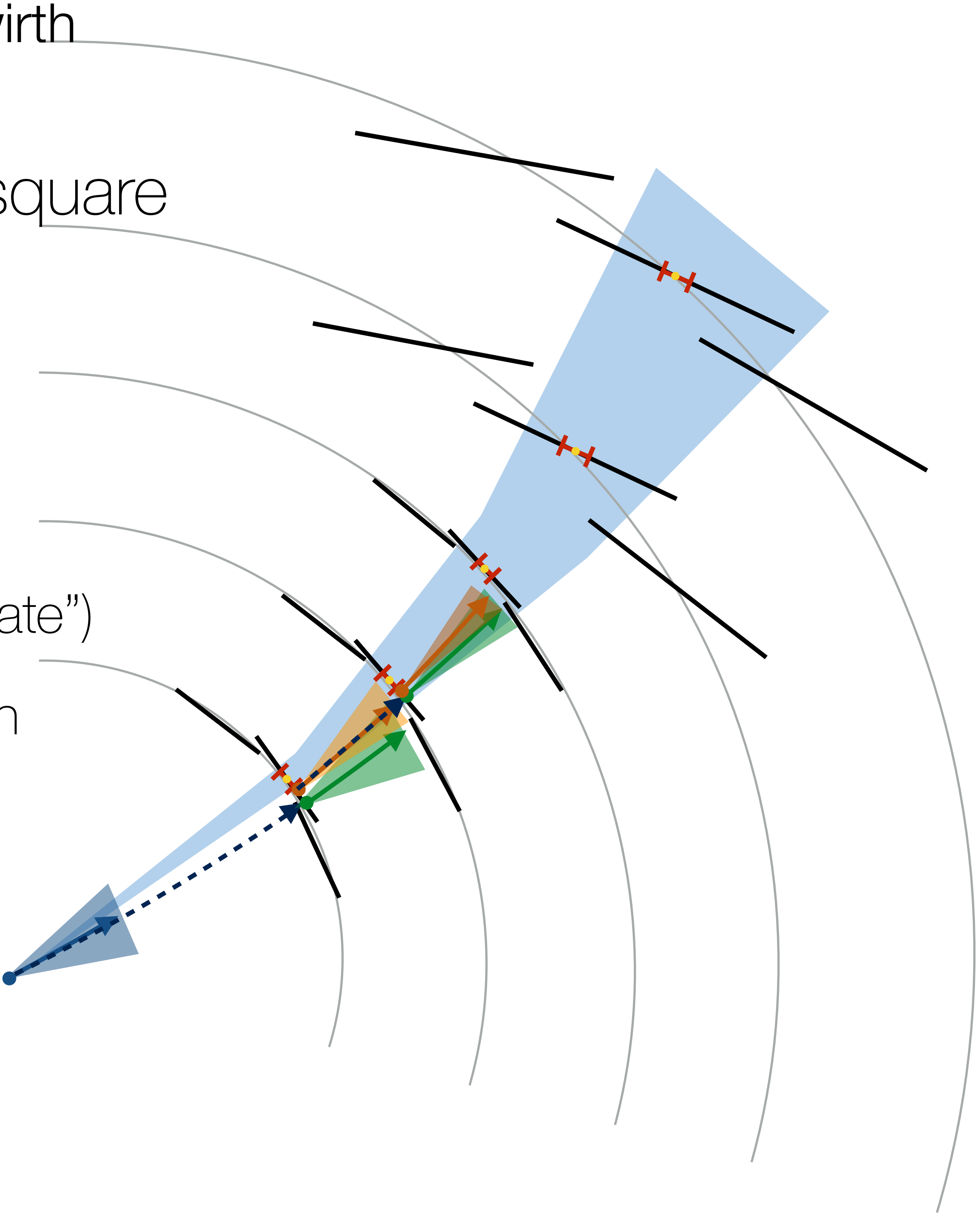
- offers an alternative solution to the large matrix inversion
  - initially developed by I. Kalman to track missiles
  - for HEP pioneered by Billoir and R. Fruehwirth
- performs a progressive way of least square estimation
  - equivalent to a χ^2 fit (if run with a smoother)
  - start with transport of track parameters (and covariances) to measurement surface, create predicted parameters ("predicted state")
  - combine/update predicted parameters with measurement to updated parameters ("filtered state")



The Kalman Filter

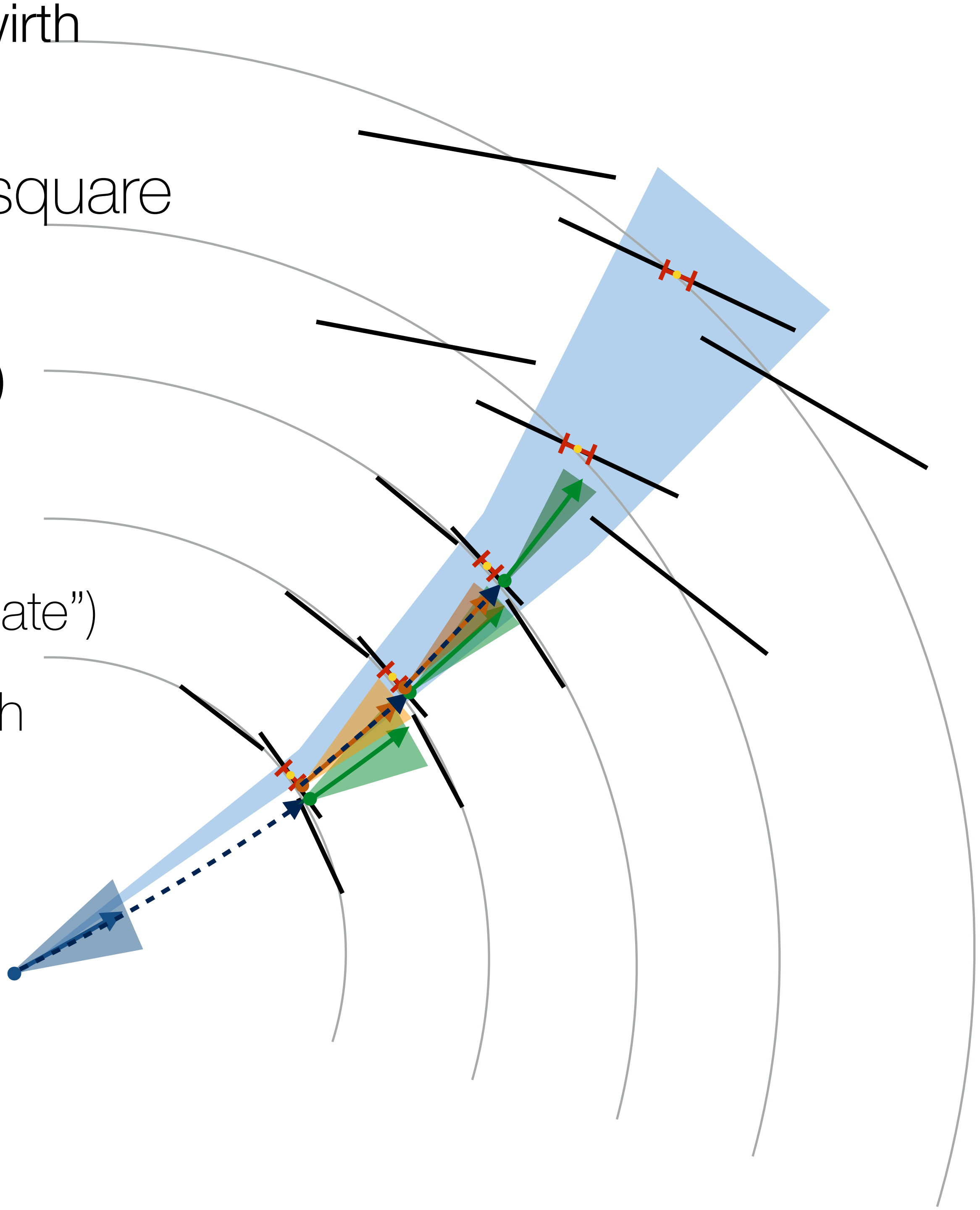
- offers an alternative solution to the large matrix inversion
  - initially developed by I. Kalman to track missiles
  - for HEP pioneered by Billoir and R. Fruehwirth

- performs a progressive way of least square estimation
  - equivalent to a χ^2 fit (if run with a smoother)
  - start with **transport** of track parameters (and covariances) to measurement surface, create **predicted parameters** ("predicted state")
  - combine/update **predicted parameters** with **measurement** to **updated parameters** ("filtered state")



The Kalman Filter

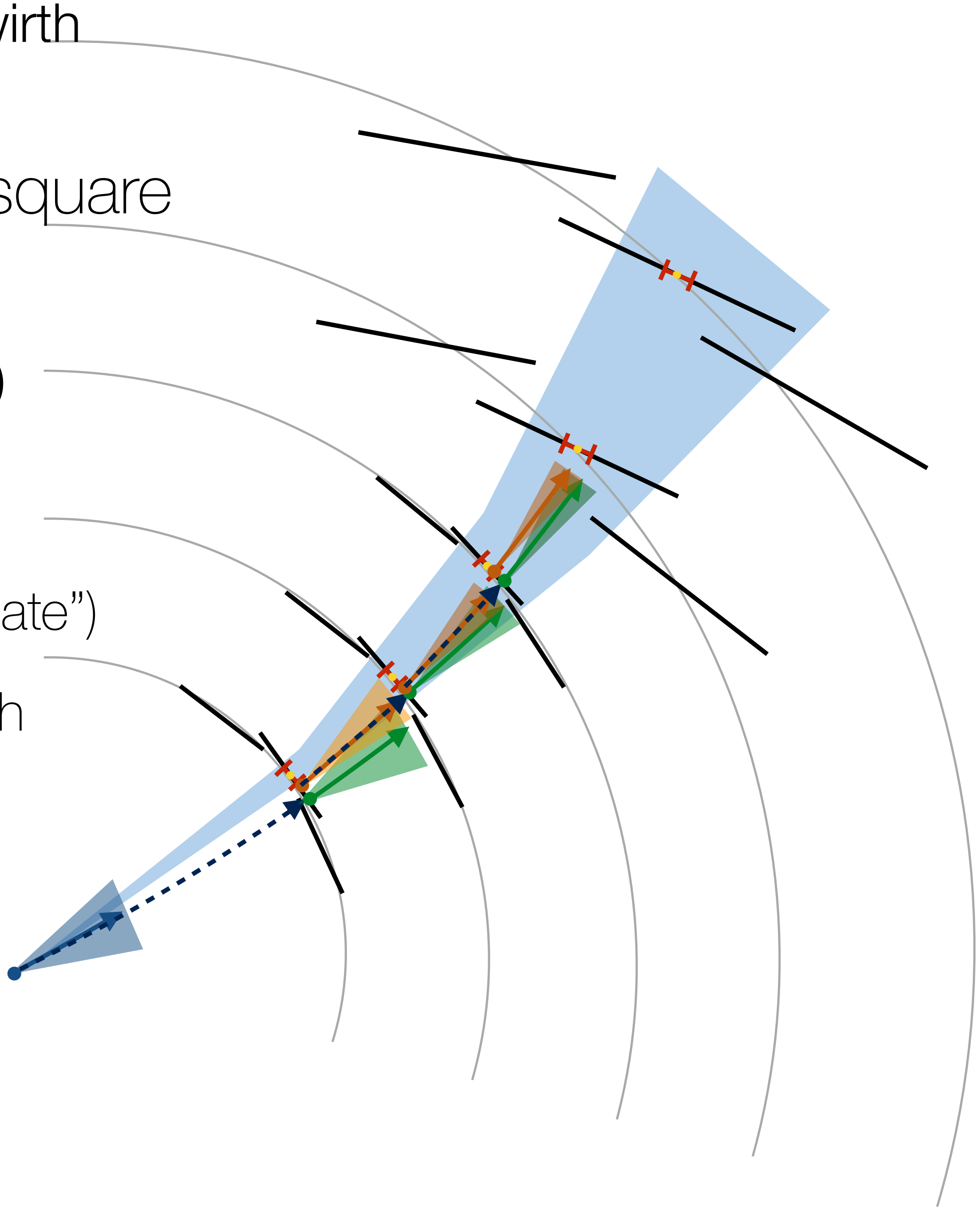
- offers an alternative solution to the large matrix inversion
  - initially developed by I. Kalman to track missiles
  - for HEP pioneered by Billoir and R. Fruehwirth
- performs a progressive way of least square estimation
  - equivalent to a χ^2 fit (if run with a smoother)
  - start with **transport** of track parameters (and covariances) to measurement surface, create **predicted parameters** ("predicted state")
  - combine/update **predicted parameters** with **measurement** to **updated parameters** ("filtered state")



The Kalman Filter

- offers an alternative solution to the large matrix inversion
  - initially developed by I. Kalman to track missiles
  - for HEP pioneered by Billoir and R. Fruehwirth

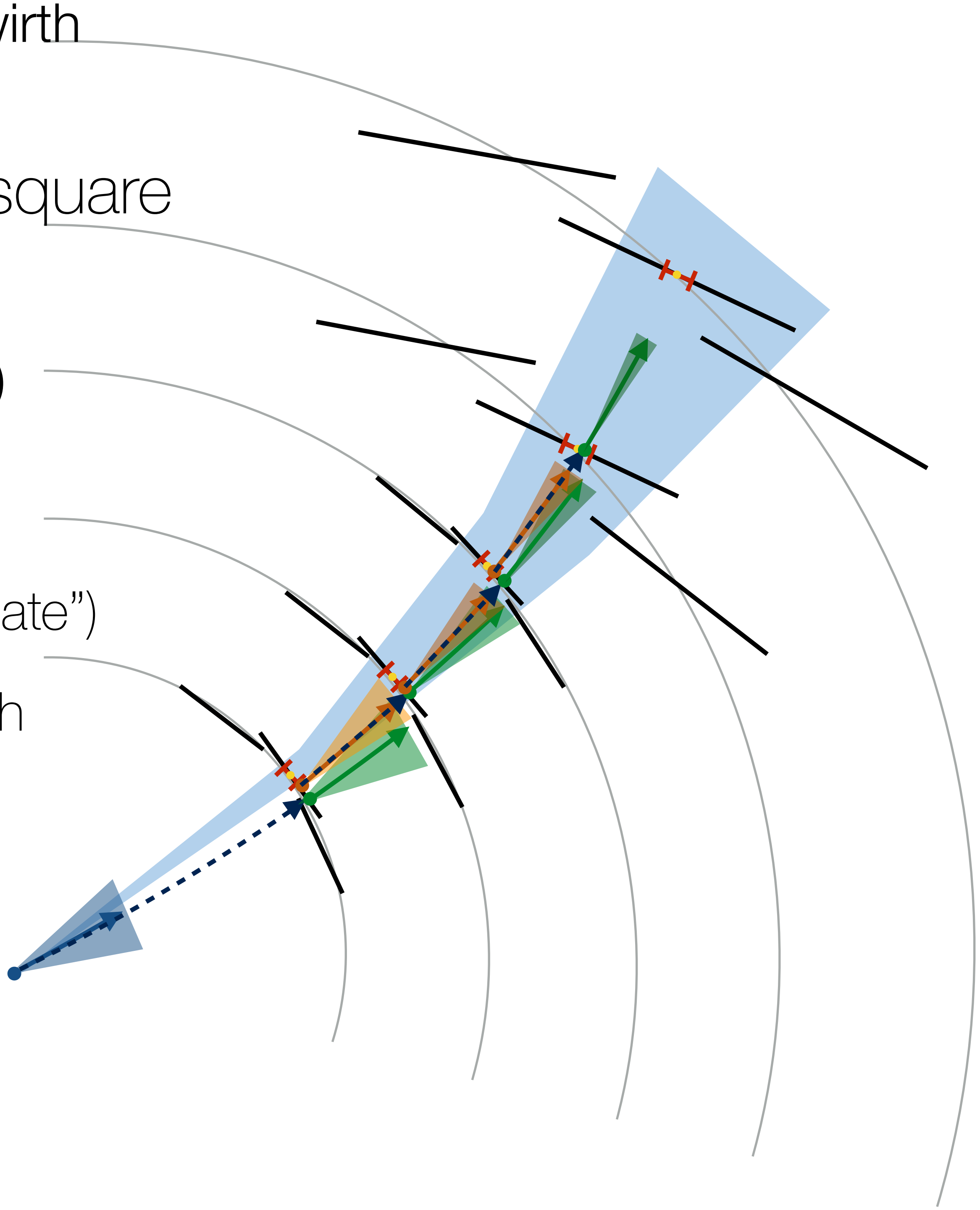
- performs a progressive way of least square estimation
  - equivalent to a χ^2 fit (if run with a smoother)
  - start with transport of track parameters (and covariances) to measurement surface, create predicted parameters ("predicted state")
  - combine/update predicted parameters with measurement to updated parameters ("filtered state")



The Kalman Filter

- offers an alternative solution to the large matrix inversion
  - initially developed by I. Kalman to track missiles
  - for HEP pioneered by Billoir and R. Fruehwirth

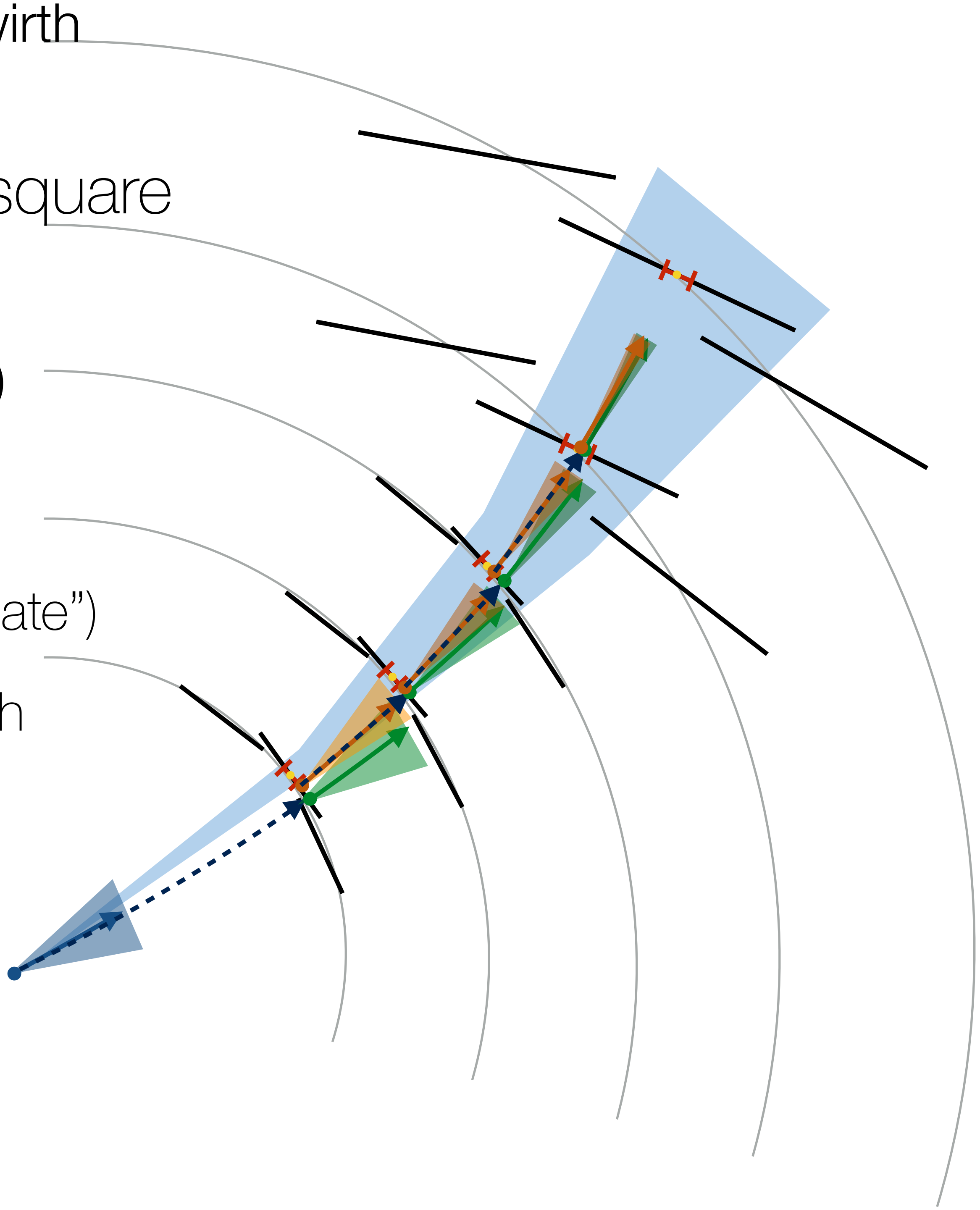
- performs a progressive way of least square estimation
  - equivalent to a χ^2 fit (if run with a smoother)
  - start with transport of track parameters (and covariances) to measurement surface, create predicted parameters ("predicted state")
  - combine/update predicted parameters with measurement to updated parameters ("filtered state")



The Kalman Filter

- offers an alternative solution to the large matrix inversion
  - initially developed by I. Kalman to track missiles
  - for HEP pioneered by Billoir and R. Fruehwirth

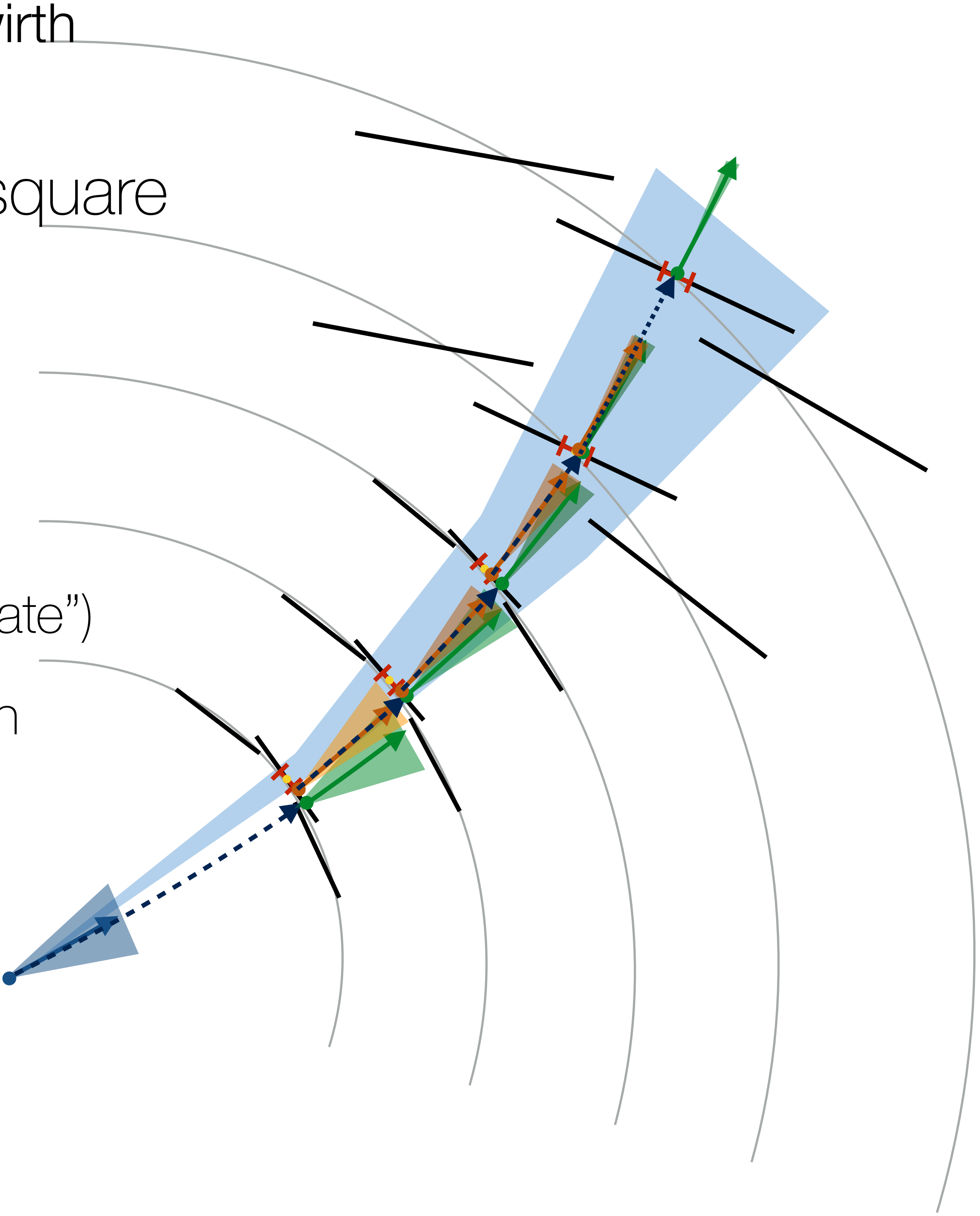
- performs a progressive way of least square estimation
  - equivalent to a χ^2 fit (if run with a smoother)
  - start with transport of track parameters (and covariances) to measurement surface, create predicted parameters ("predicted state")
  - combine/update predicted parameters with measurement to updated parameters ("filtered state")



The Kalman Filter

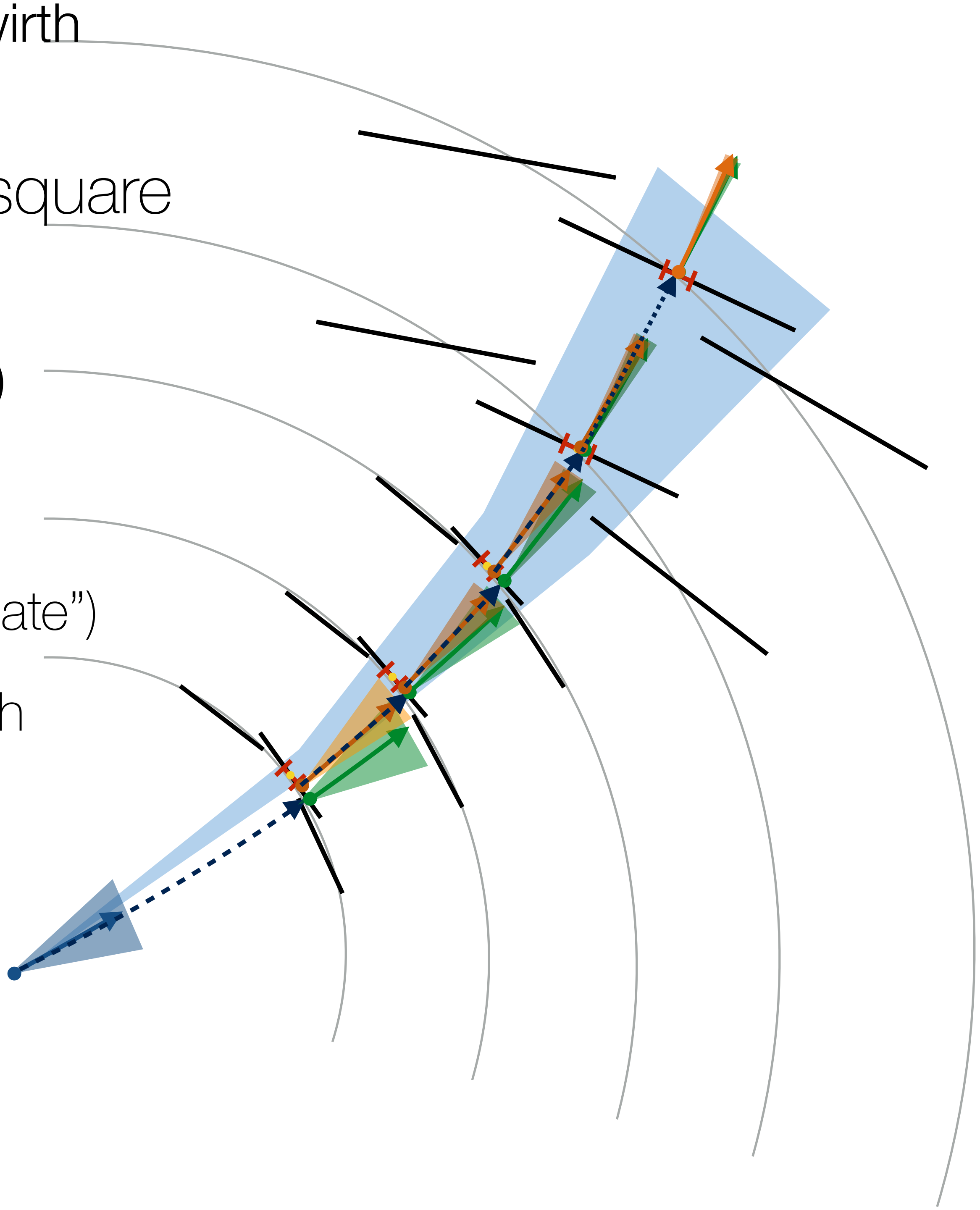
- offers an alternative solution to the large matrix inversion
  - initially developed by I. Kalman to track missiles
  - for HEP pioneered by Billoir and R. Fruehwirth

- performs a progressive way of least square estimation
  - equivalent to a χ^2 fit (if run with a smoother)
  - start with **transport** of track parameters (and covariances) to measurement surface, create **predicted parameters** ("predicted state")
  - combine/update **predicted parameters** with **measurement** to **updated parameters** ("filtered state")



The Kalman Filter

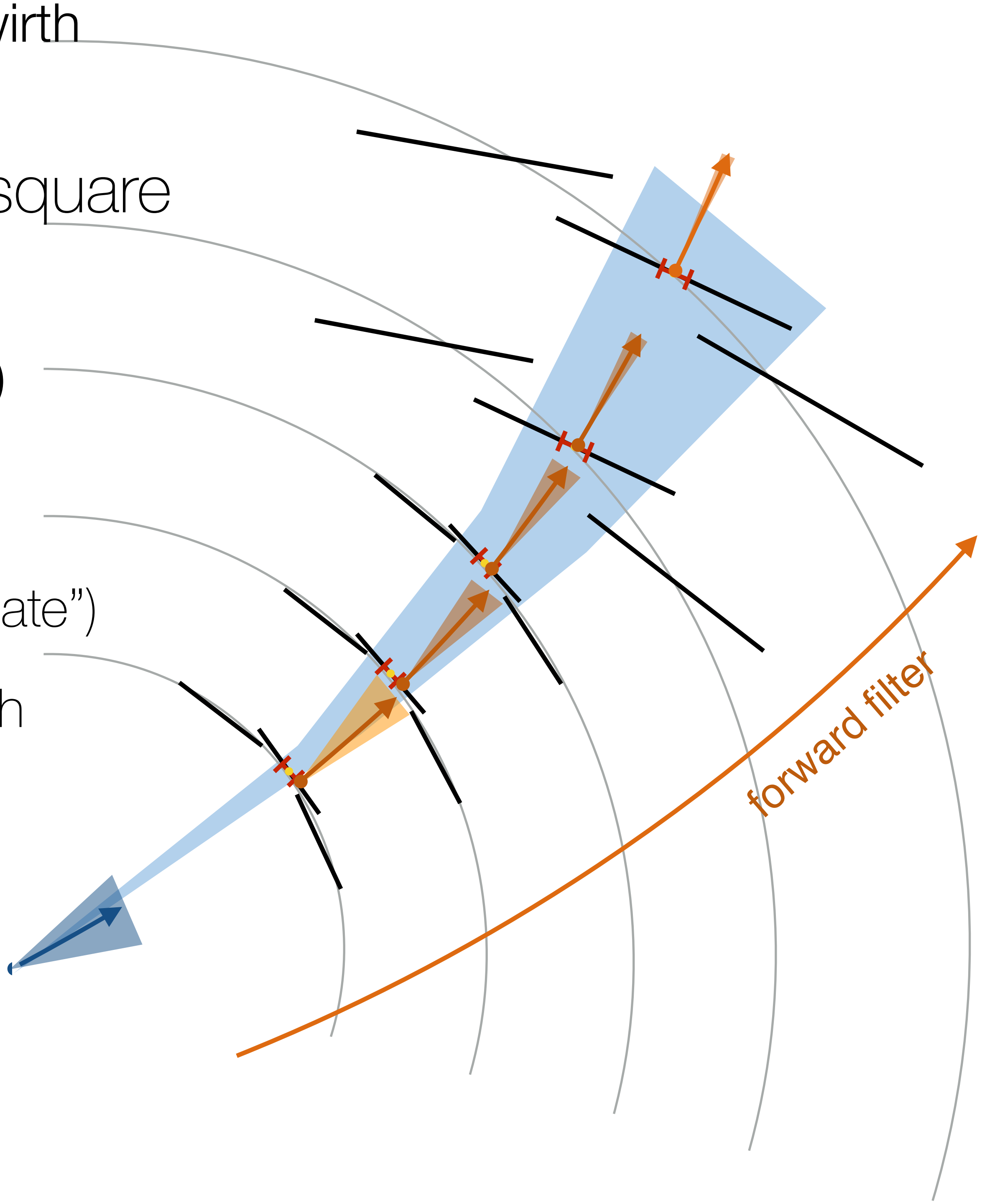
- offers an alternative solution to the large matrix inversion
  - initially developed by I. Kalman to track missiles
  - for HEP pioneered by Billoir and R. Fruehwirth
- performs a progressive way of least square estimation
  - equivalent to a χ^2 fit (if run with a smoother)
  - start with **transport** of track parameters (and covariances) to measurement surface, create **predicted parameters** (“predicted state”)
  - combine/update **predicted parameters** with **measurement** to **updated parameters** (“filtered state”)



