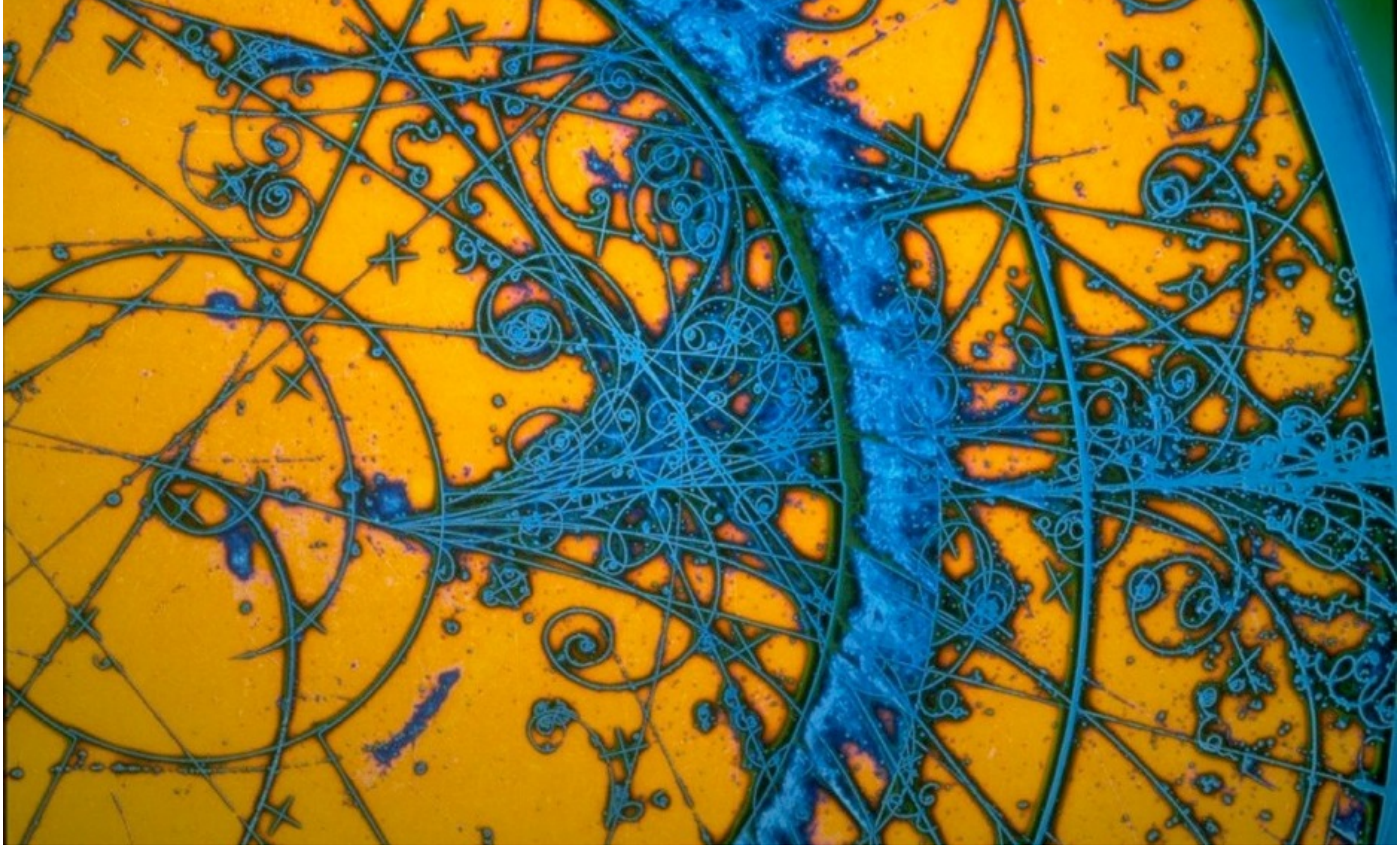


Track and Vertex reconstruction

A. Salzburger, CERN

Big European Bubble Chamber at CERN, photograph http://www.cern.ch

What is track reconstruction ?

- Track reconstruction is finding sets of measurements coming from one charged particle and building the associated trajectory through the detector. Tracks are generally used as input to higher level reconstruction objects.

  - set of measurements from charged particles

    Part 1 - basics & principle of tracking and tracking detectors

    - interaction of particles with (sensitive or not sensitive) detector material

  - finding associated measurements

    Part 2 - track finding strategies, global and local pattern recognition algorithms

  - trajectory estimation & track cleaning

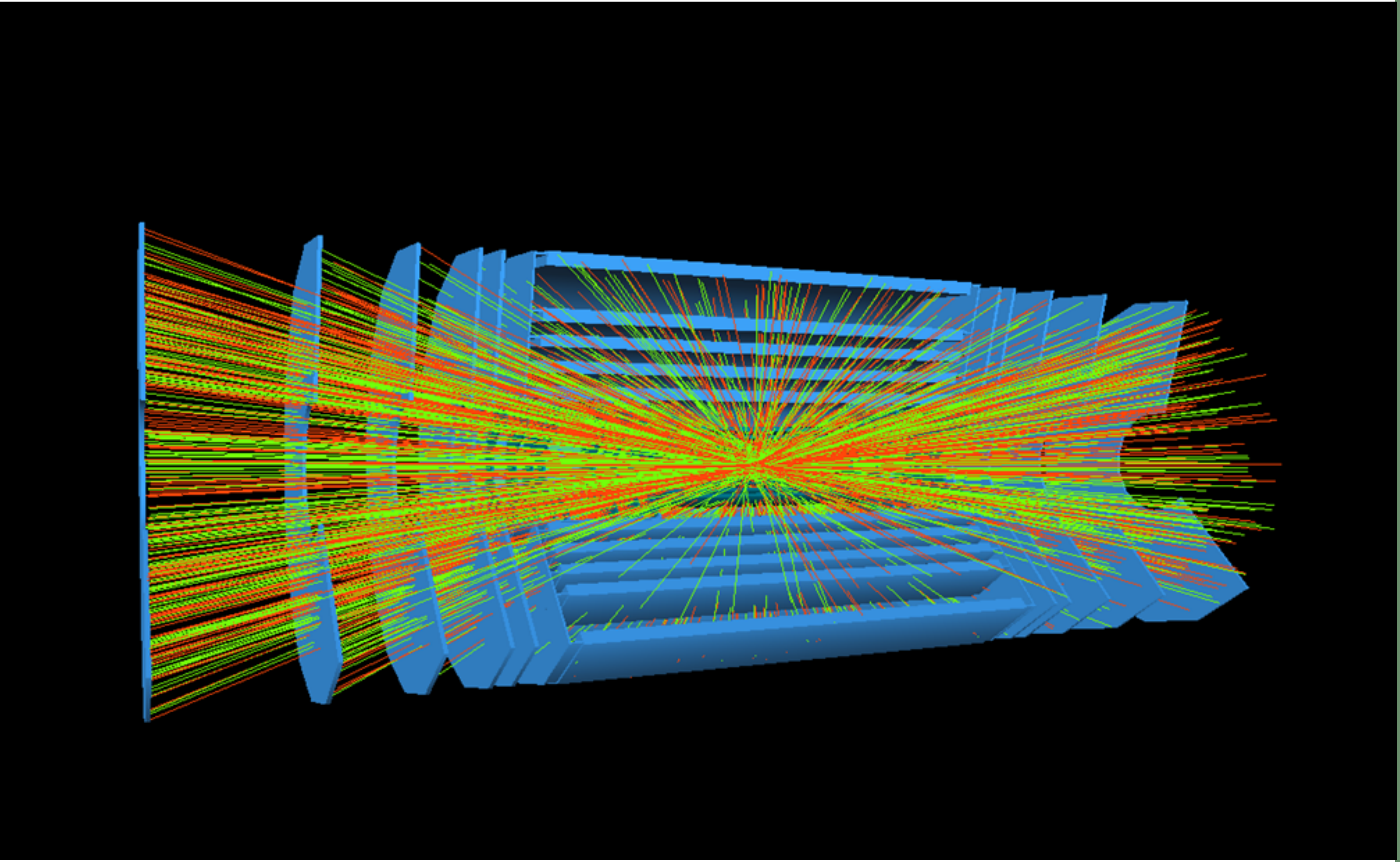
    - track fitting, fake and efficiency estimation
    - adaptive, multi-variant and specialised methods

  - tracks as input to higher level reconstruction and analysis

    Part 3 - primary and secondary vertex reconstruction

    - analysis usage
    - the reality

Part I - Basics & Tracking Detectors



ATLAS ID upgrade concept, https://inspirehep.net/record/1085976/files/itk_230.png

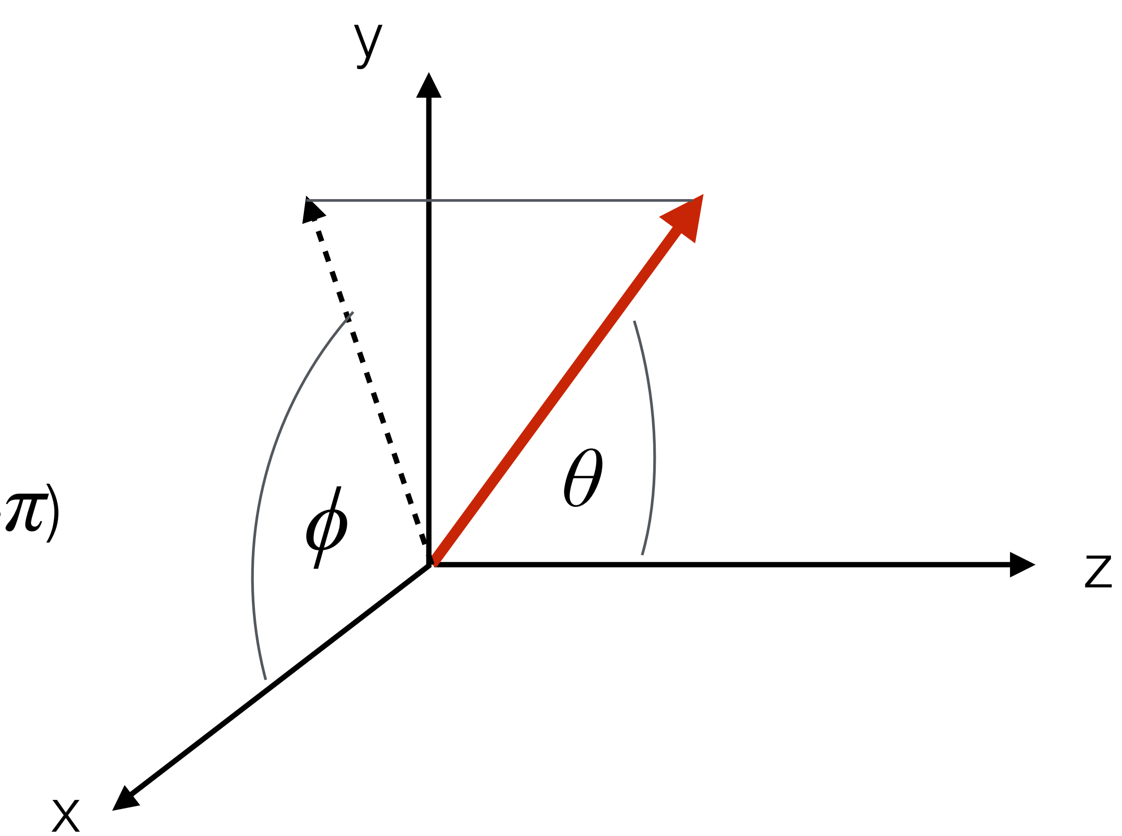
Boring - Definitions

- Let's get them out of the way …
  - coordinate systems are right-handed

    global : (x,y,z)

    local: (l_x,l_y,l_z)

  - ϕ measured in transverse plane in [-π,+π) (azimuthal angle)
  - θ is measured from z axis in [0,π] (polar angle)
  - $\lambda = \pi/2 - \theta$
  - $\eta = -\ln [\tan (\theta/2)]$ is the pseudo-rapidity (rapidity of a massless particle)

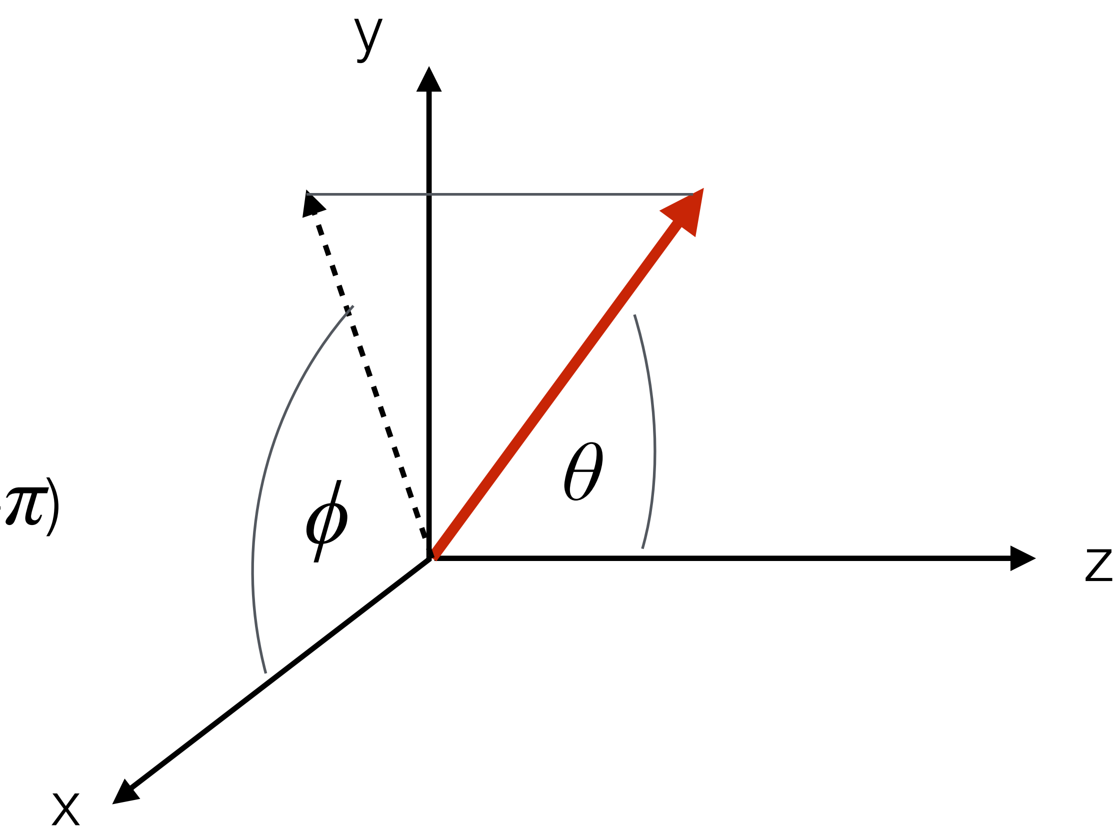


Boring - Definitions

- Let's get them out of the way …
  - coordinate systems are right-handed

    global : (x,y,z)

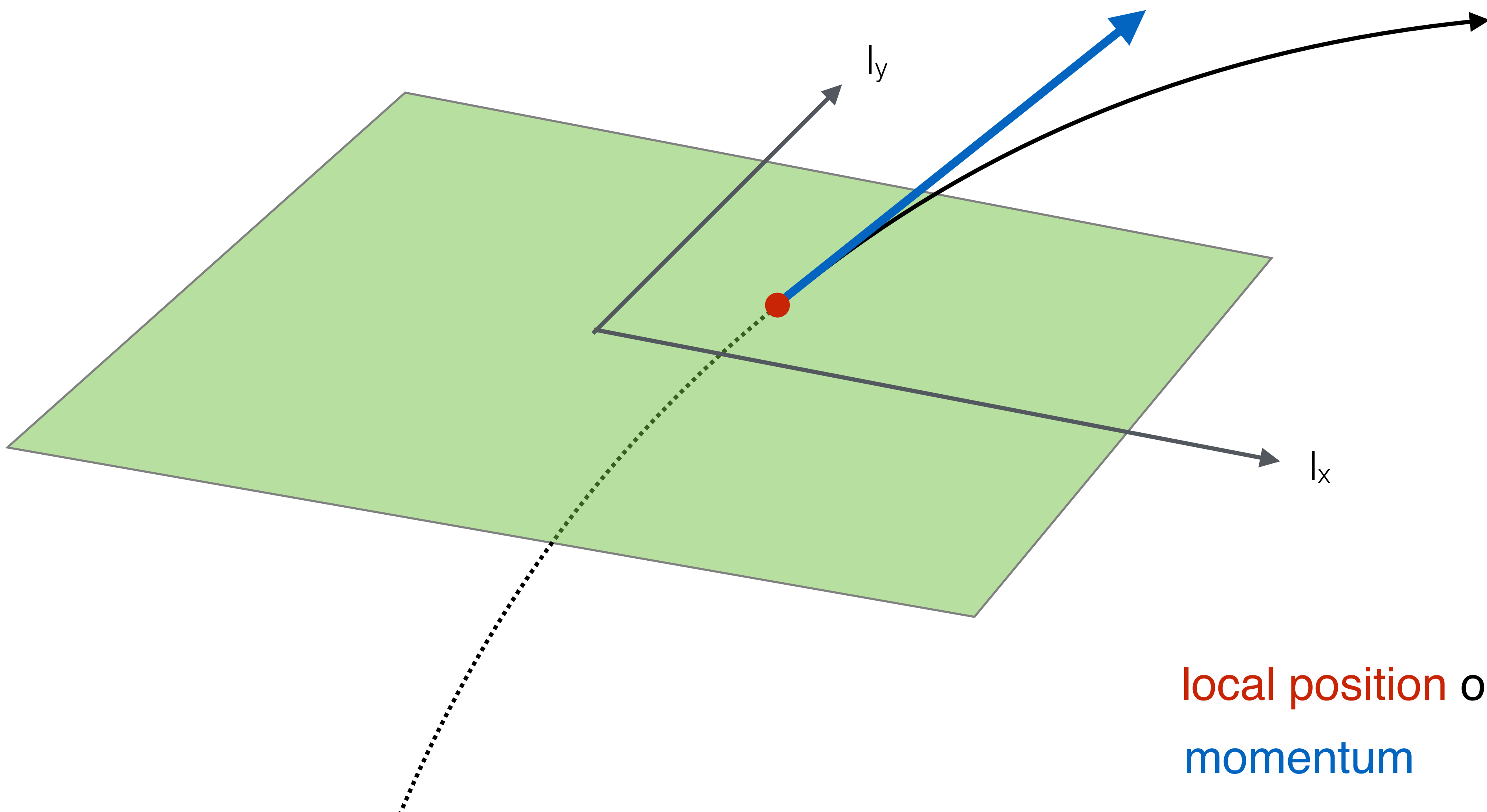
    local: (l_x,l_y,l_z)
  - ϕ measured in transverse plane in $[-\pi,+\pi)$ (azimuthal angle)
  - θ is measured from z axis in $[0,\pi]$ (polar angle)
  - $\lambda = \pi/2 - \theta$
  - $\eta = -\ln[\tan(\theta/2)]$ is the pseudo-rapidity (rapidity of a massless particle)