The Kalman Filter

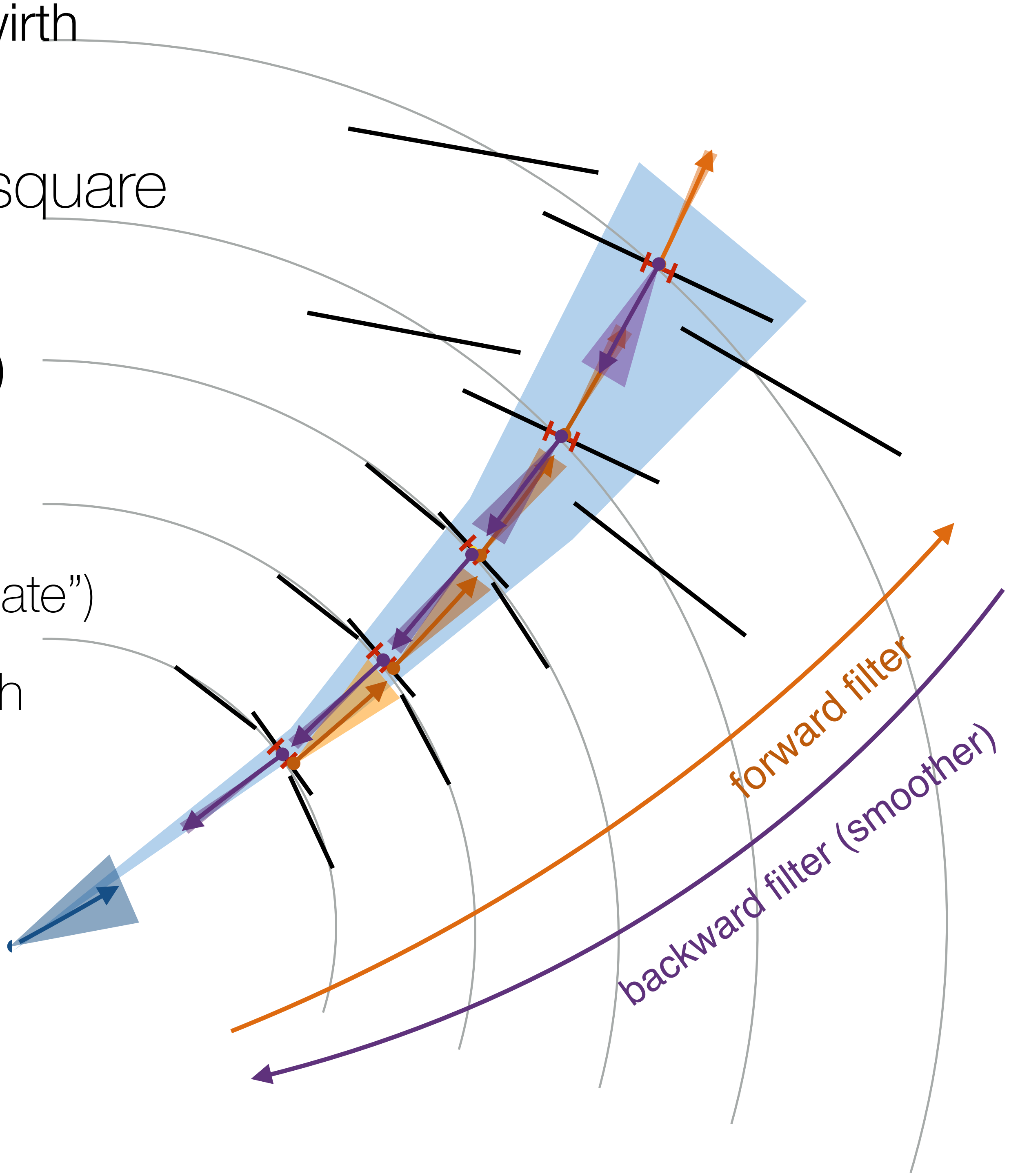
- offers an alternative solution to the large matrix inversion
  - initially developed by I. Kalman to track missiles
  - for HEP pioneered by Billoir and R. Fruehwirth

- performs a progressive way of least square estimation
  - equivalent to a χ^2 fit (if run with a smoother)
  - start with **transport** of track parameters (and covariances) to measurement surface, create **predicted parameters** ("predicted state")
  - combine/update **predicted parameters** with **measurement** to **updated parameters** ("filtered state")



The Kalman Filter

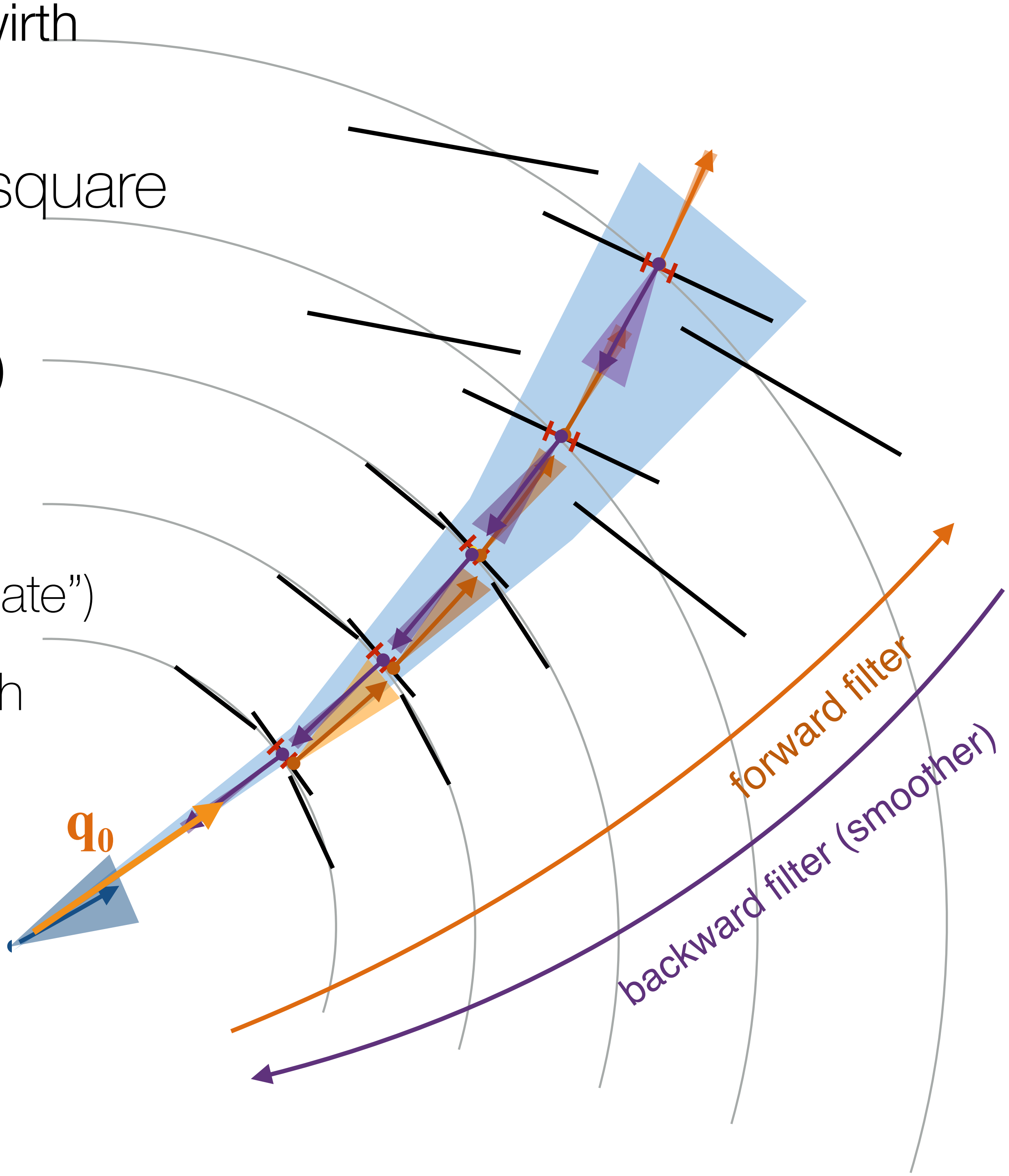
- offers an alternative solution to the large matrix inversion
  - initially developed by I. Kalman to track missiles
  - for HEP pioneered by Billoir and R. Fruehwirth
- performs a progressive way of least square estimation
  - equivalent to a χ^2 fit (if run with a smoother)
  - start with **transport** of track parameters (and covariances) to measurement surface, create **predicted parameters** ("predicted state")
  - combine/update **predicted parameters** with **measurement** to **updated parameters** ("filtered state")



The Kalman Filter

- offers an alternative solution to the large matrix inversion
  - initially developed by I. Kalman to track missiles
  - for HEP pioneered by Billoir and R. Fruehwirth

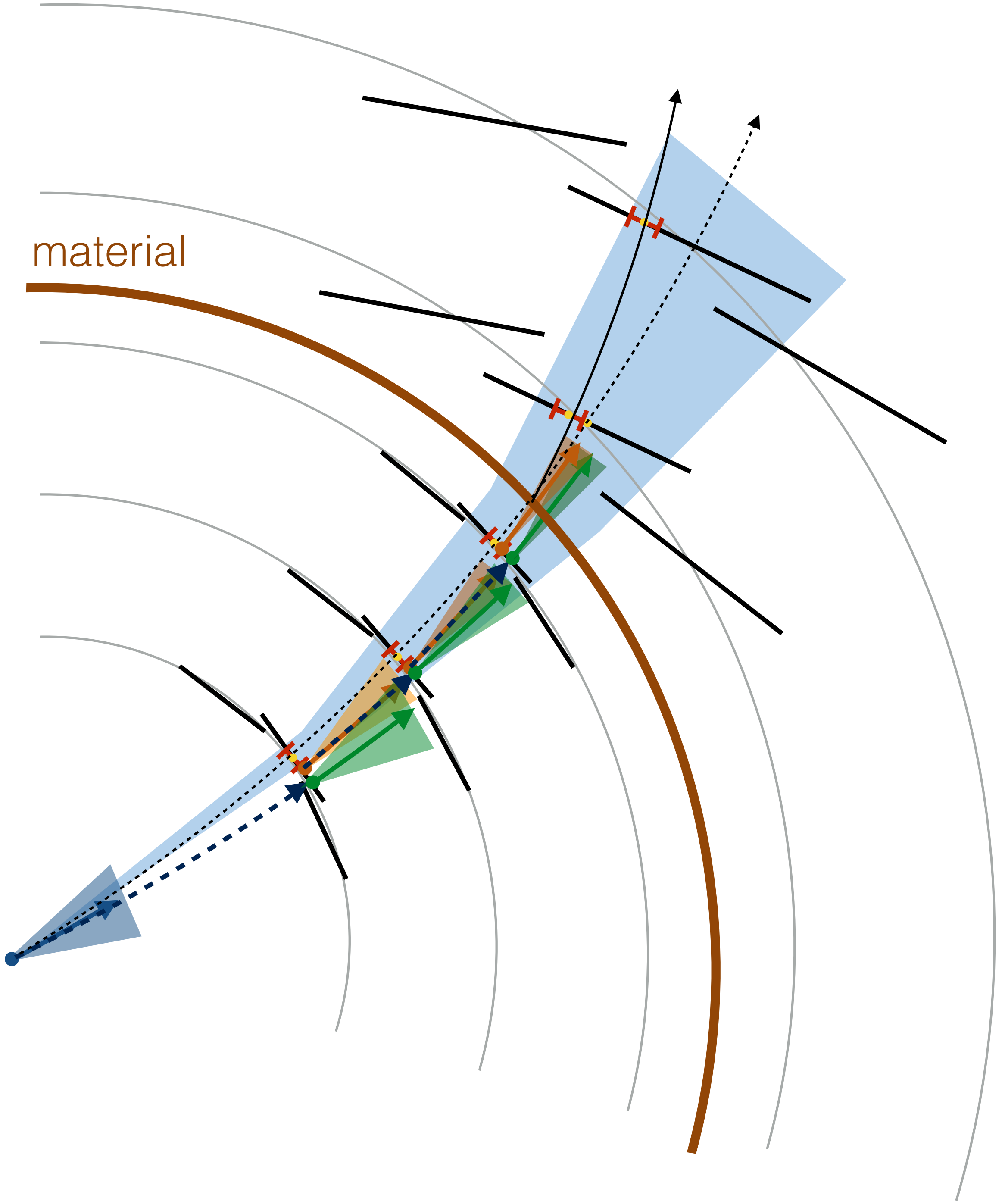
- performs a progressive way of least square estimation
  - equivalent to a χ^2 fit (if run with a smoother)
  - start with transport of track parameters (and covariances) to measurement surface, create predicted parameters ("predicted state")
  - combine/update predicted parameters with measurement to updated parameters ("filtered state")



The Kalman Filter with material

- when crossing a material layer
  - increase covariance by “noise” term according to the amount of material crossed

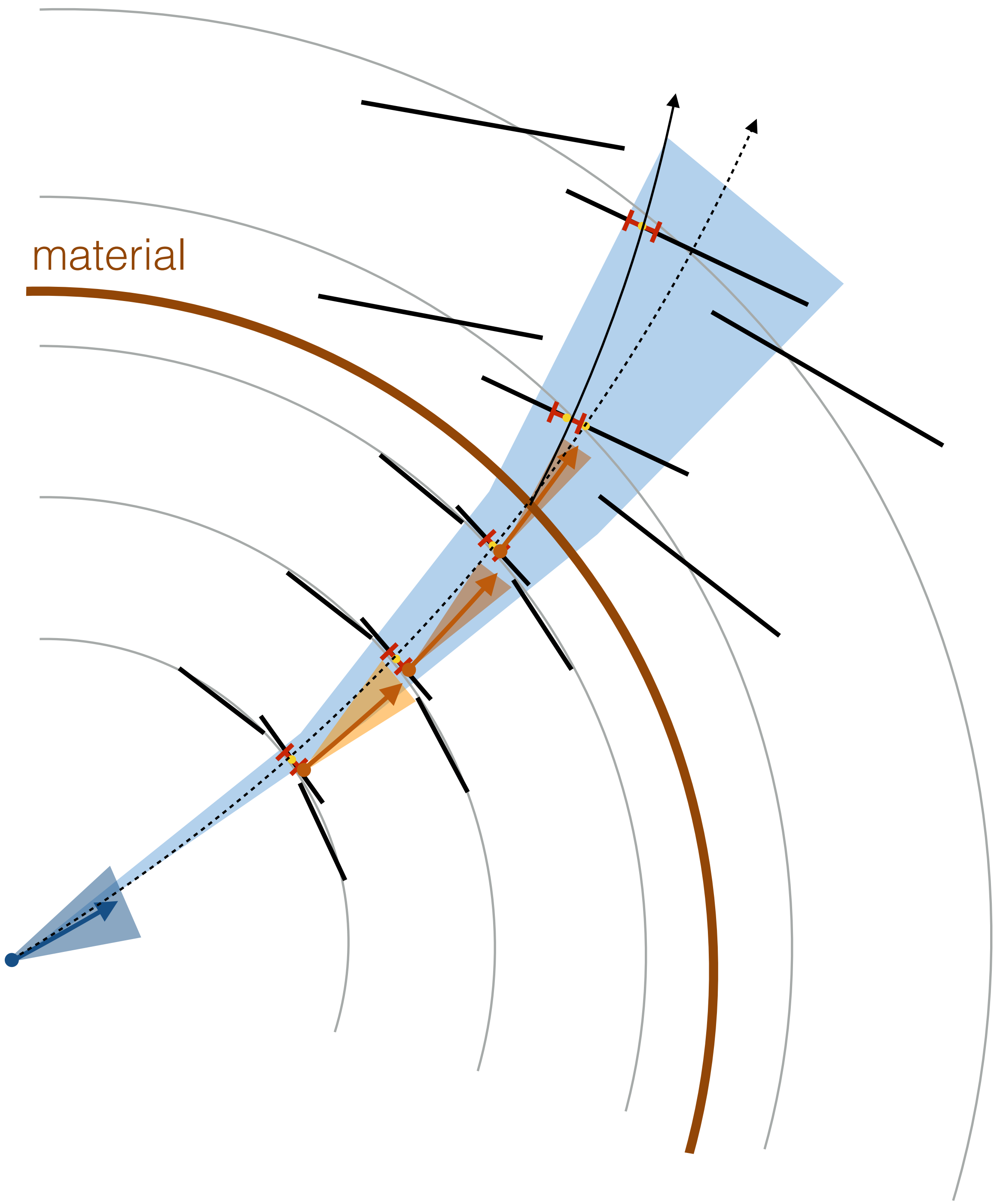
    (scattering has expected mean of 0)
  - energy loss is applied deterministically
    (additional noise term for straggling added)



The Kalman Filter with material

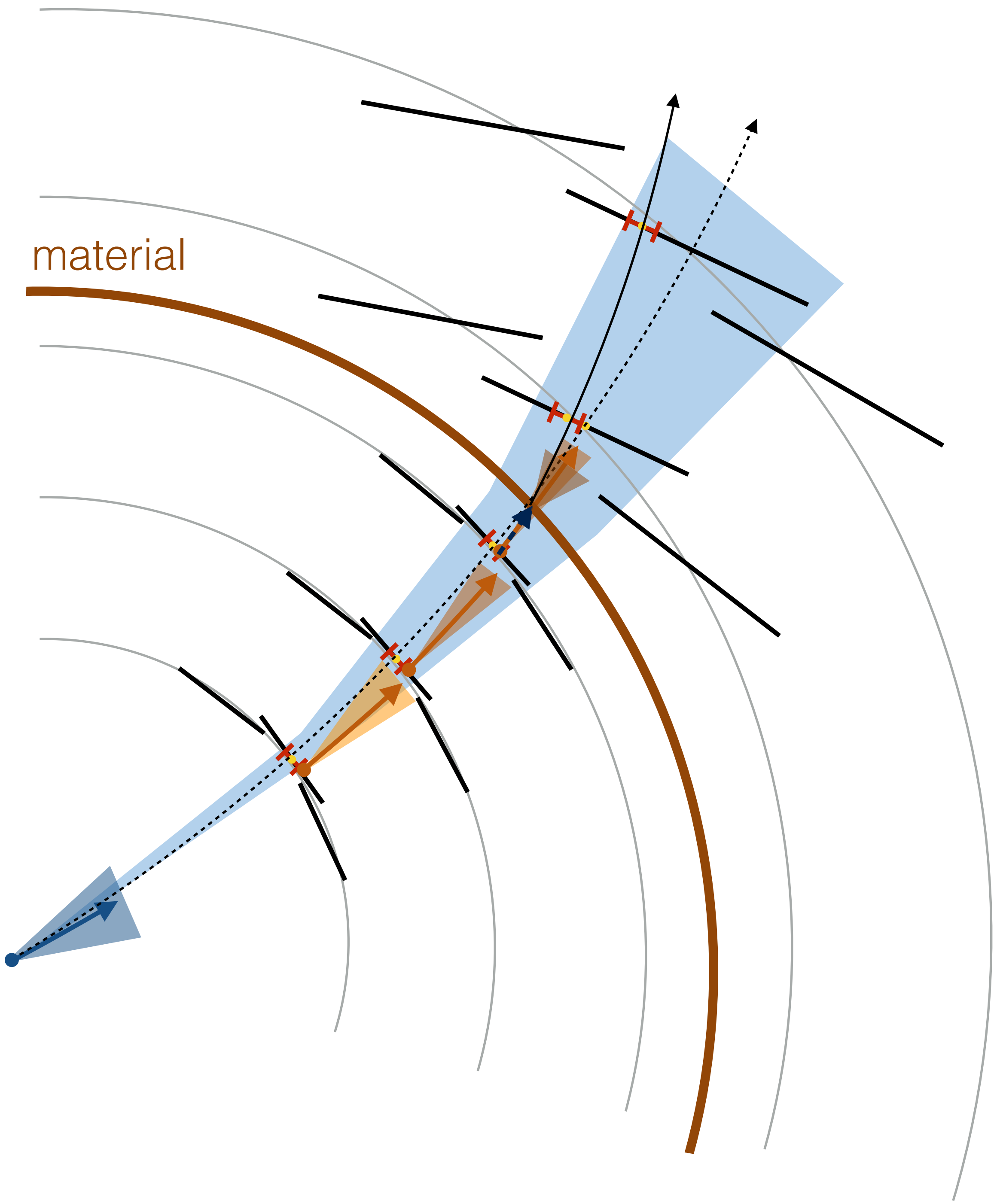
- when crossing a material layer
  - increase covariance by “noise” term according to the amount of material crossed

    (scattering has expected mean of 0)
  - energy loss is applied deterministically
    (additional noise term for straggling added)



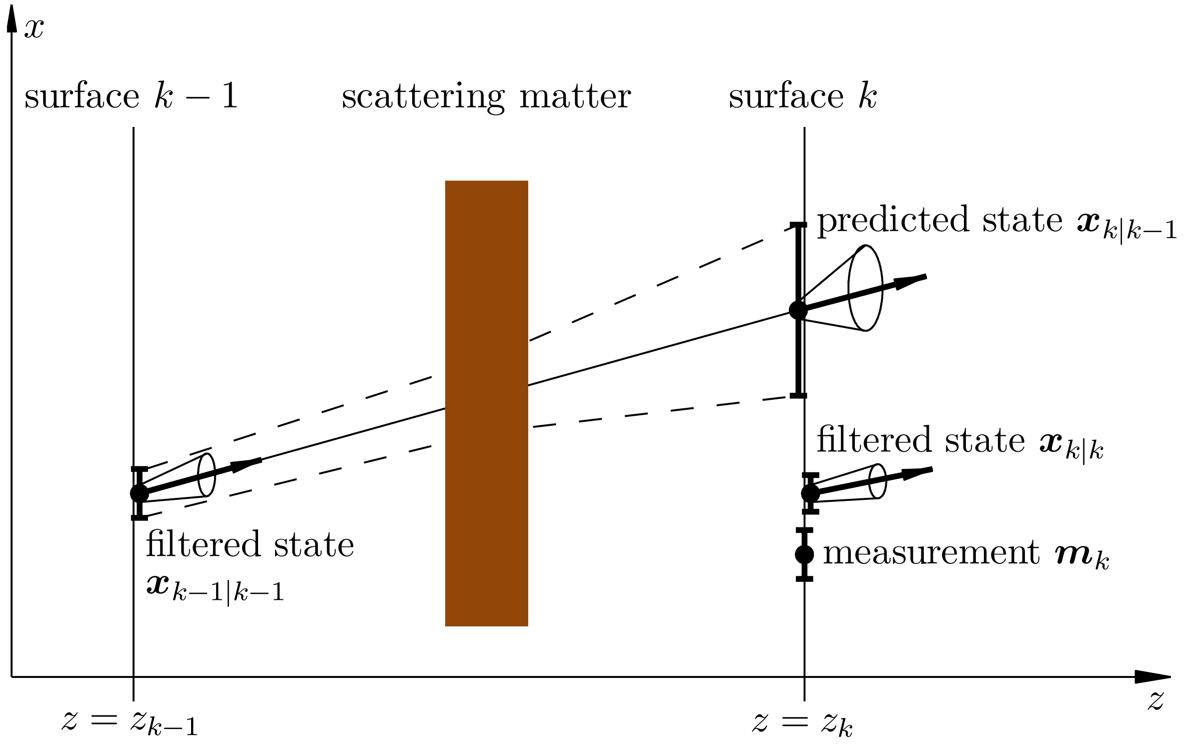
The Kalman Filter with material

- when crossing a material layer
  - increase covariance by “noise” term according to the amount of material crossed (scattering has expected mean of 0)
  - energy loss is applied deterministically (additional noise term for straggling added)

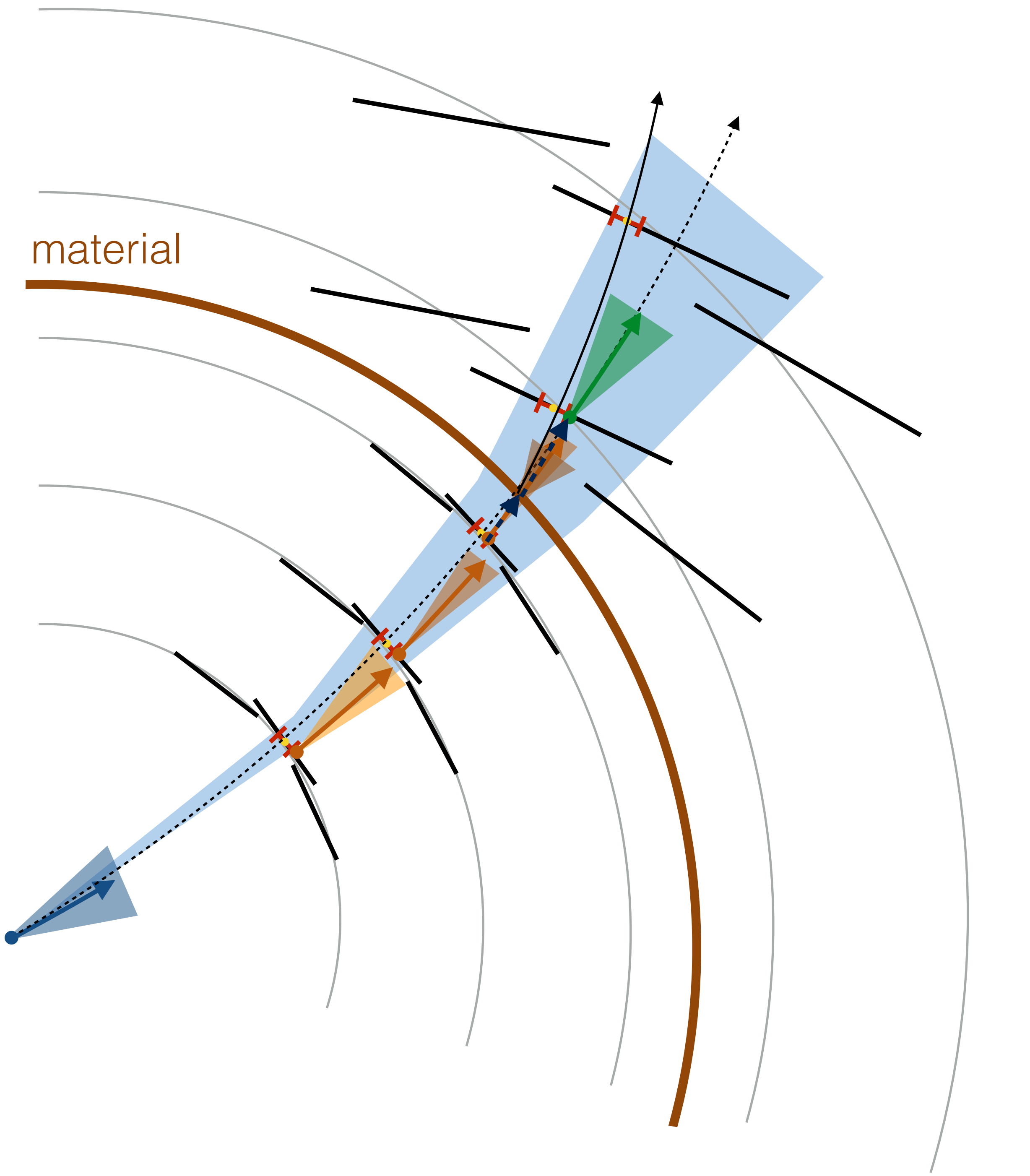


The Kalman Filter with material

- when crossing a material layer
  - increase covariance by “noise” term according to the amount of material crossed (scattering has expected mean of 0)
  - energy loss is applied deterministically (additional noise term for straggling added)



x
surface $k-1$
scattering matter
surface k
predicted state $\boldsymbol{x}_{k|k-1}$
filtered state $\boldsymbol{x}_{k|k}$
measurement $\boldsymbol{m}_k$
filtered state $\boldsymbol{x}_{k-1|k-1}$
$z = z_{k-1}$
$z = z_k$
z



The Kalman Filter in maths

- let's assume the ***k-th*** filter step
  - propagate parameters and **covariances** from *k-1* to *k* adding noise $\boldsymbol{Q}_k$ if present

$$\boldsymbol{q}_{k|k-1} = \boldsymbol{f}_{k|k-1}(\boldsymbol{q}_{k-1|k-1})$$

$$\boldsymbol{C}_{k|k-1} = \boldsymbol{F}_{k|k-1}\boldsymbol{C}_{k-1|k-1}\boldsymbol{F}^{\mathrm{T}}_{k|k-1} + \boldsymbol{Q}_k$$

  - update the prediction with measurement

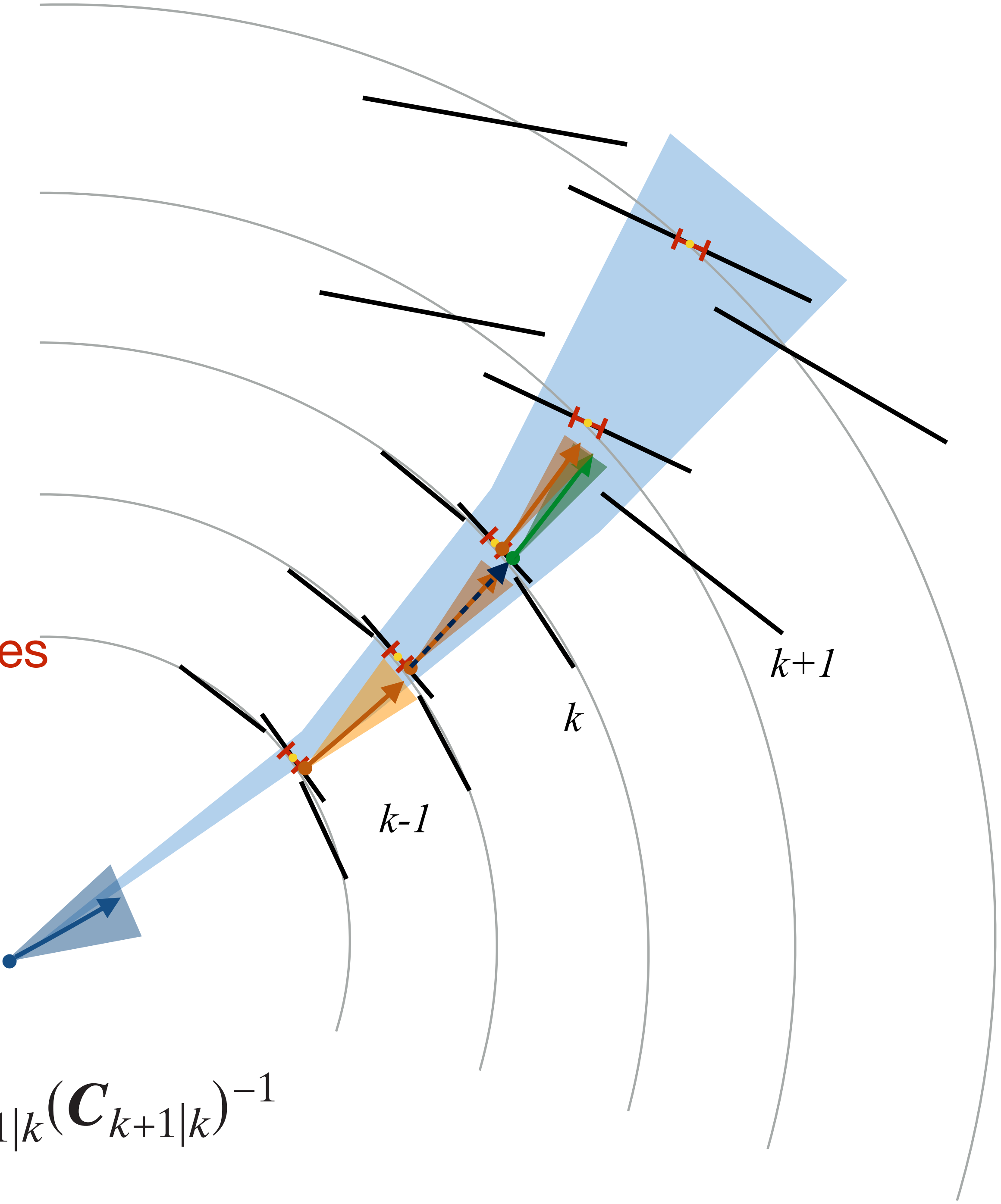
$$\boldsymbol{q}_{k|k} = \boldsymbol{q}_{k|k-1} + \boldsymbol{K}_k[\boxed{\boldsymbol{m}_k} - \boxed{\boldsymbol{h}_k}(\boldsymbol{q}_{k|k-1})]$$

$$\boldsymbol{C}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{C}_{k|k-1}$$

with gain matrix $\boldsymbol{K}_k$:

$$\boldsymbol{K}_k = \boldsymbol{C}_{k|k-1}\boxed{\boldsymbol{H}^{\mathrm{T}}_k}(\boxed{\boldsymbol{G}_k} + \boldsymbol{H}_k\boldsymbol{C}_{k|k-1}\boldsymbol{H}^{\mathrm{T}}_k)^{-1}$$

mapping measurement covariances



- run the smoother from *k+1* to *k*

$$\boldsymbol{q}_{k|n} = \boldsymbol{q}_{k|k} + \boldsymbol{A}_k(\boldsymbol{q}_{k+1|n} - \boldsymbol{q}_{k+1|k})$$

$$\boldsymbol{C}_{k|n} = \boldsymbol{C}_{k|k} - \boldsymbol{A}_k(\boldsymbol{C}_{k+1|k} - \boldsymbol{C}_{k+1|n})\boldsymbol{A}^{\mathrm{T}}_k$$

with smoother gain matrix $\boldsymbol{A}_k$: $\boldsymbol{A}_k = \boldsymbol{C}_{k|k}\boldsymbol{F}^{\mathrm{T}}_{k+1|k}(\boldsymbol{C}_{k+1|k})^{-1}$

Reminder

Wait a second …

- Global χ^2 fitter and Kalman filter are least squares estimators that rely on gaussian errors:

$\boldsymbol{G_k}$ the covariance of measurement $\mathbf{m}_k$

$\boldsymbol{Q}_k$ the noise addition due to material effects (Kalman filter)

$\sum_i \delta\theta_i{}^T Q_i^{-1} \delta\theta_i$ χ^2 contribution from scattering angles (χ^2 fitter)

Reminder

Wait a second …

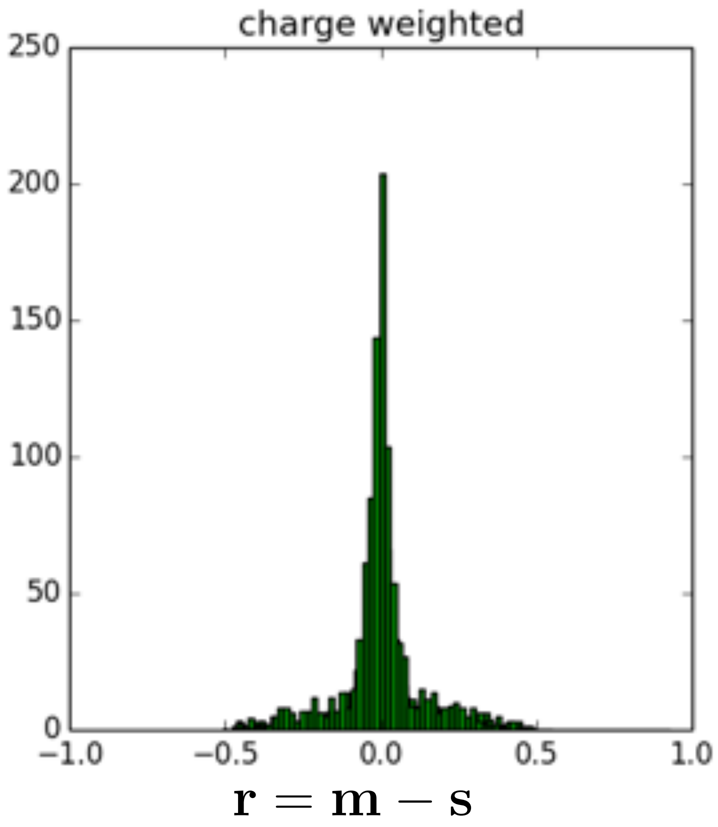
- Global χ^2 fitter and Kalman filter are least squares estimators that rely on gaussian errors:

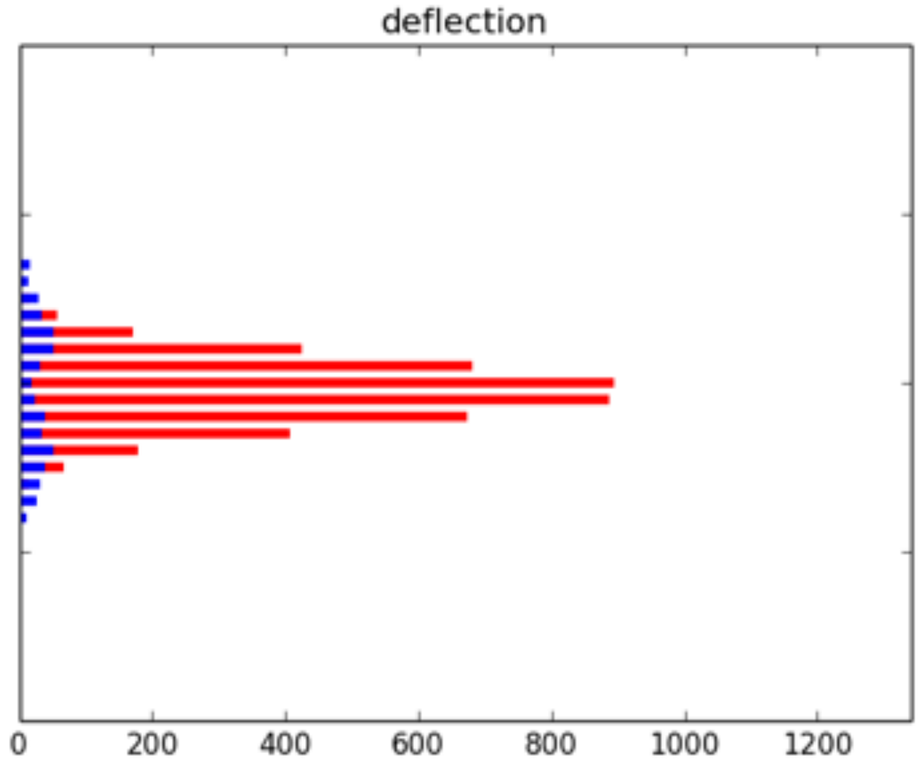
$\boldsymbol{G_k}$ the covariance of measurement $\mathbf{m}_k$

$\boldsymbol{Q}_k$ the noise addition due to material effects (Kalman filter)

$\sum_i \delta\theta_i^T Q_i^{-1} \delta\theta_i$ χ^2 contribution from scattering angles (χ^2 fitter)

neither of them are !





Outliers

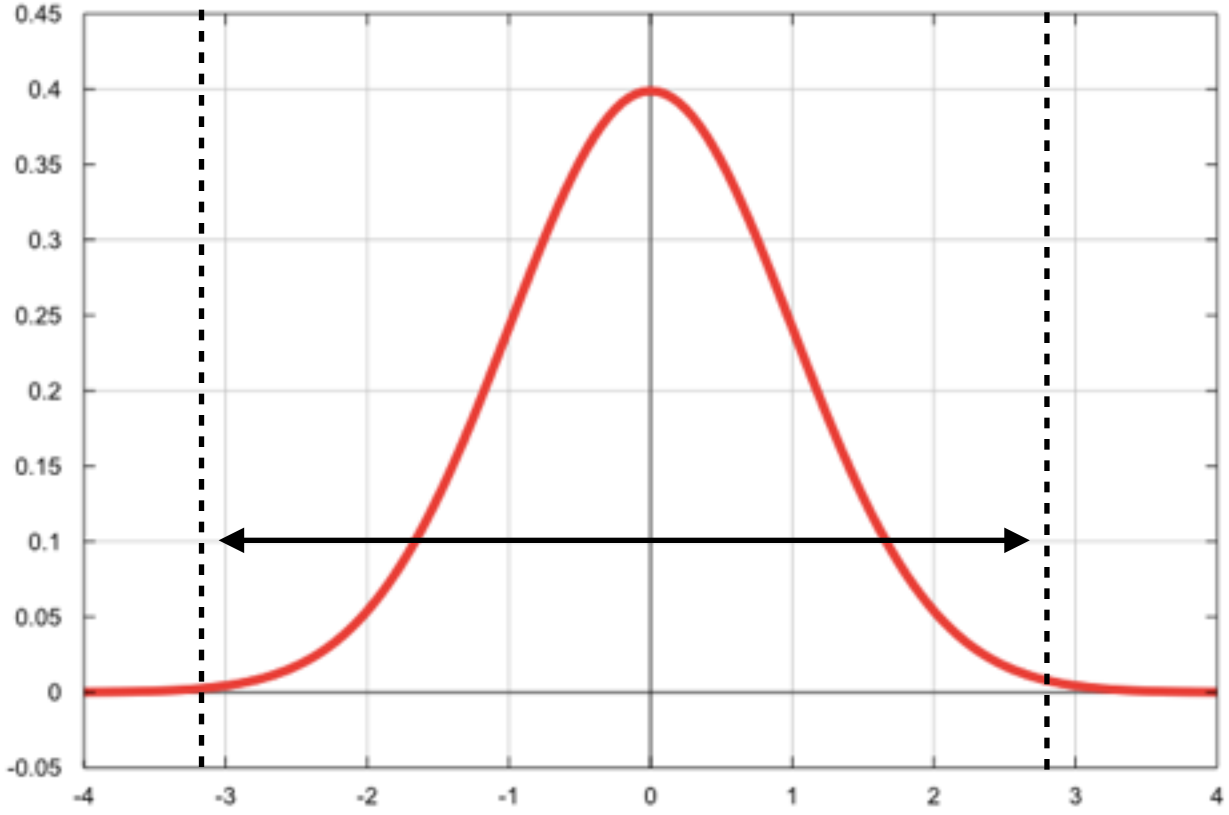
- What is a compatible measurement ?
  - first of all: that's a definition, usually bound to a χ^2 compatibility cut
  - assuming a perfect gaussian system:

    there is a probability of hits being outside any range, usually defined

    as <u>outliers</u> if found by pattern recognition by rejected by fit
  - non-gaussian tails increase the outlier probability

    non-gaussian measurement p.d.f.

    or non-gaussian noise effects increase the risk of outliers



- Outliers do not contribute to the track fit
  - they are a good quality measure of the track though

Understanding the track fit output

- Track fit yields
  - fit quality measure, usually χ^2 over number of degrees of freedom
  - fitted parameters (e.g. expressed at perigee) and associated error matrix

$$\mathbf{q} = (d_0, z_0, \phi, \theta, q/p)$$

$$\mathbf{C} = \begin{pmatrix} \sigma^2(d_0) & cov(d_0, z_0) & cov(d_0, \phi) & cov(d_0, \theta) & cov(d_0, q/p) \\ . & \sigma^2(z_0) & cov(z_0, \phi) & cov(z_0, \theta) & cov(z_0, q/p) \\ . & . & \sigma^2(\phi) & cov(\phi, \theta) & cov(\phi, q/p) \\ . & . & . & \sigma^2(\theta) & cov(\theta, q/p) \\ . & . & . & . & \sigma^2(q/p) \end{pmatrix}$$

Understanding the track fit output

- Track fit yields
  - fit quality measure, usually χ^2 over number of degrees of freedom
  - fitted parameters (e.g. expressed at perigee) and associated error matrix

$$\mathbf{q} = (d_0, z_0, \phi, \theta, q/p)$$

$$\mathbf{C} = \begin{pmatrix} \sigma^2(d_0) & cov(d_0, z_0) & cov(d_0, \phi) & cov(d_0, \theta) & cov(d_0, q/p) \\ . & \sigma^2(z_0) & cov(z_0, \phi) & cov(z_0, \theta) & cov(z_0, q/p) \\ . & . & \sigma^2(\phi) & cov(\phi, \theta) & cov(\phi, q/p) \\ . & . & . & \sigma^2(\theta) & cov(\theta, q/p) \\ . & . & . & . & \sigma^2(q/p) \end{pmatrix}$$

diagonal elements: **errors on the parameters**

Understanding the track fit output

- Track fit yields
  - fit quality measure, usually χ^2 over number of degrees of freedom
  - fitted parameters (e.g. expressed at perigee) and associated error matrix

$$\mathbf{q} = (d_0, z_0, \phi, \theta, q/p)$$

$$\mathbf{C} = \begin{pmatrix} \sigma^2(d_0) & cov(d_0, z_0) & cov(d_0, \phi) & cov(d_0, \theta) & cov(d_0, q/p) \\ . & \sigma^2(z_0) & cov(z_0, \phi) & cov(z_0, \theta) & cov(z_0, q/p) \\ . & . & \sigma^2(\phi) & cov(\phi, \theta) & cov(\phi, q/p) \\ . & . & . & \sigma^2(\theta) & cov(\theta, q/p) \\ . & . & . & . & \sigma^2(q/p) \end{pmatrix}$$

diagonal elements: **errors on the parameters**

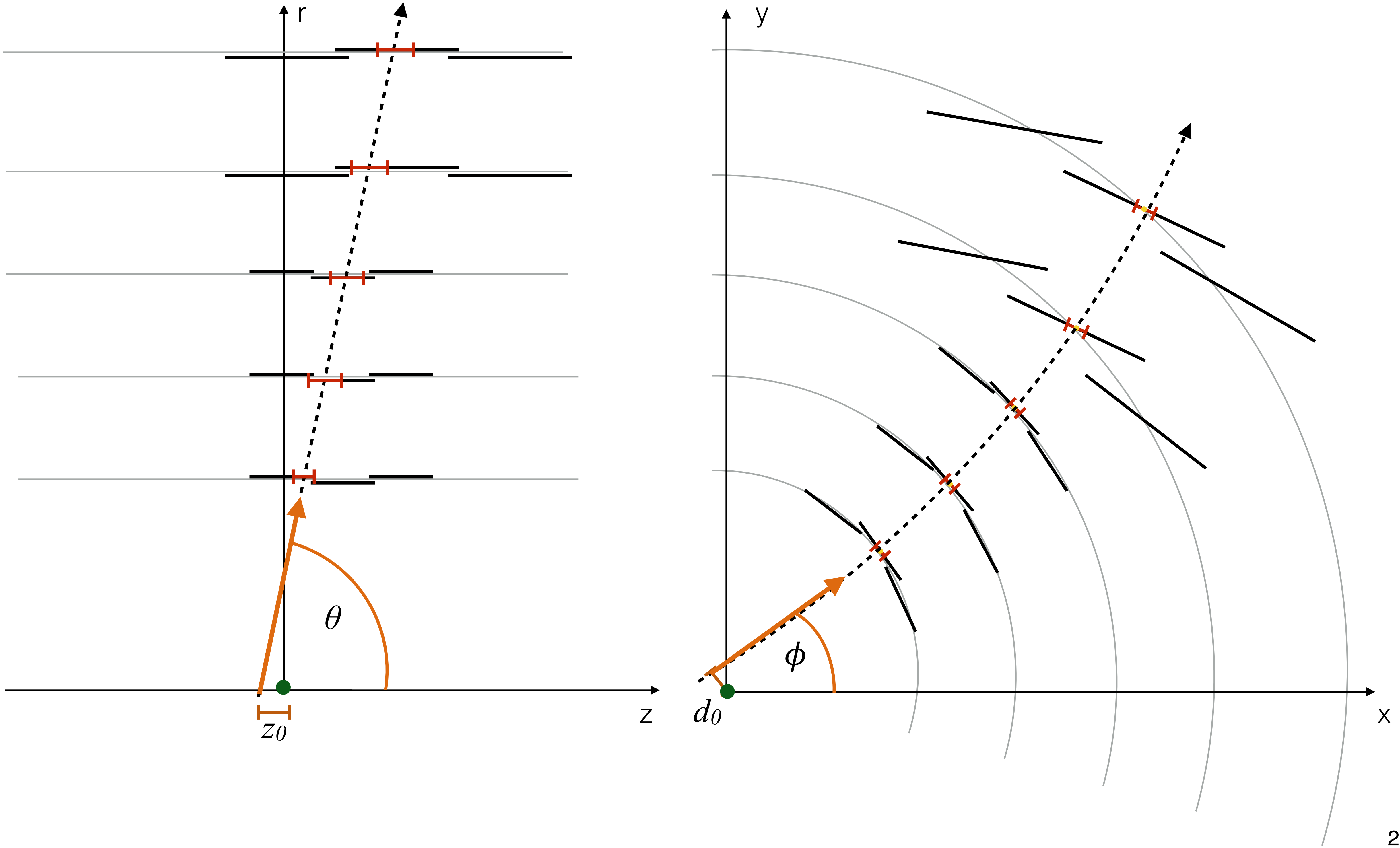
off-diagonal elements: **include the correlations between the parameters**

$$cov(q_i, q_k) = \varrho_{ik}\sigma_i\sigma_k$$

correlation coefficient

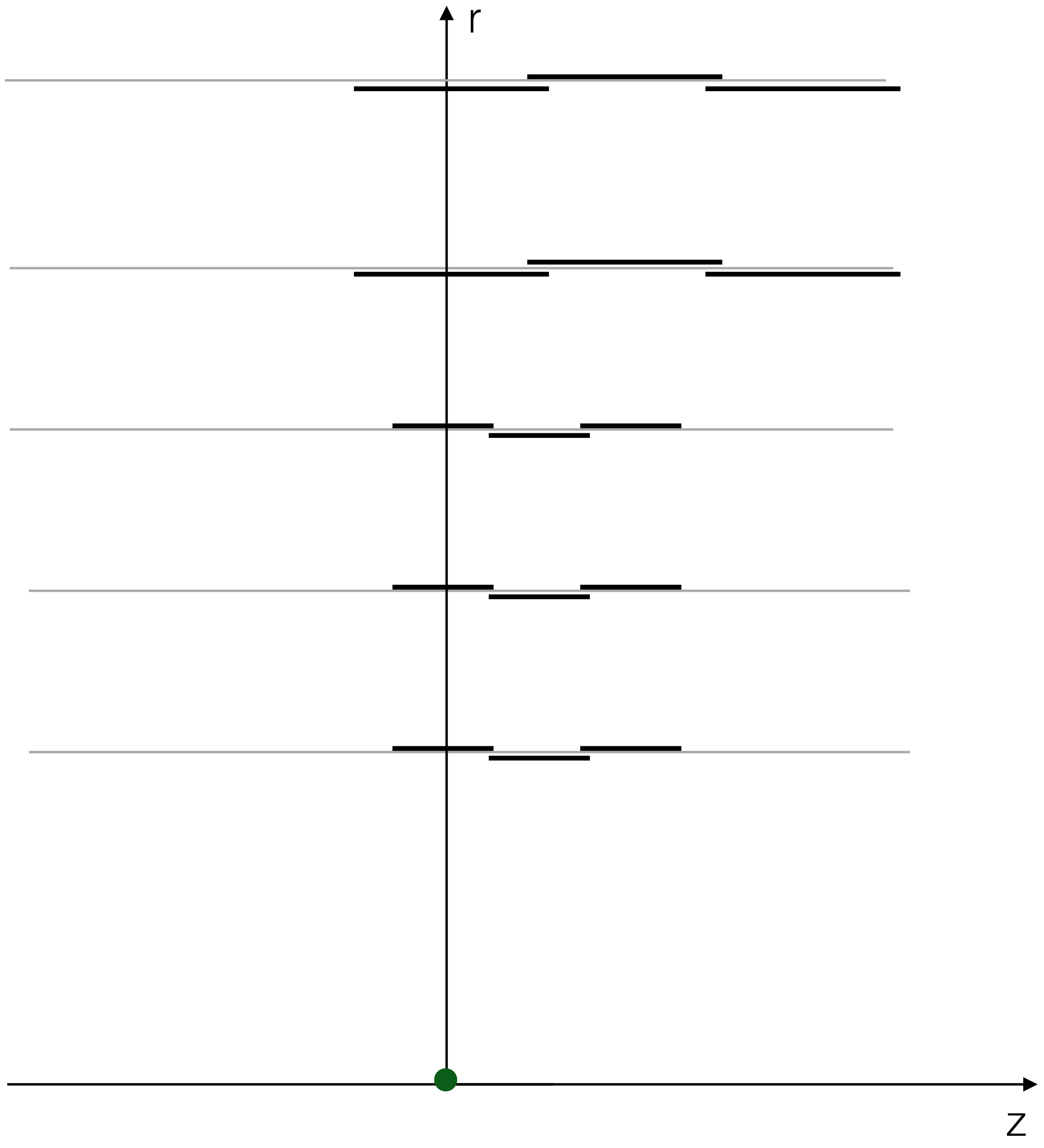
Understanding the track fit output: parameters

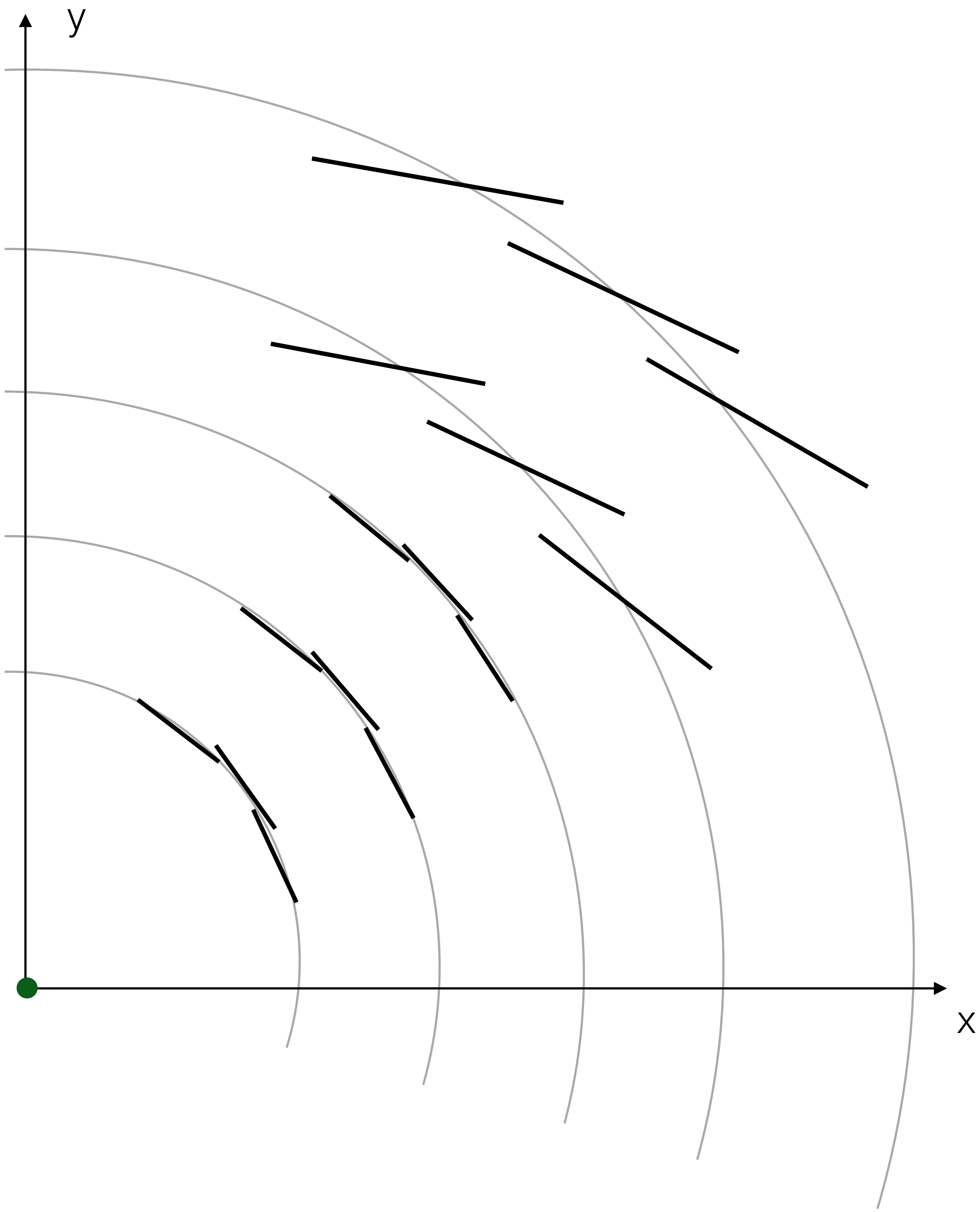
- What do large impact parameters mean ?
  - imagine a neutral particle decaying somewhere in the detector



Understanding the track fit output: parameters

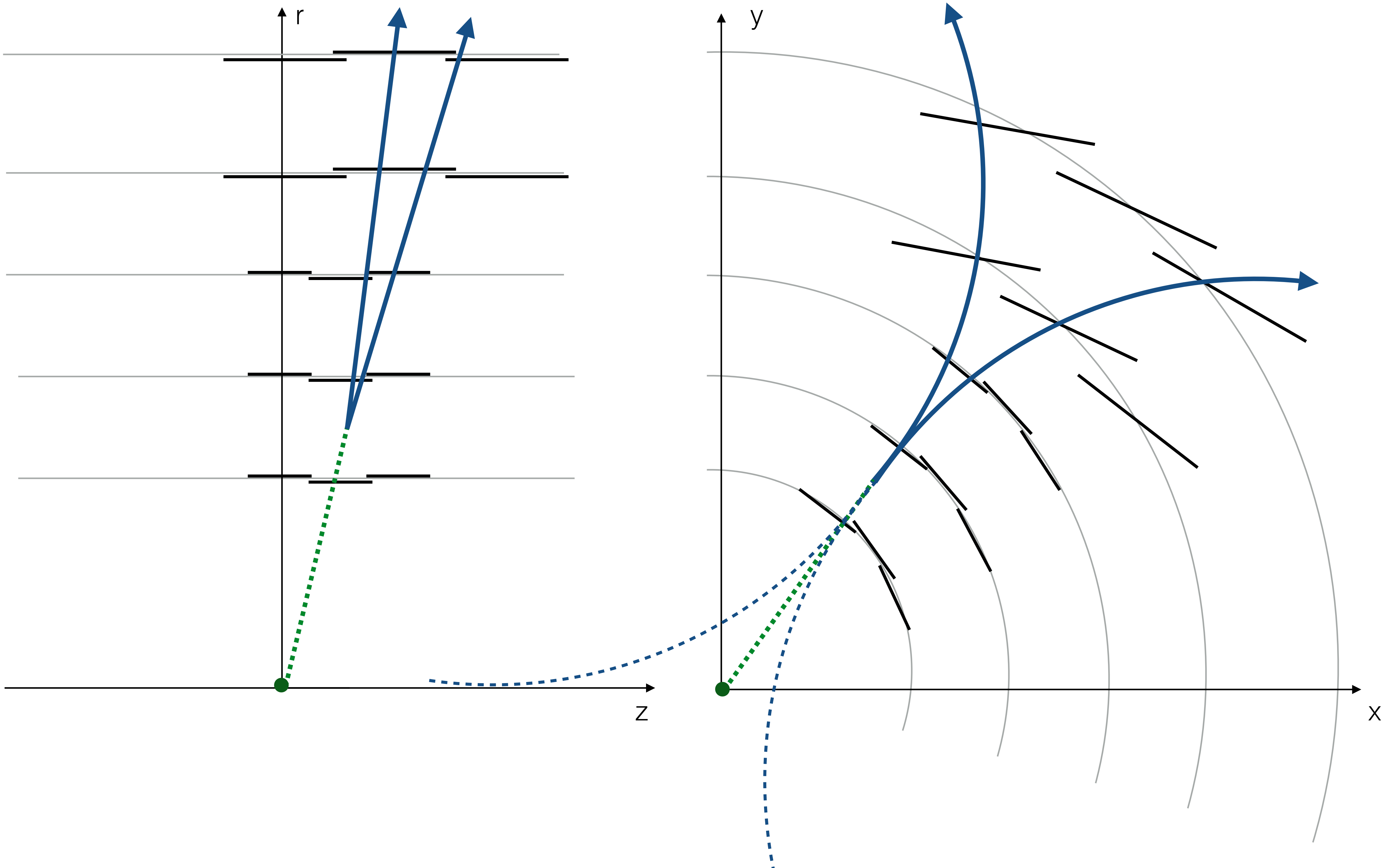
- What do large impact parameters mean ?





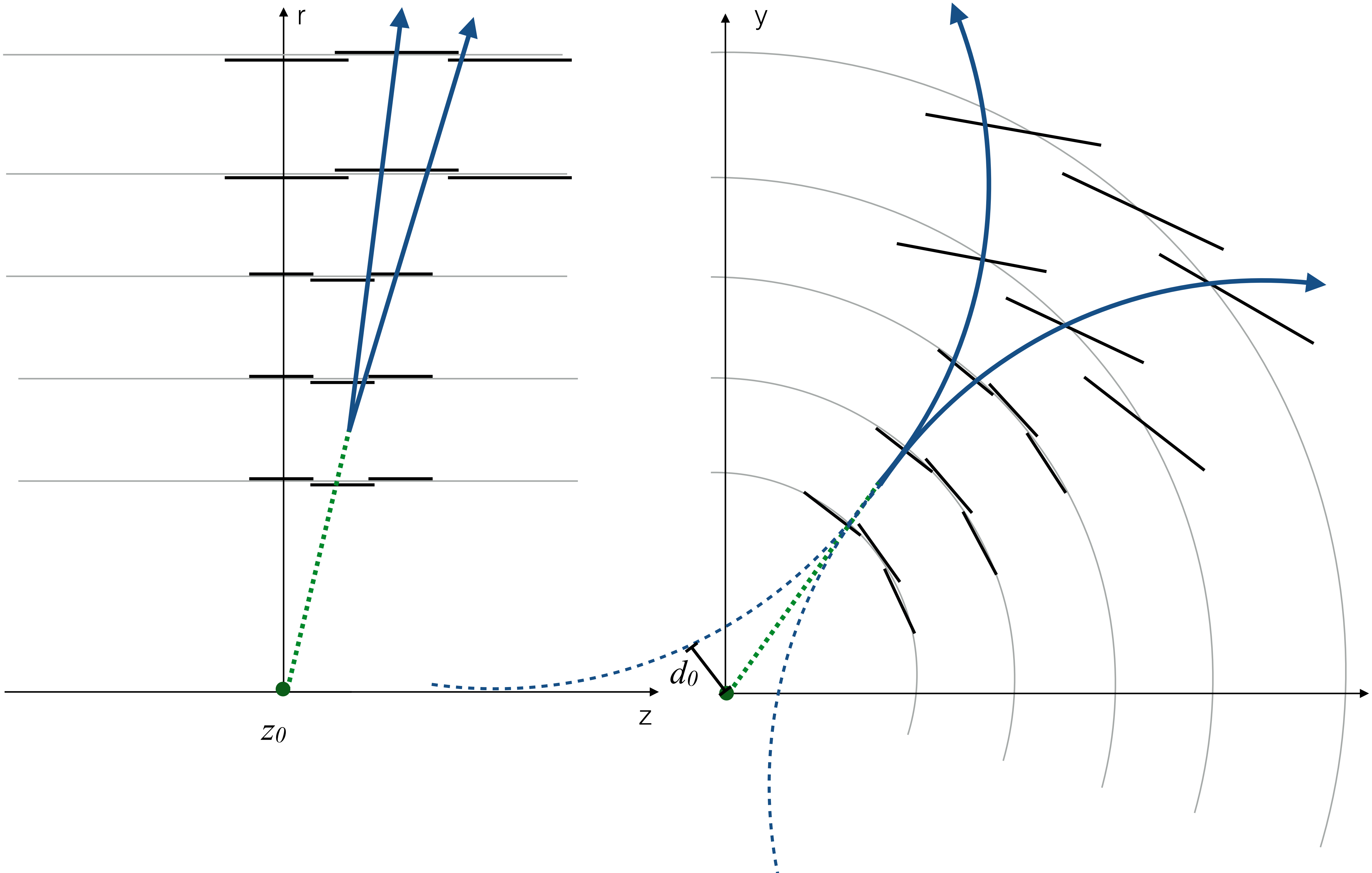
Understanding the track fit output: parameters

- What do large impact parameters mean ?



Understanding the track fit output: <u>parameters</u>

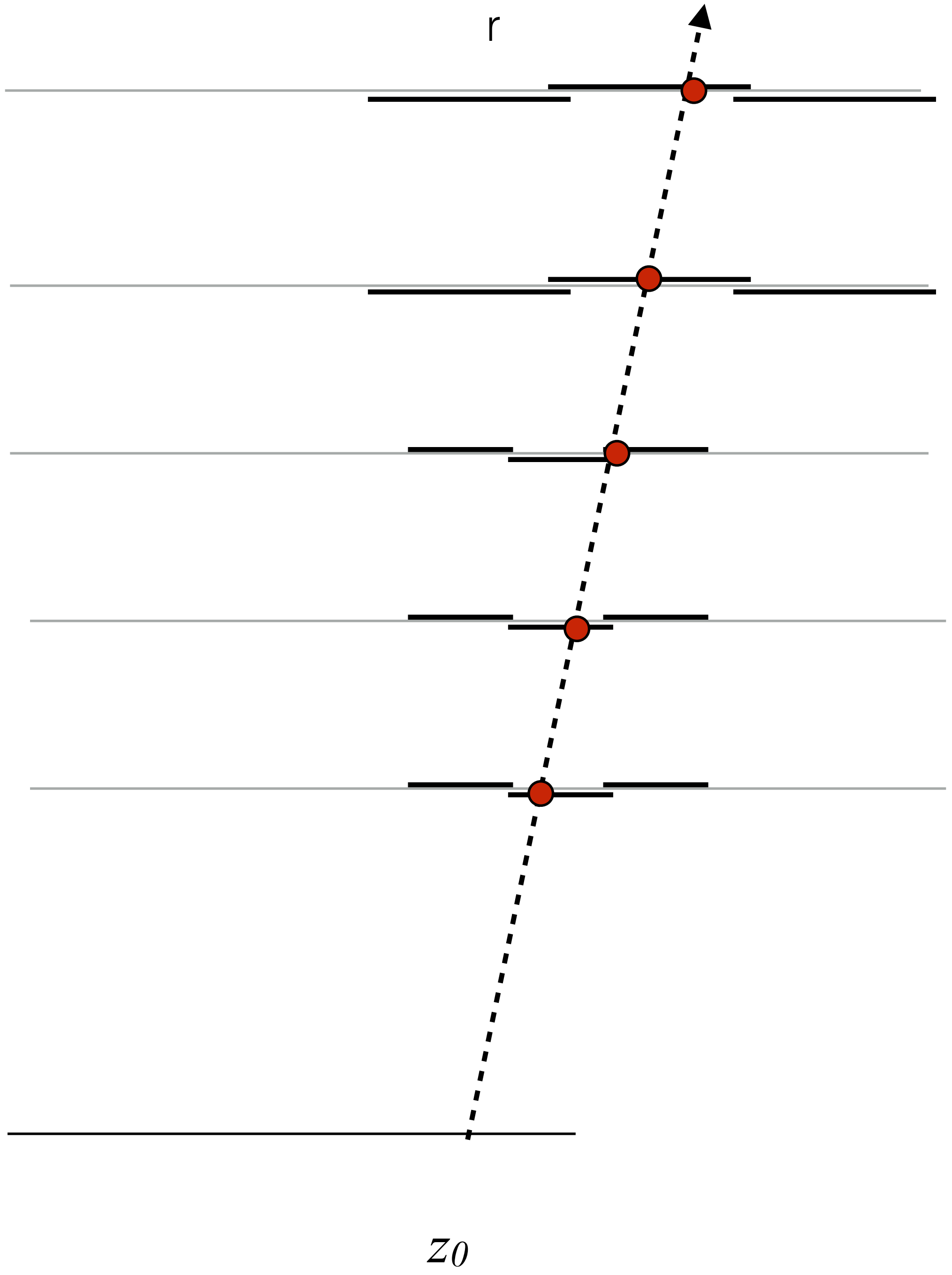
- What do large impact parameters mean ?