 $= Andreas$

Track parameterisation

- When bound to a **surface**, a trajectory of a charged particle needs in a magnetic field **five parameters** to be defined



local position on surface

momentum

charge

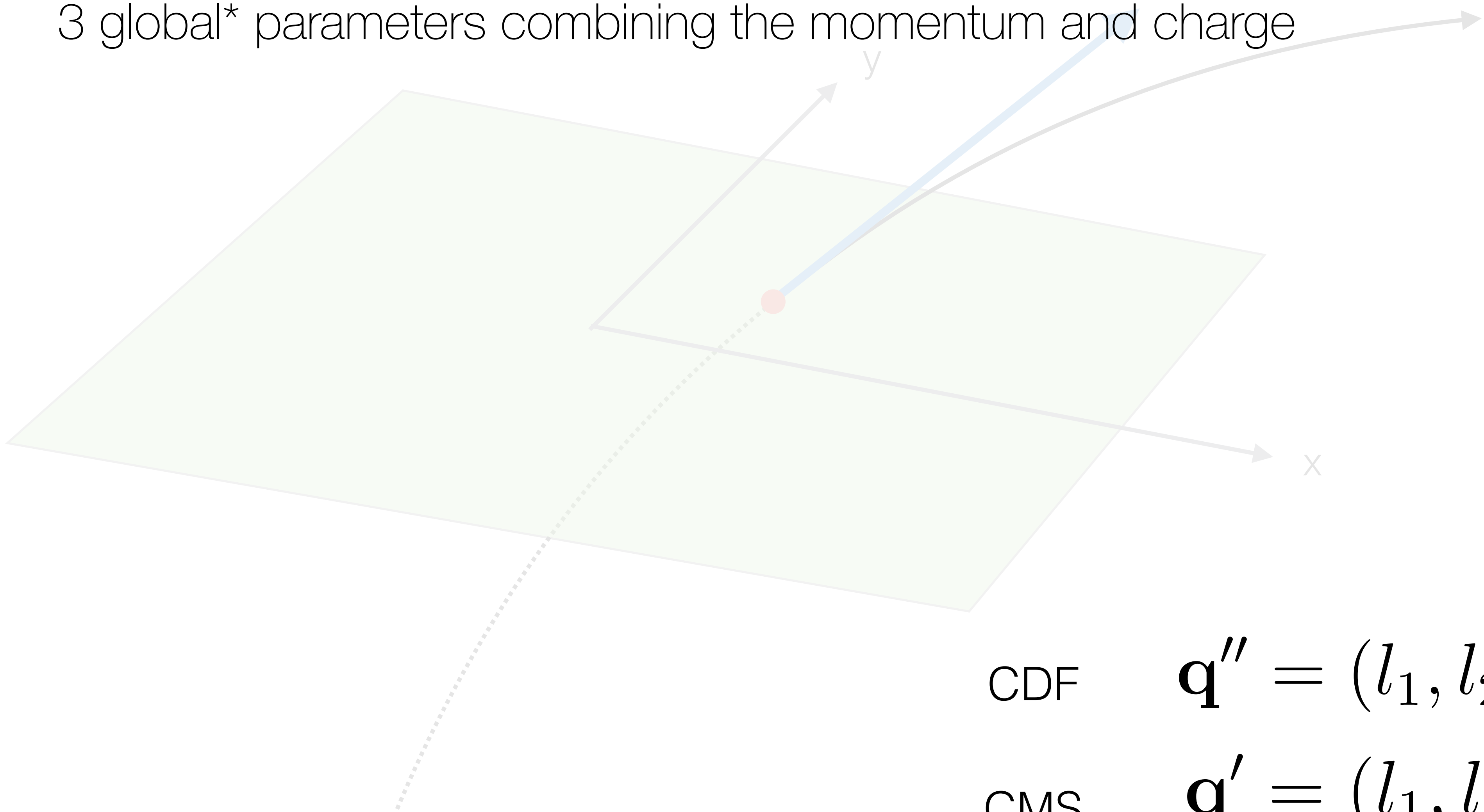
$$\mathbf{q} = \boxed{(l_1, l_2,}\boxed{\phi, \theta, q/p)}$$

Track parameterisation

- there is a certain level of freedom in the actual parameterisation
  - general feature:

    2 local* parameters bound to the surface

    3 global* parameters combining the momentum and charge



CDF $\mathbf{q}'' = (l_1, l_2, \phi, \cot(\theta), C)$

CMS $\mathbf{q}' = (l_1, l_2, \phi, \lambda, q/p)$

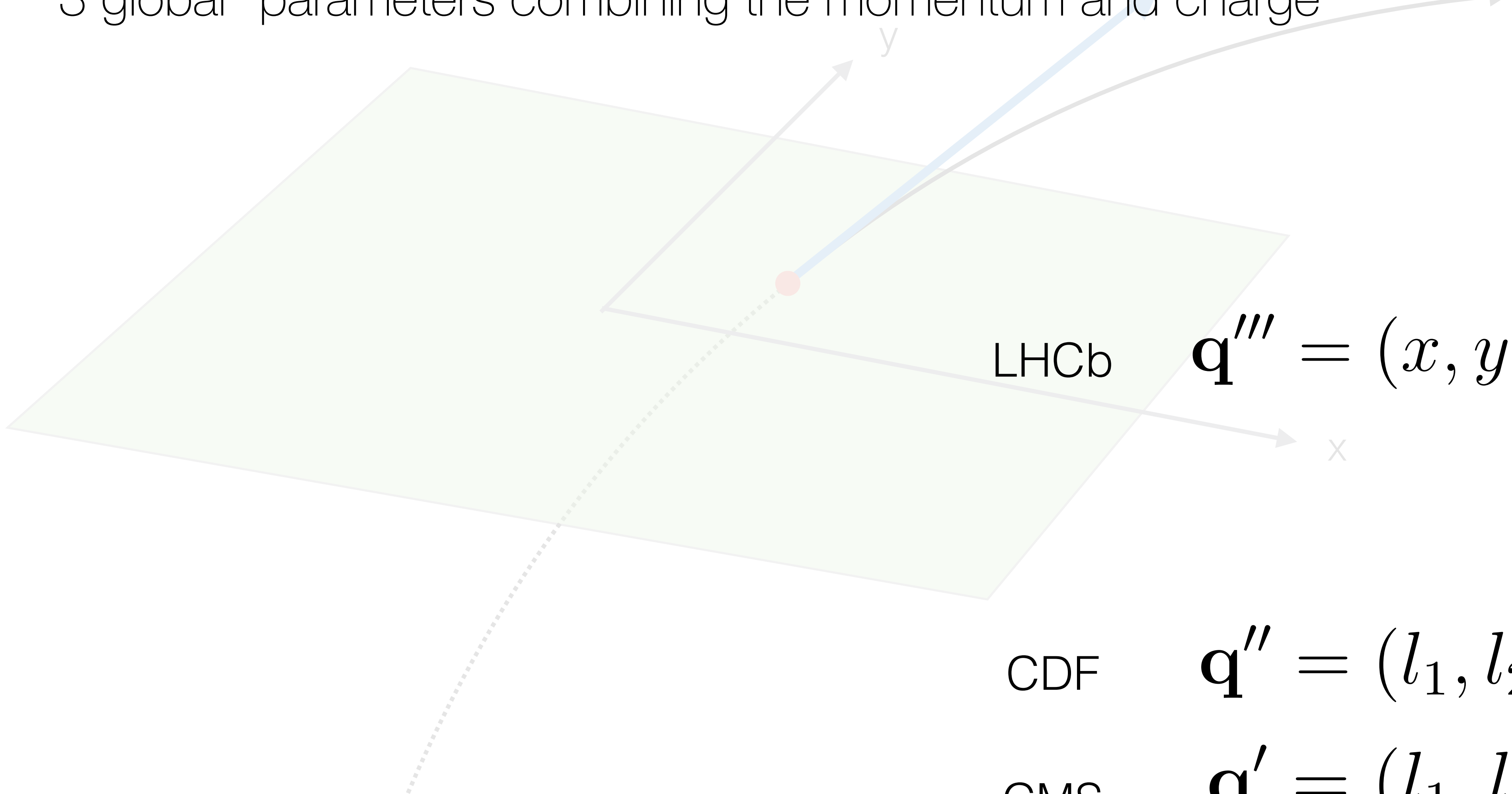
ATLAS $\mathbf{q} = (l_1, l_2, \phi, \theta, q/p)$

Track parameterisation

- there is a certain level of freedom in the actual parameterisation
  - general feature:

    2 local* parameters bound to the surface

    3 global* parameters combining the momentum and charge



LHCb $$\mathbf{q}''' = (x, y, t_x, t_y, q/p)$$

$$t_{x(y)} = \frac{\partial p}{\partial x(y)}$$

CDF $$\mathbf{q}'' = (l_1, l_2, \phi, \cot(\theta), C)$$

CMS $$\mathbf{q}' = (l_1, l_2, \phi, \lambda, q/p)$$

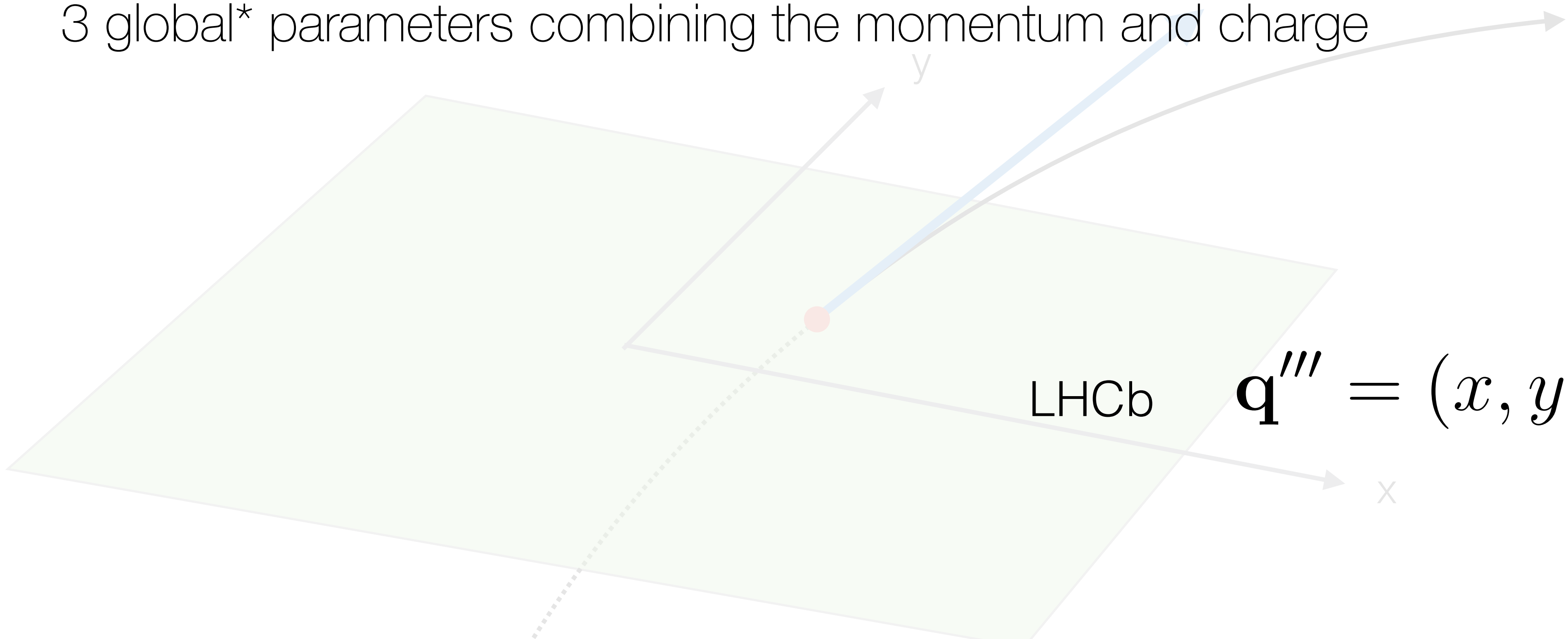
ATLAS $$\mathbf{q} = (l_1, l_2, \phi, \theta, q/p)$$

Track parameterisation

- there is a certain level of freedom in the actual parameterisation
  - general feature:

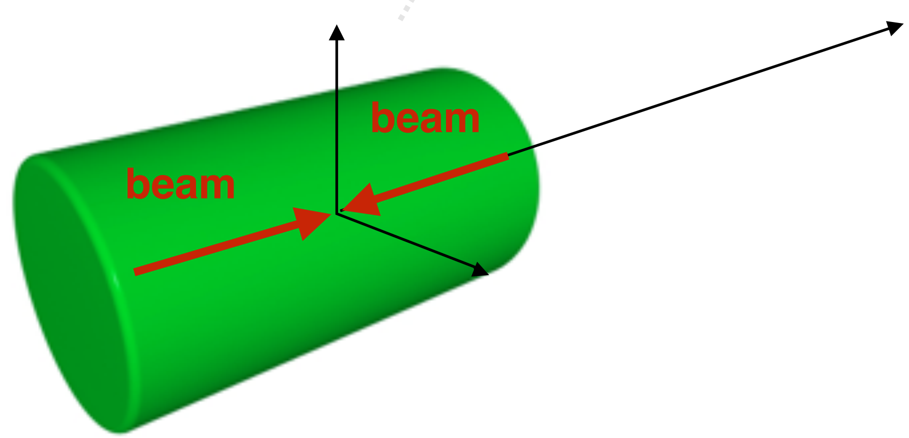
    2 local* parameters bound to the surface

    3 global* parameters combining the momentum and charge



LHCb $$\mathbf{q}''' = (x, y, t_x, t_y, q/p)$$

$$t_{x(y)} = \frac{\partial p}{\partial x(y)}$$



CDF $$\mathbf{q}'' = (l_1, l_2, \phi, \cot(\theta), C)$$

CMS $$\mathbf{q}' = (l_1, l_2, \phi, \lambda, q/p)$$

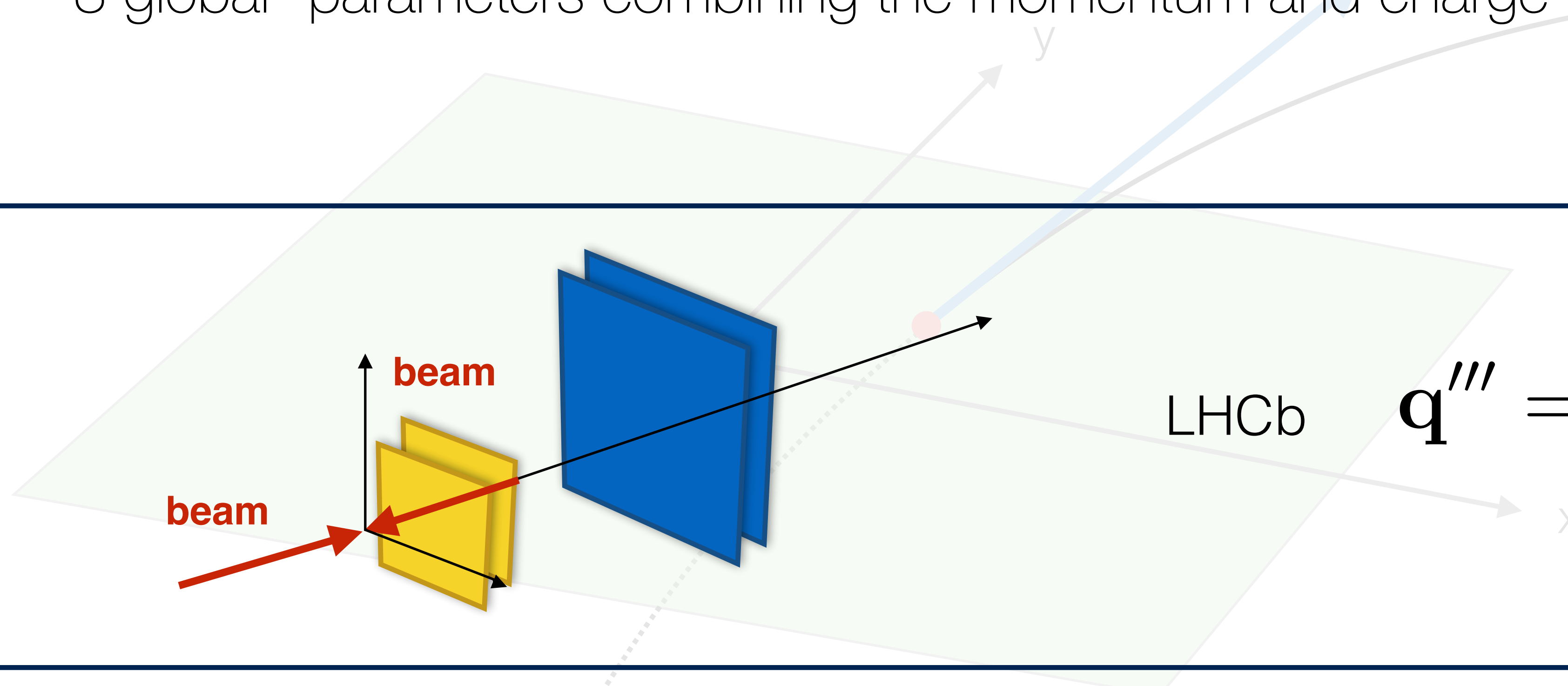
ATLAS $$\mathbf{q} = (l_1, l_2, \phi, \theta, q/p)$$

Track parameterisation

- there is a certain level of freedom in the actual parameterisation
  - general feature:

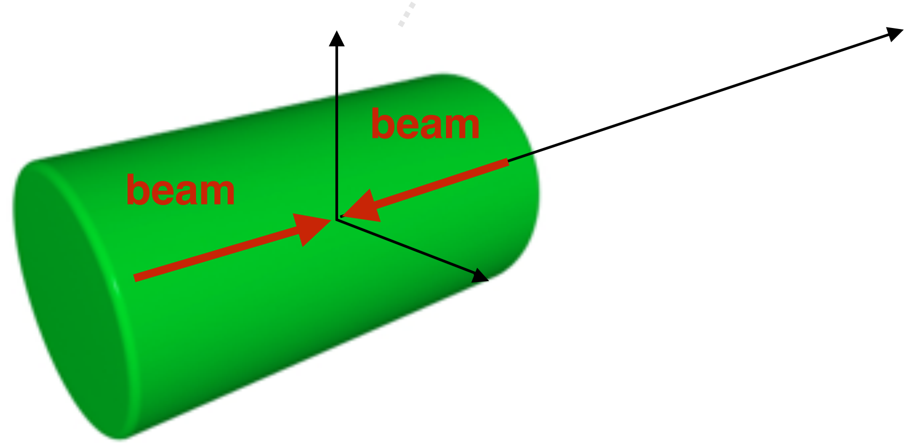
    2 local* parameters bound to the surface

    3 global* parameters combining the momentum and charge



LHCb $$\mathbf{q}''' = (x, y, t_x, t_y, q/p)$$

$$t_{x(y)} = \frac{\partial p}{\partial x(y)}$$



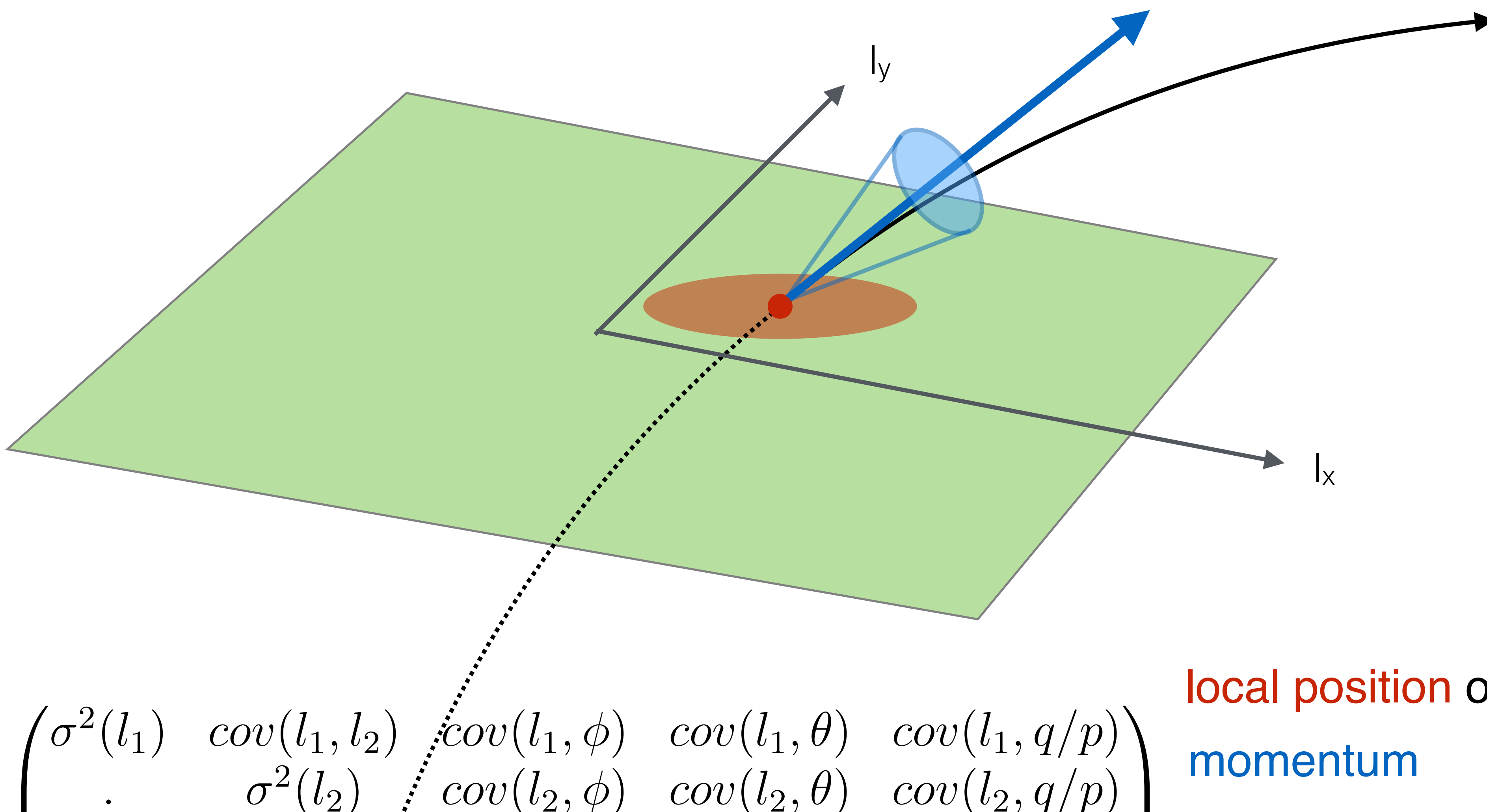
CDF $$\mathbf{q}'' = (l_1, l_2, \phi, \cot(\theta), C)$$