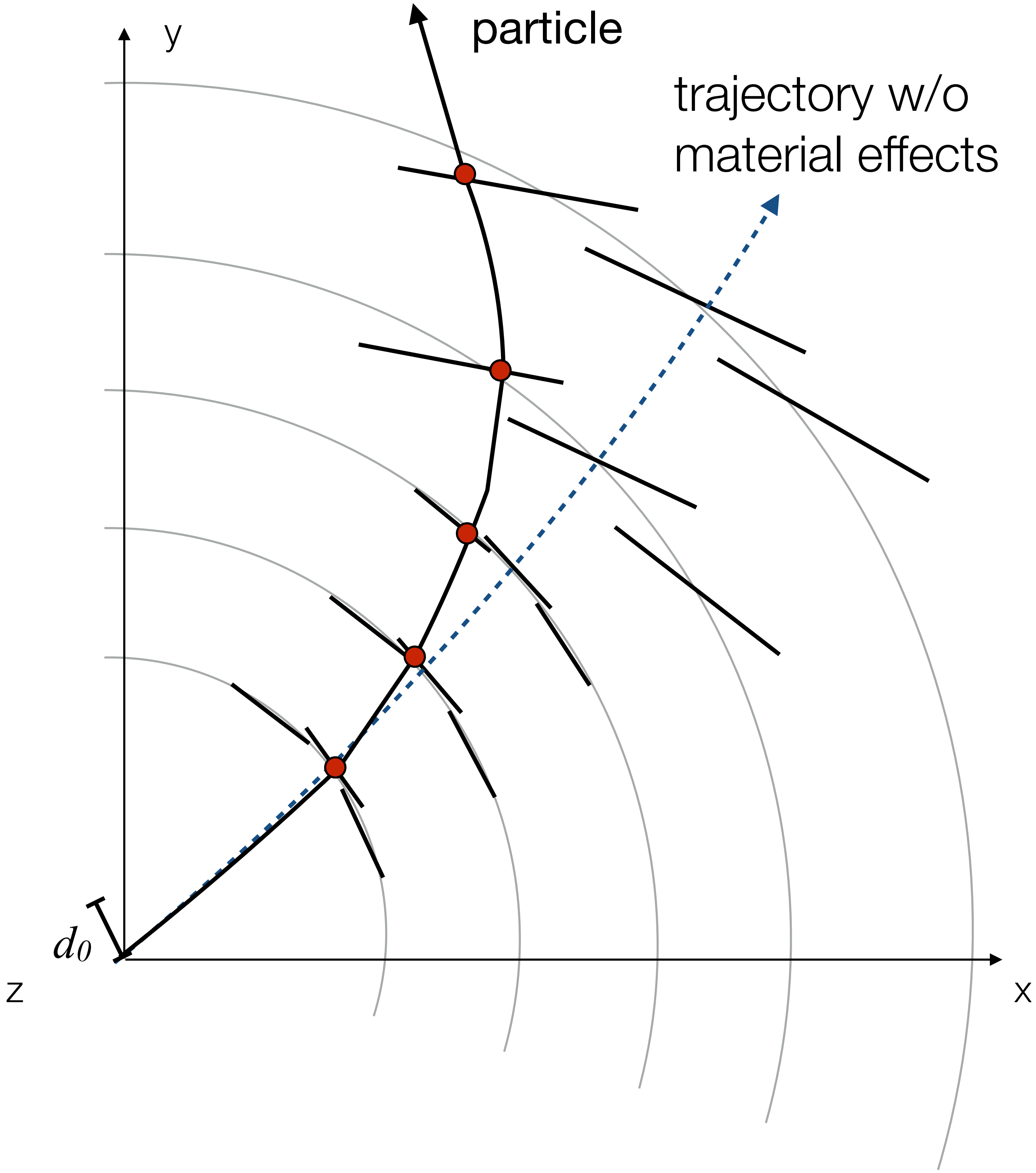


INTERLUDE

Understanding the track fit output: parameters

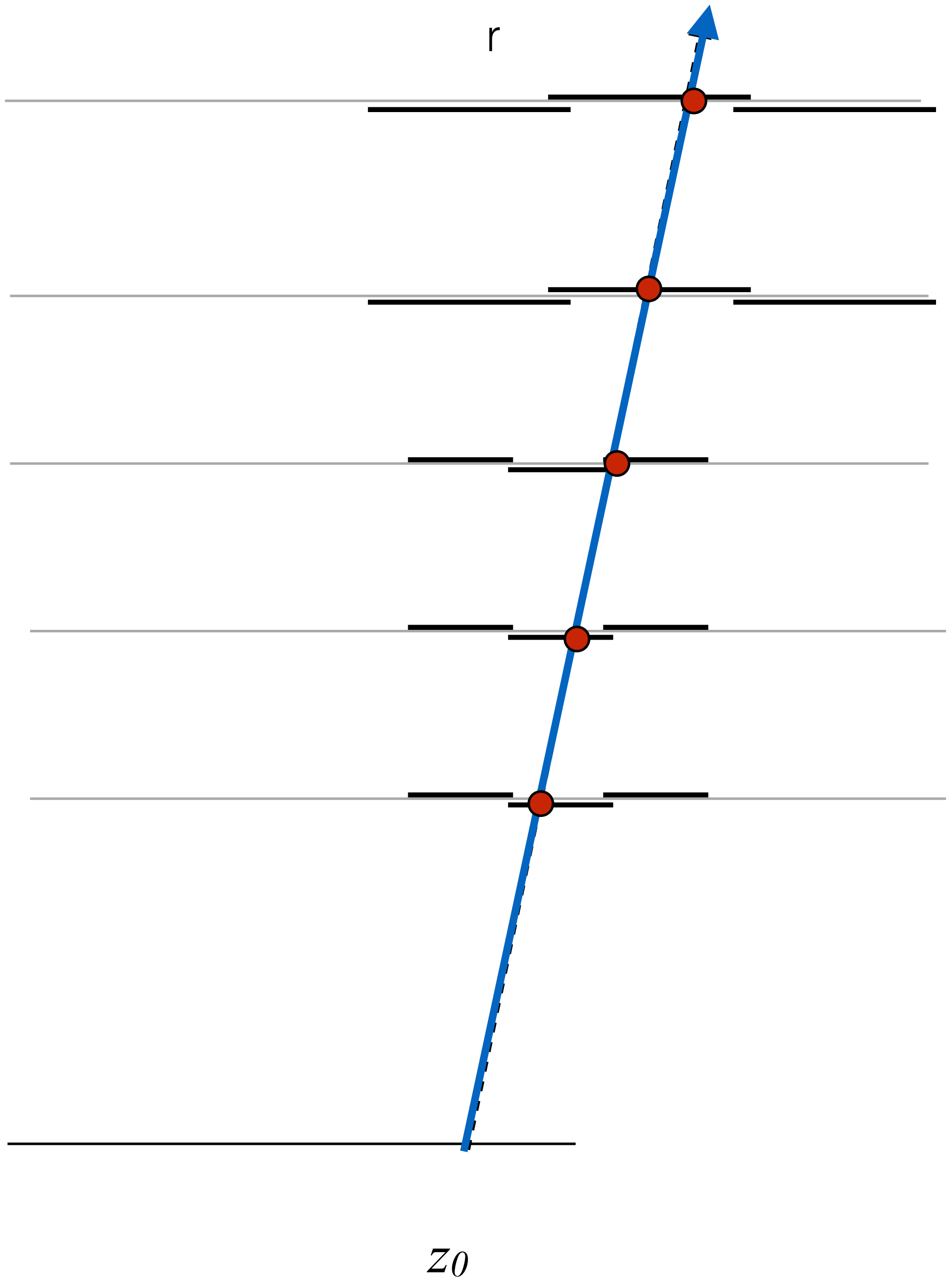
- What do large impact parameters mean ?
  - imagine some significant energy loss

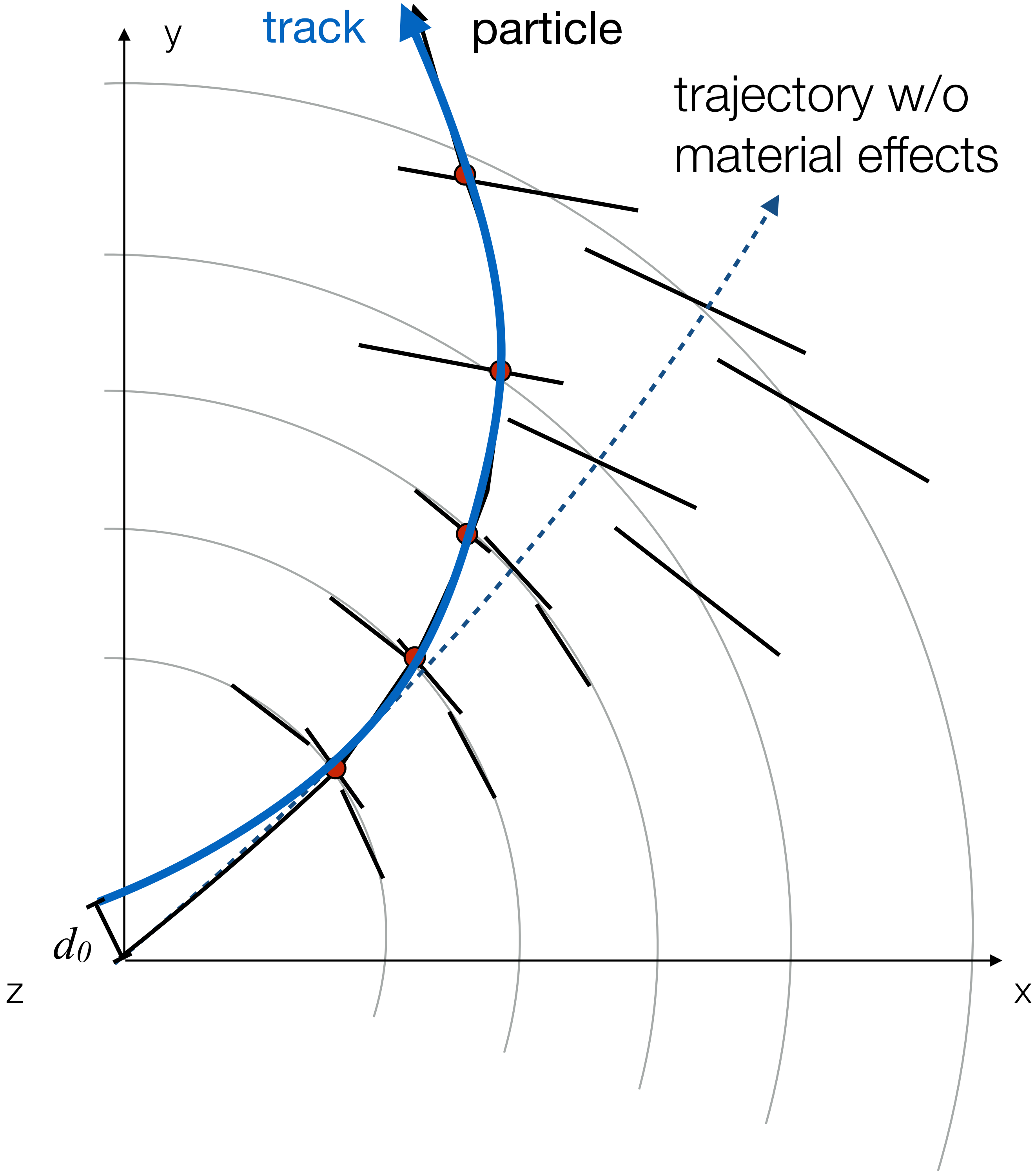




Understanding the track fit output: parameters

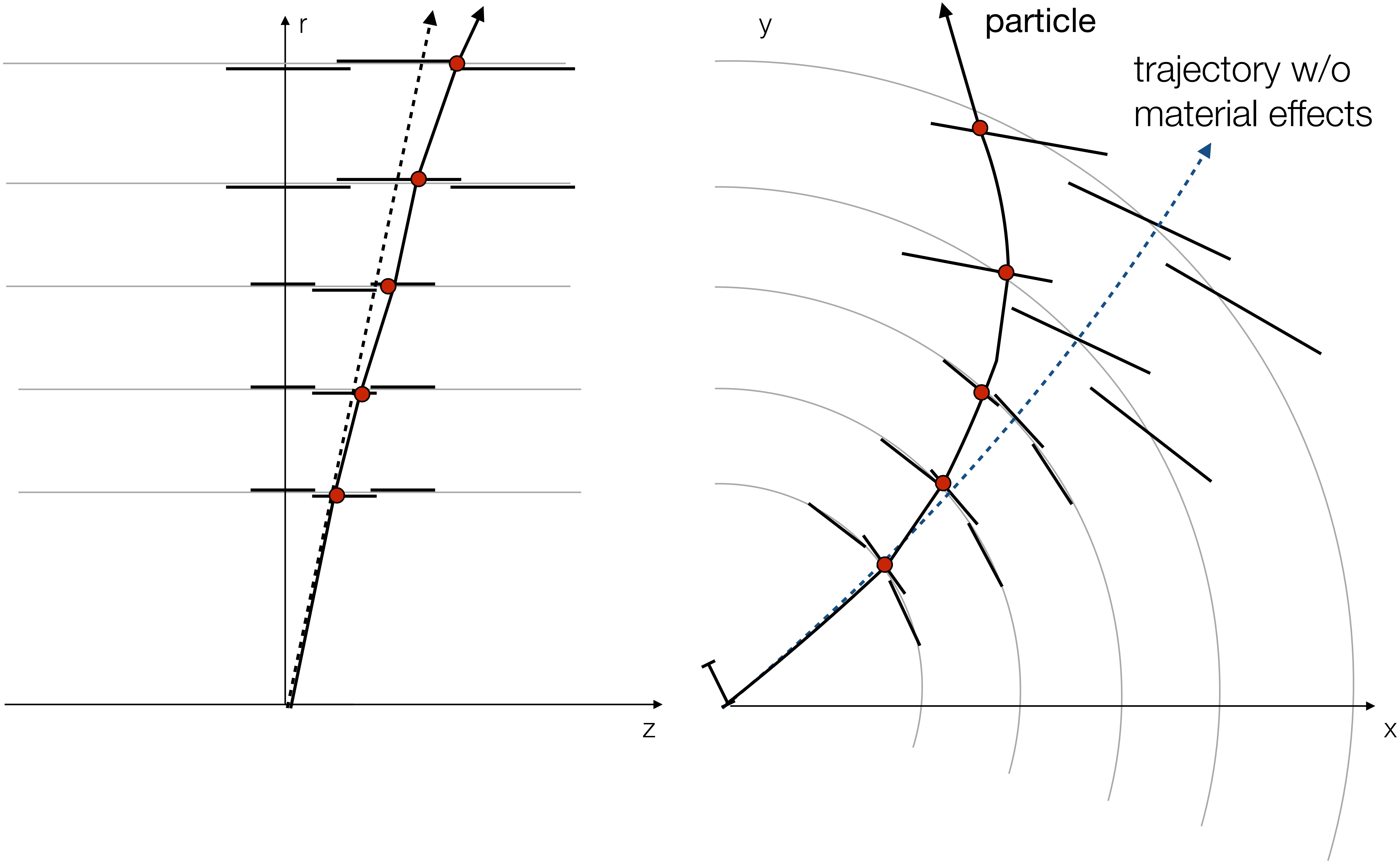
- What do large impact parameters mean ?
  - imagine some significant energy loss





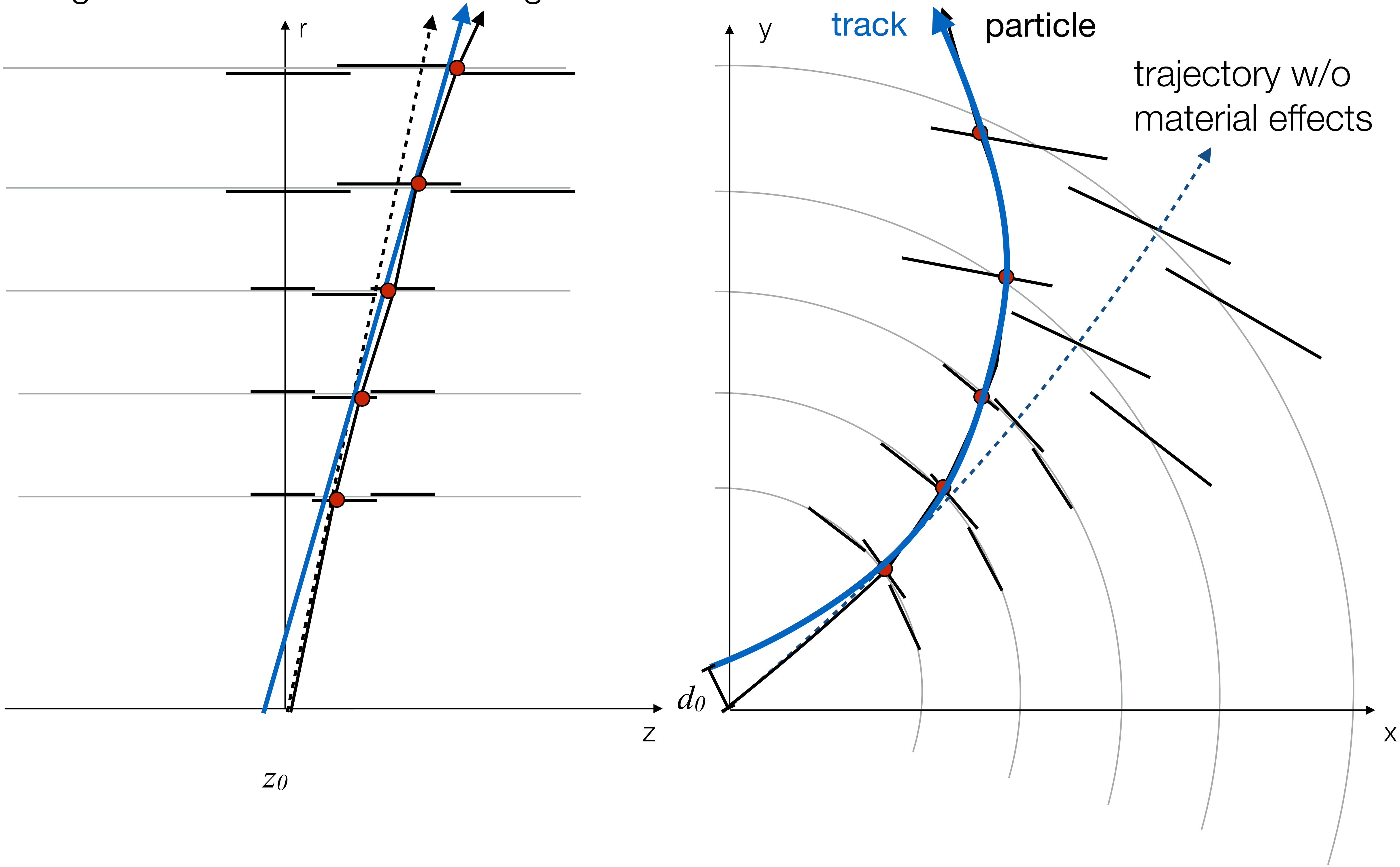
Understanding the track fit output: parameters

- What do large impact parameters mean ?
  - imagine an unfortunate scattering chain



Understanding the track fit output: parameters

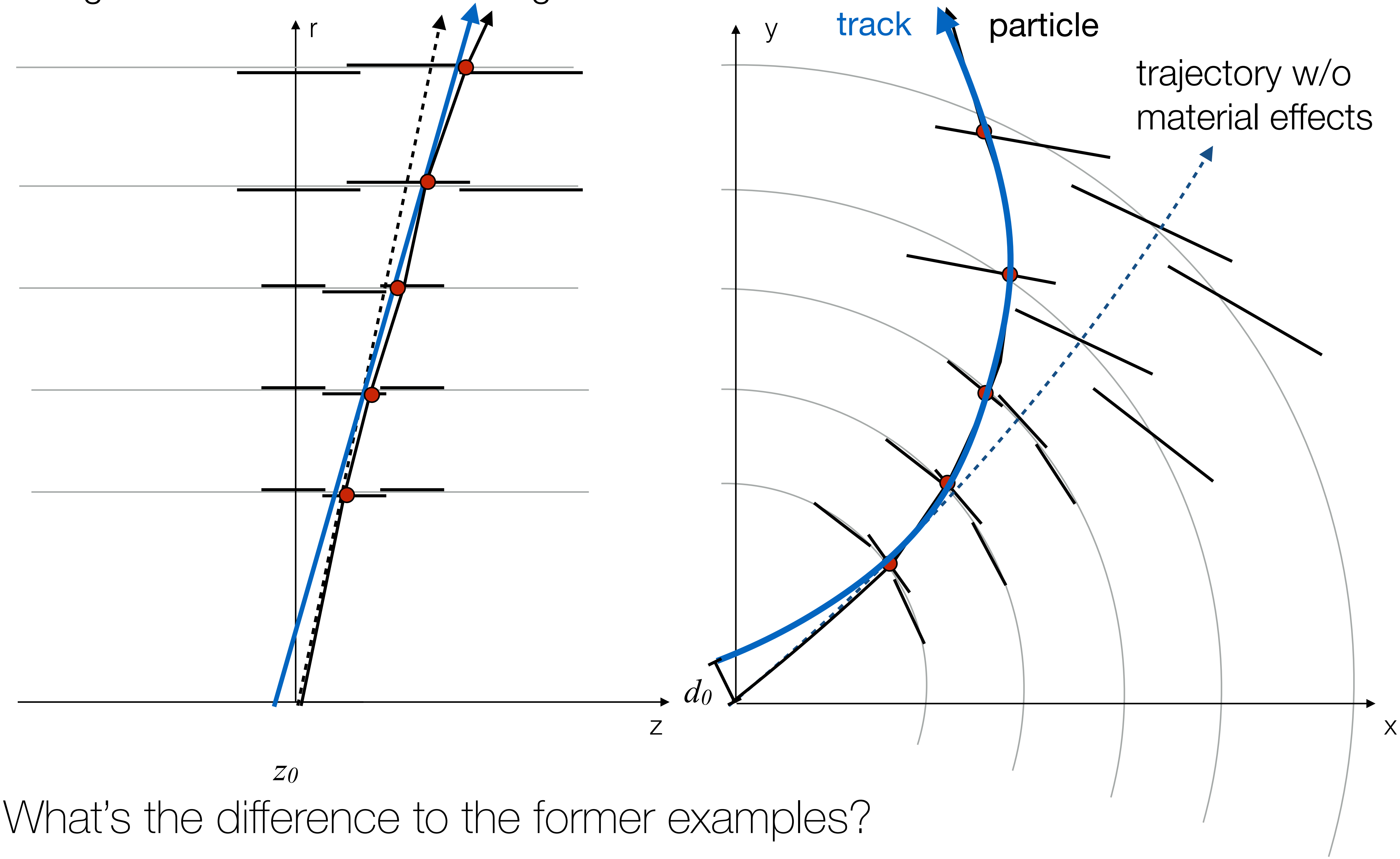
- What do large impact parameters mean ?
  - imagine an unfortunate scattering chain



INTERLUDE

Understanding the track fit output: parameters

- What do large impact parameters mean ?
  - imagine an unfortunate scattering chain

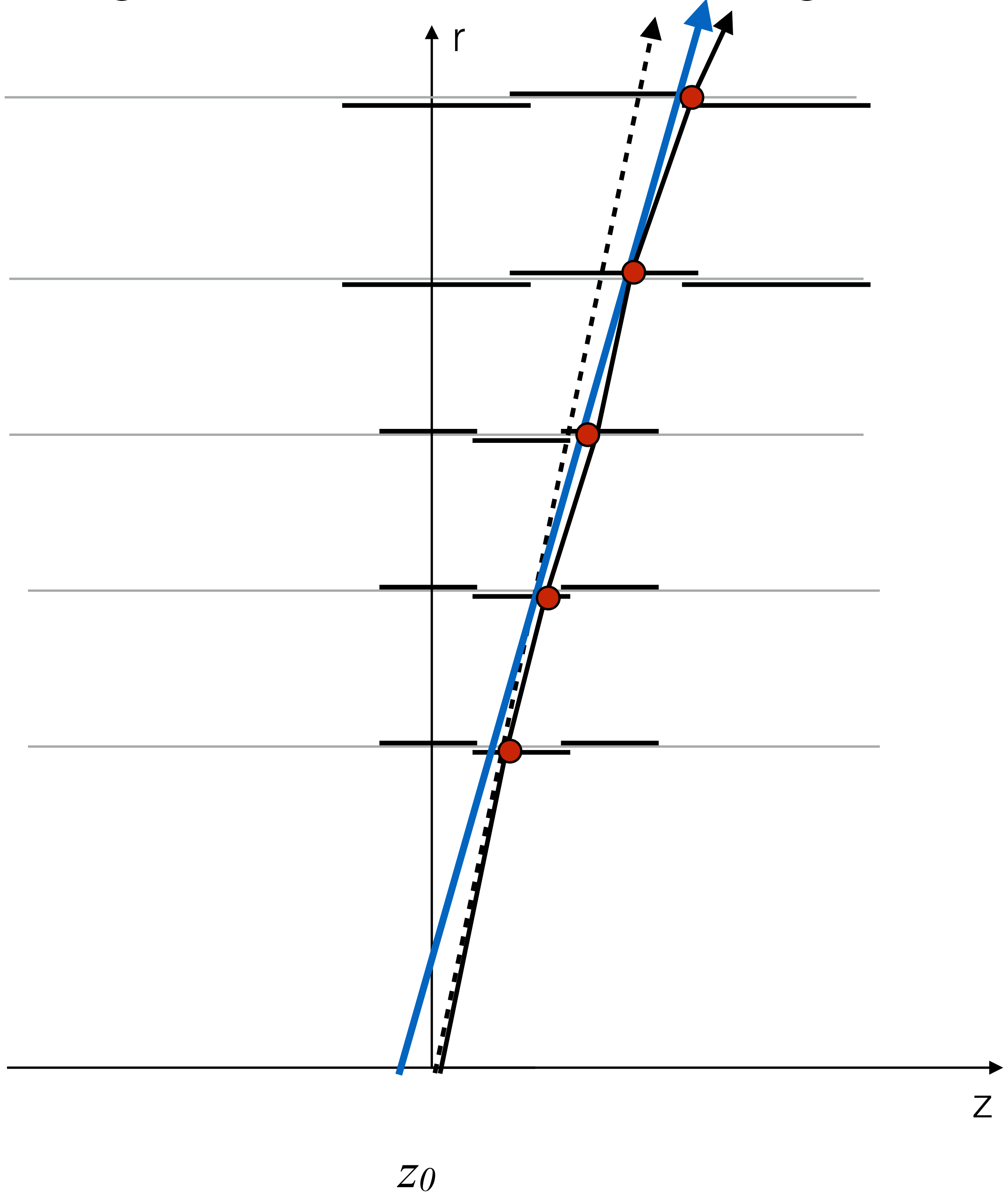


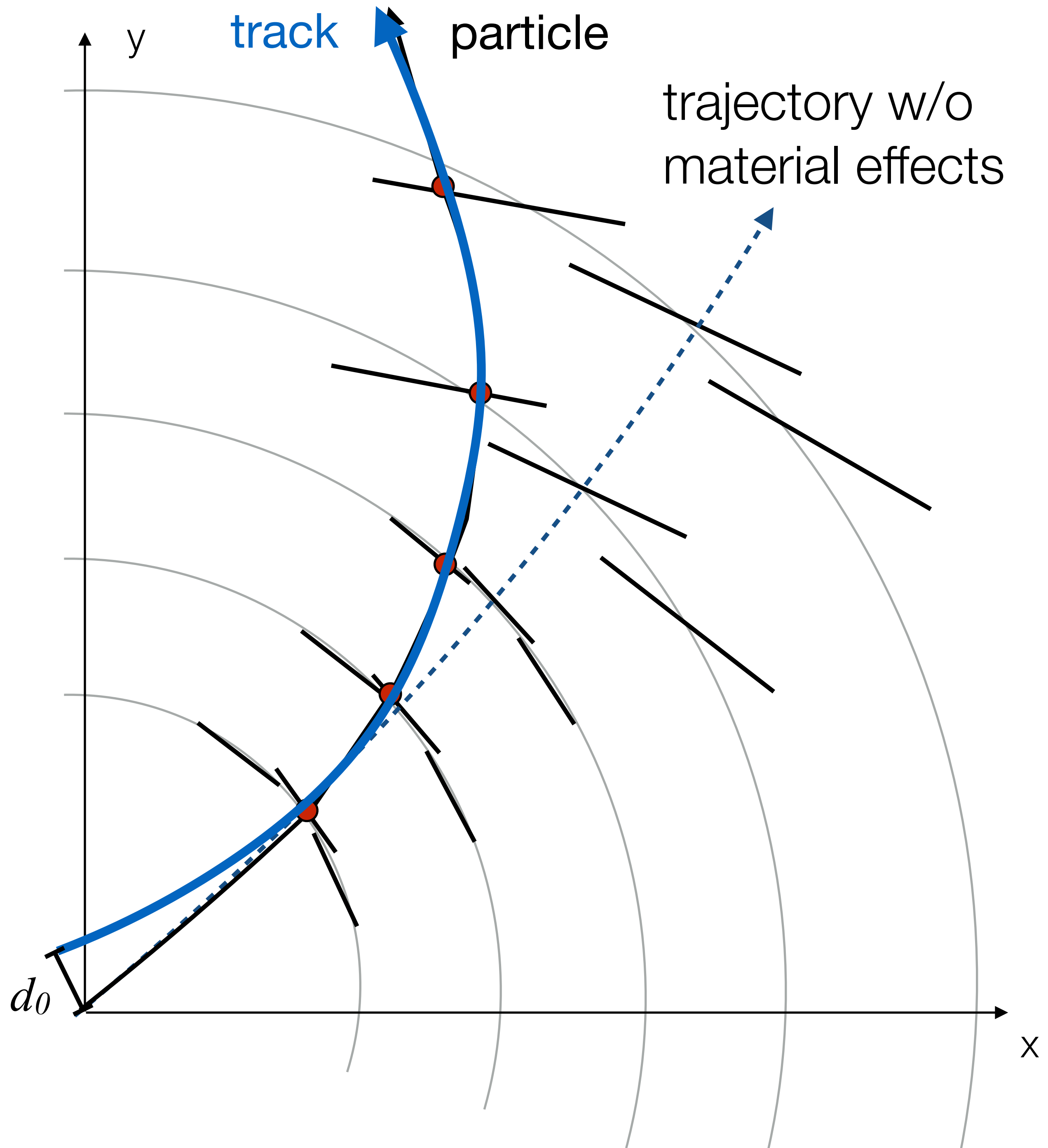
- What's the difference to the former examples?