CMS $$\mathbf{q}' = (l_1, l_2, \phi, \lambda, q/p)$$

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

Track parameterisation with uncertainties

- When bound to a **surface**, a trajectory of a charged particle needs in a magnetic field **five parameters** ($\mathbf{q}$) to be defined: track parameters



$$\mathbf{C} = \begin{pmatrix} \sigma^2(l_1) & cov(l_1,l_2) & cov(l_1,\phi) & cov(l_1,\theta) & cov(l_1,q/p) \\ . & \sigma^2(l_2) & cov(l_2,\phi) & cov(l_2,\theta) & cov(l_2,q/p) \\ . & . & \sigma^2(\phi) & cov(\phi,\theta) & cov(\phi,q/p) \\ . & . & . & \sigma^2(\theta) & cov(\theta,q/p) \\ . & . & . & . & \sigma^2(q/p) \end{pmatrix}$$

local position on surface

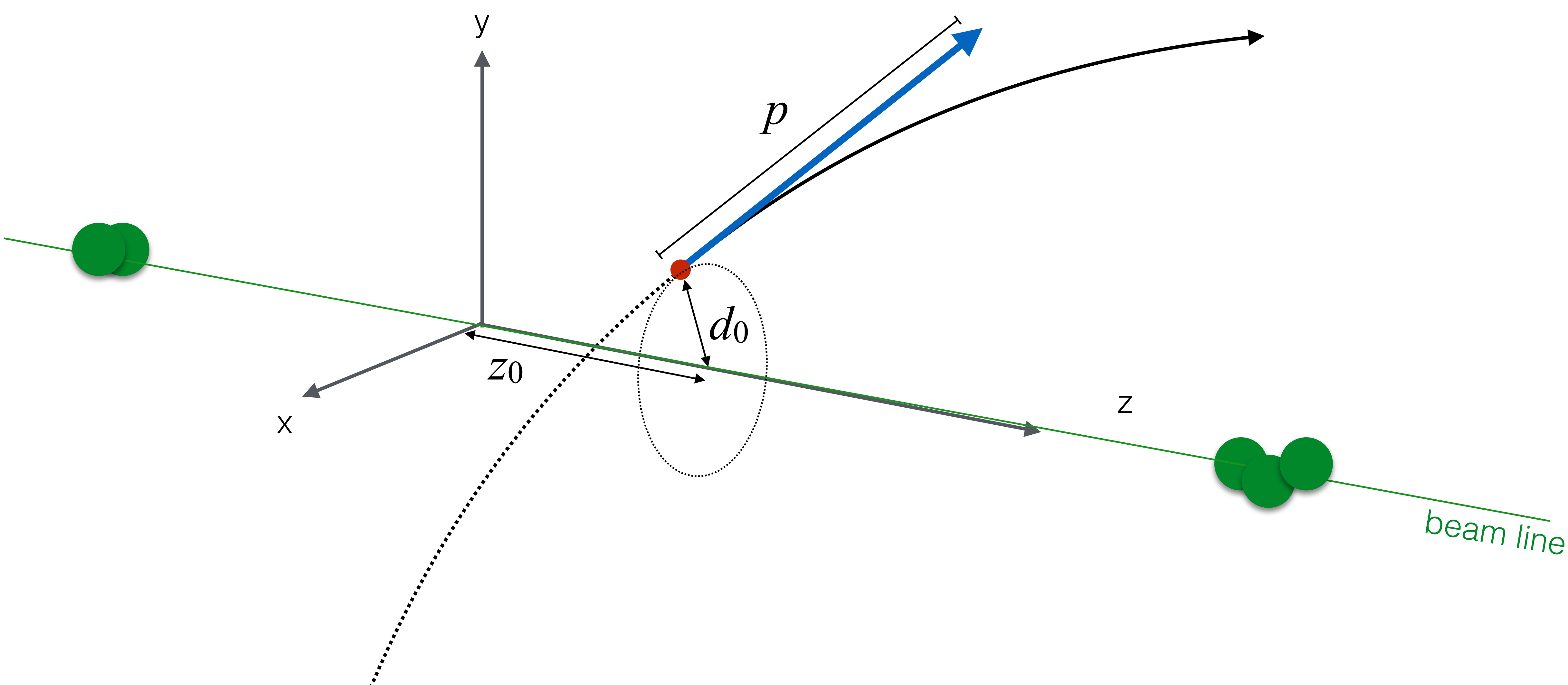
momentum

charge

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

The special one: the Perigee

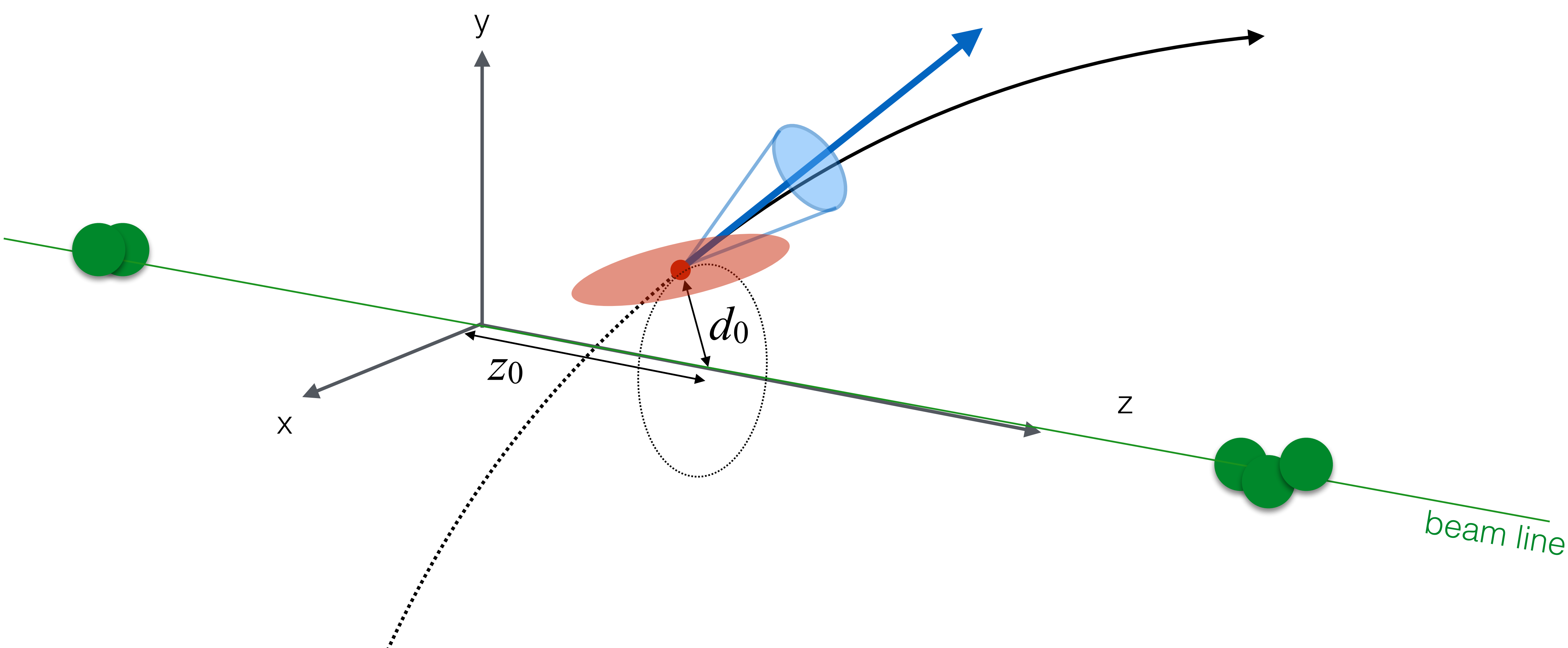
- Perigee representation
  - parameterisation of closest approach to a reference line:
    transverse (d_0) and longitudinal (z_0) impact parameter



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

The special one: the Perigee with uncertainties

- Perigee representation



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

Part 1 - Tracking Detectors

- Track reconstruction in central tracking devices
  - high granular detectors as close as possible to the beam-beam interaction region

    usually hermetic detector design (although dependent on experimental setup)
  - objective is to measure a precise localisation of the charged particle on a certain detection device, e.g.
    - planar detectors, e.g.

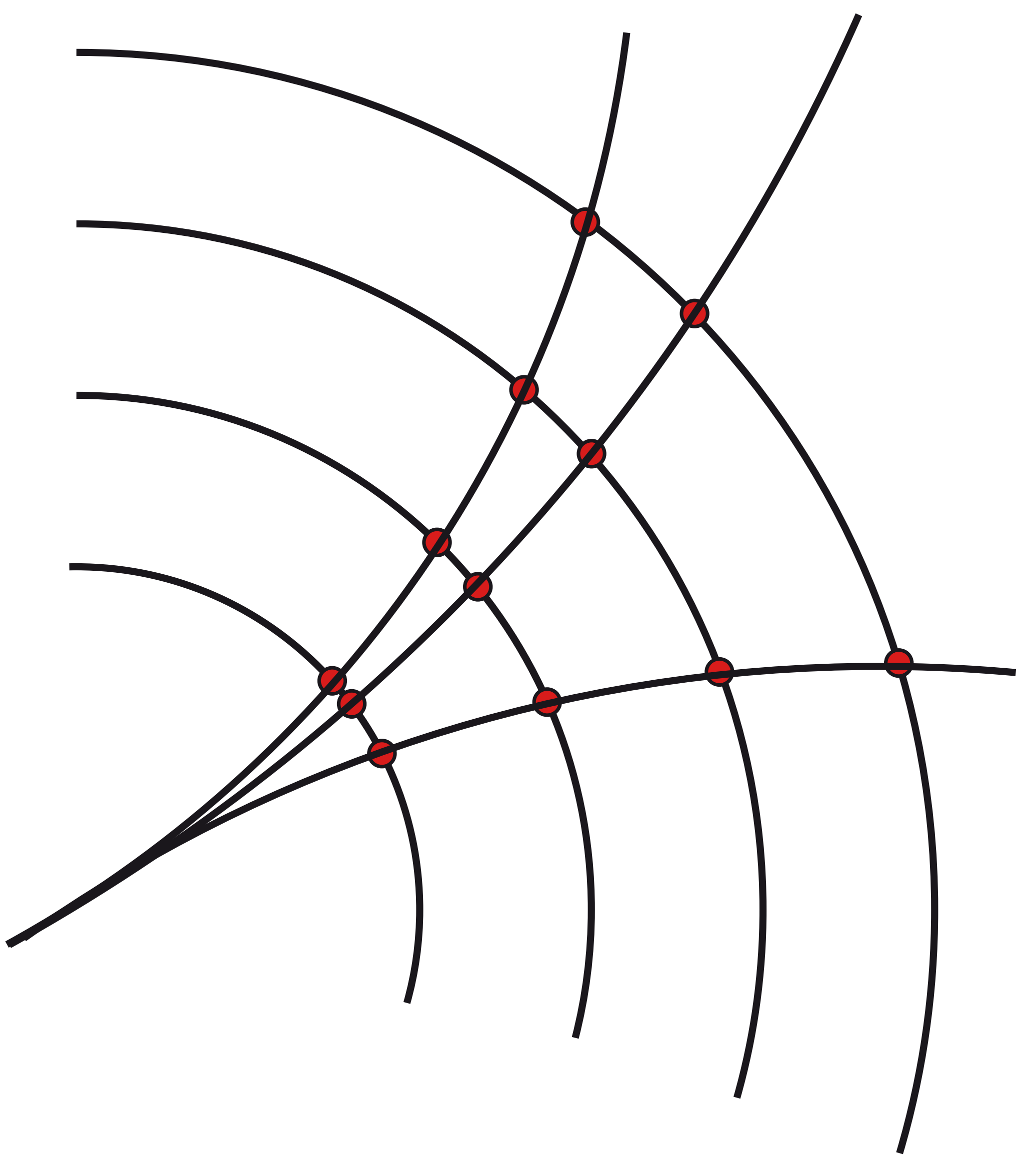
      semiconductor based pixels, strip
    - panar drift detector. e.g. micromegas
    - drift tube detectors
    - time projection chamber (TPC)

Part 1 - Tracking Detectors

- Track reconstruction in central tracking devices
  - high granular detectors as close as possible to the beam-beam interaction region

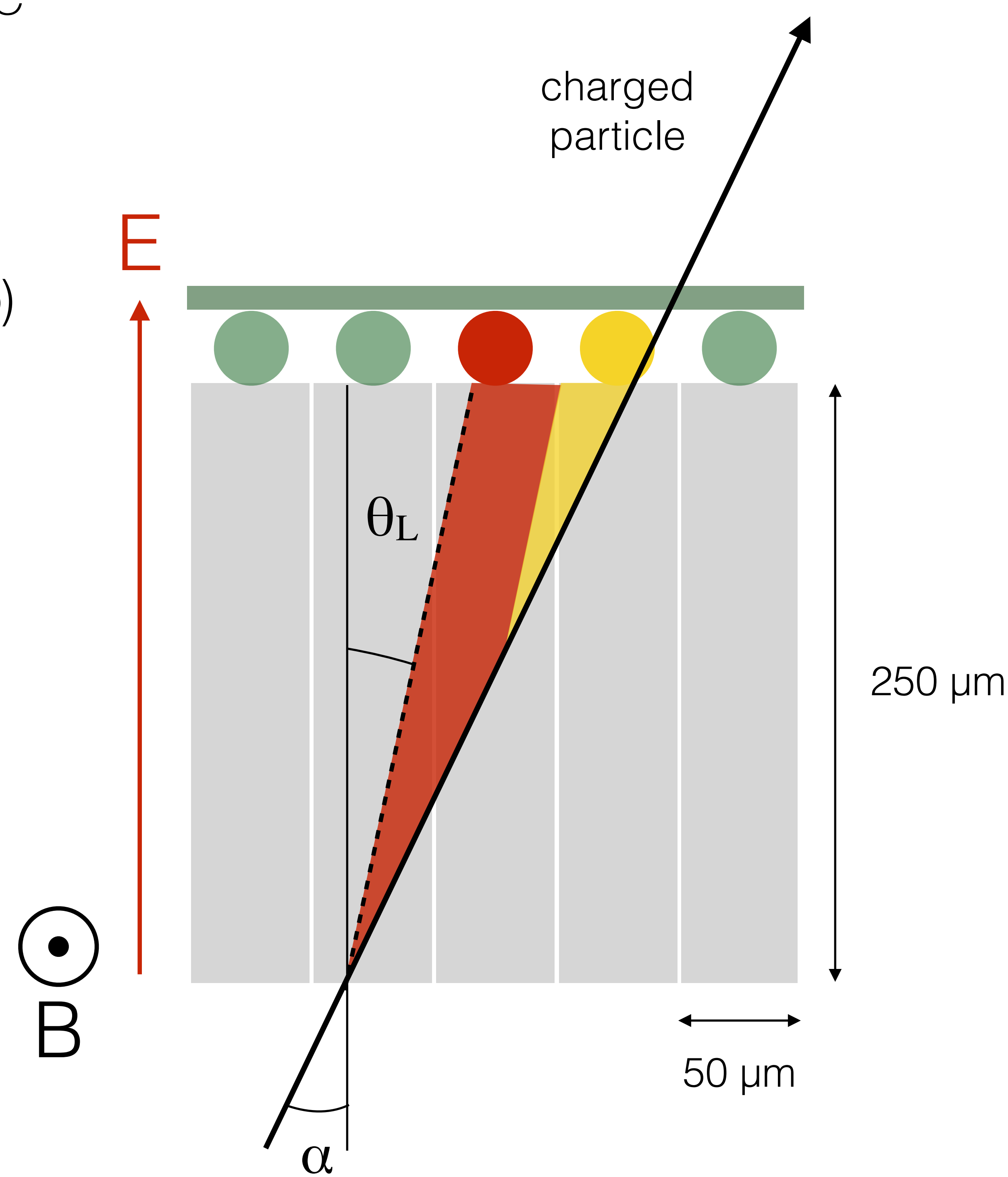
    usually hermetic detector design (although dependent on experimental setup)
  - objective is to measure a precise localisation of the charged particle on a certain detection device, e.g.
    - planar detectors, e.g.

      semiconductor based pixels, strip
    - panar drift detector. e.g. micromegas
    - drift tube detectors
    - time projection chamber (TPC)



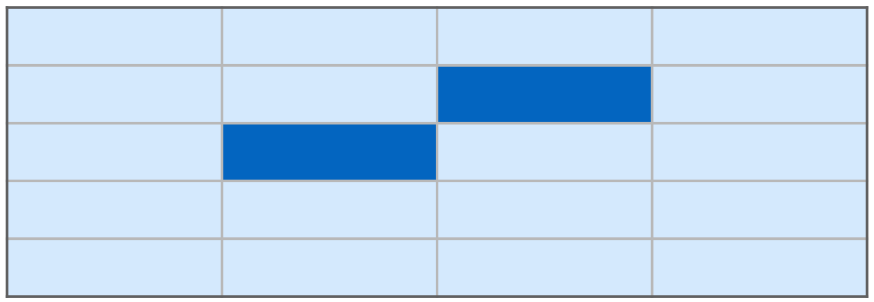
Semiconductor based detectors

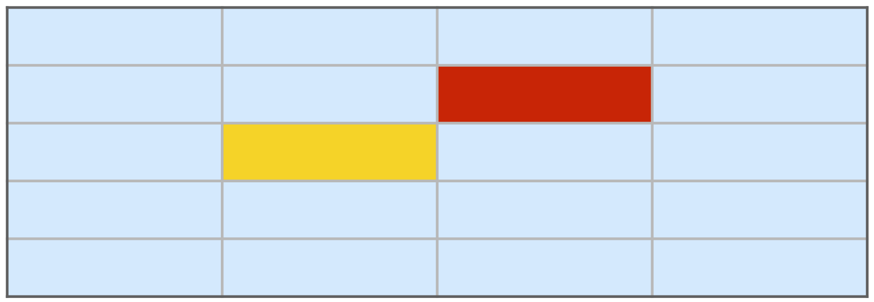
- LHC innermost tracking devices are planar silicon detectors:
  - exist as pixel and strip detectors (they need a local pattern recognition to find clusters of connected pixels/strip)
  - **ionisation** of the silicon through charged particle (primary and secondary ionisation)
  - drift of deposited charge to readout surface using an electric field (**E**)
  - when embedded in magnetic field (**B**), drift deflection by Lorentz angle θ_L