A. Salzburger - Track and Vertex Reconstruction - Part 2 - HCPSS Aug 11-22, 2014

Understanding the track fit output: parameters

- What do large impact parameters mean ?
  - imagine an unfortunate scattering chain

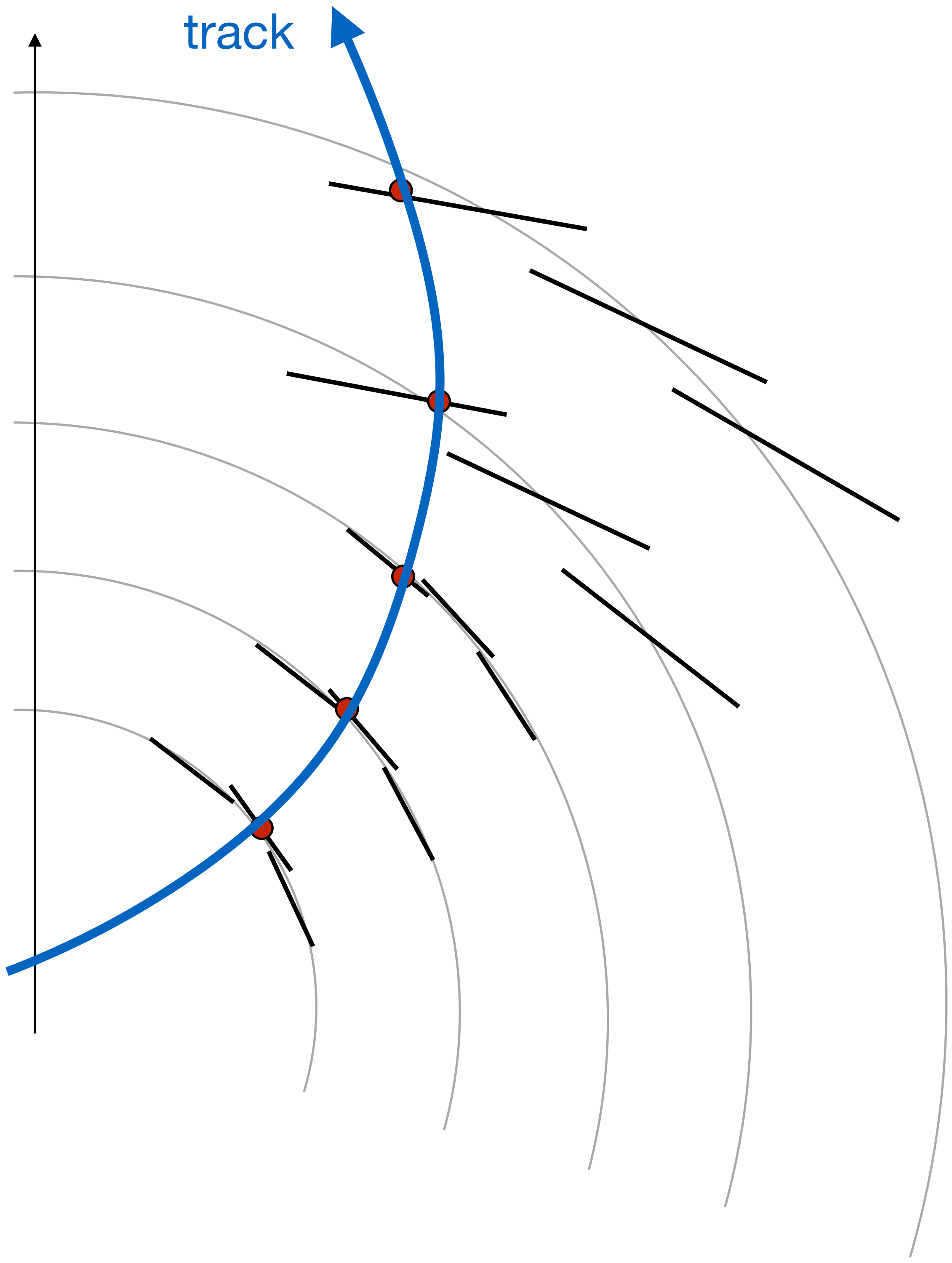




- What's the difference to the former examples? This is actually a large Δd_0, Δz_0

Understanding the track fit output: momentum

- Assume homogenous magnetic field B

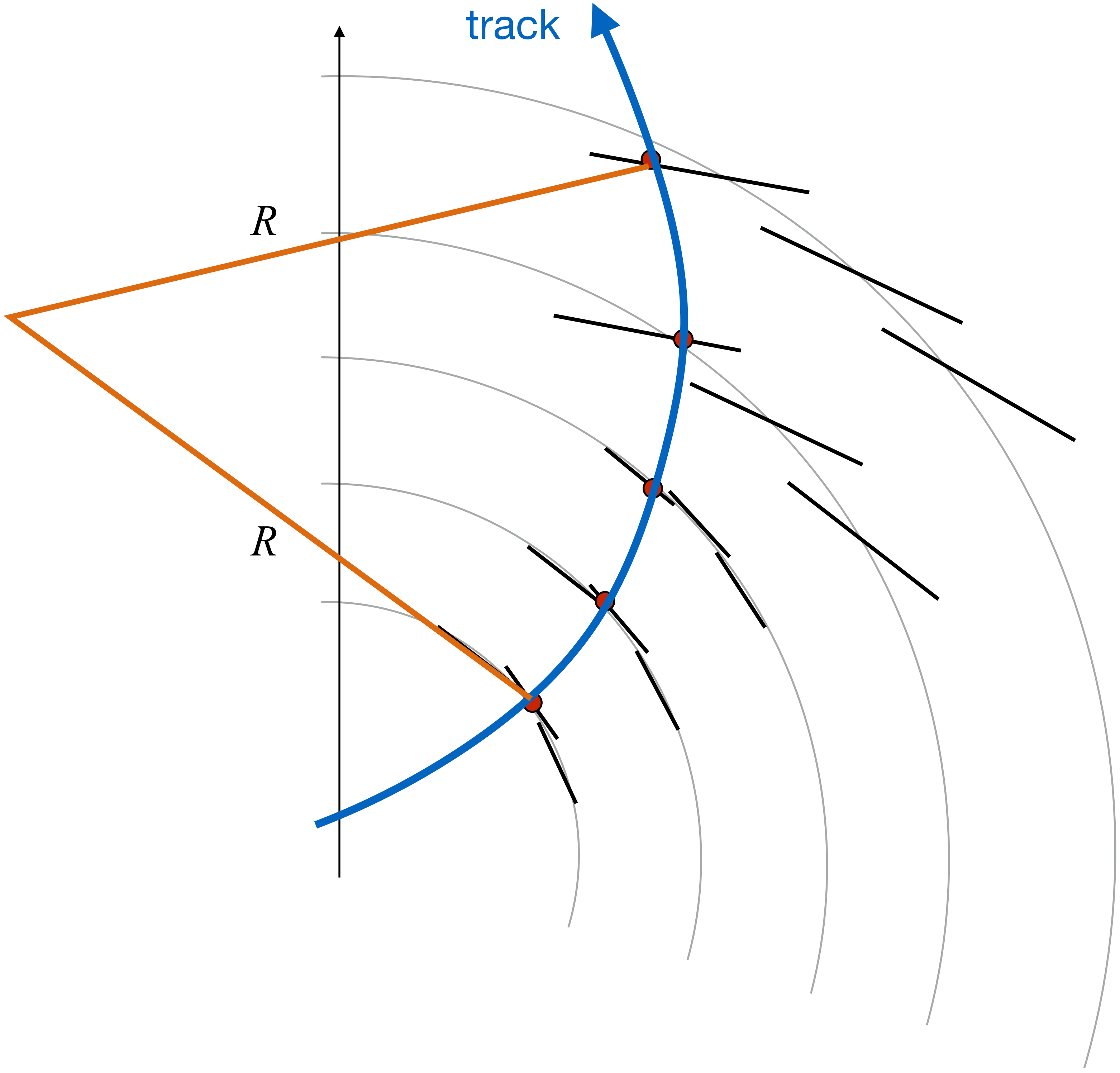


Understanding the track fit output: momentum

- Assume homogenous magnetic field B

$$\frac{d^2\mathbf{r}}{ds^2} = \frac{q}{p}\left[\frac{d\mathbf{r}}{ds} \times \mathbf{B}(\mathbf{r})\right]$$

$$\boxed{p_{\mathrm{T}} = \kappa B R}$$



Understanding the track fit output: momentum

- Assume homogenous magnetic field B

$$\frac{d^2\mathbf{r}}{ds^2} = \frac{q}{p}\left[\frac{d\mathbf{r}}{ds} \times \mathbf{B}(\mathbf{r})\right]$$

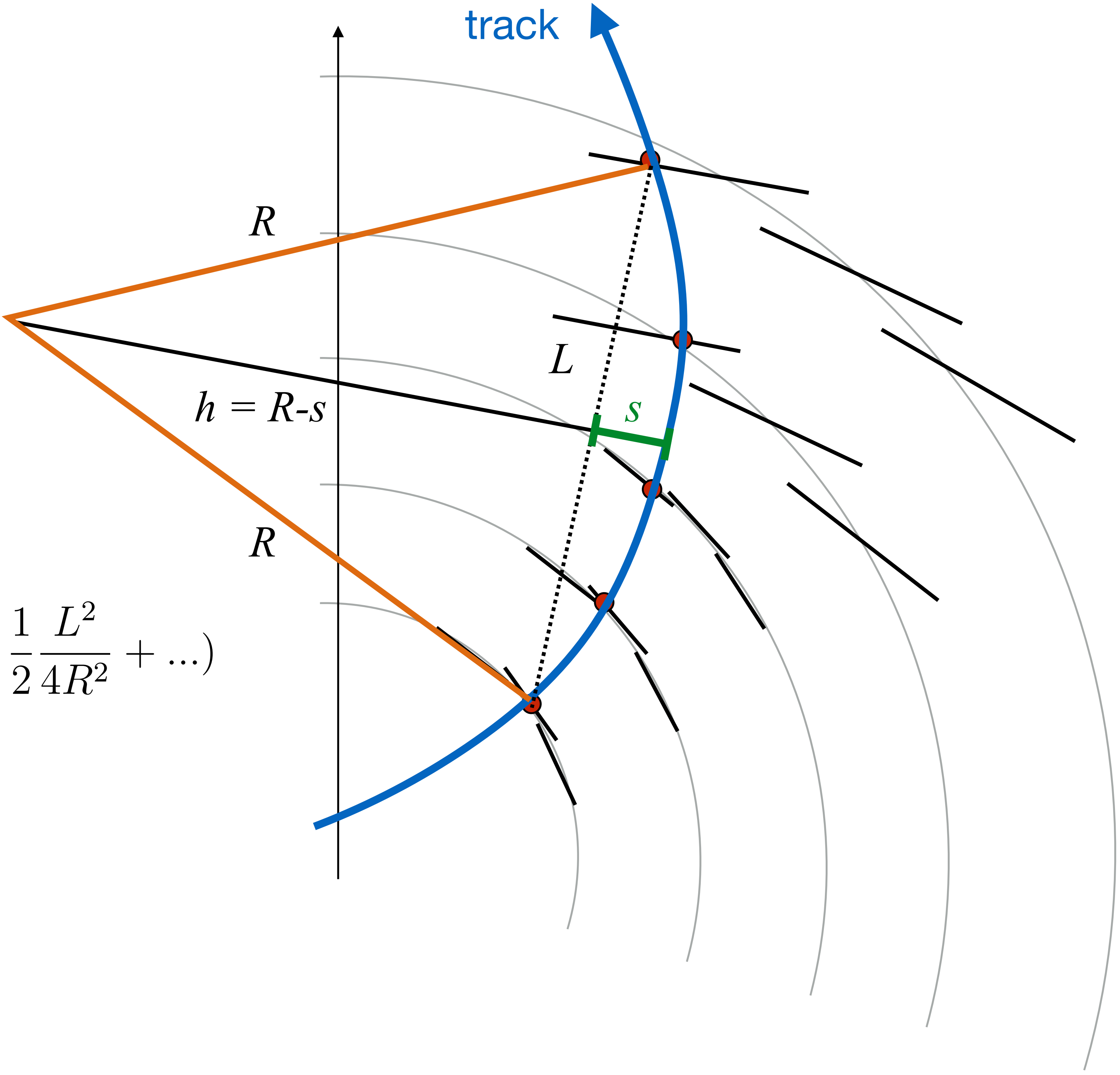
$$\boxed{p_{\mathrm{T}} = \kappa B R}$$

- transverse momentum measurement is a sagitta measurement

$$h^2 = R^2 - \left(\frac{L}{2}\right)^2 = R^2(1 - \frac{L^2}{4R^2})$$

$$h = R\left(1 - \frac{L^2}{4R^2})\right)^{-\frac{1}{2}} \approx R(1 - \frac{1}{2}\frac{L^2}{4R^2} + ...)$$

$$\boxed{s = R - h = \frac{L^2}{8R}}$$

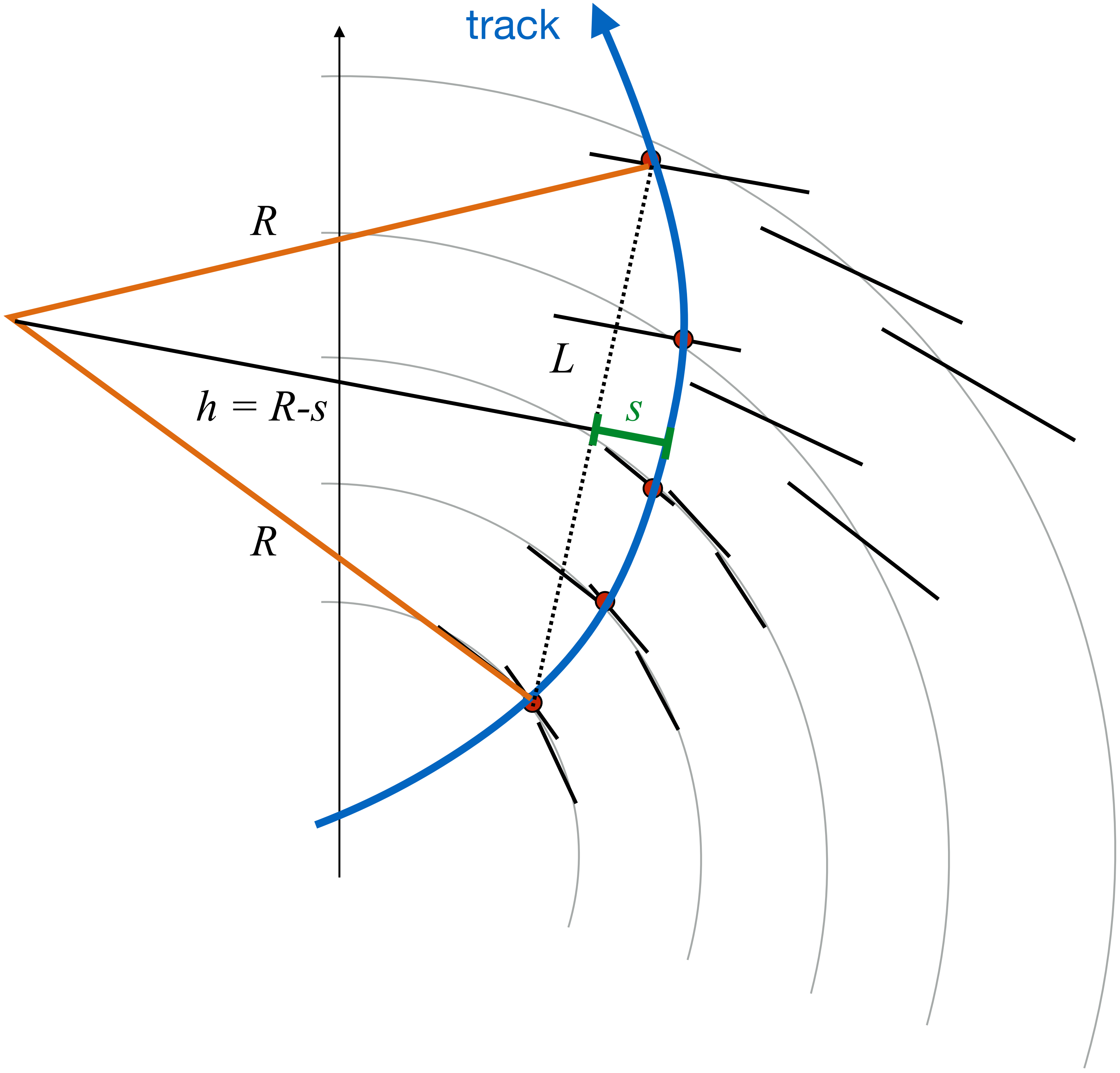


Understanding the track fit output: momentum

- Transverse momentum & sagitta

$$s = R - h = \frac{L^2}{8R}$$

$$p_{\mathrm{T}} = \kappa B R = \kappa B \frac{L^2}{8s}$$



Understanding the track fit output: momentum

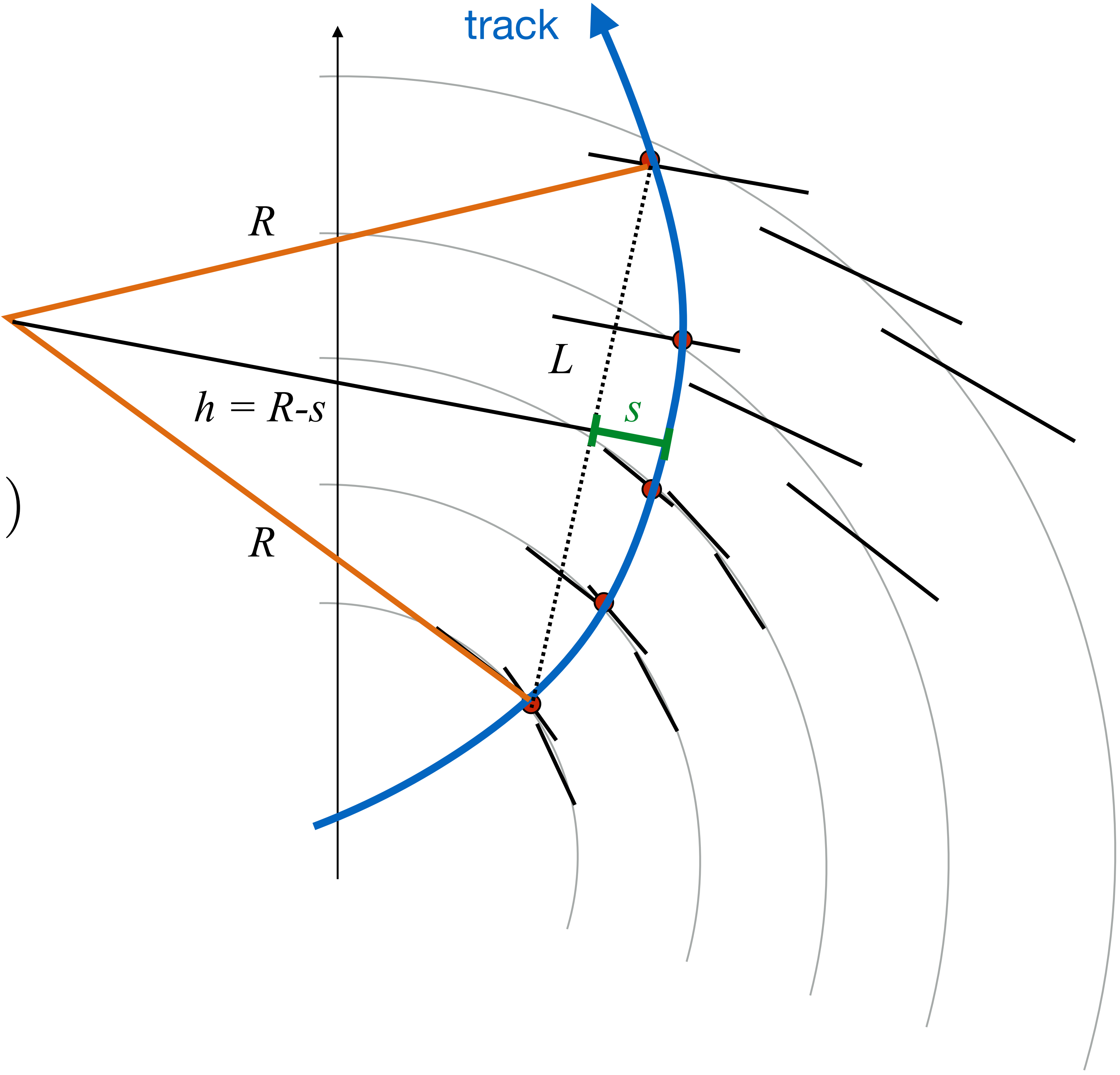
- Transverse momentum & sagitta

$$s = R - h = \frac{L^2}{8R}$$

$$p_{\mathrm{T}} = \kappa B R = \kappa B \frac{L^2}{8s}$$

- Yields measurement uncertainty on p_{T}

$$\frac{\sigma(p_{\mathrm{T}})}{p_{\mathrm{T}}} = \frac{8 p_{\mathrm{T}}}{\kappa B L^2} \sigma(s)$$



Understanding the track fit output: momentum

- Transverse momentum & sagitta

$$s = R - h = \frac{L^2}{8R}$$

$$p_{\mathrm{T}} = \kappa B R = \kappa B \frac{L^2}{8s}$$

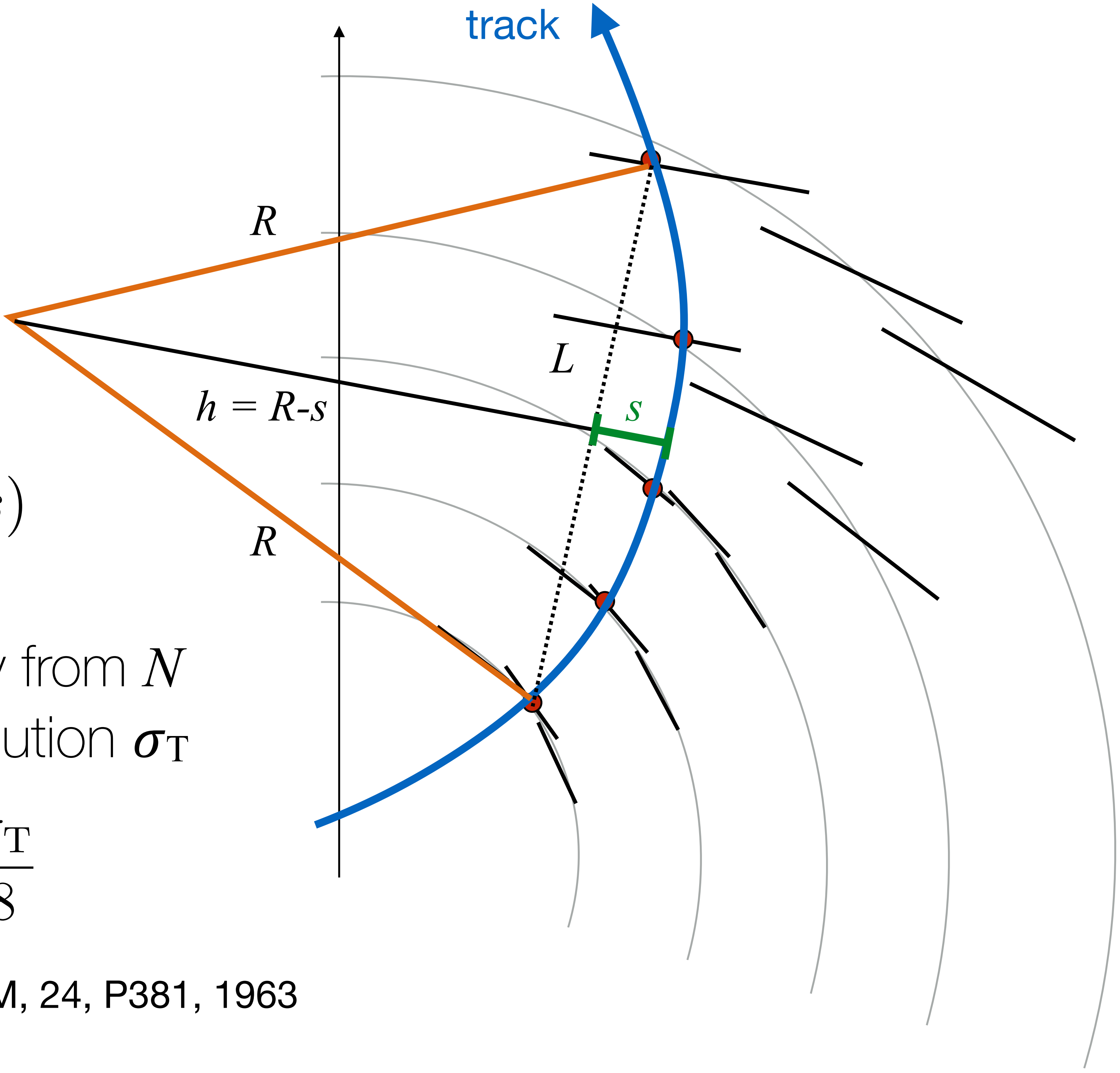
- Yields measurement uncertainty on p_{T}

$$\frac{\sigma(p_{\mathrm{T}})}{p_{\mathrm{T}}} = \frac{8 p_{\mathrm{T}}}{\kappa B L^2} \sigma(s)$$

- With a sagitta uncertainty from N measurements with resolution $\boldsymbol{\sigma}_{\mathrm{T}}$

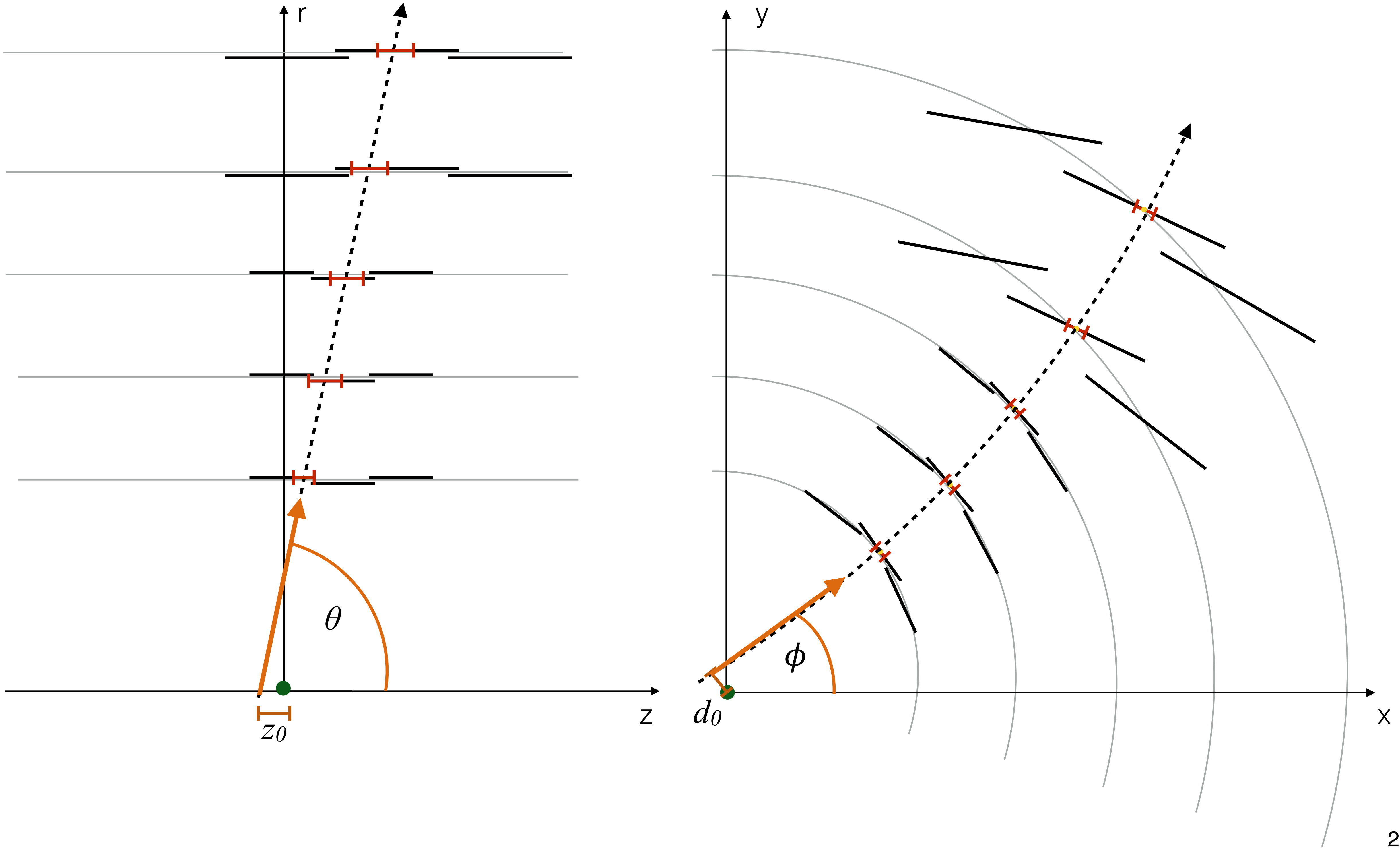
$$\sigma(s_N) = \sqrt{\frac{A_N}{N+4}} \frac{\sigma_{\mathrm{T}}}{8}$$

with $A_N = 720$ (Gluckstern factor), NIM, 24, P381, 1963



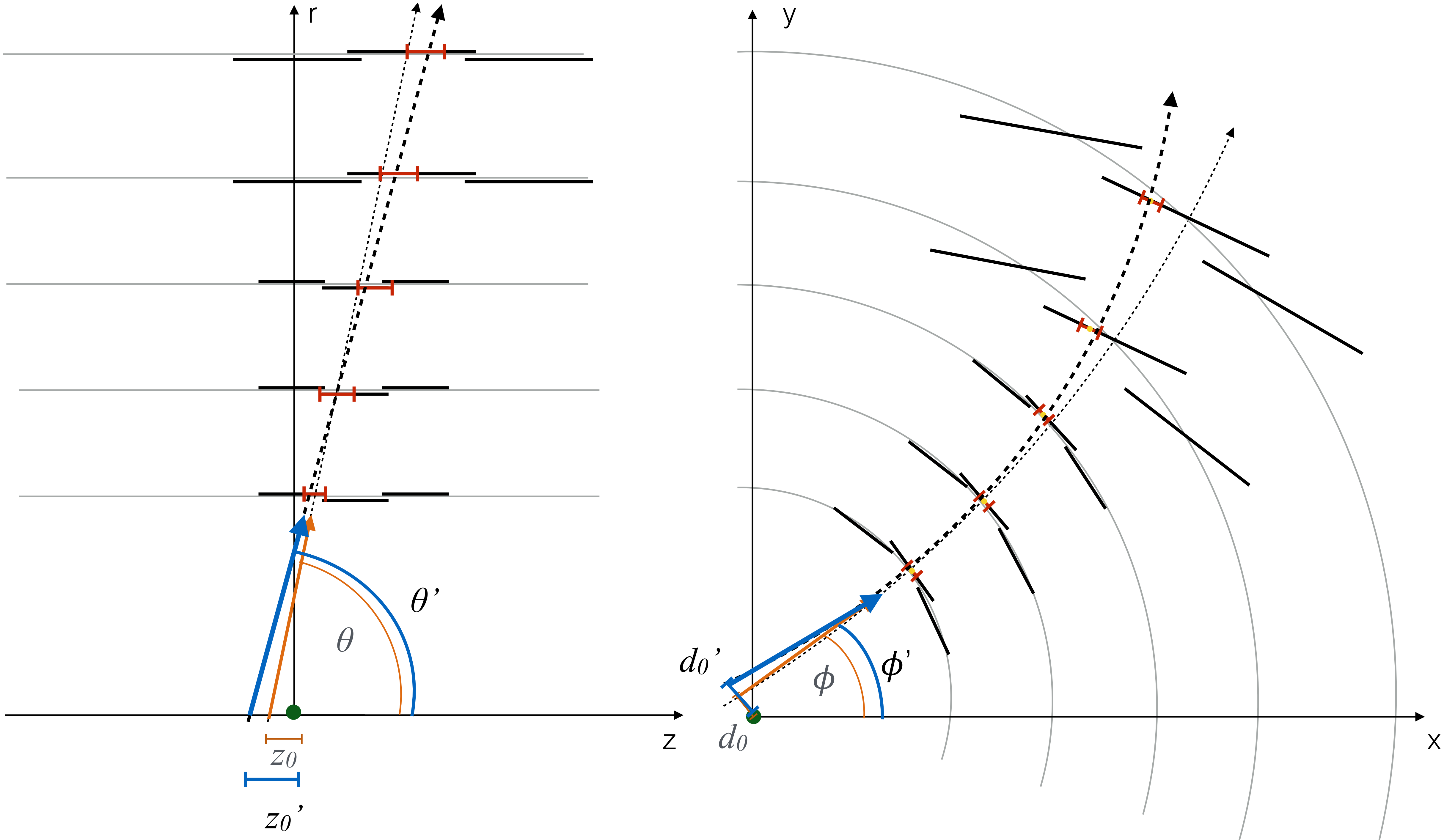
Understanding the track fit output: correlations

- Assuming a helical track model (solenoidal magnetic field)



Understanding the track fit output: correlations

- Assuming a helical track model (solenoidal magnetic field)



Understanding the track fit output: correlations

- What can we say about the correlations ?
  - d_0 correlates strongly with ϕ
  - z_0 correlates strongly with θ
  - q/p correlates with d_0 and ϕ

    via the transverse component p_T
  - q/p correlates with z_0 and θ

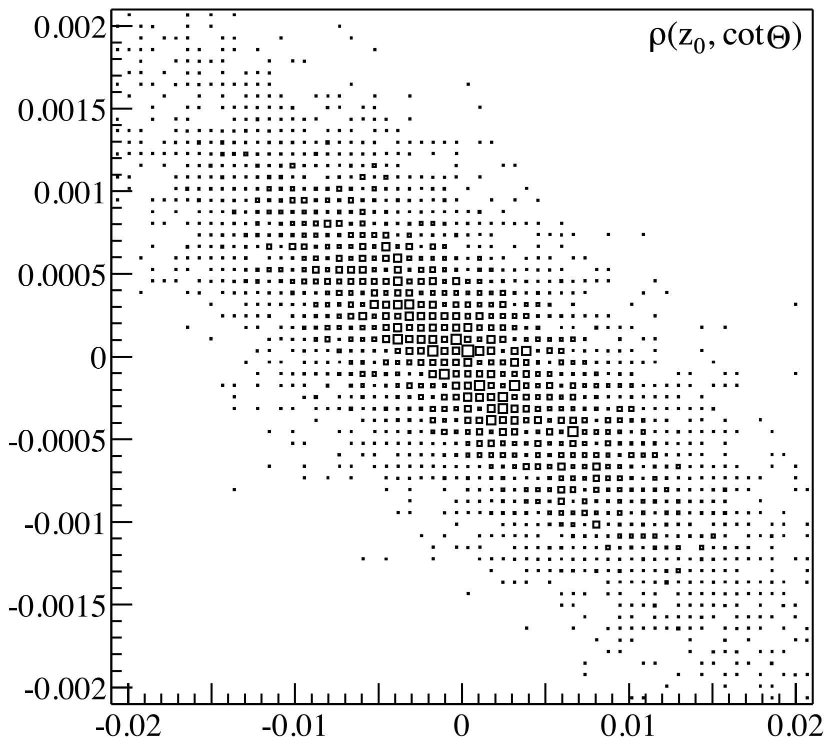
    via the longitudinal component p_L

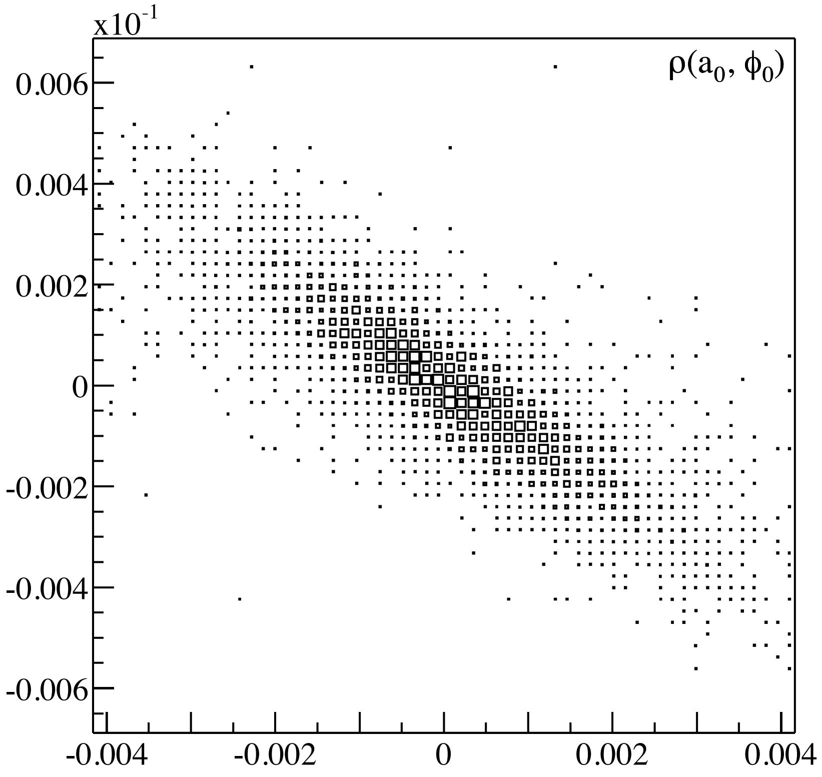
Understanding the track fit output: correlations

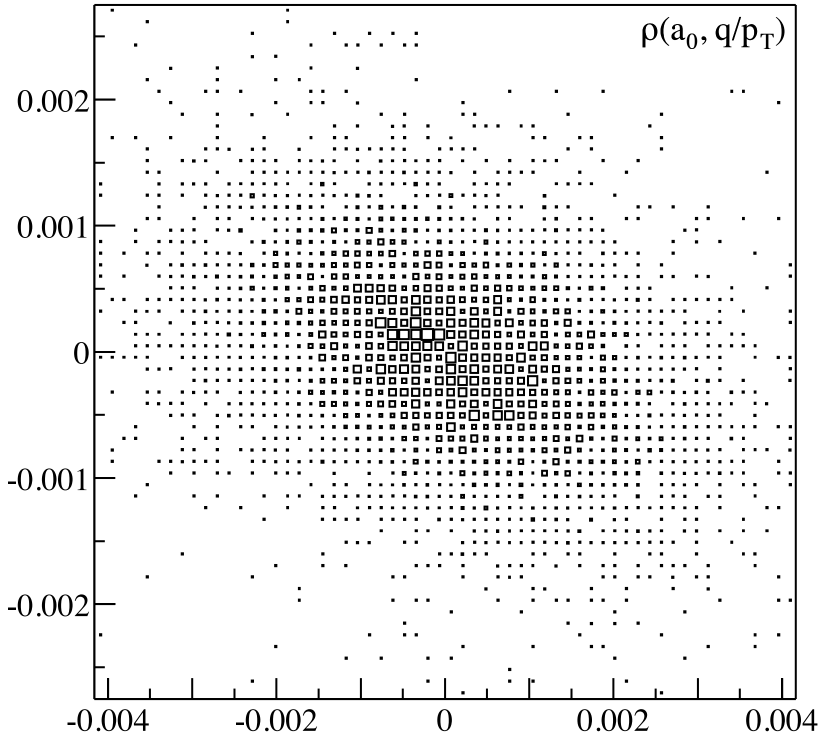
- What can we say about the correlations ?
  - d_0 correlates strongly with ϕ
  - z_0 correlates strongly with θ
  - q/p correlates with d_0 and ϕ

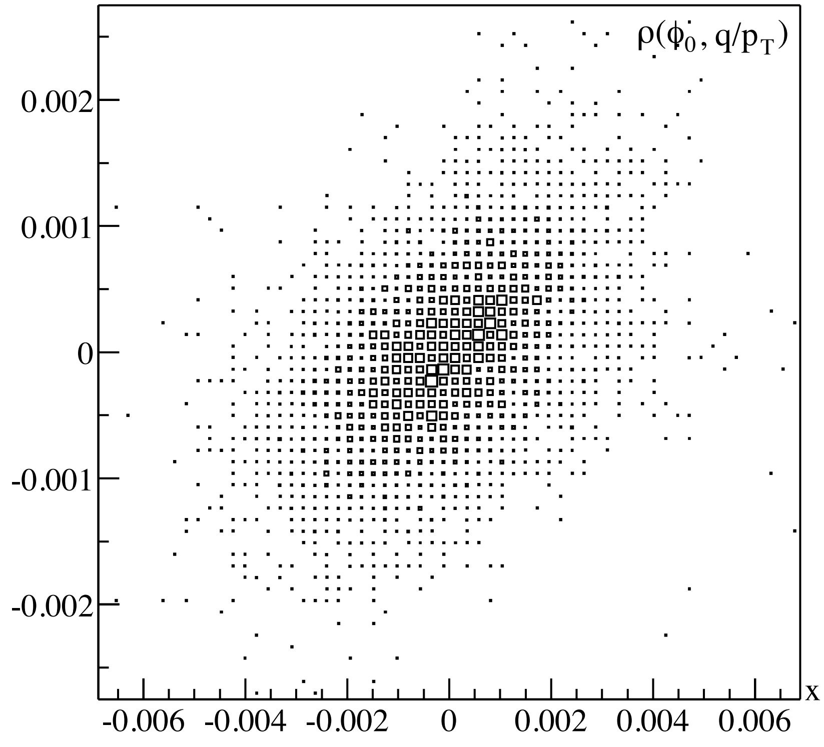
    via the transverse component p_T
  - q/p correlates with z_0 and θ

    via the longitudinal component p_L



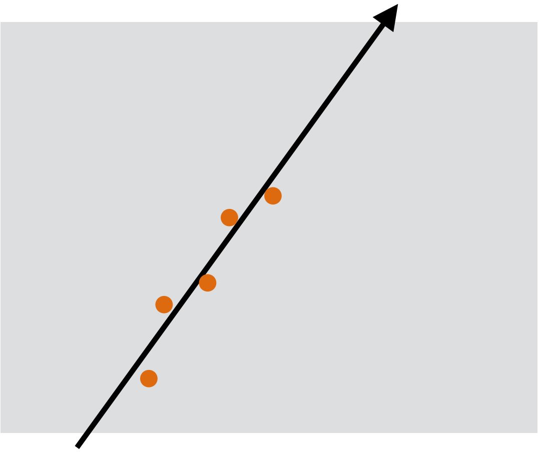






Highly non-gaussian systems

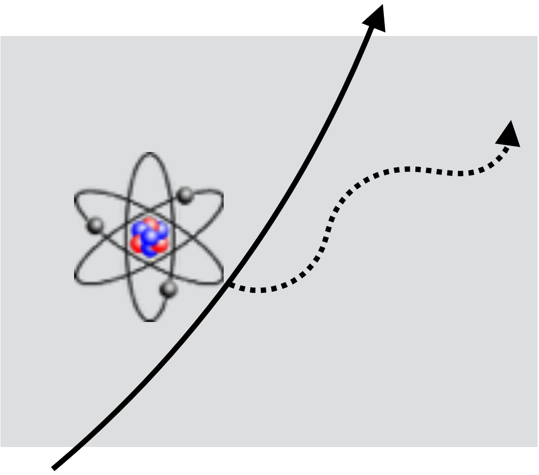
- Non-gaussian measurement errors can be kept under control
  - after all: we build the detector

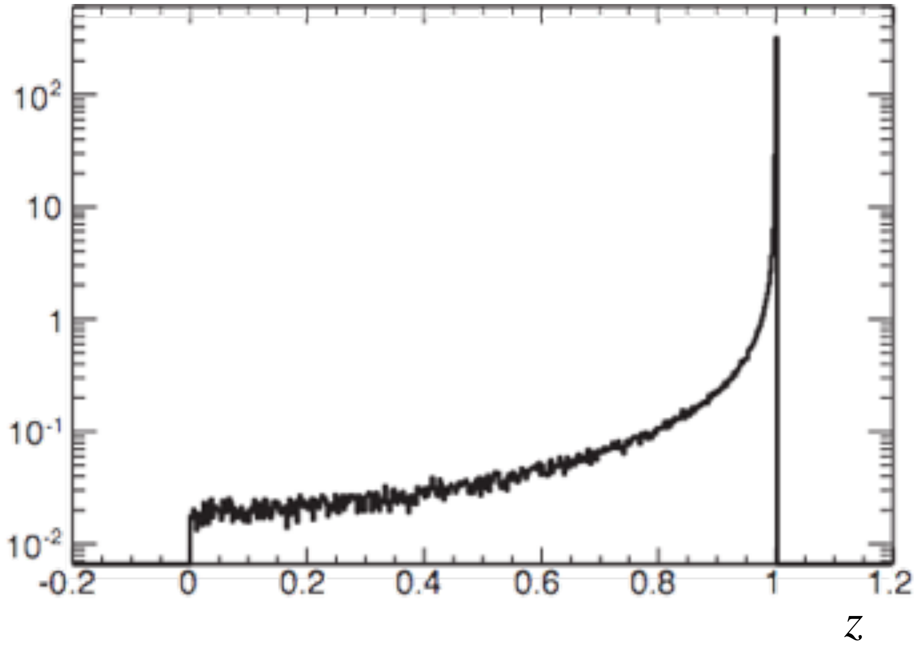


- Non-gaussian material effects are a real problem
  - multiple scattering has only small gaussian tails
  - energy loss is non-gaussian:

    -> ionization loss is Landau distributed, but fortunately $\Delta E << E$

    -> remember: bremsstrahlung is a dramatic effect

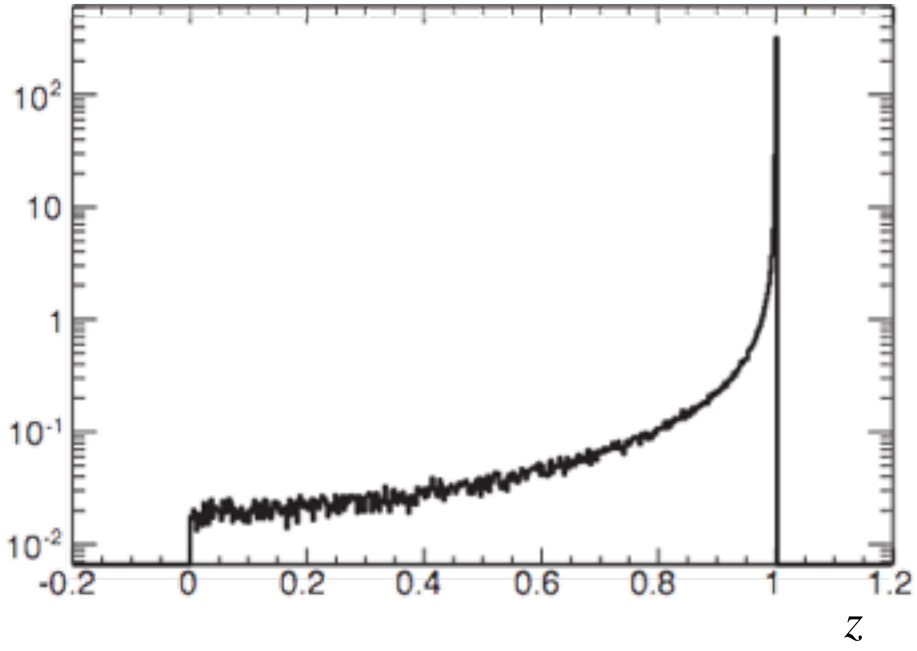


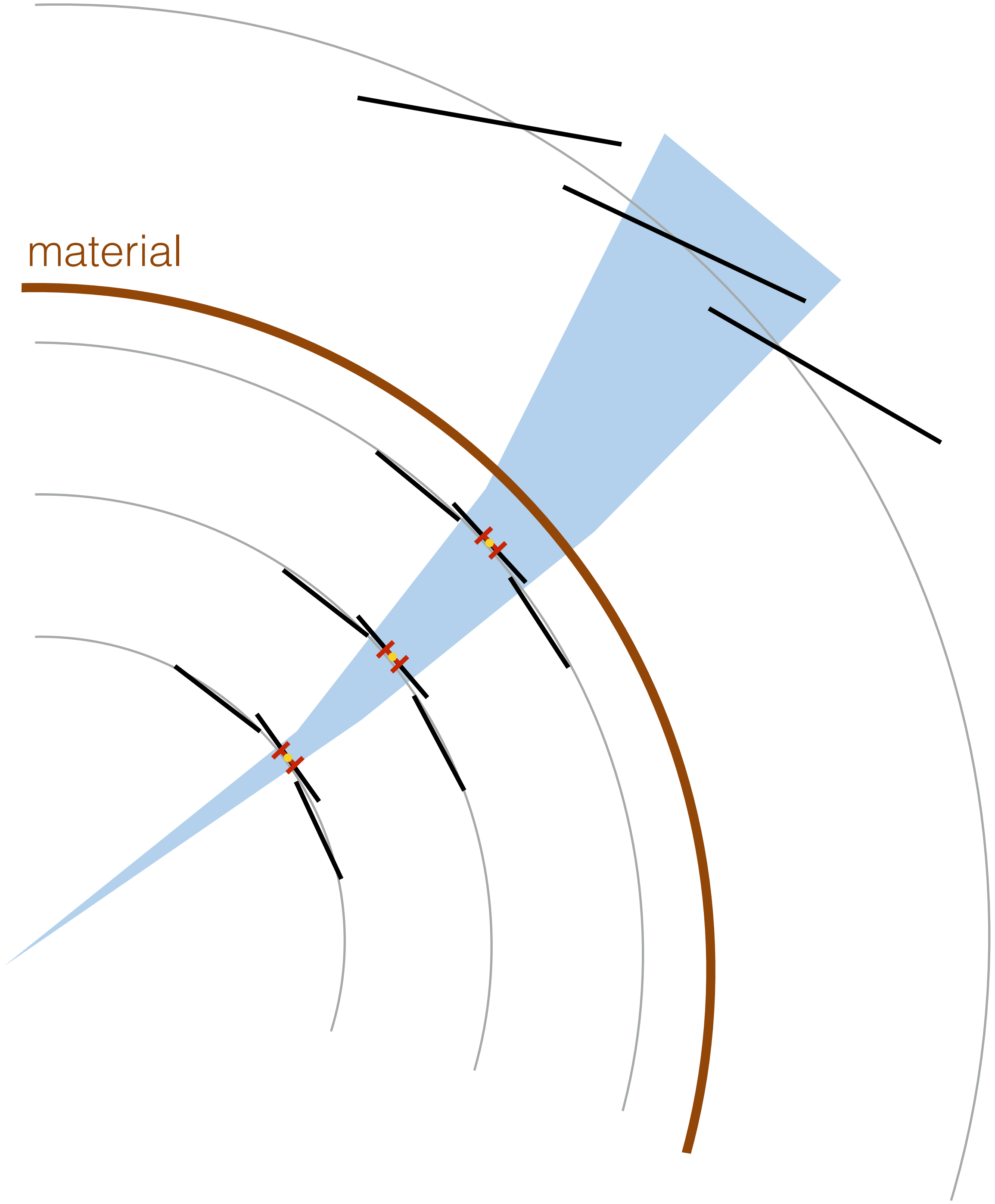


Highly non-gaussian systems

- Large energy loss leads to an effective deflection of the particle
  - dramatic change of curvature

- It is a stochastic effect
  - how to estimate a compatibility of a hit with the track model ?

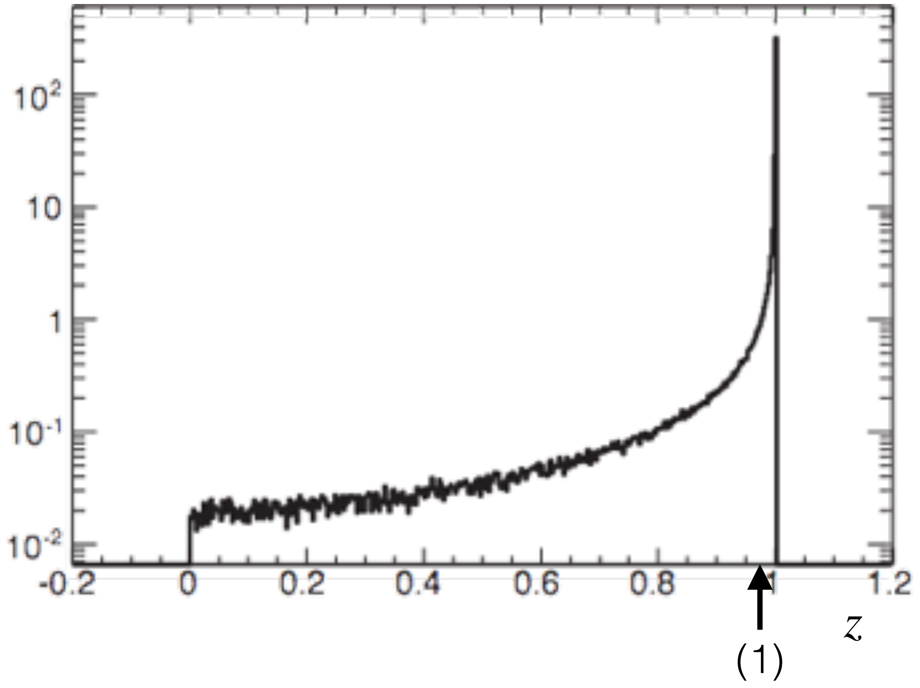


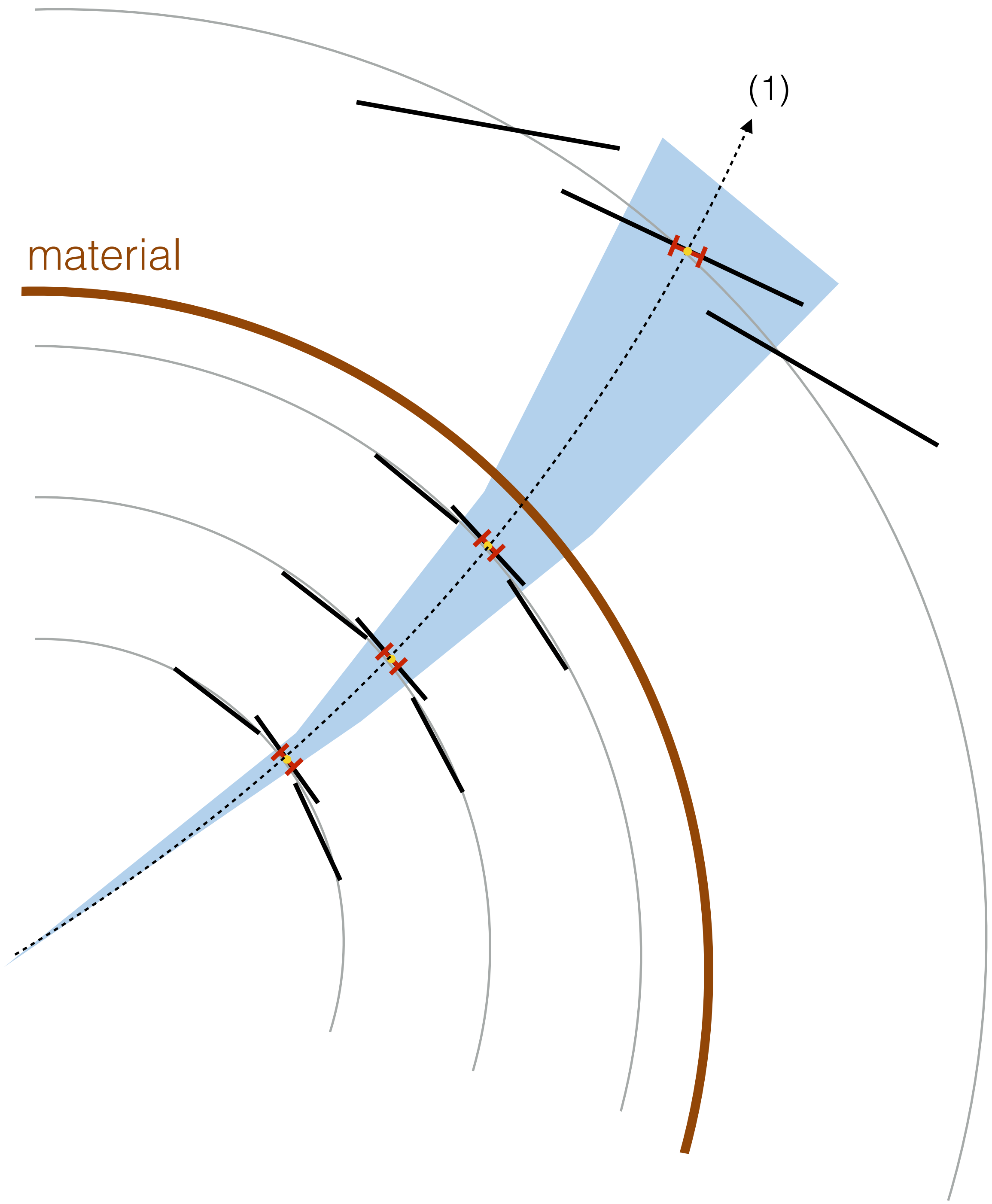


Highly non-gaussian systems

- Large energy loss leads to an effective deflection of the particle
  - dramatic change of curvature

- It is a stochastic effect
  - how to estimate a compatibility of a hit with the track model ?

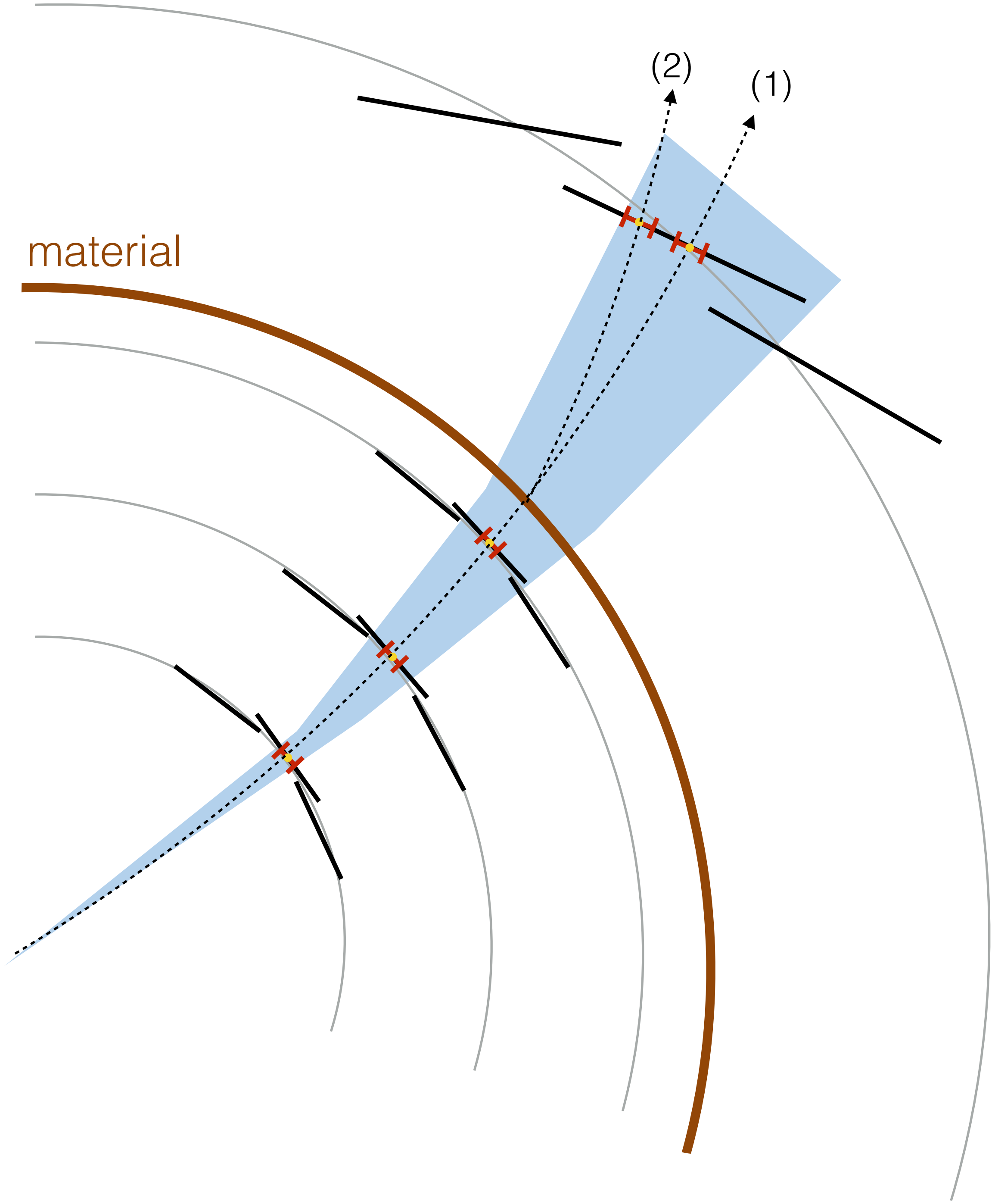


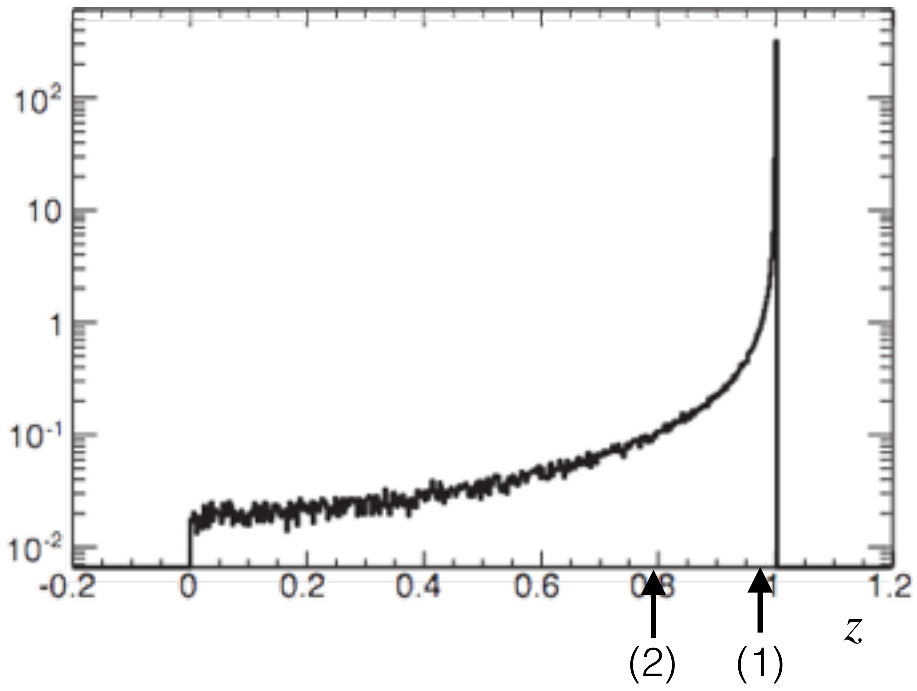


Highly non-gaussian systems

- Large energy loss leads to an effective deflection of the particle
  - dramatic change of curvature

- It is a stochastic effect
  - how to estimate a compatibility of a hit with the track model ?

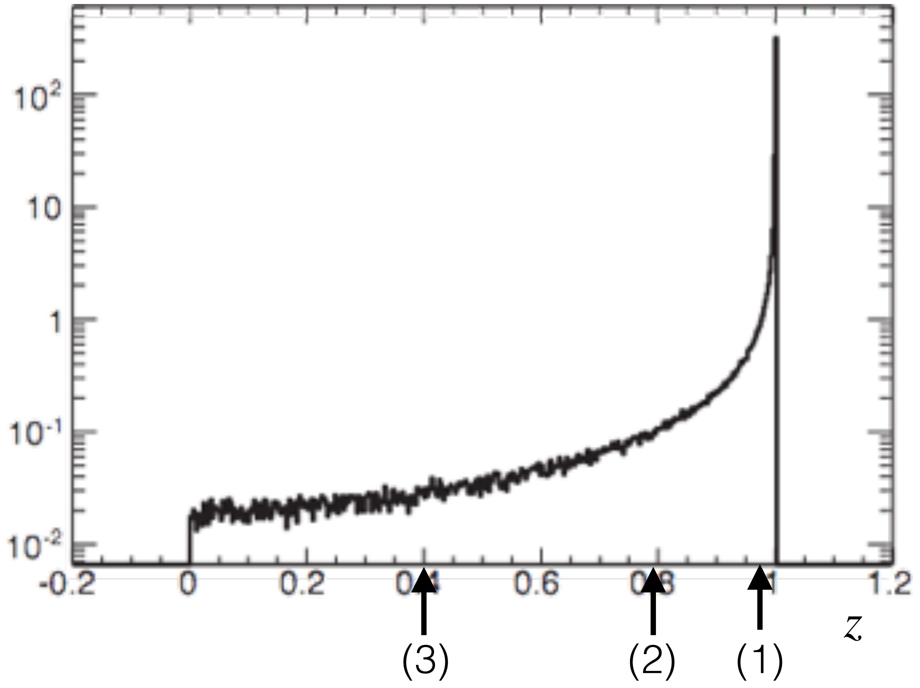


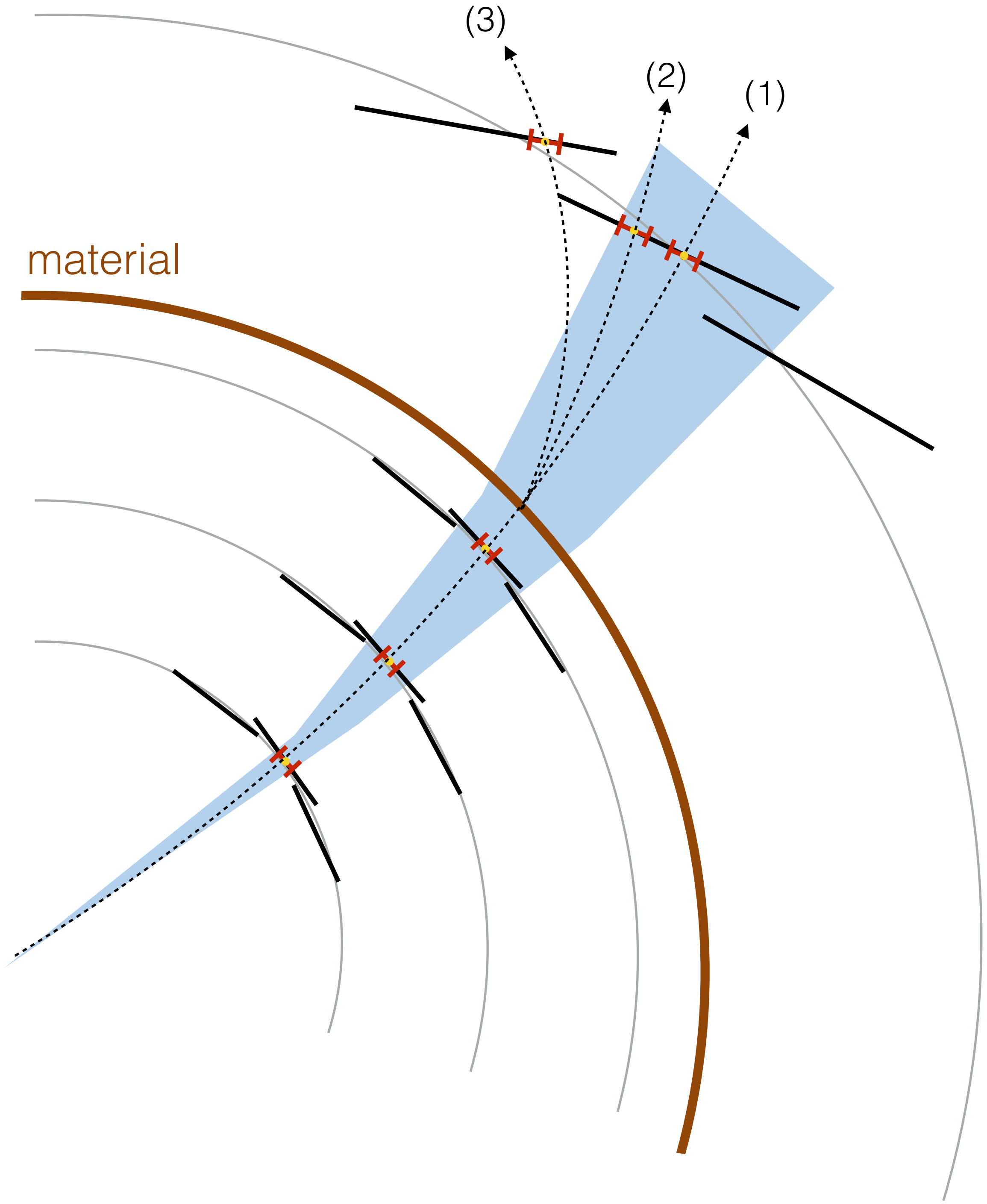


Highly non-gaussian systems

- Large energy loss leads to an effective deflection of the particle
  - dramatic change of curvature

- It is a stochastic effect
  - how to estimate a compatibility of a hit with the track model ?

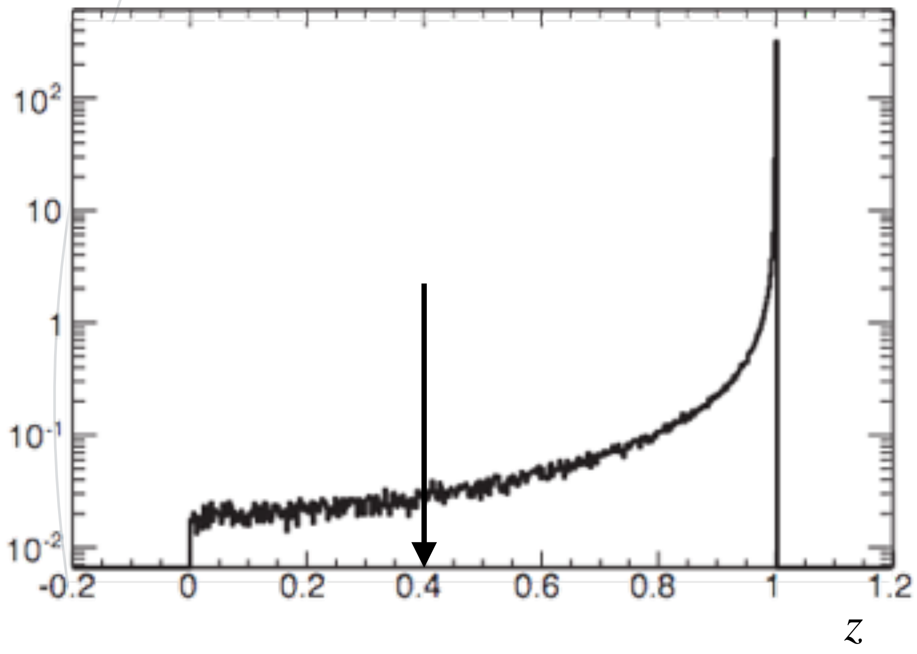


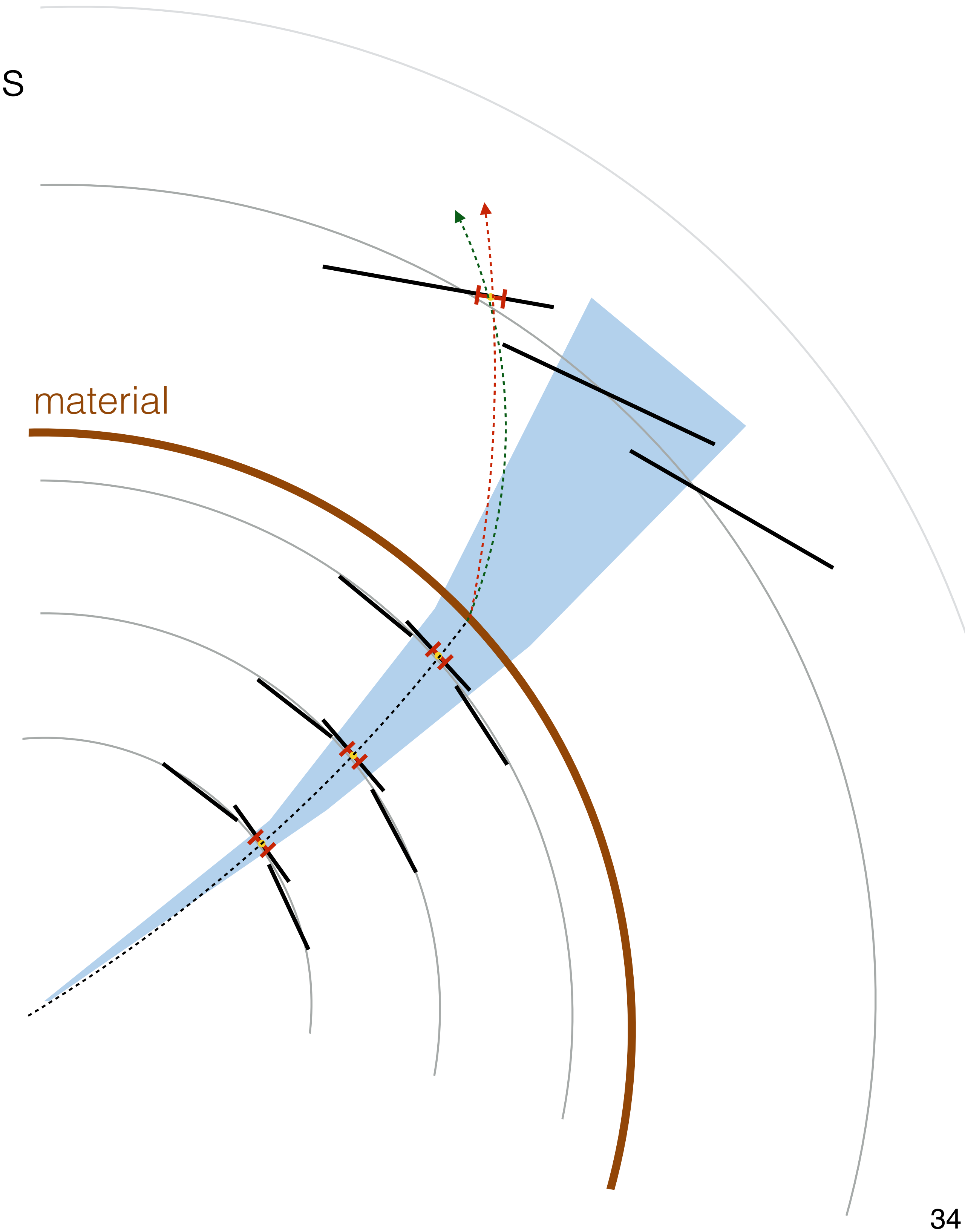


Highly non-gaussian systems

- Trying a naive global χ^2 fit
  - needs a large scattering angle / energy loss to compensate this change of curvature

    is it a change of curvature ?
    is it a deflection ?
    are hits from one particle ?