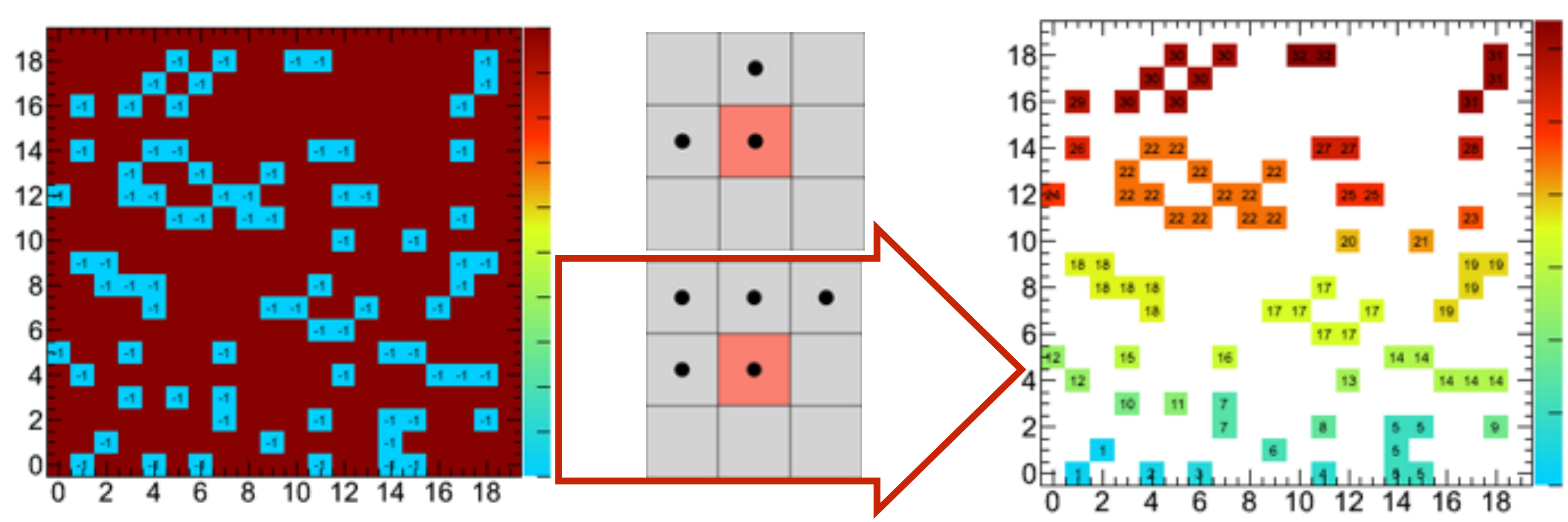
Planar detectors: cluster finding

- LHC innermost tracking devices are planar silicon detectors:
  - either pixel or strip technology with binary (on/off) or non-binary readout (e.g. charge collected by time over readout threshold)



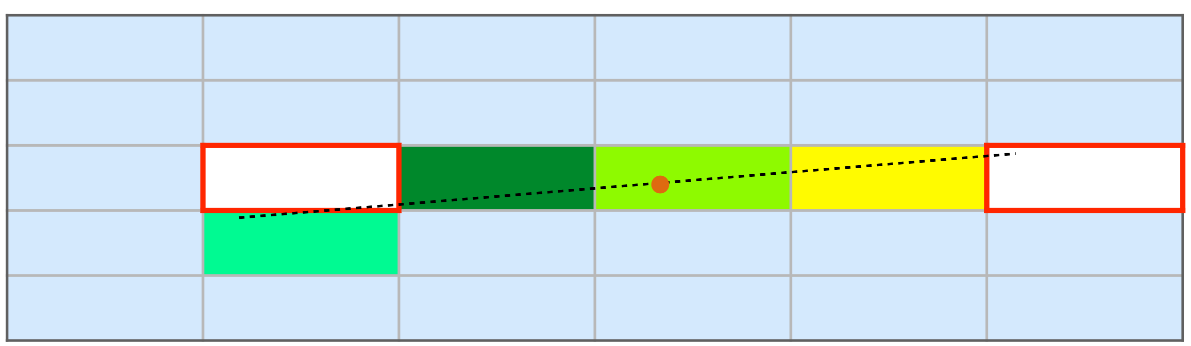


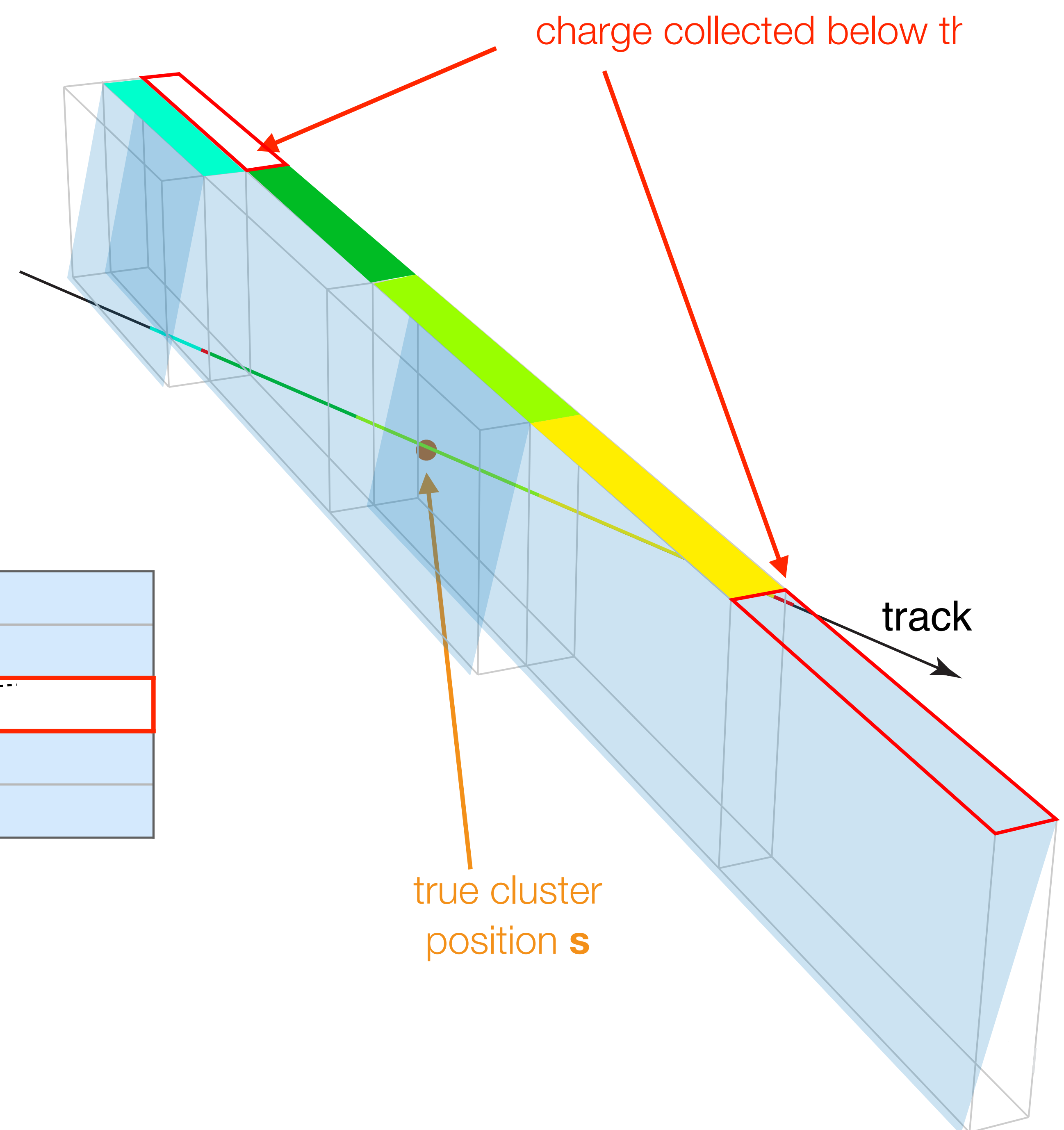
  - more than one pixel/strip can be traversed by one particle: clustering needed usually performed with a *connected component analysis (4-cell, 8-cell connectivity)*
  - example of connected component labelling with 8-cell connectivity:



Planar detectors: cluster building

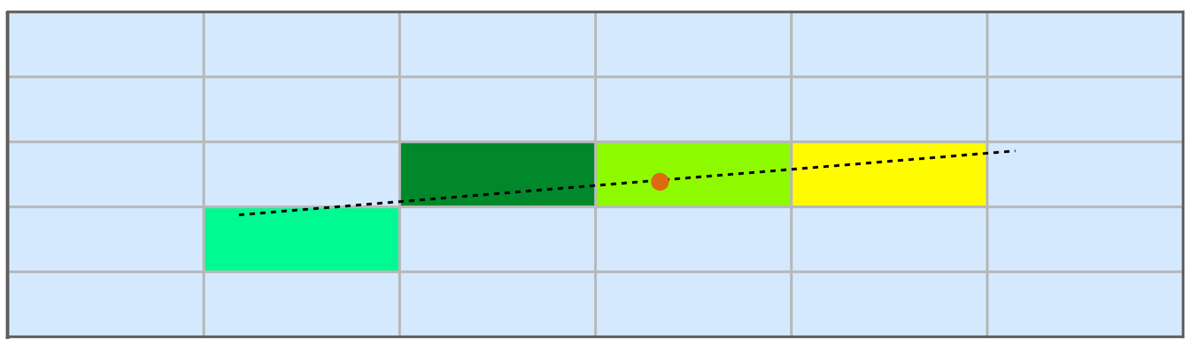
- multiple cells hit can be used to increase measurement precision

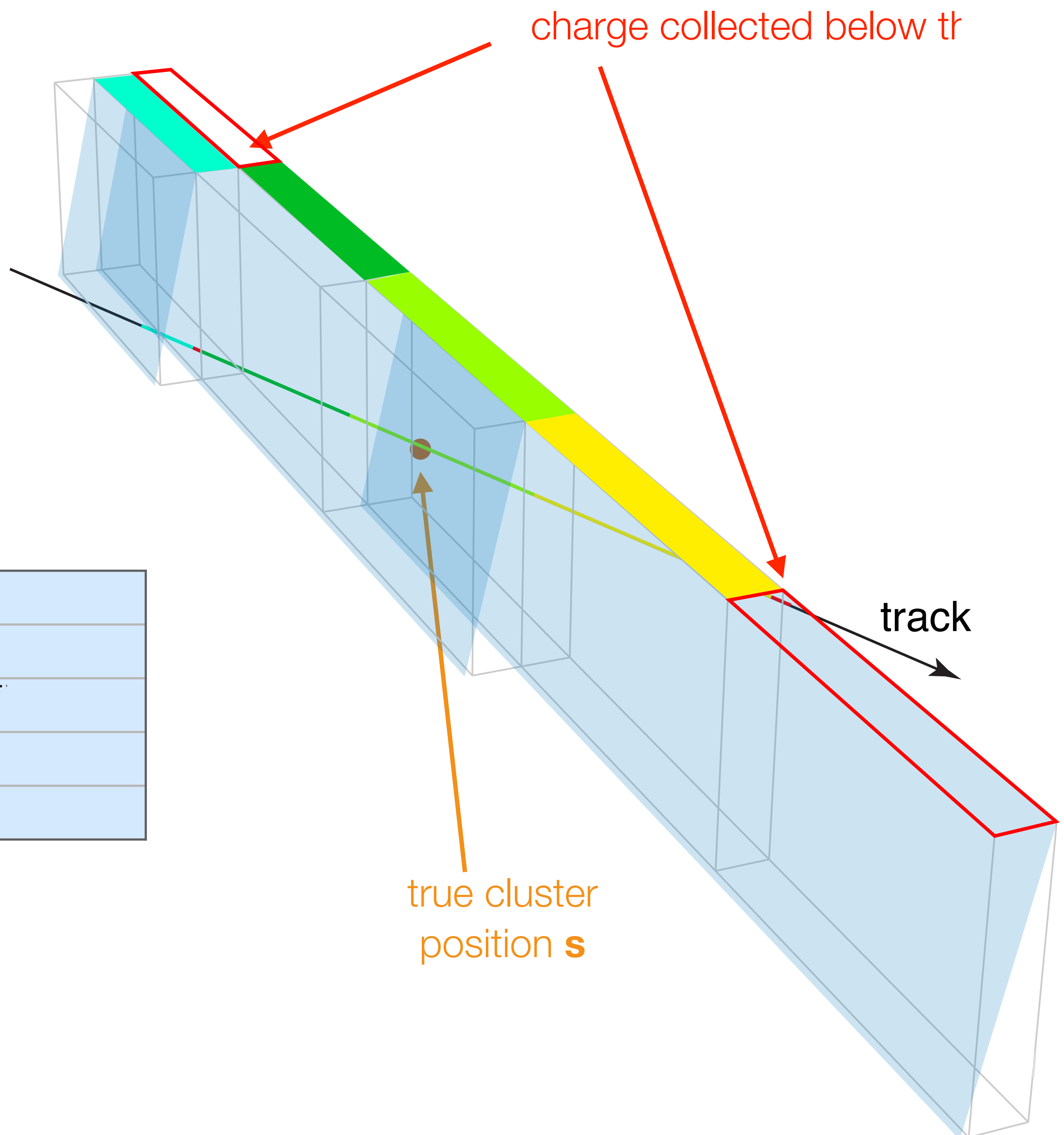




Planar detectors: cluster building

- multiple cells hit can be used to increase measurement precision





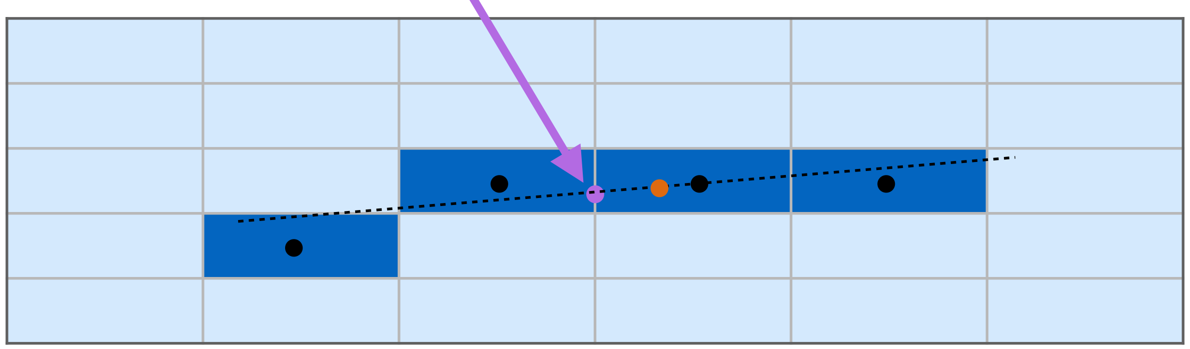
Planar detectors: cluster building

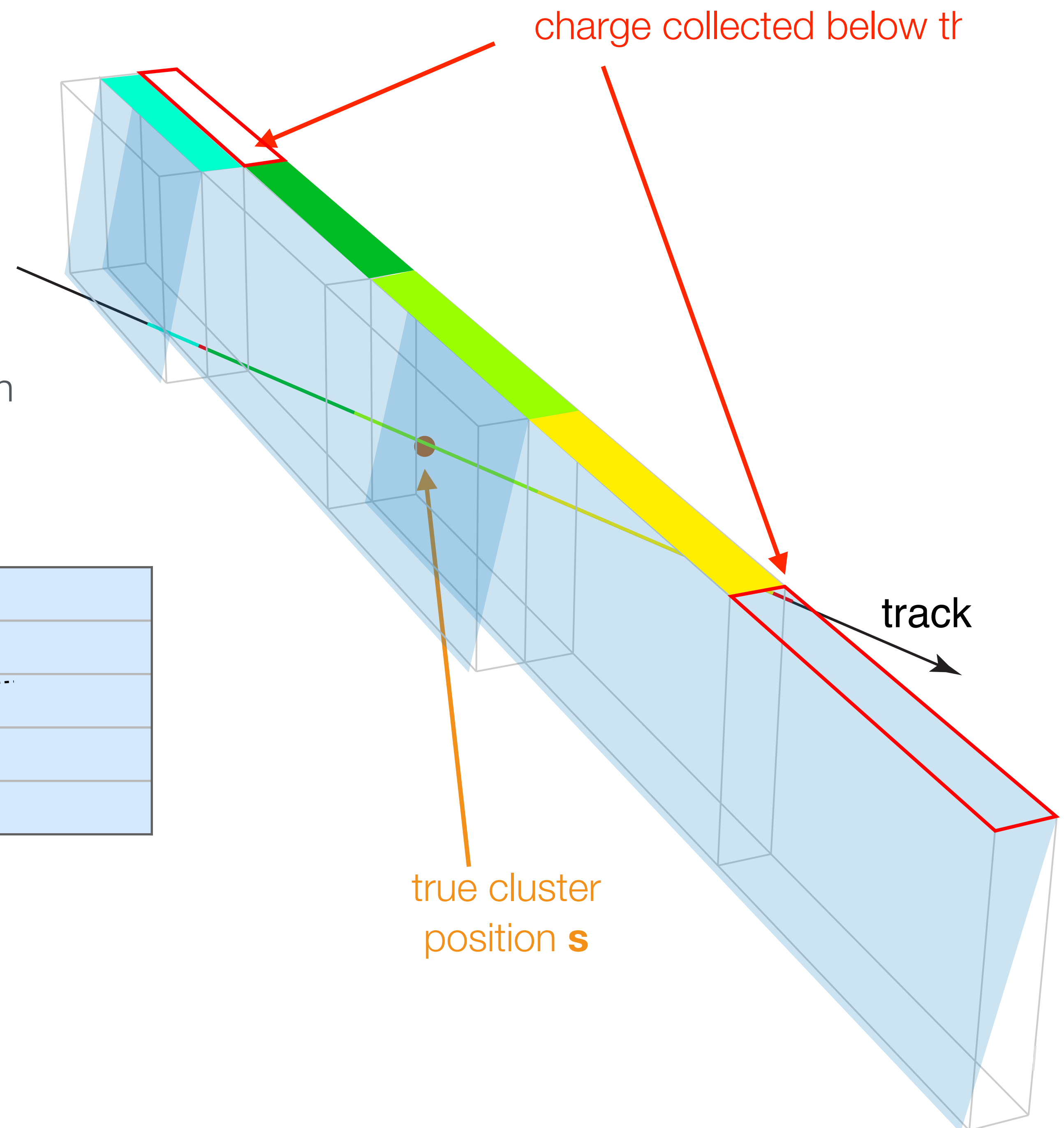
- multiple cells hit can be used to increase measurement precision

the binary approach:

i-th pixel position

measurement $\boxed{\mathbf{m}} = \frac{1}{N} \sum_{i=1,N} \boxed{l_i}$





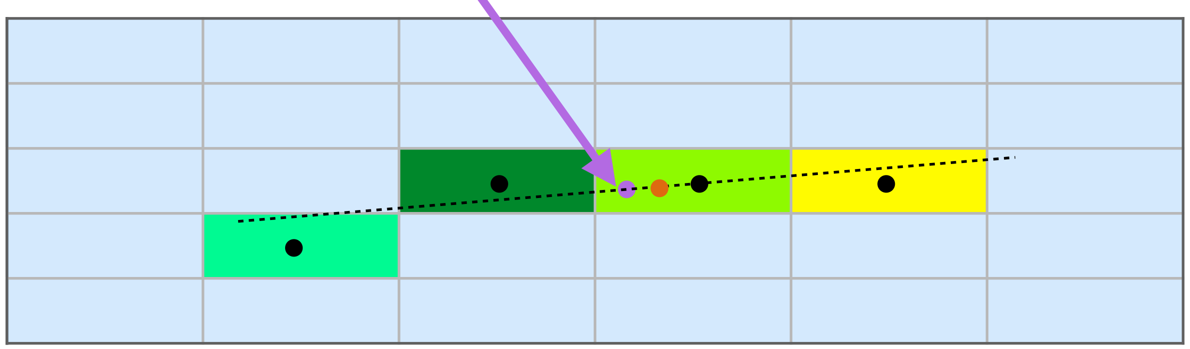
Planar detectors: cluster building

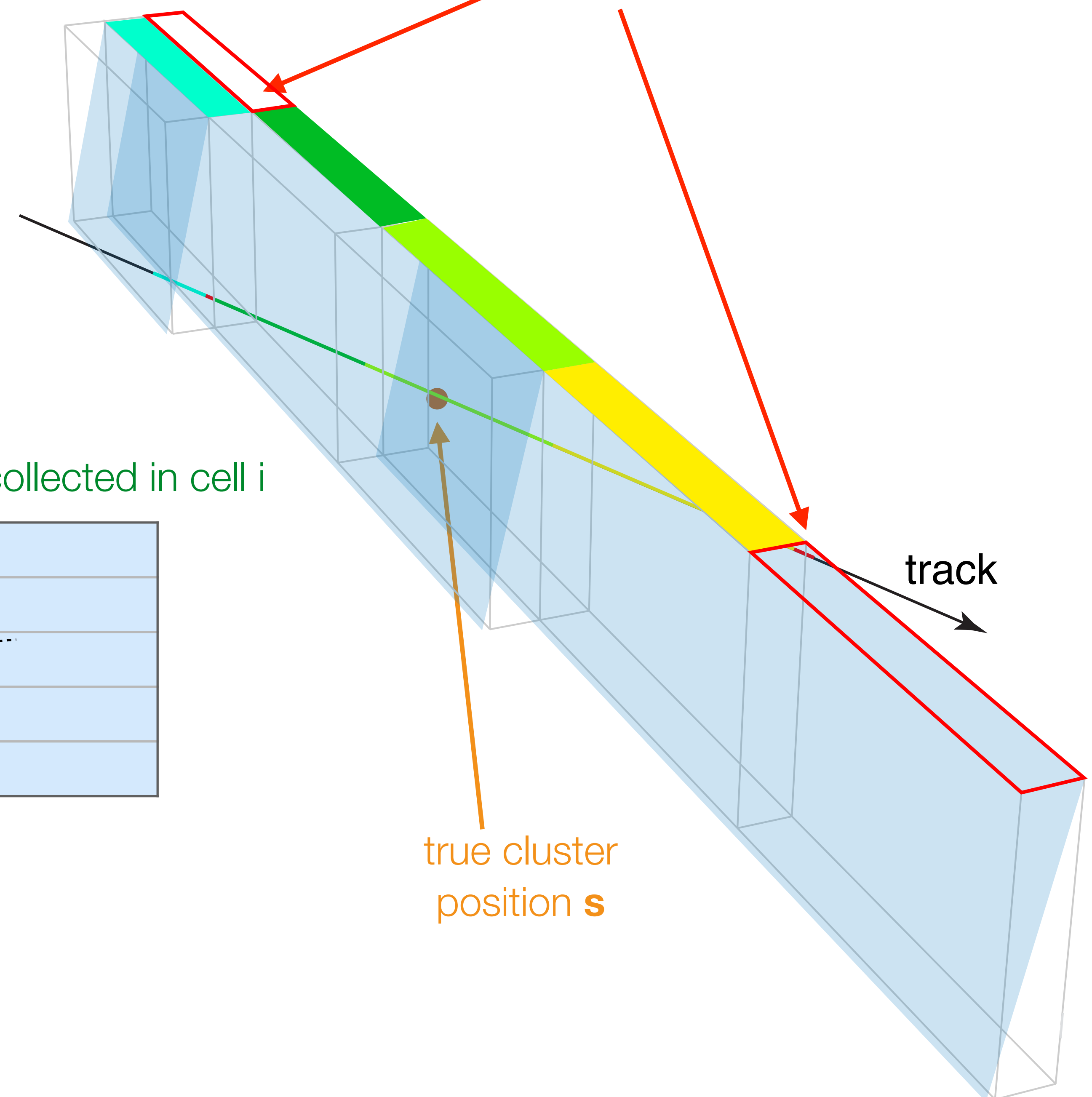
- multiple cells hit can be used to increase measurement precision

the charge-weighted approach :

$$\boxed{\mathbf{m}} = \frac{1}{\sum_{i=1,N} q_i} \sum_{i=1,N} \boxed{q_i} l_i$$

charge collected in cell i





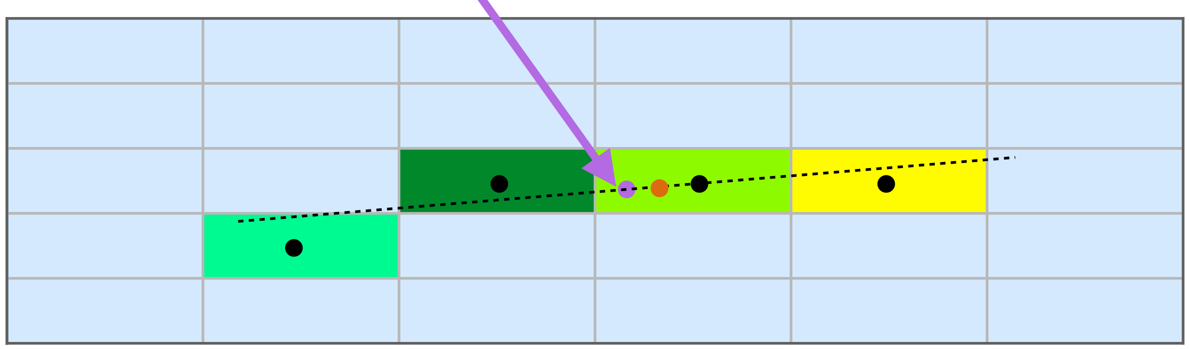
Planar detectors: cluster building

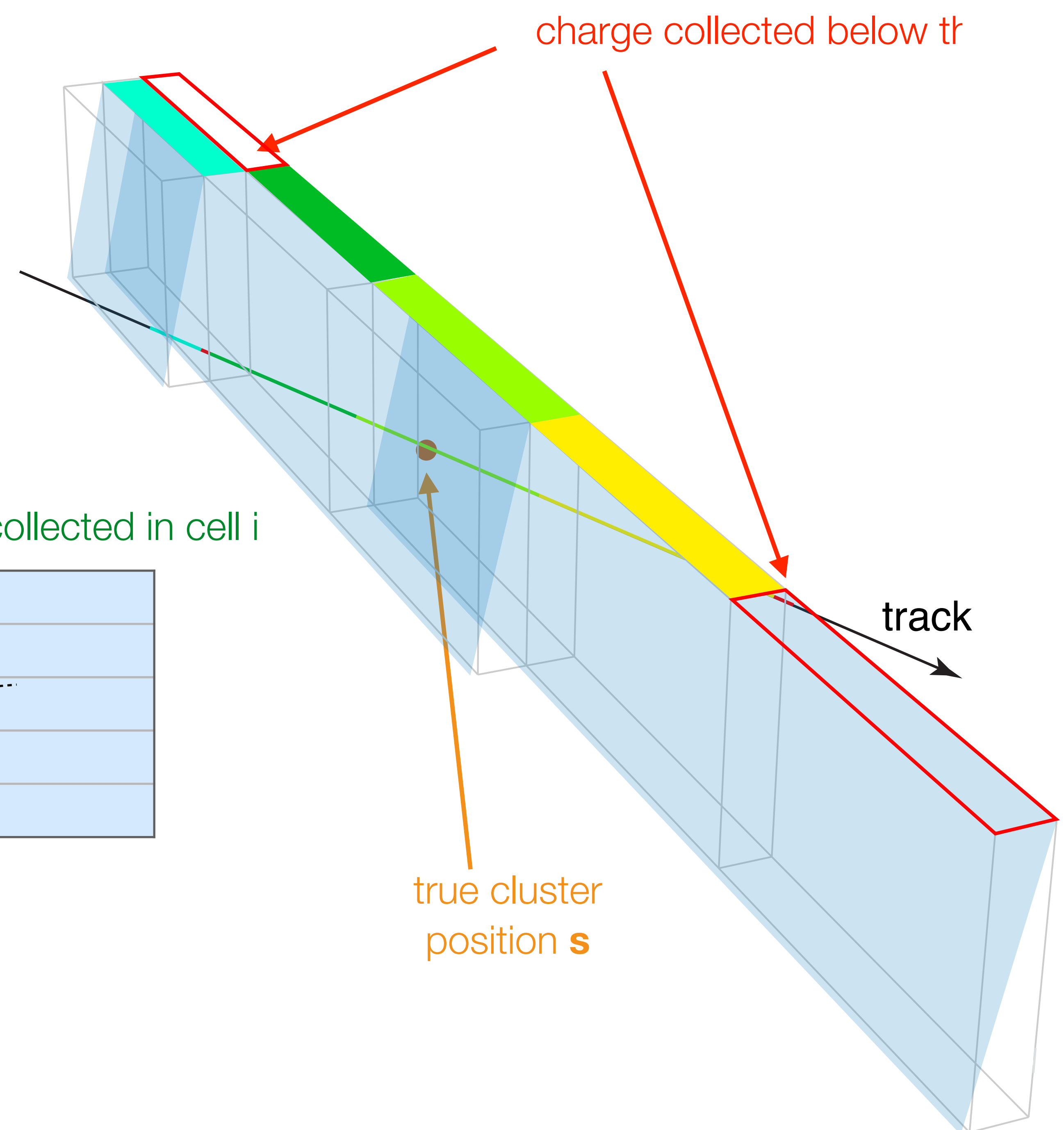
- multiple cells hit can be used to increase measurement precision

the charge-weighted approach :

$$\boxed{\mathbf{m}} = \frac{1}{\sum_{i=1,N} q_i} \sum_{i=1,N} \boxed{q_i} l_i$$

charge collected in cell i



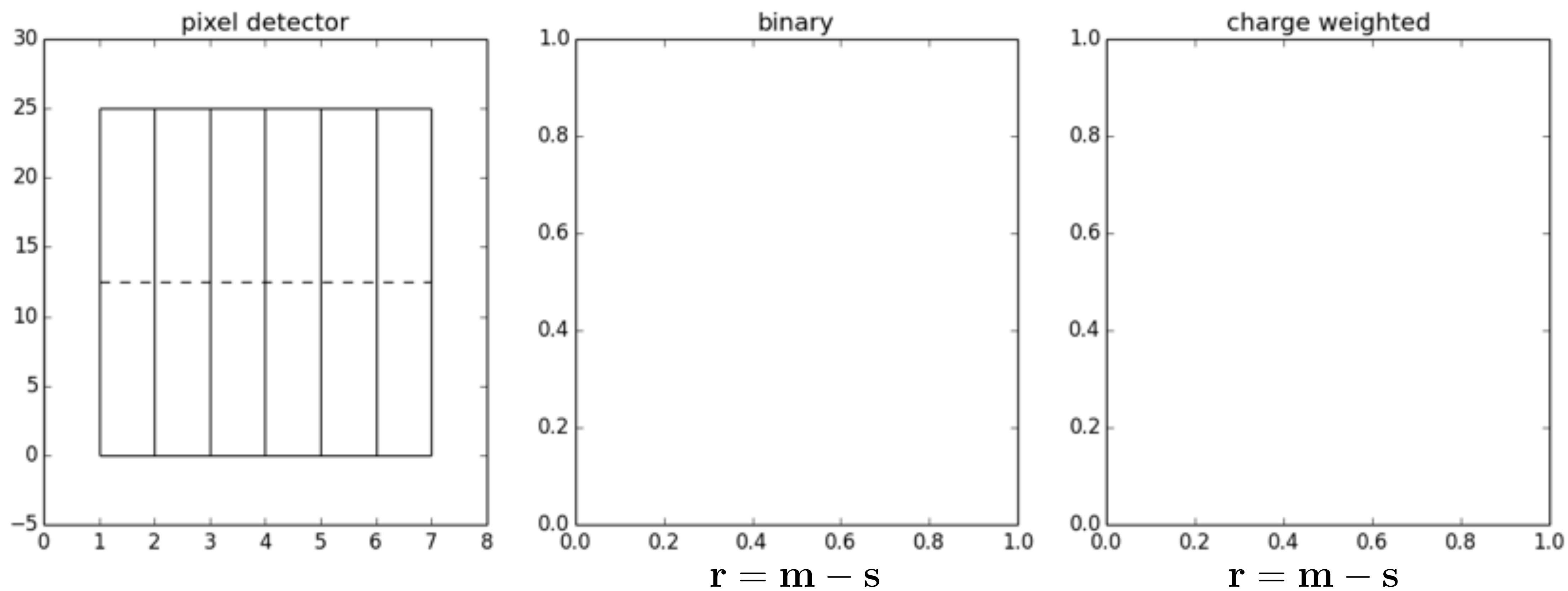


which one is better ?

let's measure it using the residuum

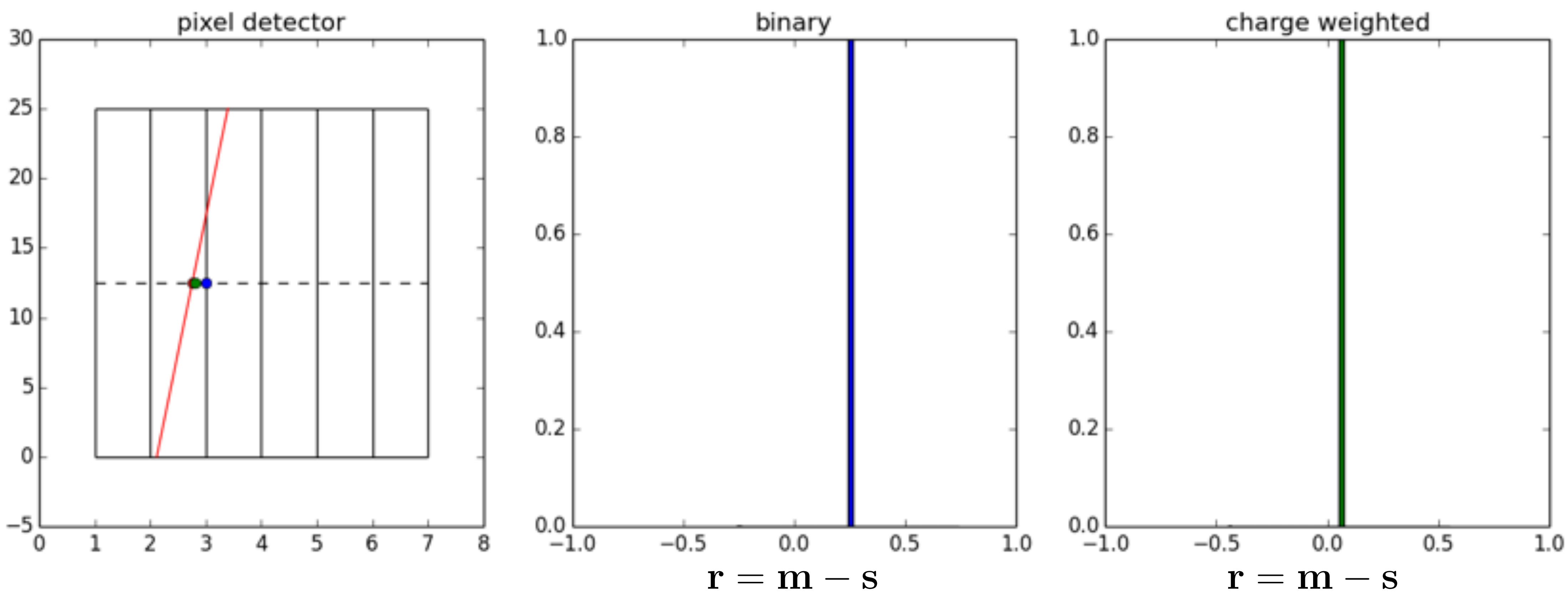
$$\mathbf{r} = \mathbf{m} - \mathbf{s}$$

MC Toy: pixel clustering



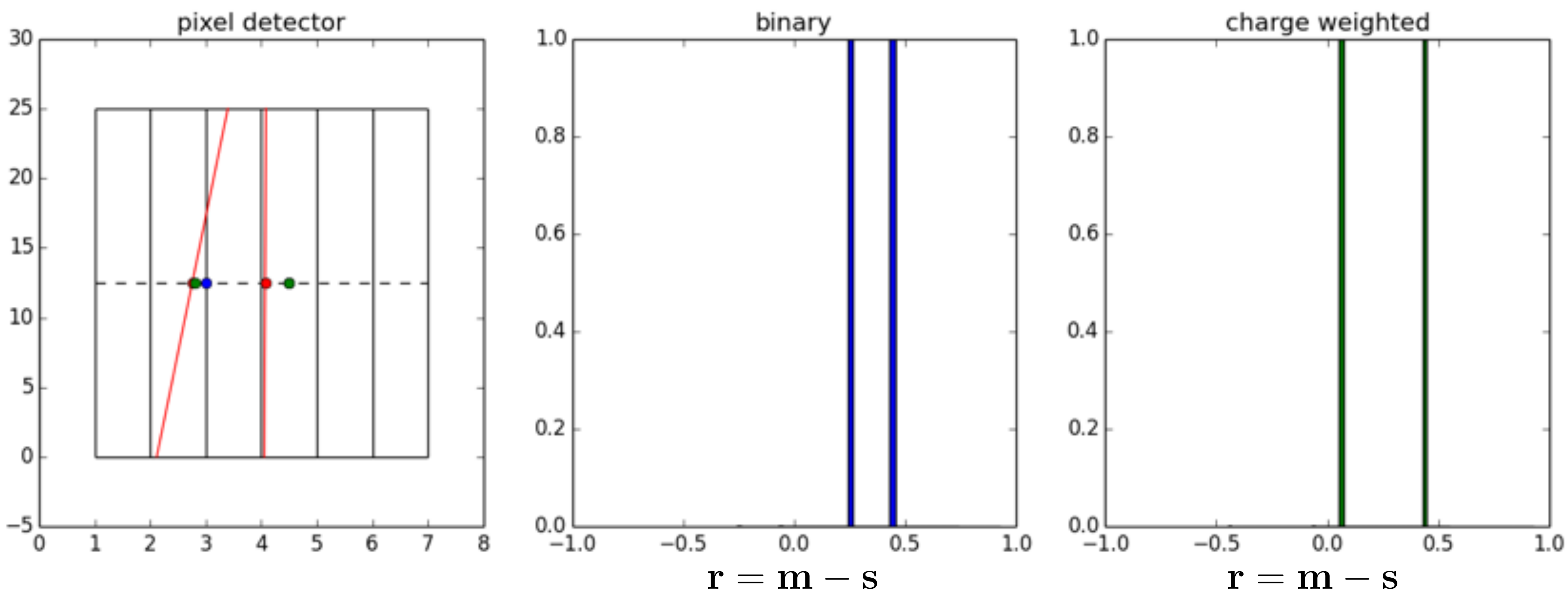

```
salzburg$ ipython -i --matplotlib=osx PixelClustering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 1000 )
```