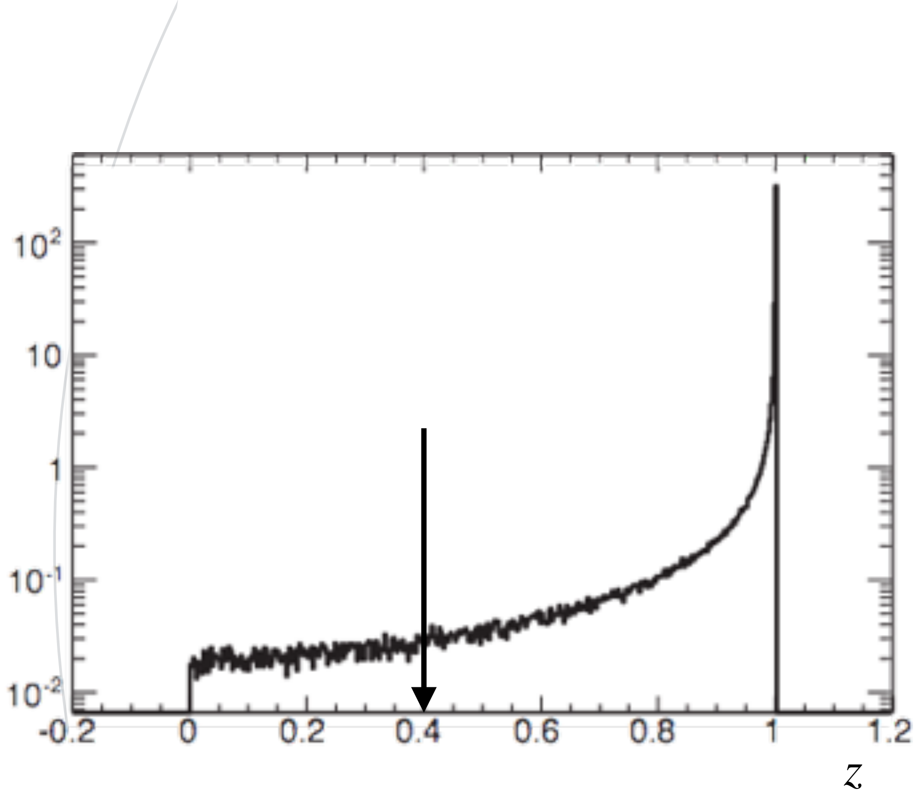


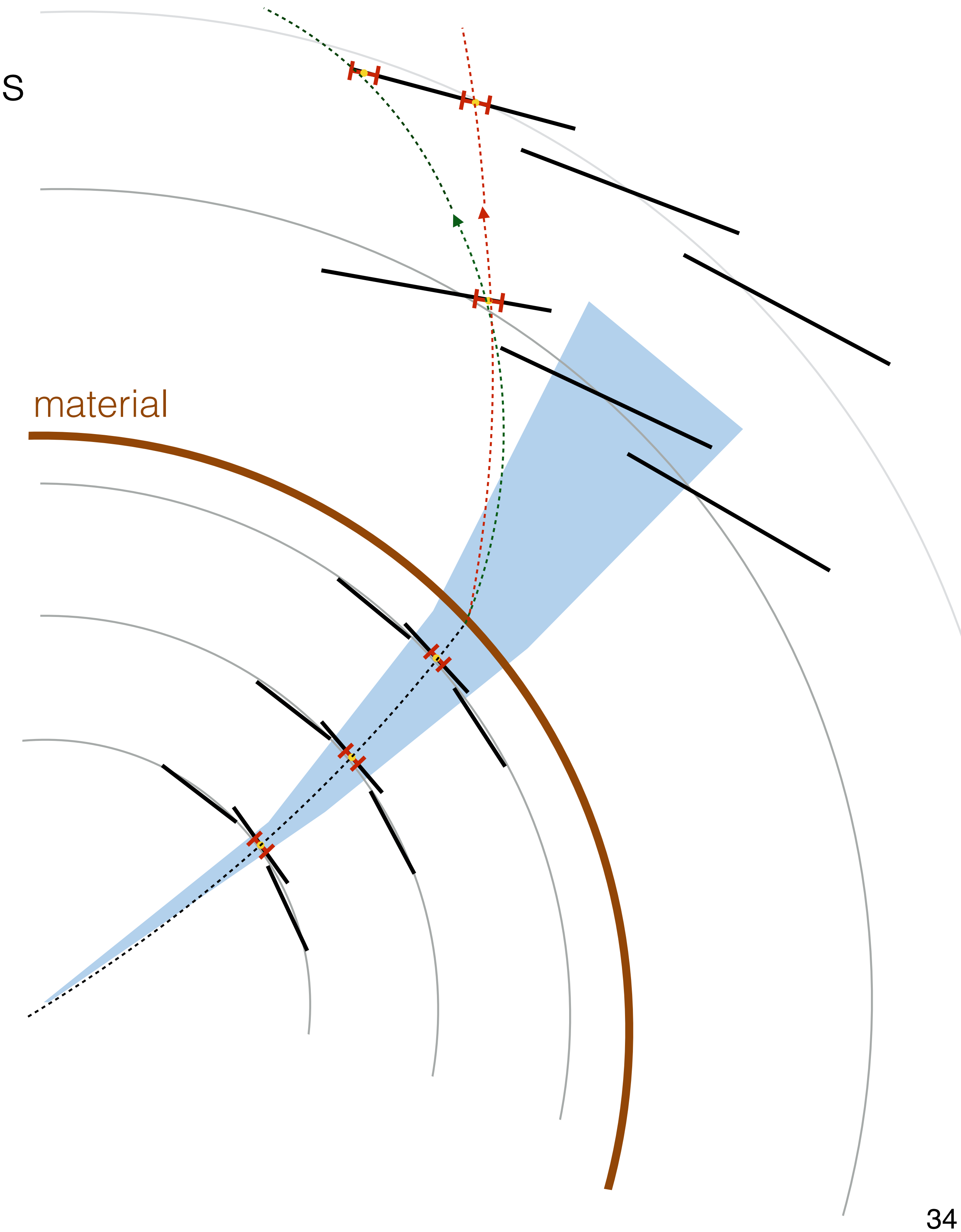
Highly non-gaussian systems

- Trying a naive global χ^2 fit
  - needs a large scattering angle / energy loss to compensate this change of curvature

is it a change of curvature ?
is it a deflection ?
are hits from one particle ?

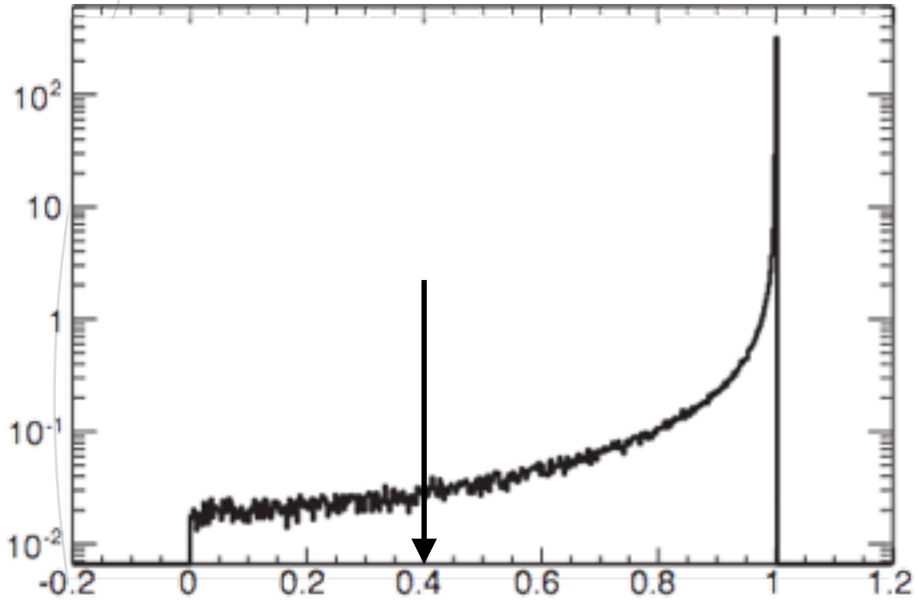
additional measurements help

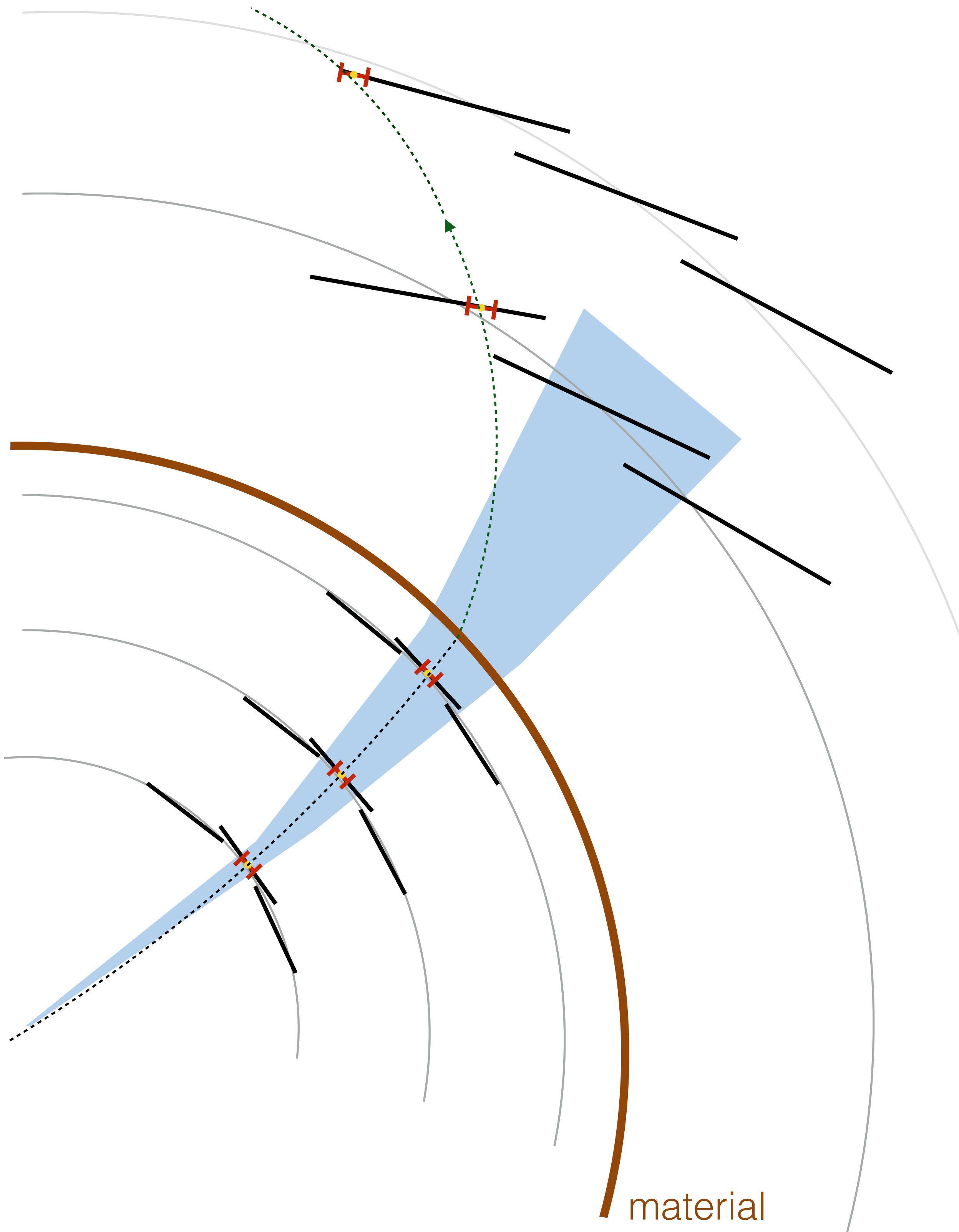




Highly non-gaussian systems

- Trying a naive global χ^2 fit
  - needs a large energy loss to compensate this change of curvature



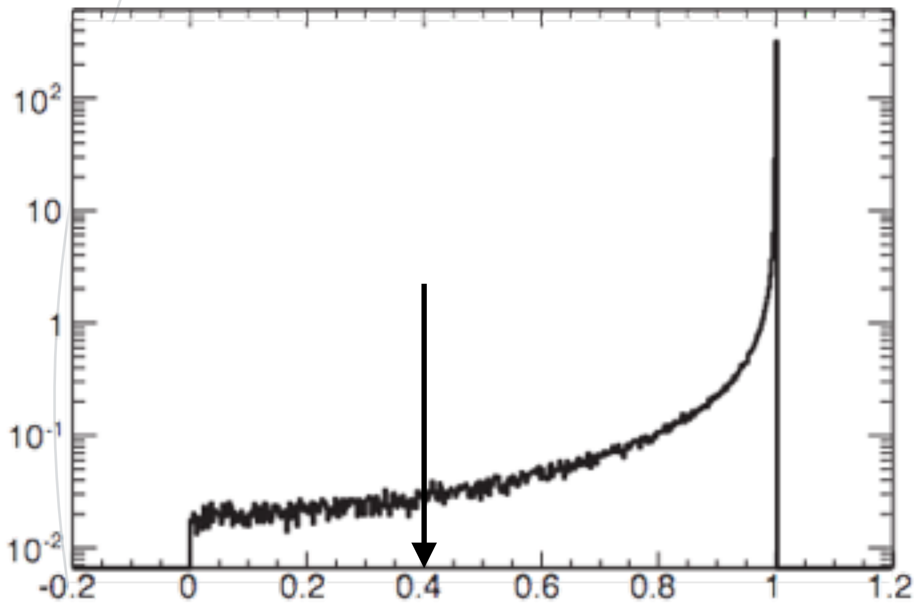


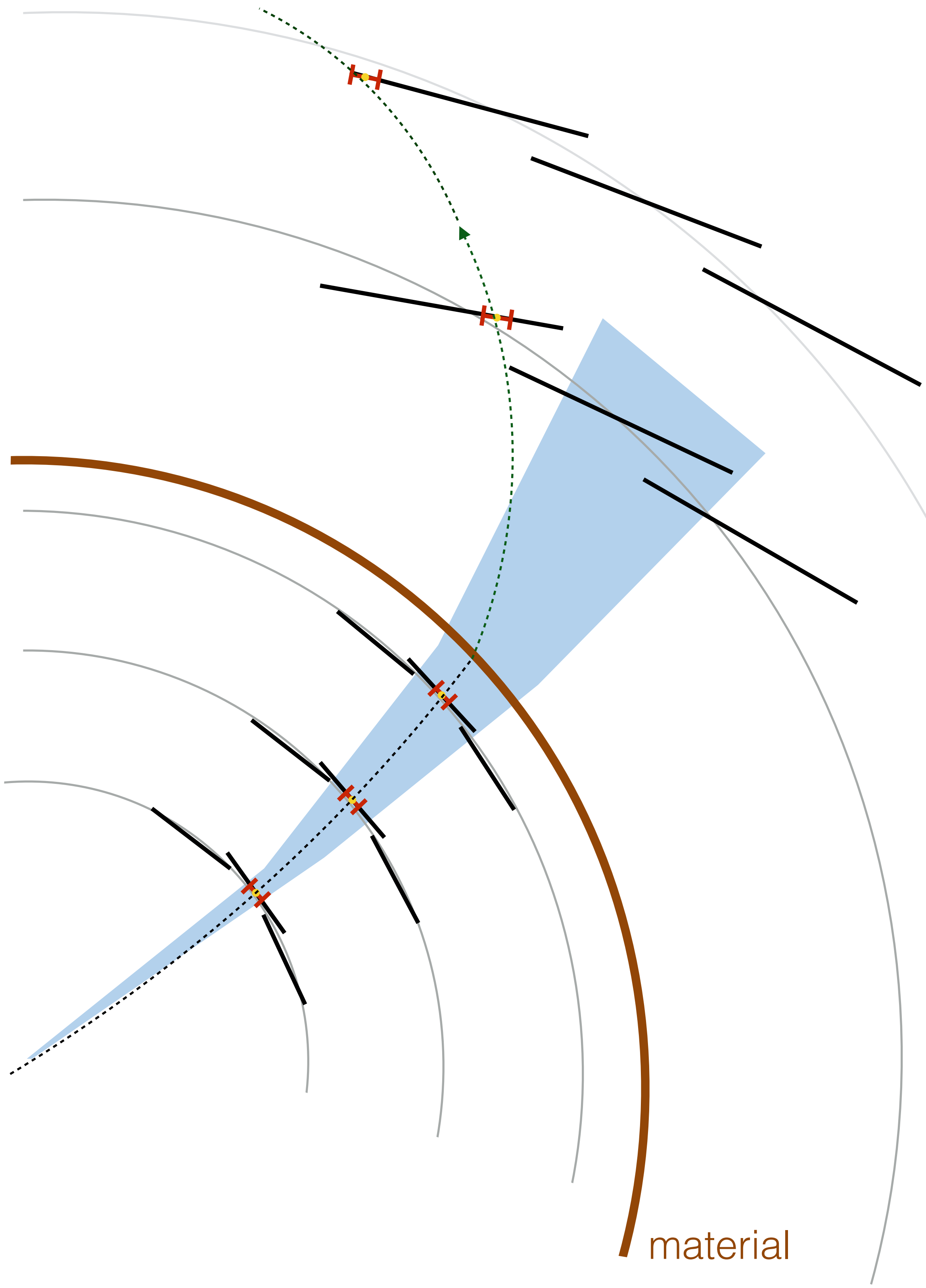
Highly non-gaussian systems

- Trying a naive global χ^2 fit
  - needs a large energy loss to compensate this change of curvature
- modification of the χ^2 function

$$\chi^2 = \sum_k \Delta m_k^T G_K^{-1} \Delta m_k + \sum_i \delta\theta_i^T Q_i^{-1} \delta\theta_i$$

with: $\Delta m_k = m_k - d_k(\mathbf{q}, \delta\theta_i)$



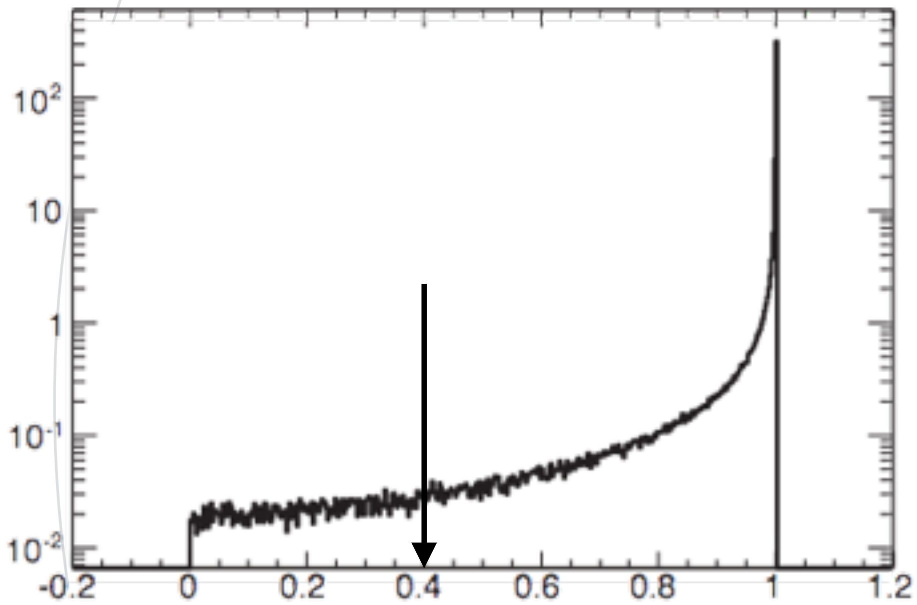


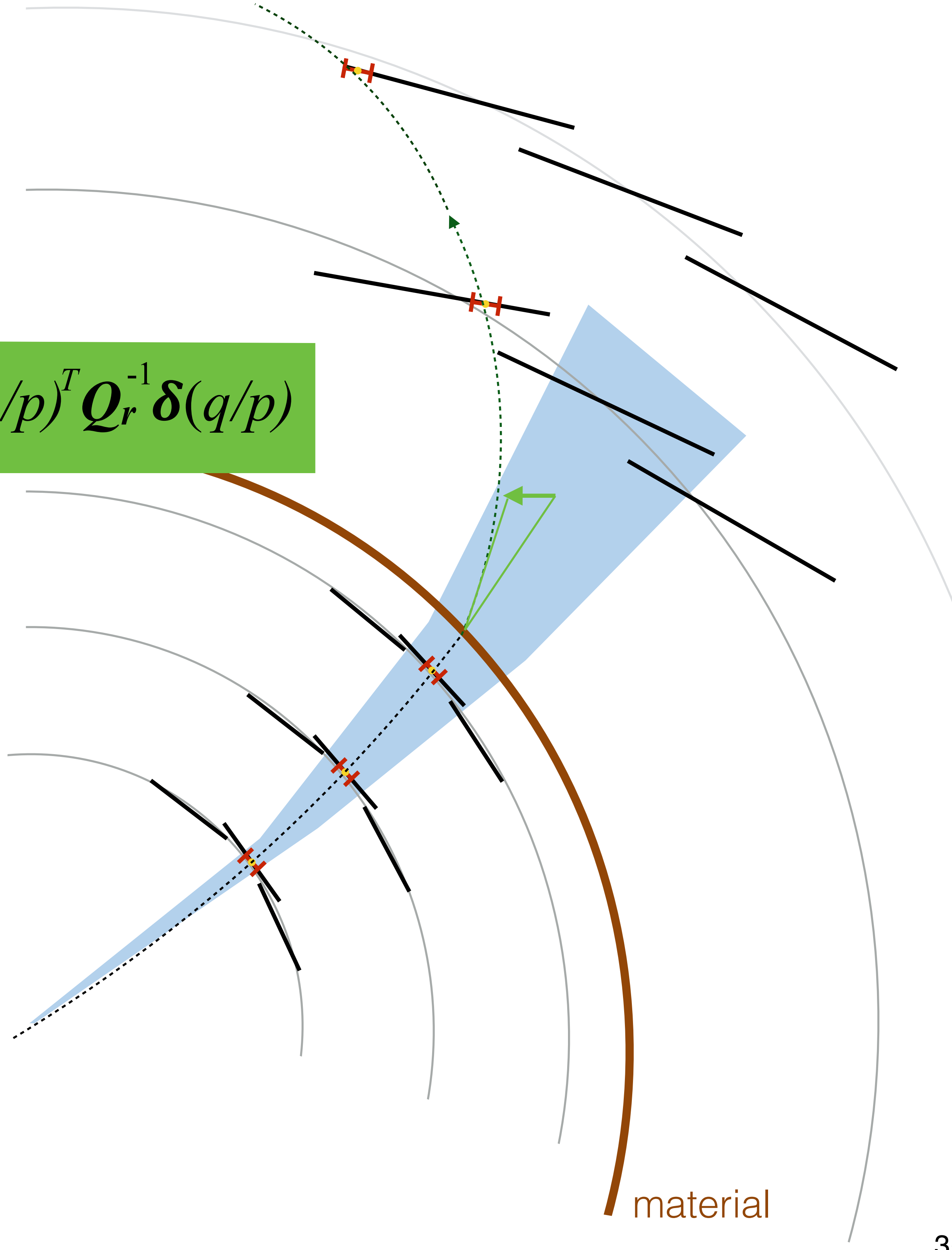
Highly non-gaussian systems

- Trying a naive global χ^2 fit
  - needs a large energy loss
    to compensate this change of curvature
- modification of the χ^2 function

$$\chi^2 = \sum_k \Delta m_k^T G_K^{-1} \Delta m_k + \sum_i \delta\theta_i^T Q_i^{-1} \delta\theta_i + \boldsymbol{\delta}(q/p)^T \boldsymbol{Q}_r^{-1} \boldsymbol{\delta}(q/p)$$

with: $\Delta m_k = m_k - d_k\left(\mathbf{q}, \delta\theta_i, \boldsymbol{\delta}(q/p)\right)$



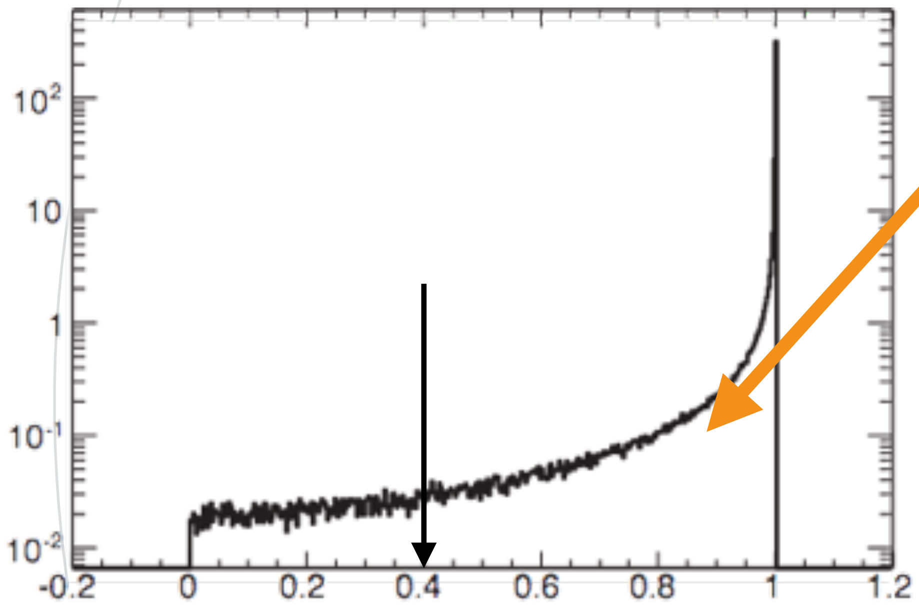


Highly non-gaussian systems

- Trying a naive global χ^2 fit
  - needs a large energy loss
    to compensate this change of curvature
- modification of the χ^2 function

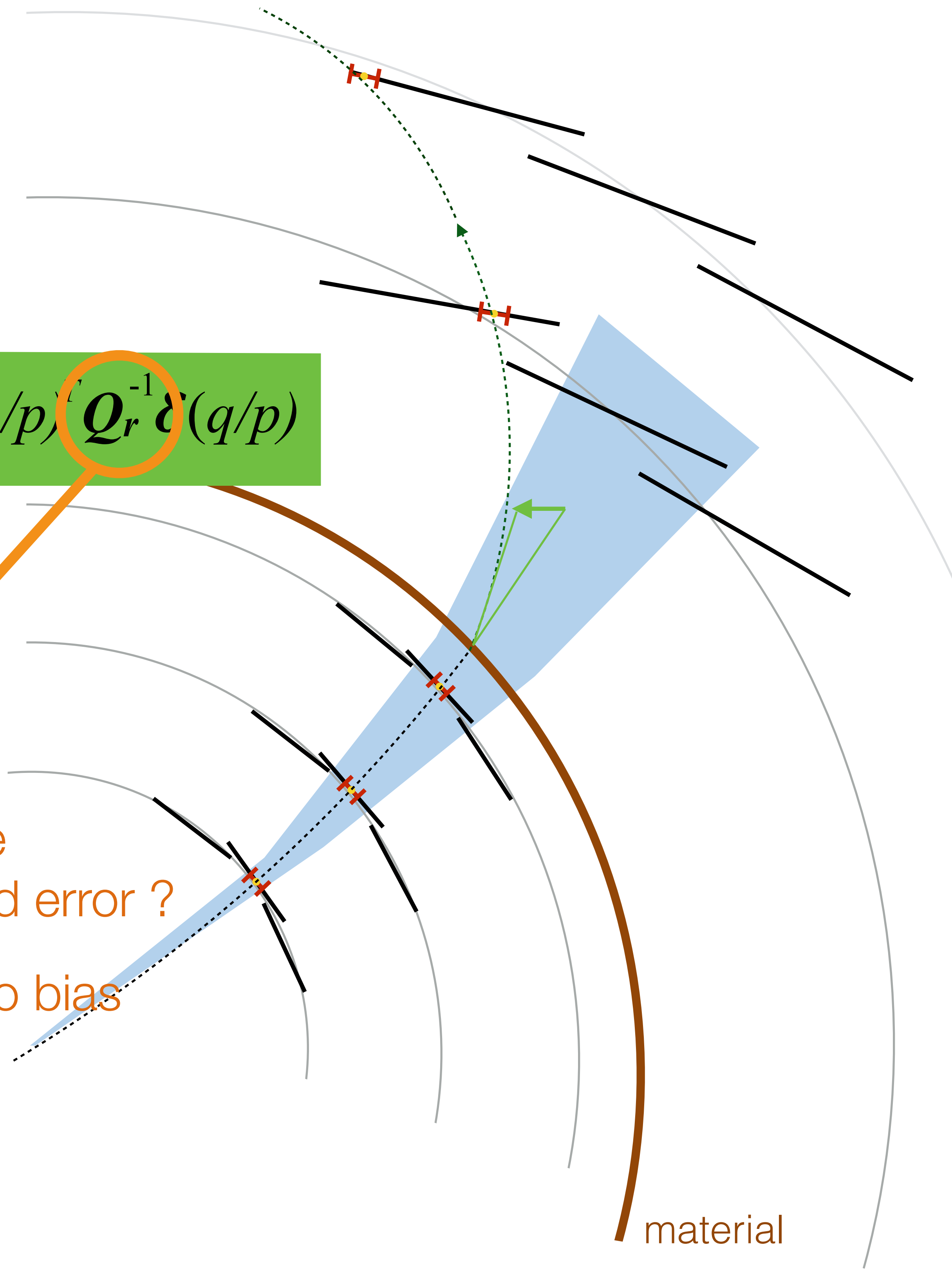
$$\chi^2 = \sum_k \Delta m_k^T G_K^{-1} \Delta m_k + \sum_i \delta\theta_i^T Q_i^{-1} \delta\theta_i + \boldsymbol{\delta}(q/p)^T \boldsymbol{Q_r}^{-1} \boldsymbol{\delta}(q/p)$$

with: $\Delta m_k = m_k - d_k\left(\mathbf{q}, \delta\theta_i, \boldsymbol{\delta}(q/p)\right)$



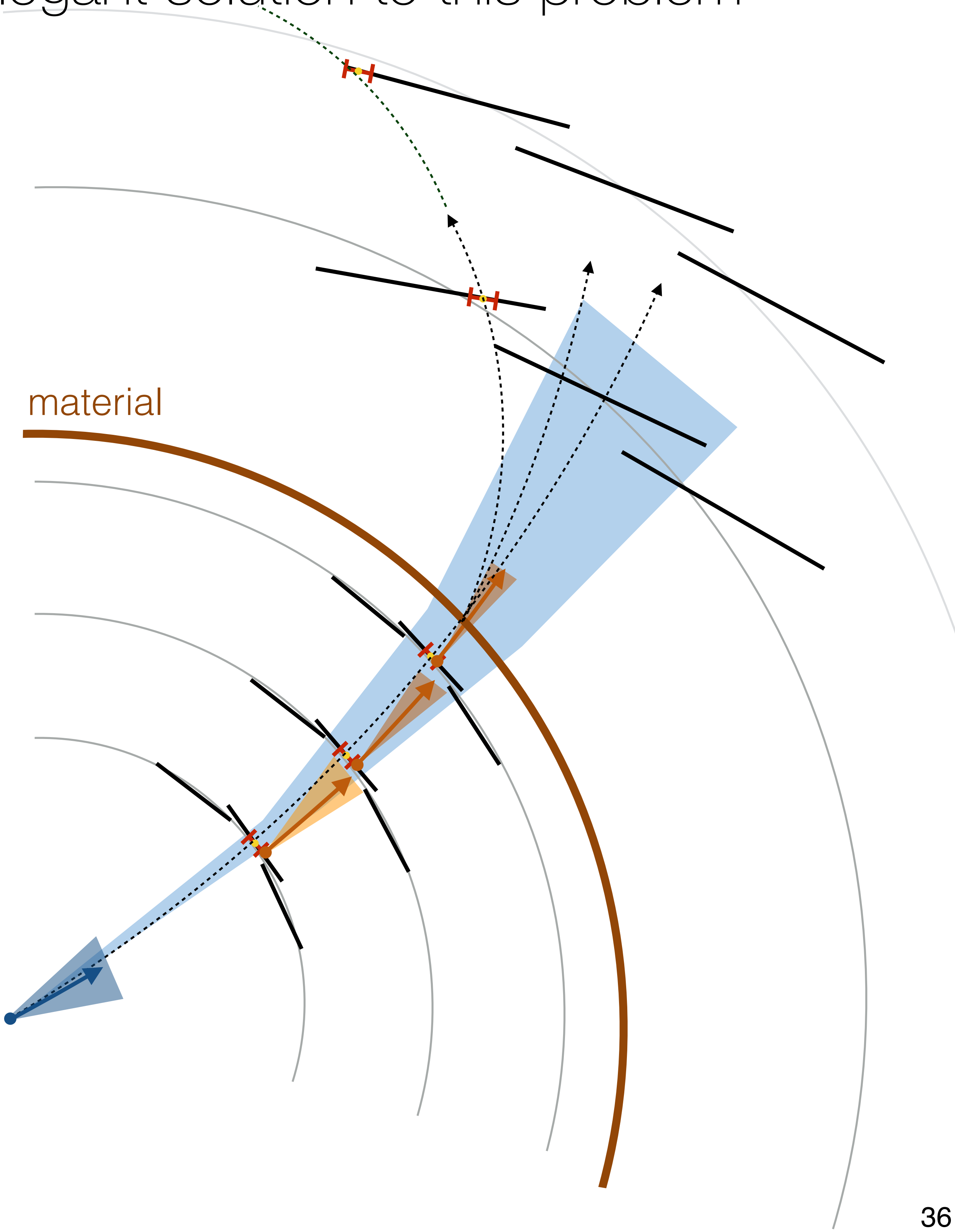
what's the associated error ?