MC Toy: pixel clustering



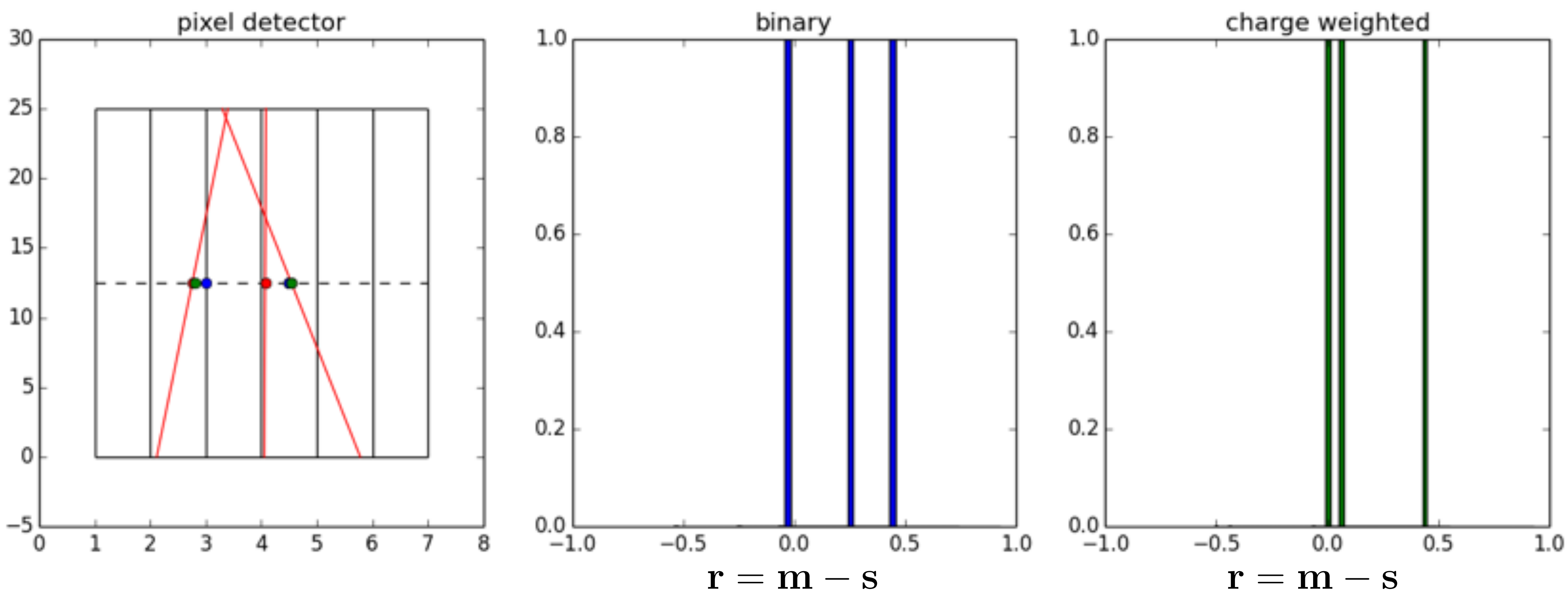

```
salzburg$ ipython -i --matplotlib=osx PixelClustering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 1000 )
```

MC Toy: pixel clustering



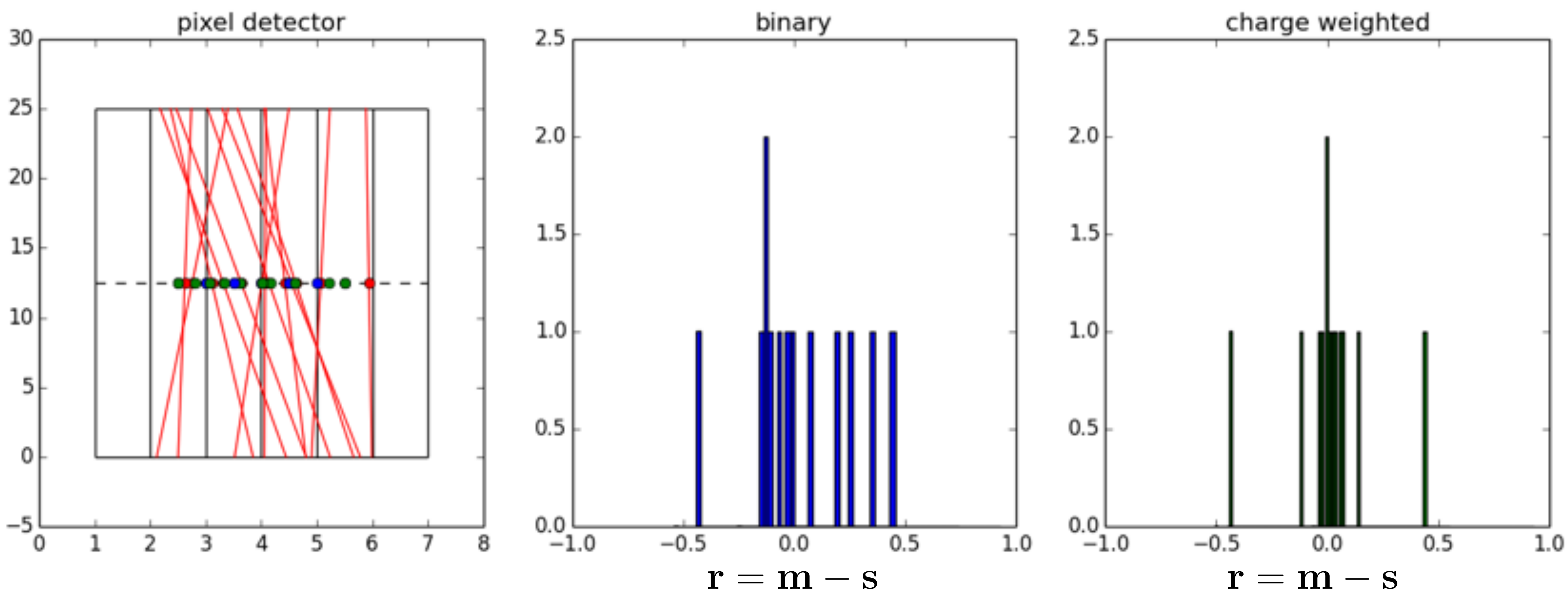

```
salzburg$ ipython -i --matplotlib=osx PixelClustering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 1000 )
```

MC Toy: pixel clustering



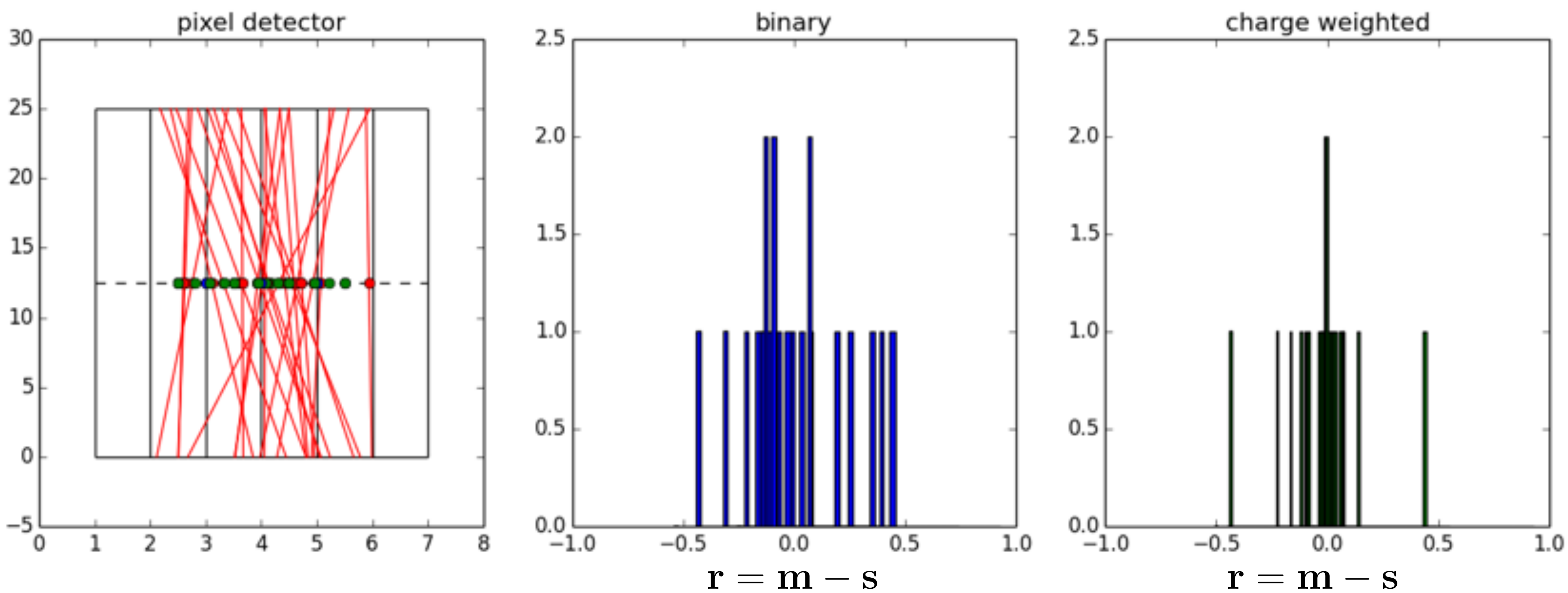

```
salzburg$ ipython -i --matplotlib=osx PixelClustering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 1000 )
```

MC Toy: pixel clustering



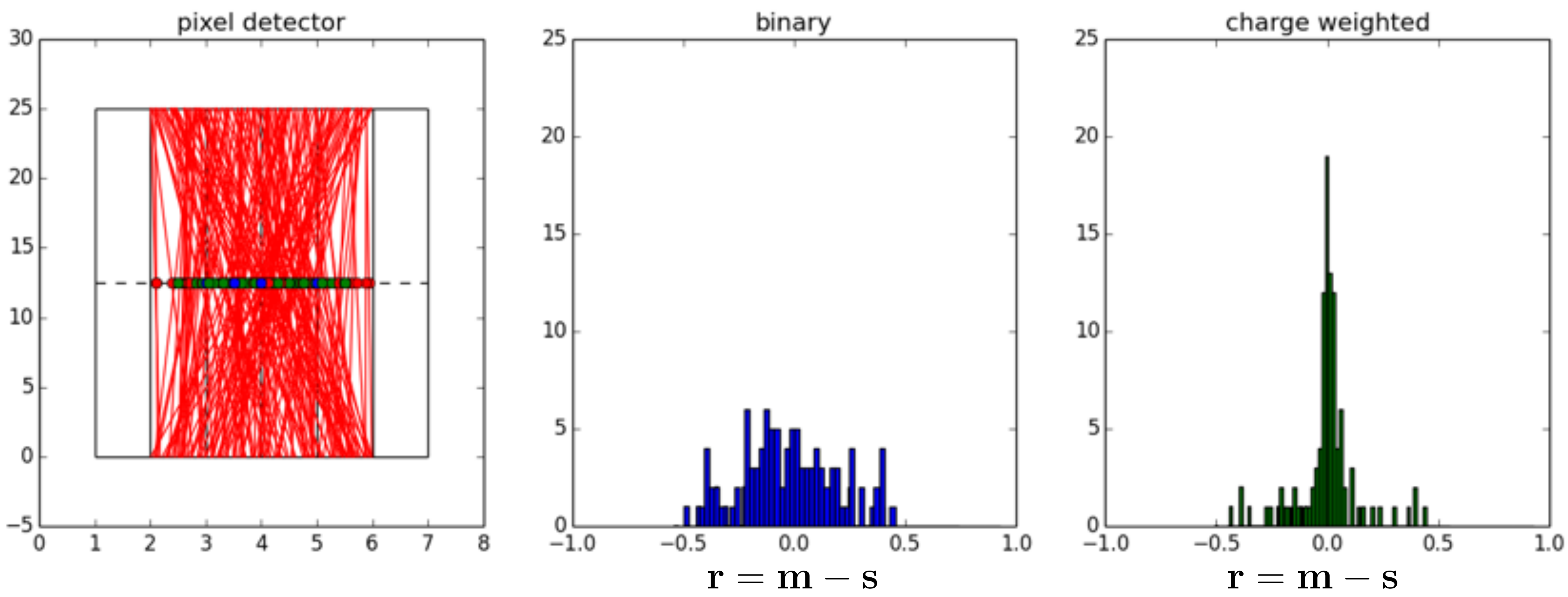

```
salzburg$ ipython -i --matplotlib=osx PixelClustering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 1000 )
```

MC Toy: pixel clustering



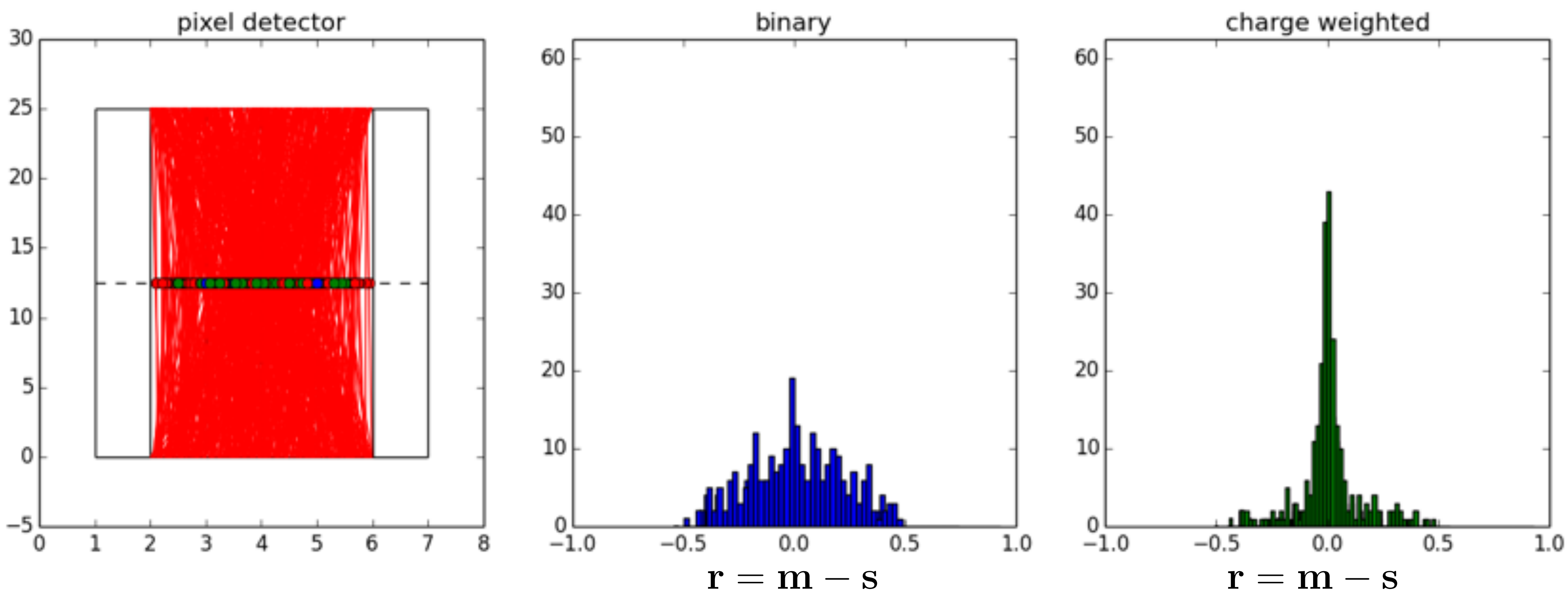

```
salzburg$ ipython -i --matplotlib=osx PixelClustering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 1000 )
```

MC Toy: pixel clustering



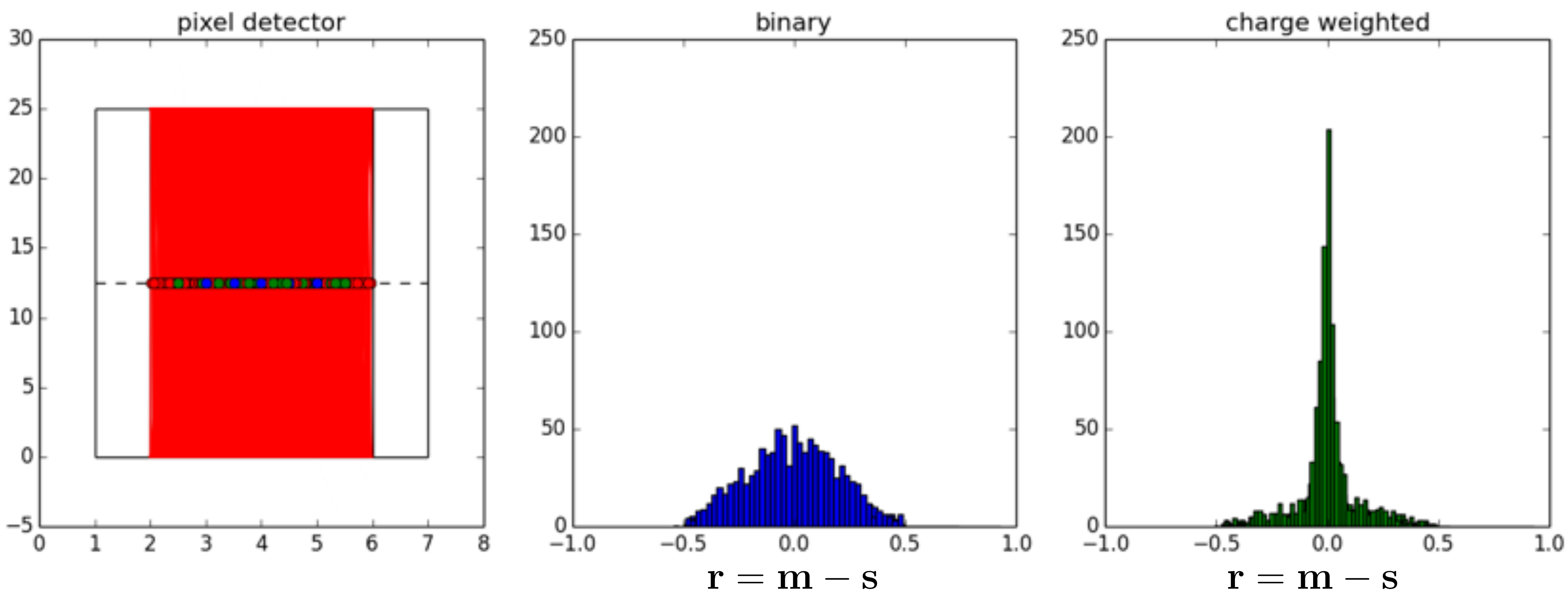

```
salzburg$ ipython -i --matplotlib=osx PixelClustering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 1000 )
```

MC Toy: pixel clustering



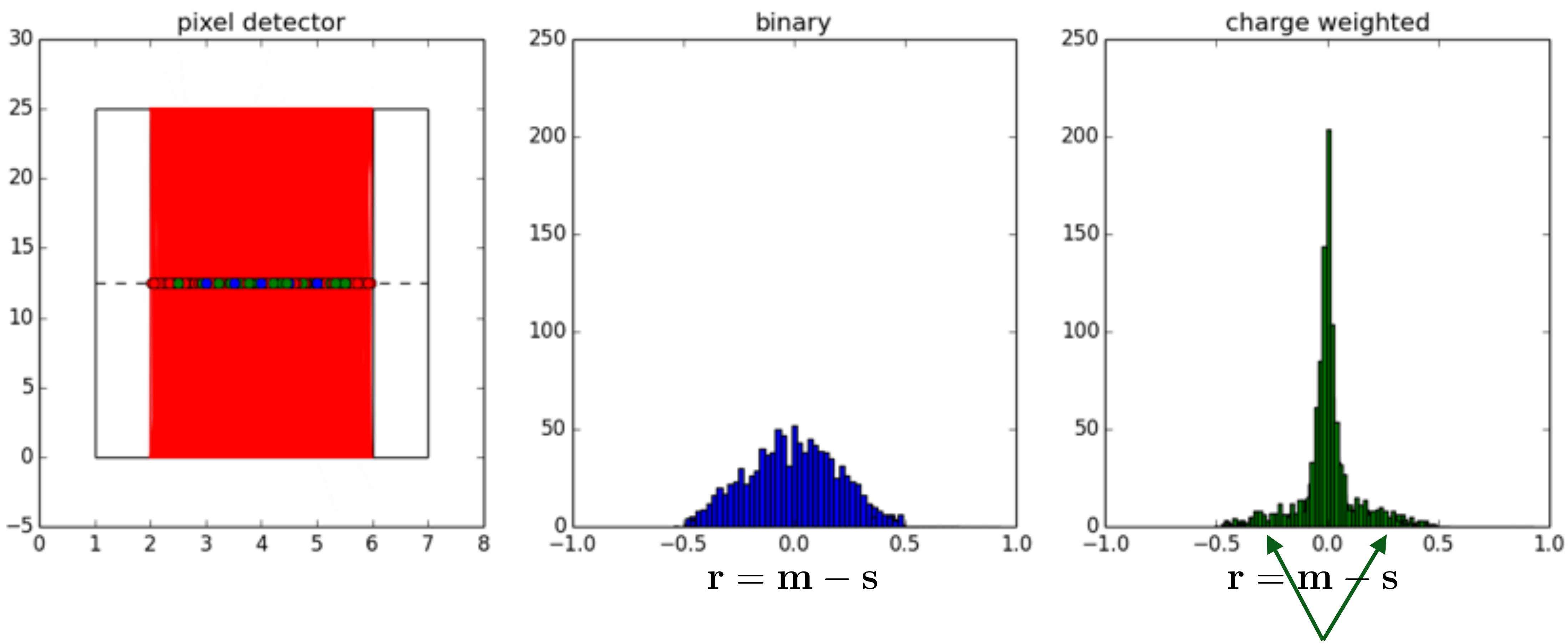

```
salzburg$ ipython -i --matplotlib=osx PixelClustering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 1000 )
```