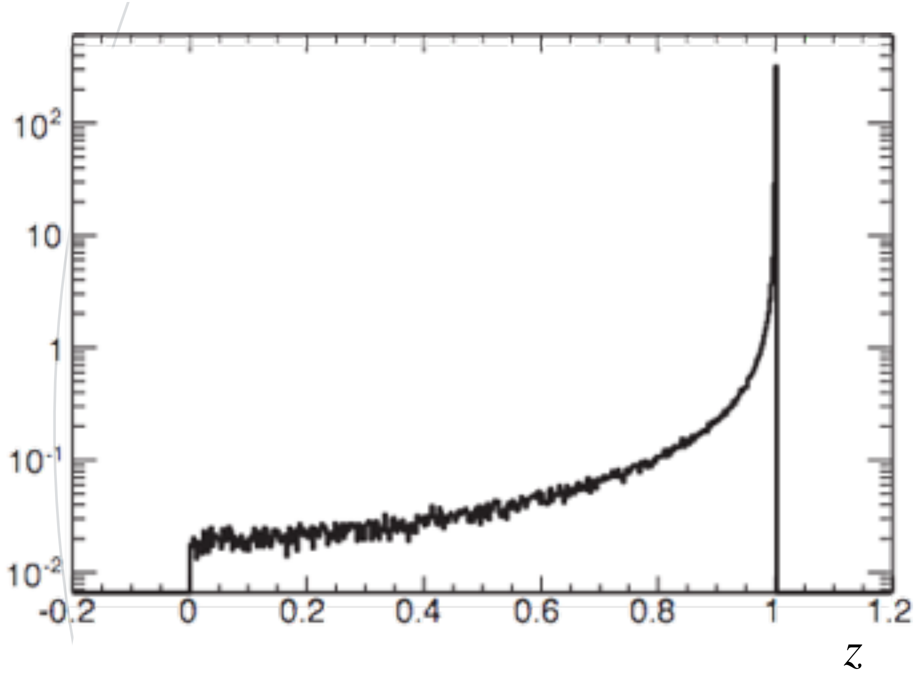
how not to bias the fit ?



The Gaussian Sum Filter

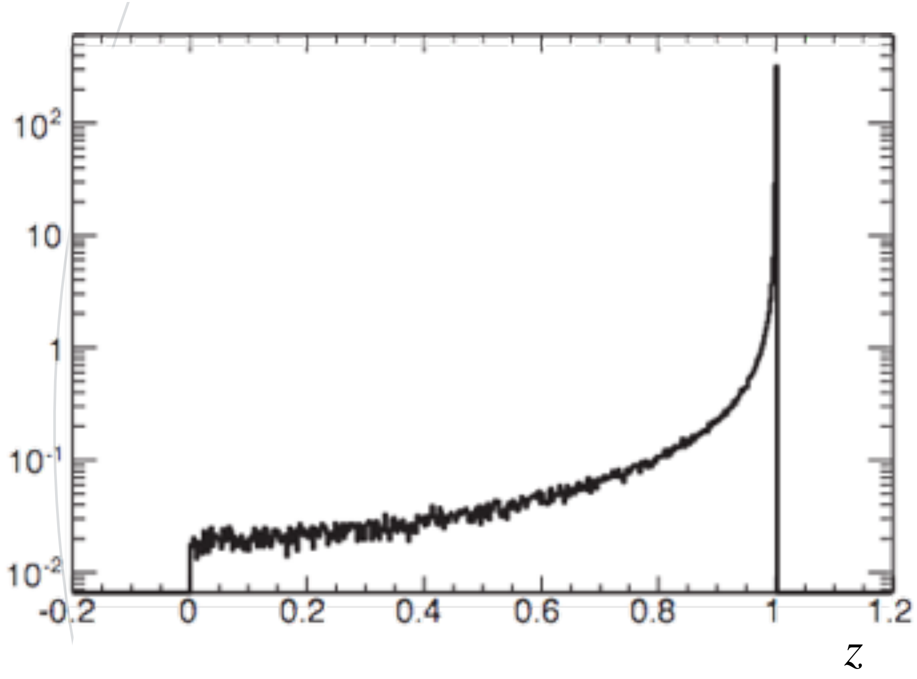
- Kalman filter formalism offers a very elegant solution to this problem

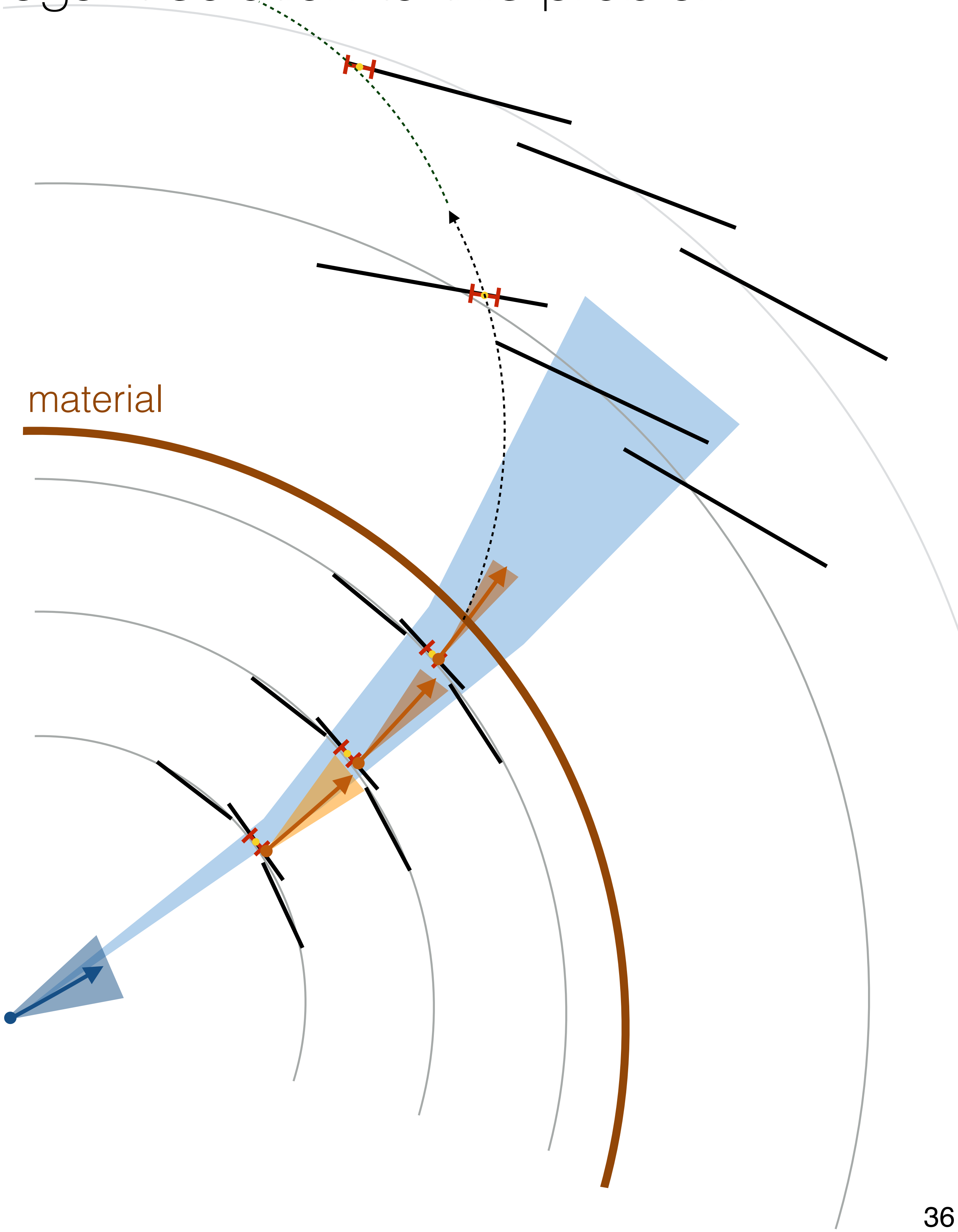




The Gaussian Sum Filter

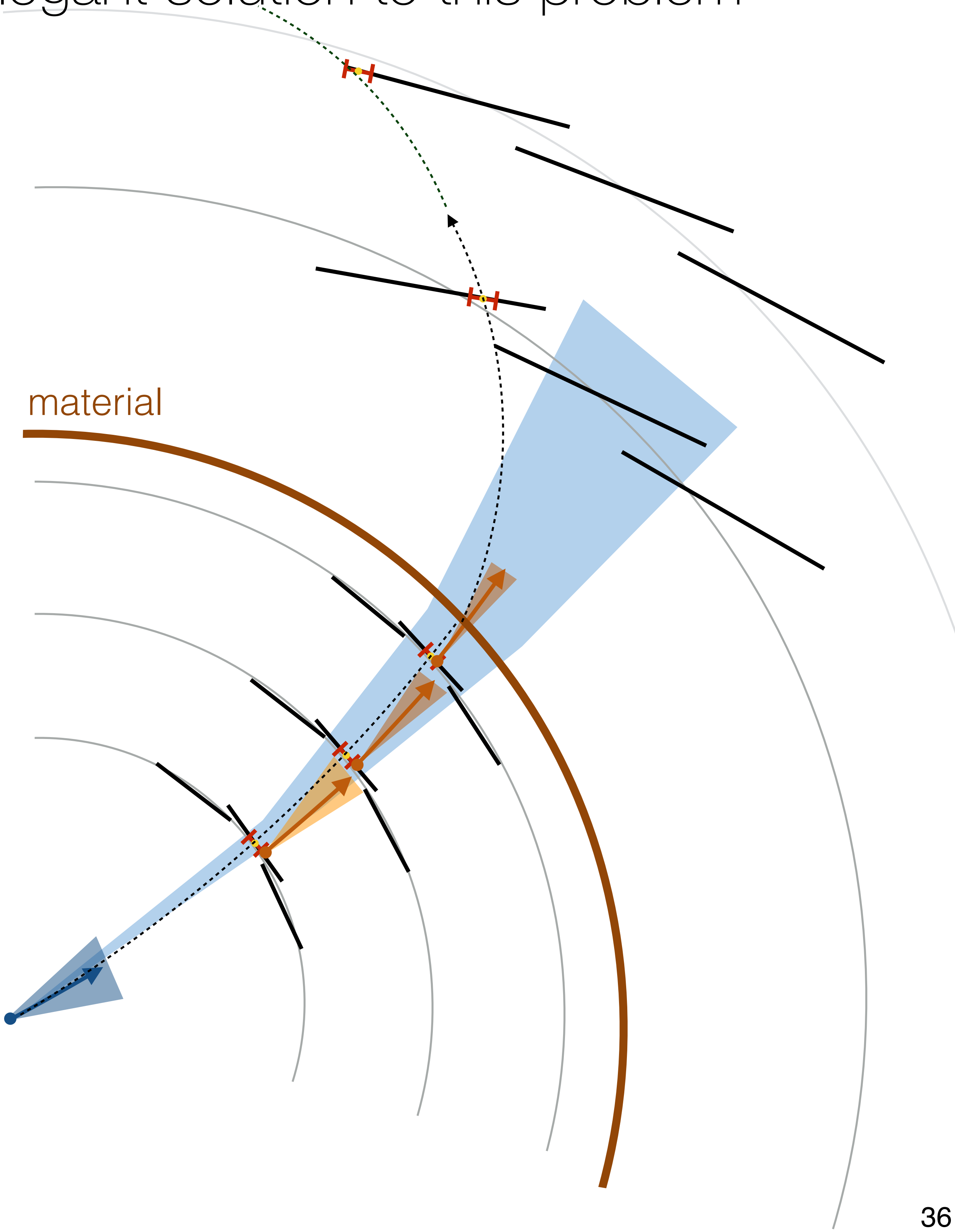
- Kalman filter formalism offers a very elegant solution to this problem

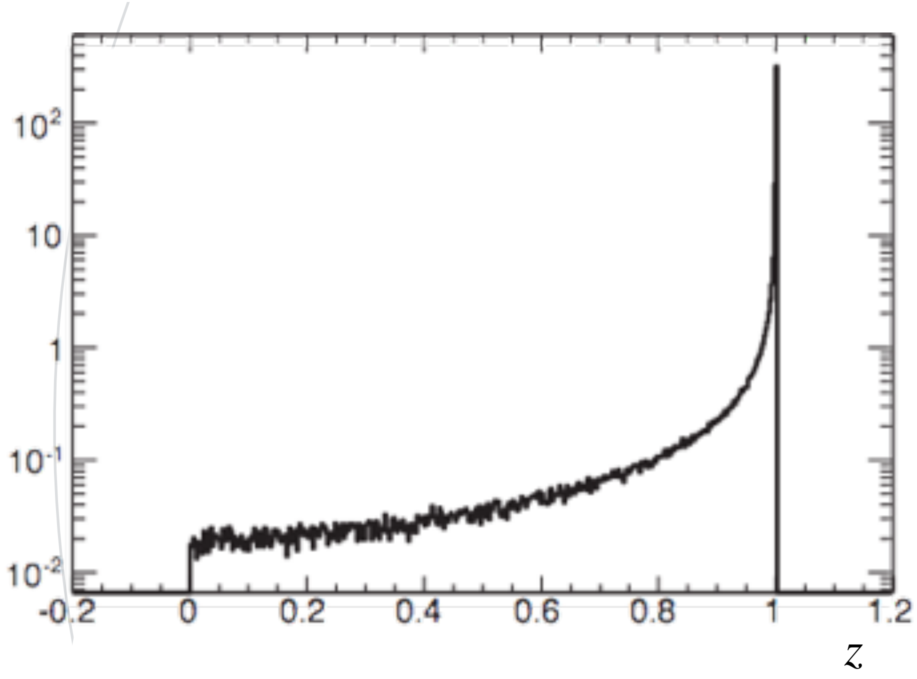




The Gaussian Sum Filter

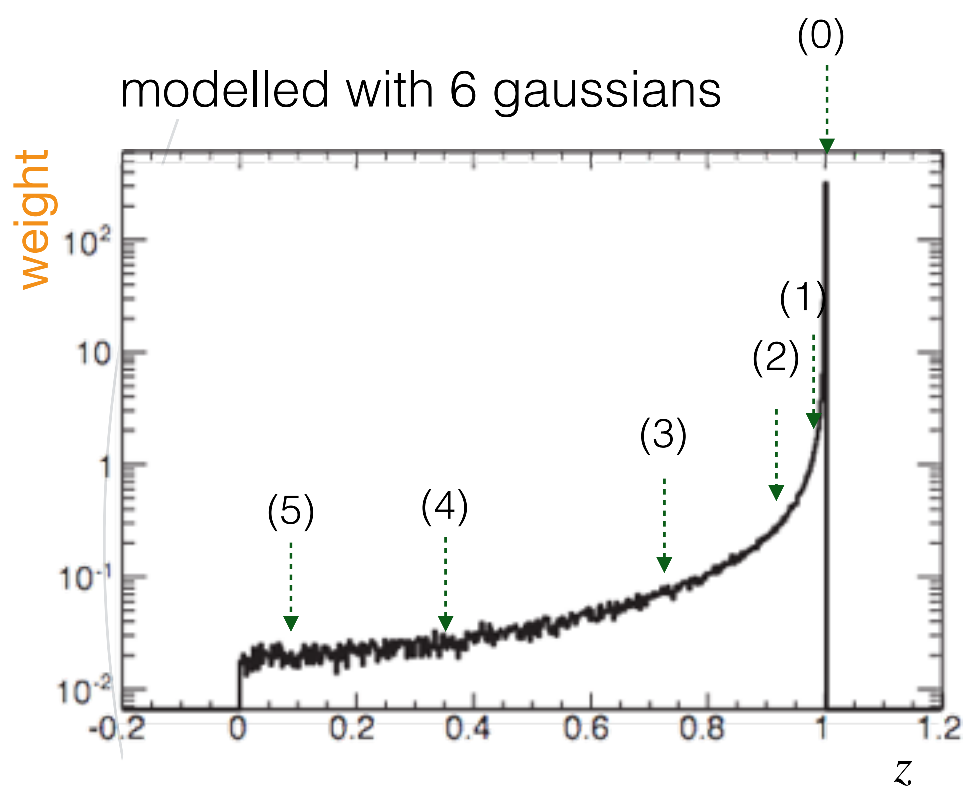
- Kalman filter formalism offers a very elegant solution to this problem

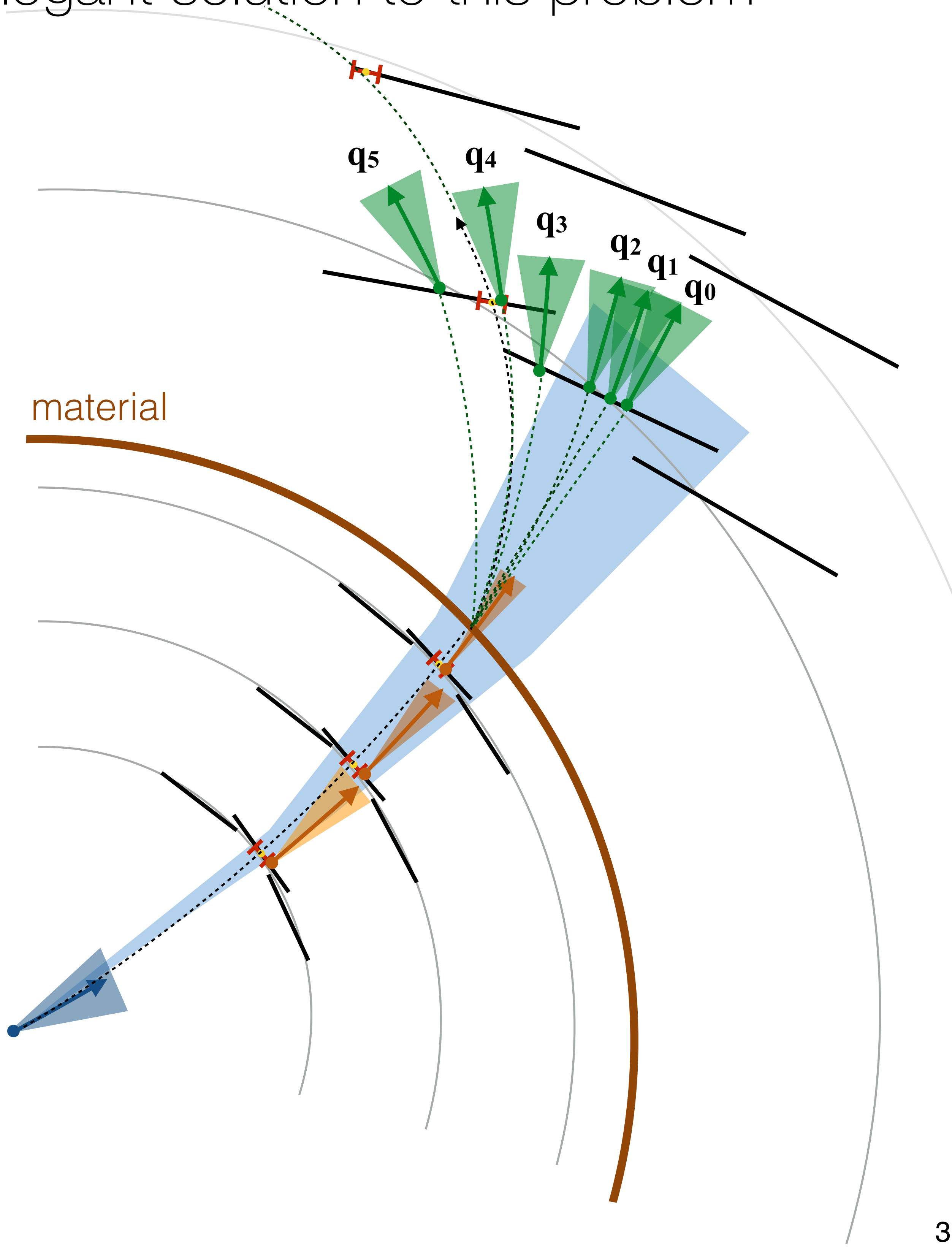




The Gaussian Sum Filter

- Kalman filter formalism offers a very elegant solution to this problem
  - fork the Kalman filter at the material layer into multiple components with weights and propagate them individually

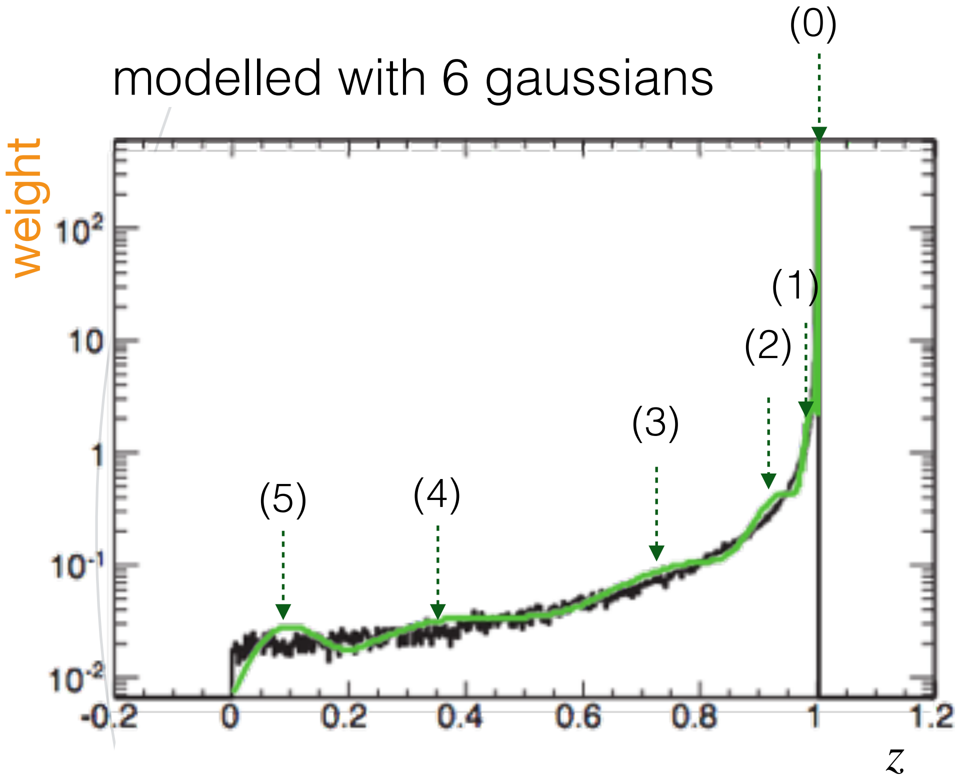


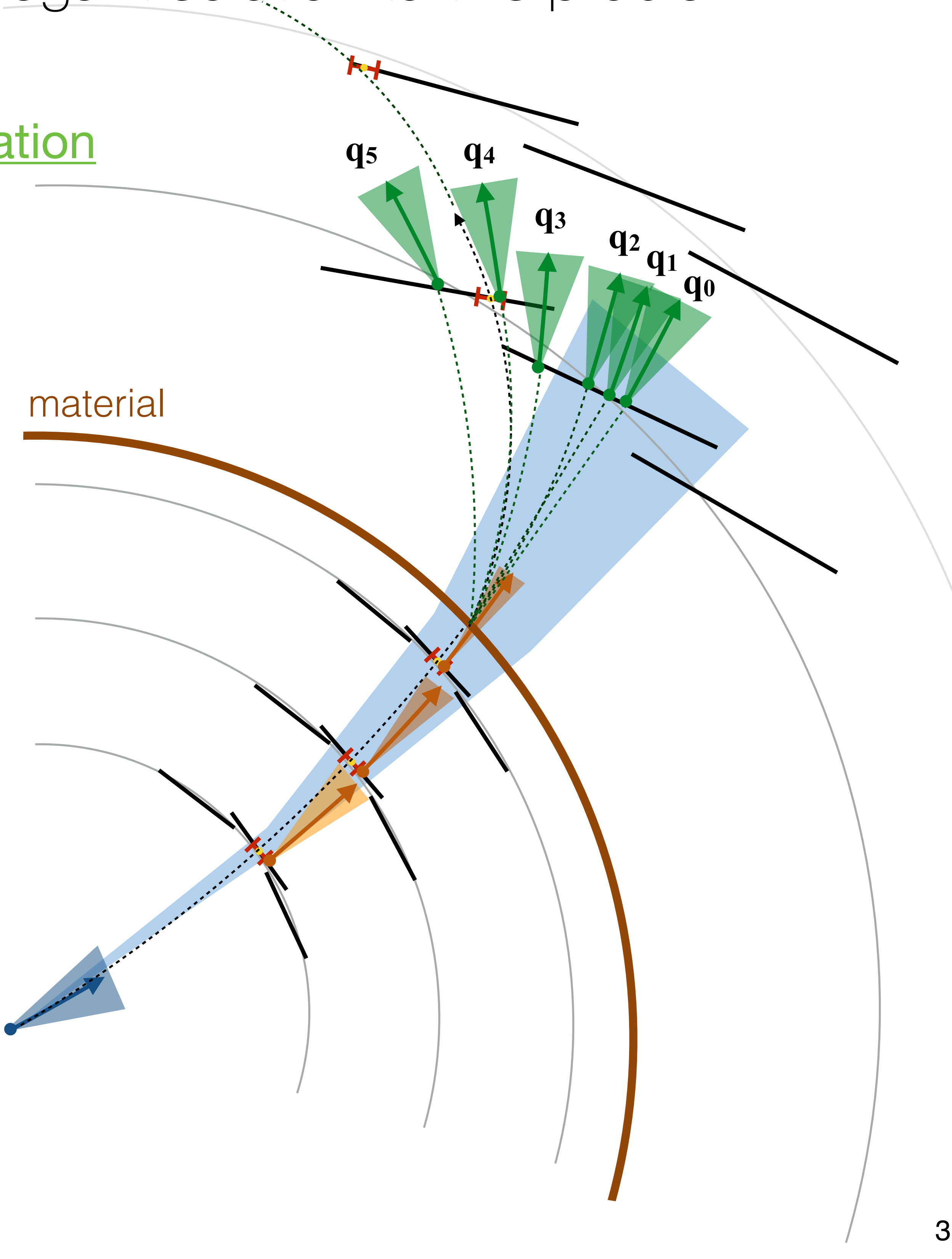


The Gaussian Sum Filter

- Kalman filter formalism offers a very elegant solution to this problem
  - modelling of non-gaussian noise through multivariant (gaussian) approximation
  - fork the Kalman filter at the material layer into multiple components with weights and propagate them individually

modelled with 6 gaussians

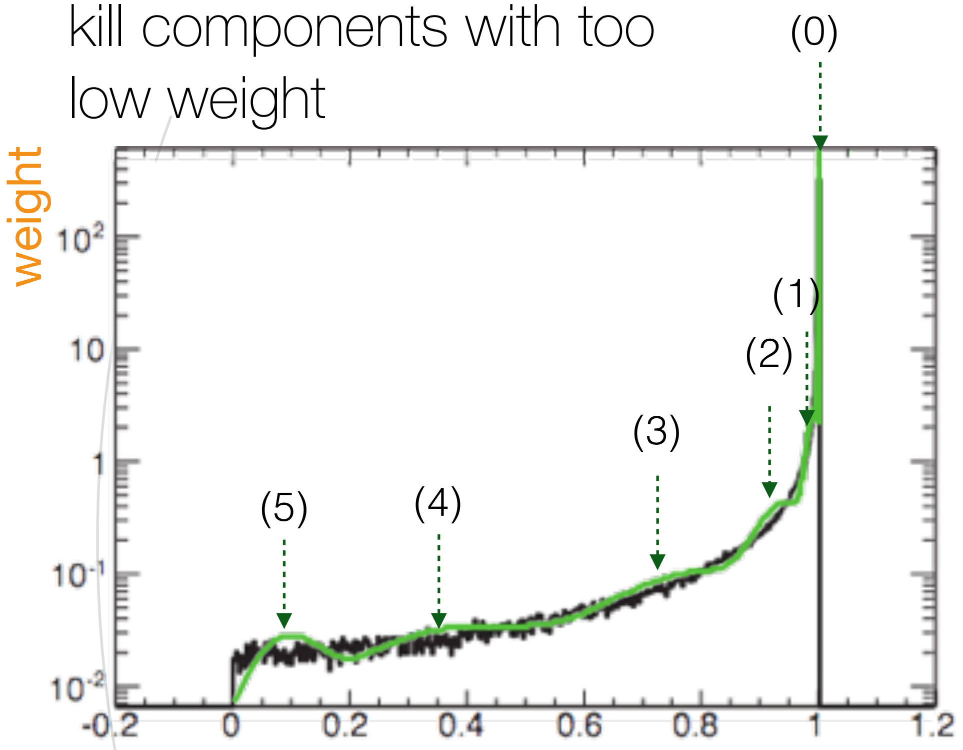


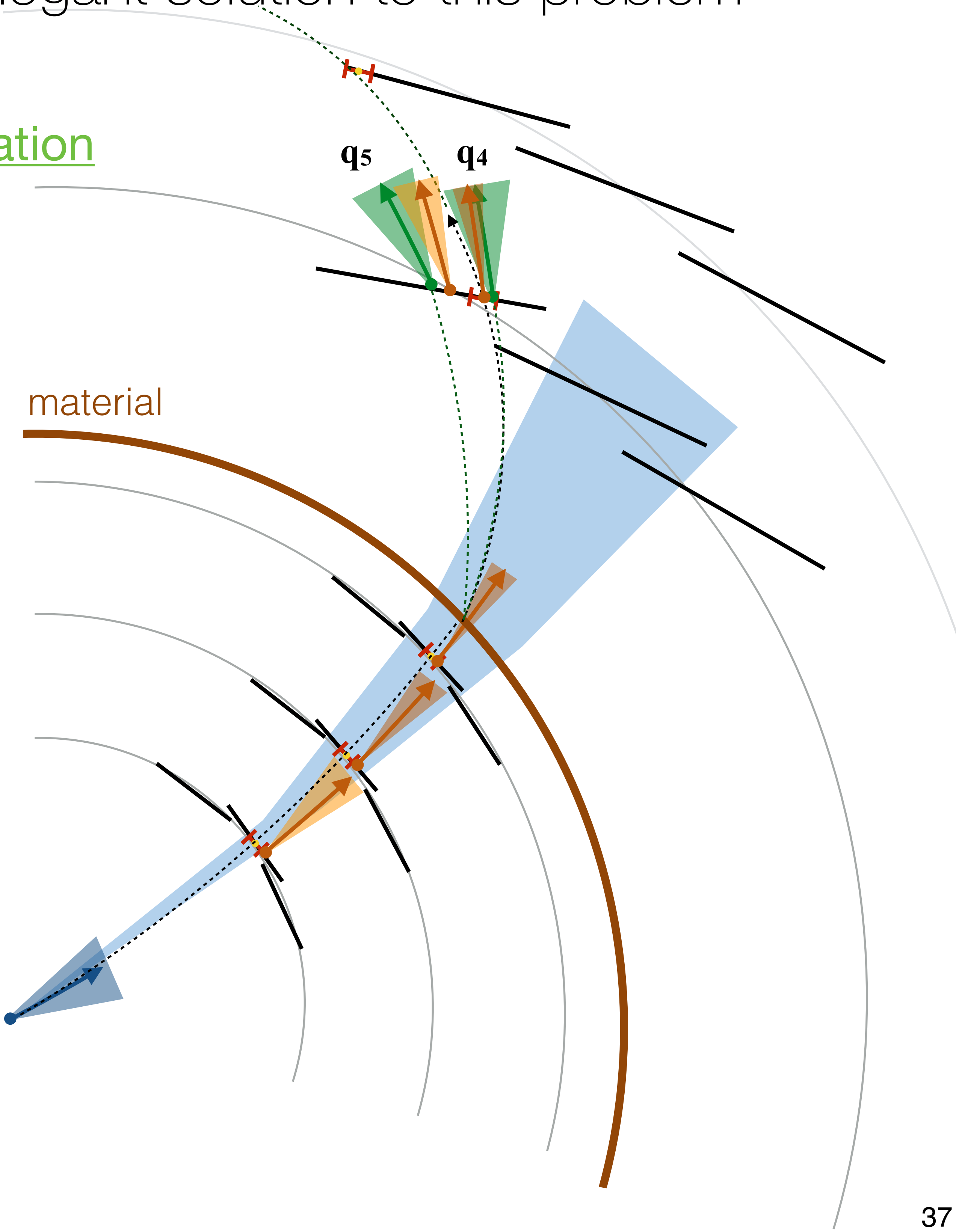


The Gaussian Sum Filter

- Kalman filter formalism offers a very elegant solution to this problem
  - modelling of non-gaussian noise through multivariant (gaussian) approximation
  - fork the Kalman filter at the material layer into multiple components with weights and propagate them individually
  - update each component and re-evaluate the weight depending on compatibility

    kill components with too low weight





Recap of today

- We've found tracks
  - global and local pattern recognition algorithms
- We've fitted those tracks
  - least squares estimator fit,
    e.g. global χ^2 minimazation, Kalman filter
- Discussed the fit output
- Touched upon "ghost tracks"
  - we will hear a bit more about that though
- Dedicated electron fitting