MC Toy: pixel clustering




```
salzburg$ ipython -i --matplotlib=osx PixelClustering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 1000 )
```

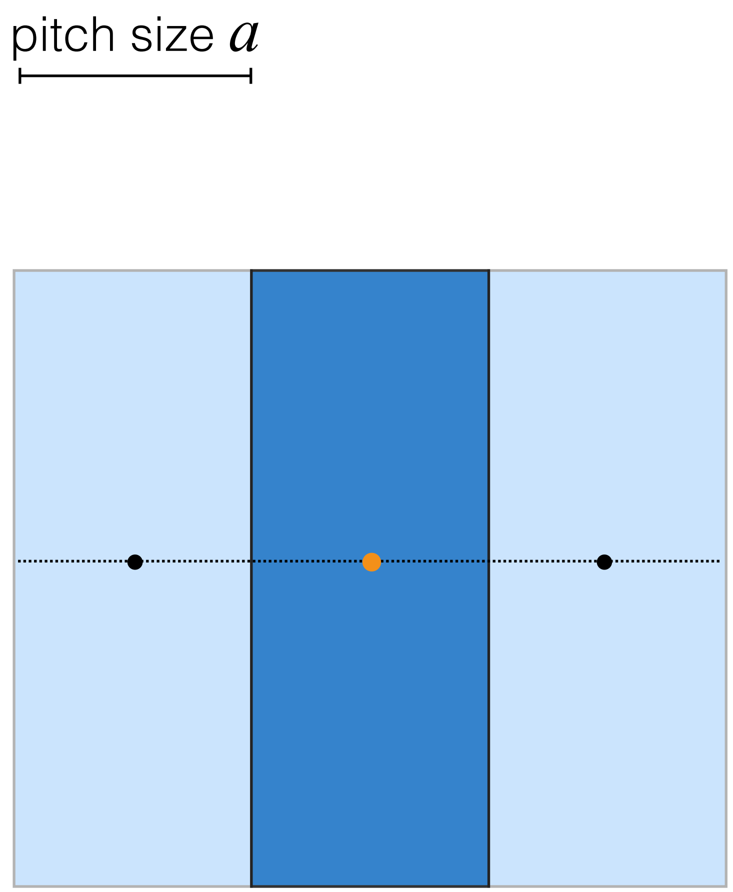
MC Toy: pixel clustering




```
salzburg$ ipython -i --matplotlib=osx PixelClustering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 1000 )
```

MC Toy: pixel clustering

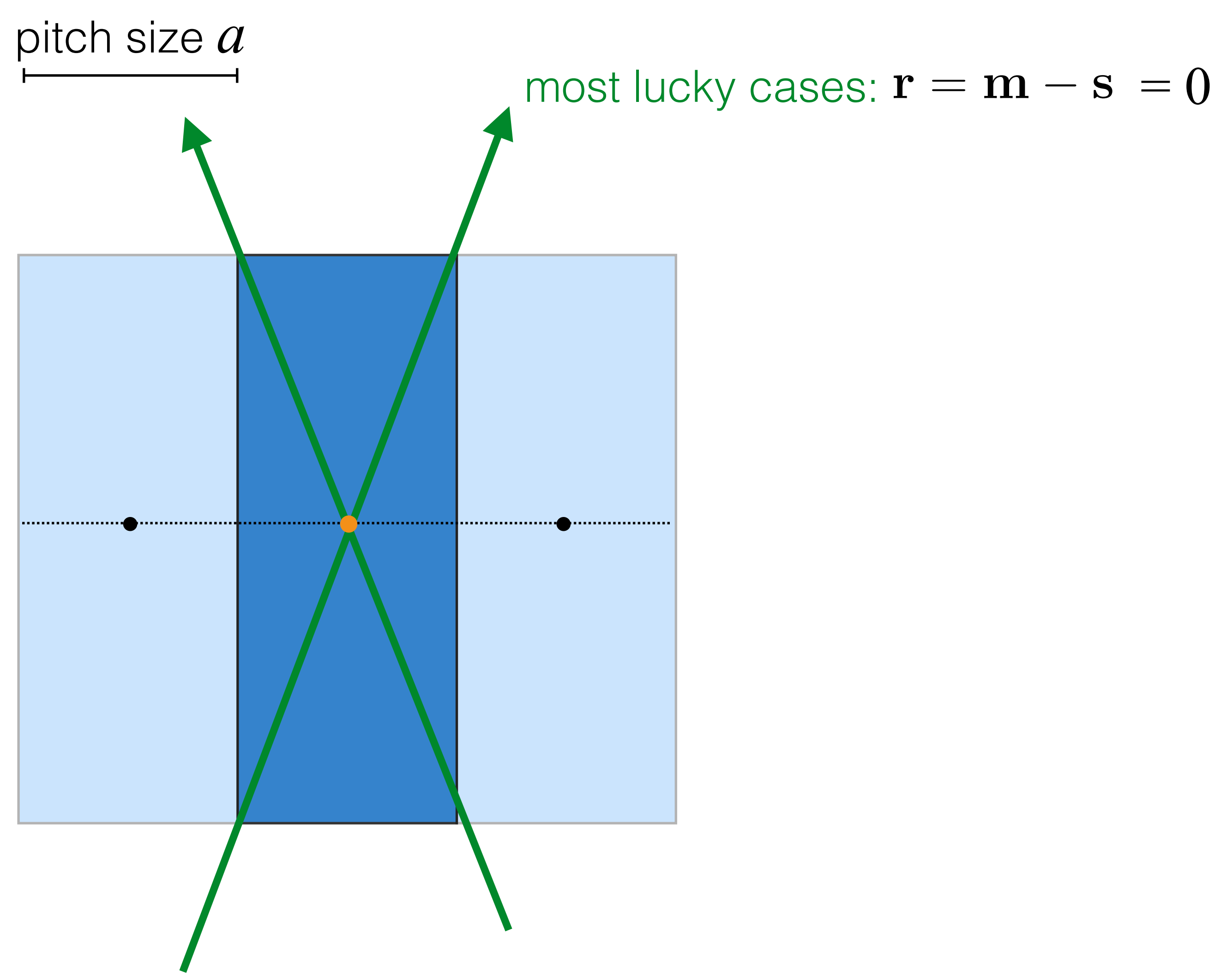
- what can we say about the uncertainty of the measurement ?



binary case, single pixel cluster

MC Toy: pixel clustering

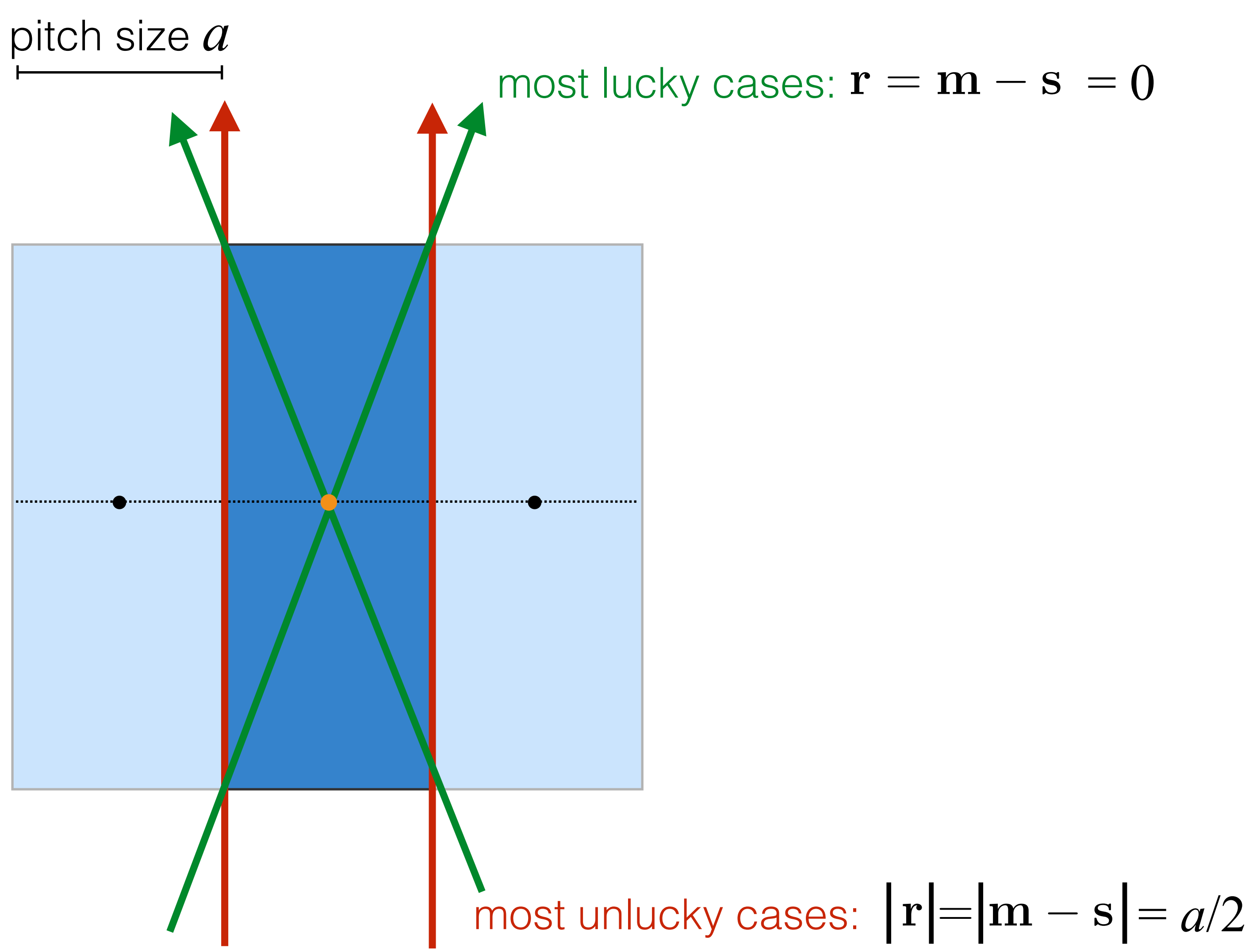
- what can we say about the uncertainty of the measurement ?



binary case, single pixel cluster

MC Toy: pixel clustering

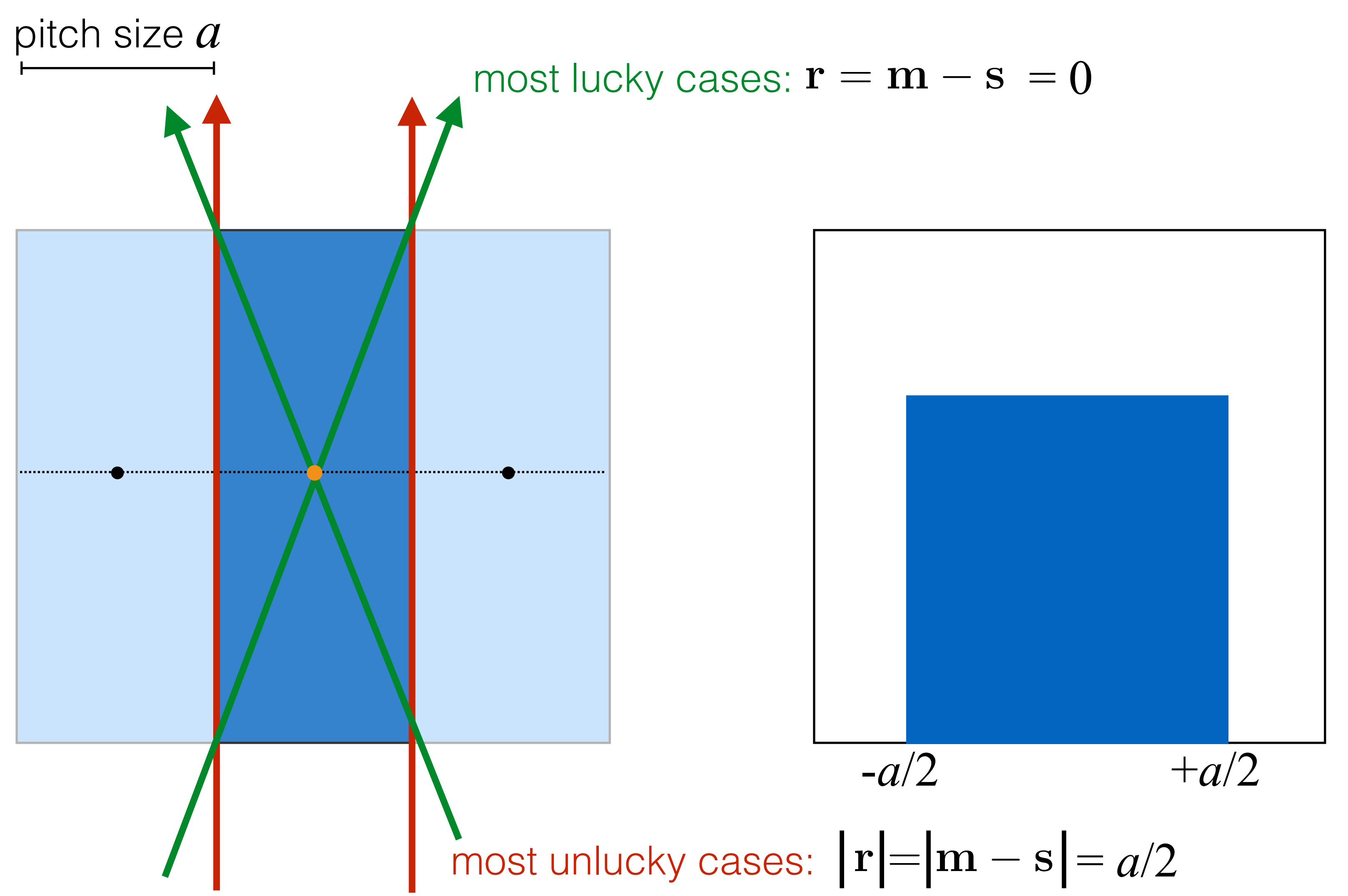
- what can we say about the uncertainty of the measurement ?



binary case, single pixel cluster

MC Toy: pixel clustering

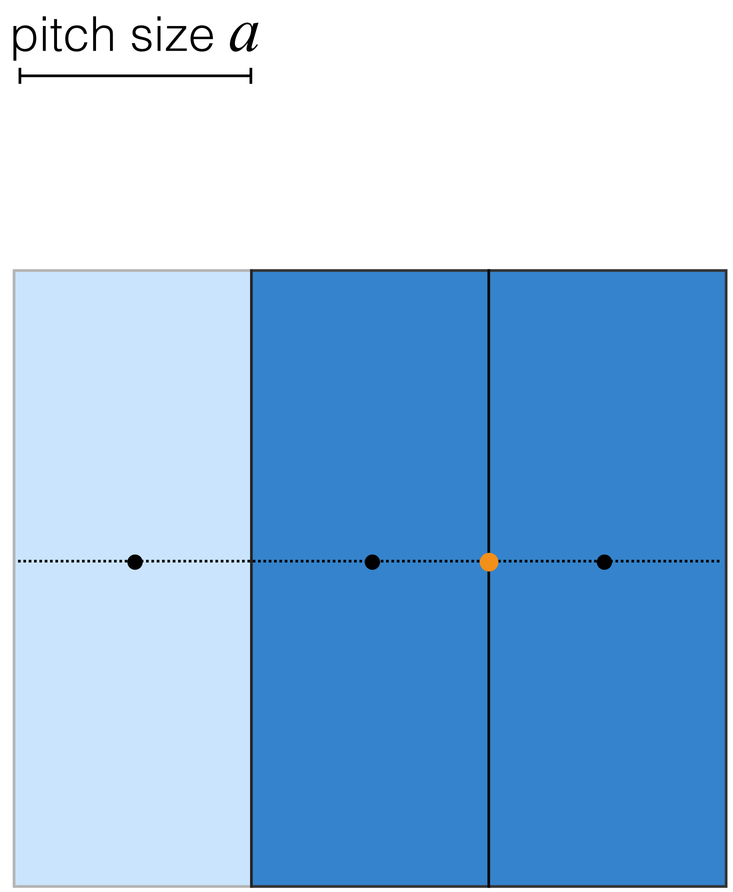
- what can we say about the uncertainty of the measurement ?



binary case, single pixel cluster

MC Toy: pixel clustering

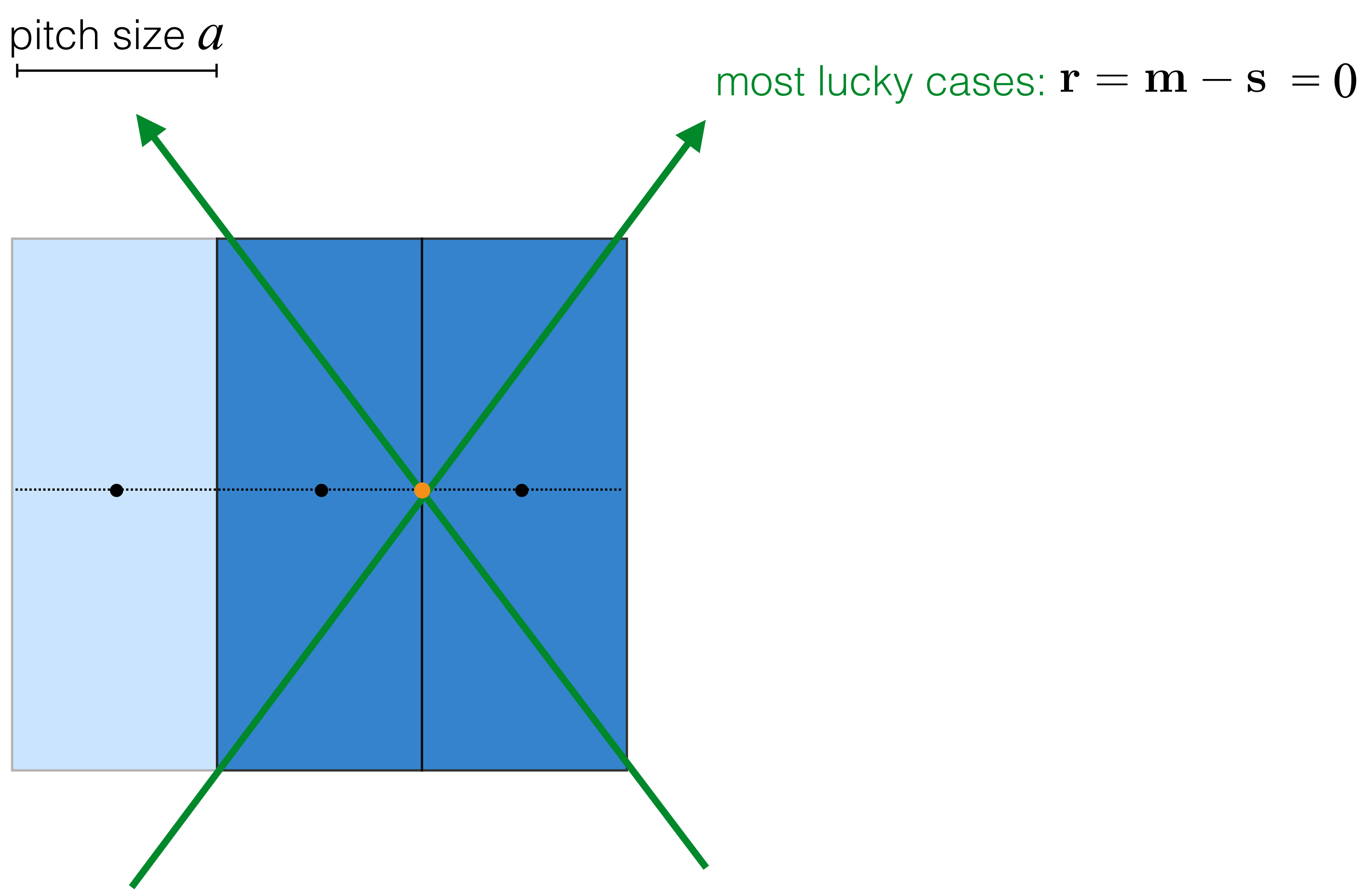
- what can we say about the uncertainty of the measurement ?



binary case, 2-pixel cluster

MC Toy: pixel clustering

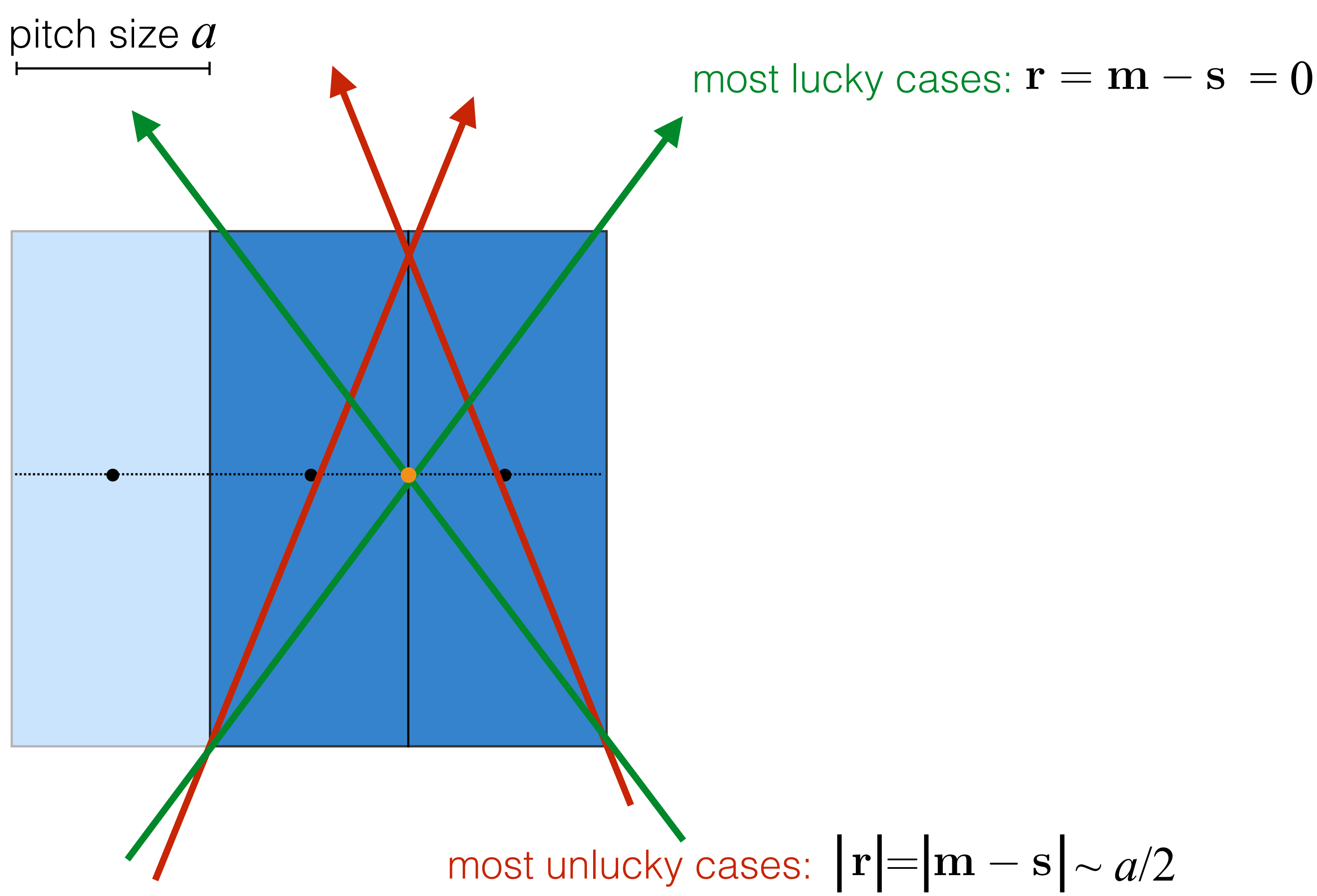
- what can we say about the uncertainty of the measurement ?



binary case, 2-pixel cluster

MC Toy: pixel clustering

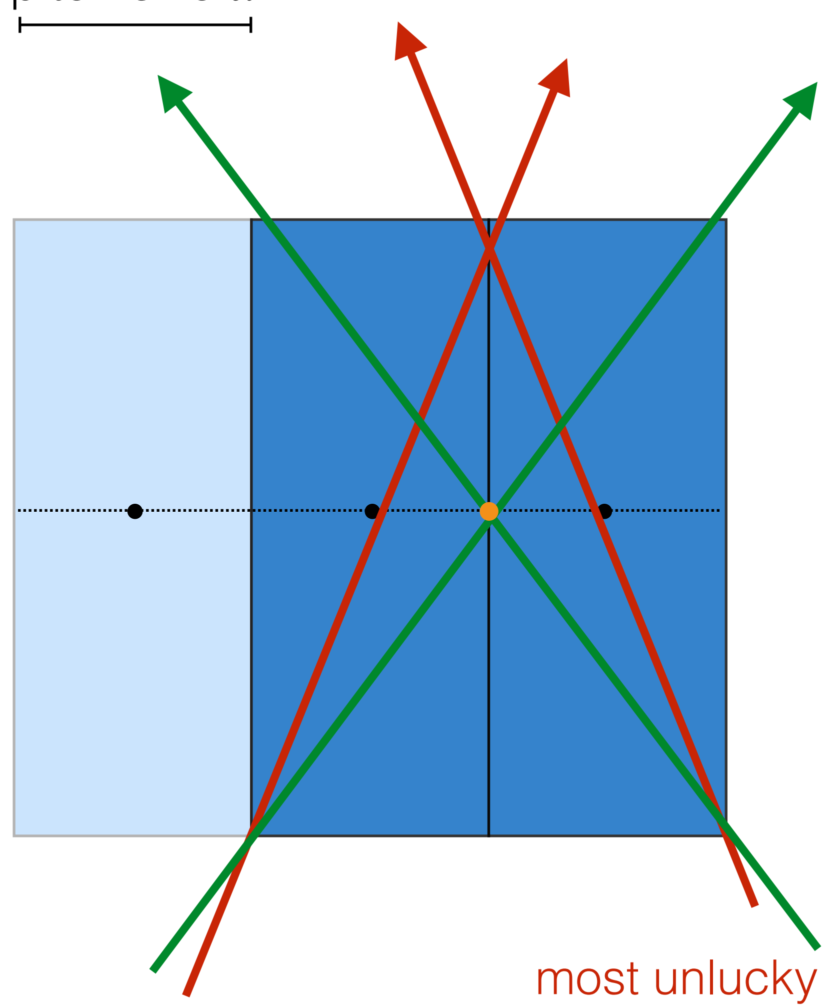
- what can we say about the uncertainty of the measurement ?

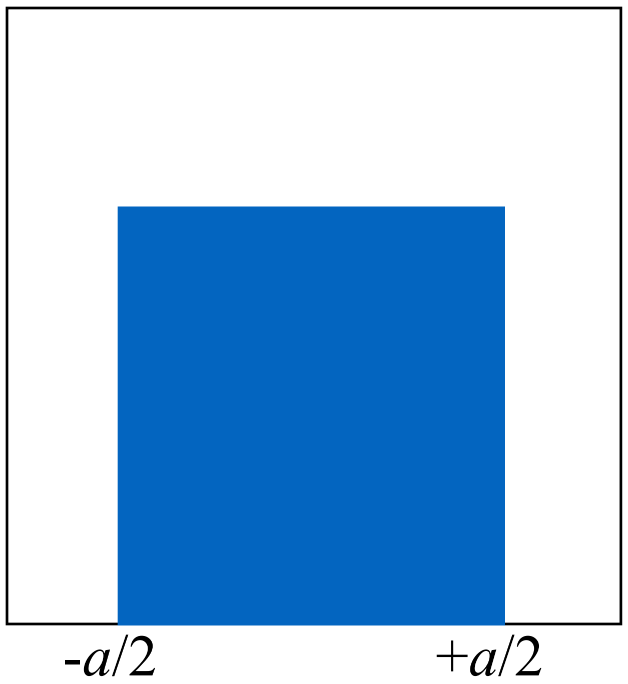


binary case, 2-pixel cluster

MC Toy: pixel clustering

- what can we say about the uncertainty of the measurement ?

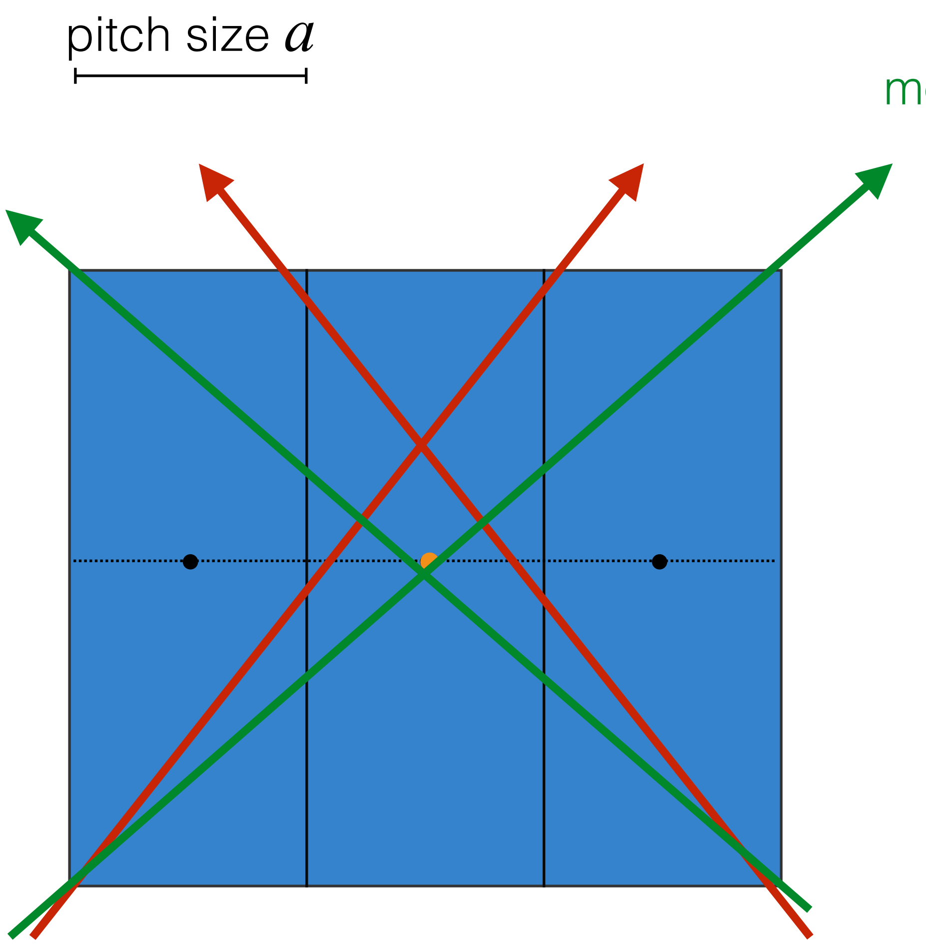




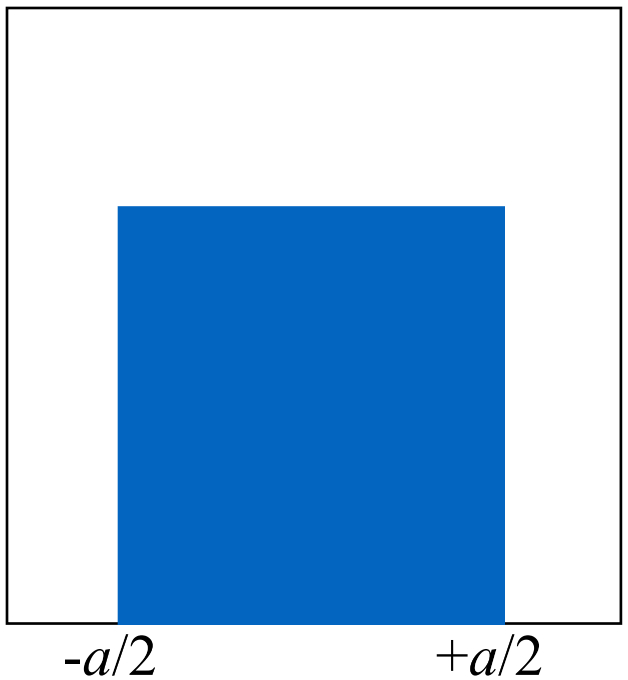
binary case, 2-pixel cluster

MC Toy: pixel clustering

- what can we say about the uncertainty of the measurement ?



most lucky cases: $\mathbf{r} = \mathbf{m} - \mathbf{s} = 0$



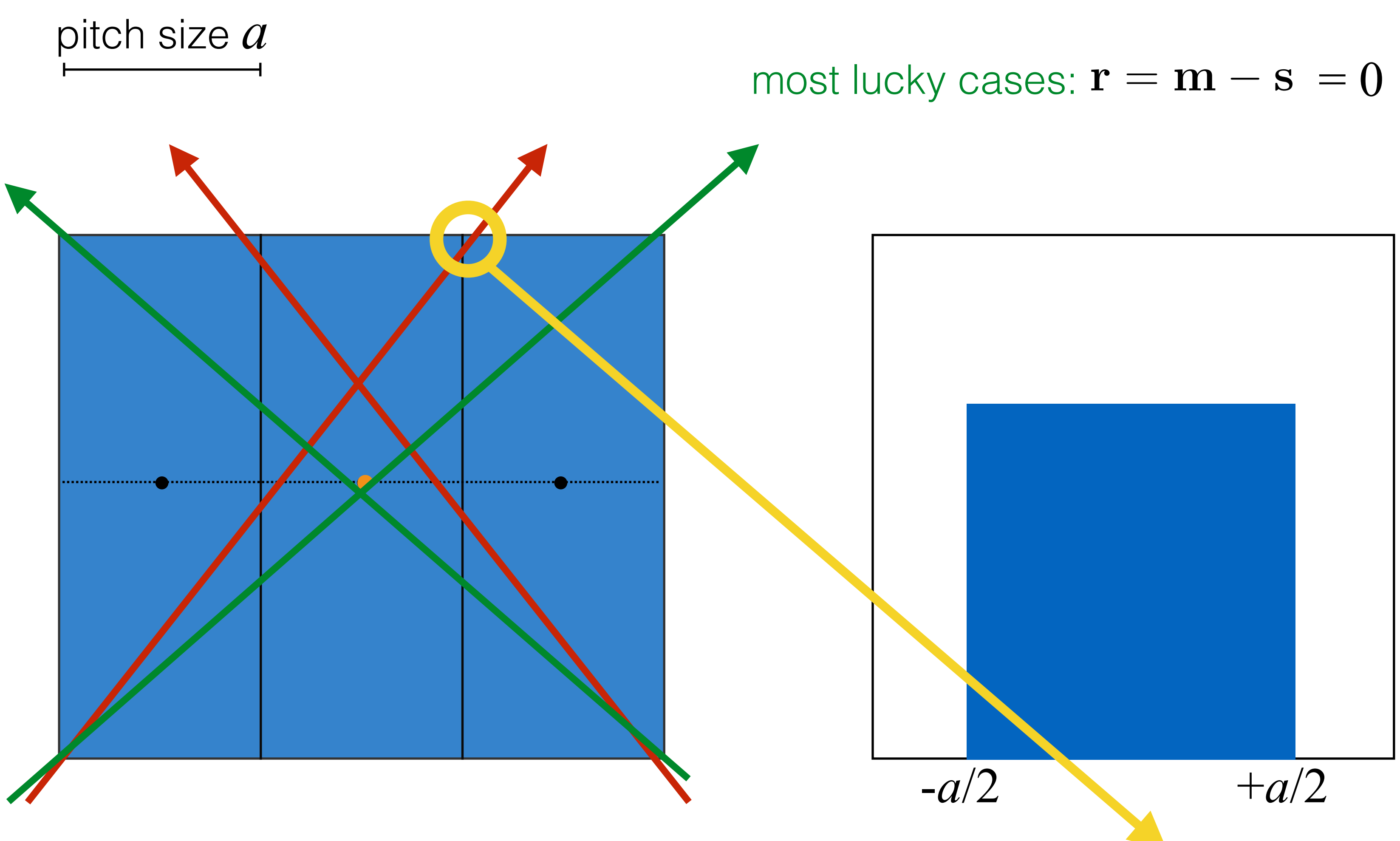
most unlucky cases: $|\mathbf{r}| = |\mathbf{m} - \mathbf{s}| \sim a/2$

binary case, n-pixel cluster

MC Toy: pixel clustering

- what can we say about the uncertainty of the measurement ?

most lucky cases: $\mathbf{r} = \mathbf{m} - \mathbf{s} = 0$



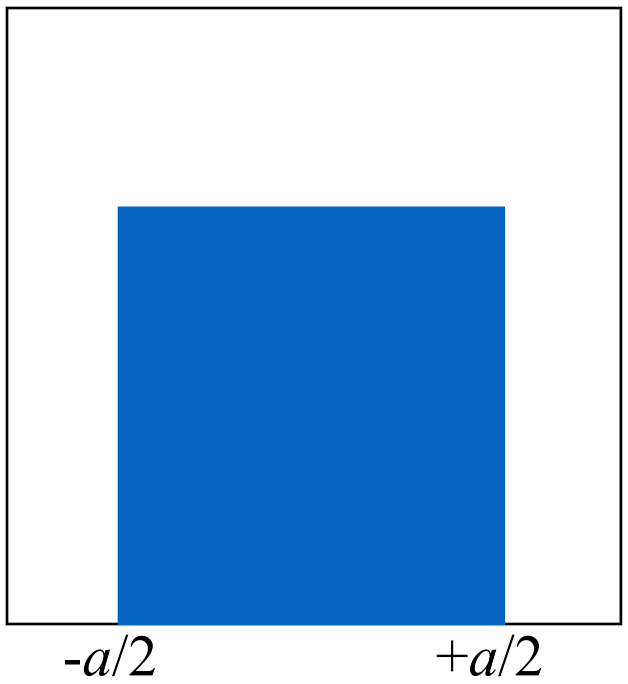
most unlucky cases: $|\mathbf{r}| = |\mathbf{m} - \mathbf{s}| \sim a/2$

usually, a minimum path length
is required to deposit enough charge,
turns the biggest error into < $a/2$

binary case, n-pixel cluster

MC Toy: pixel clustering

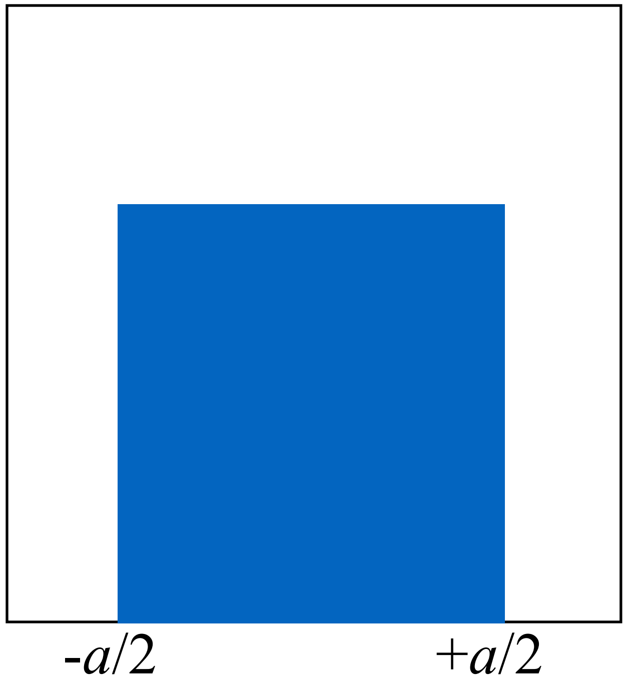
- what's the variance of a uniform distribution between $-a/2$ and $a/2$?



MC Toy: pixel clustering

- what's the variance of a uniform distribution between *-a/2* and *a/2 ?*

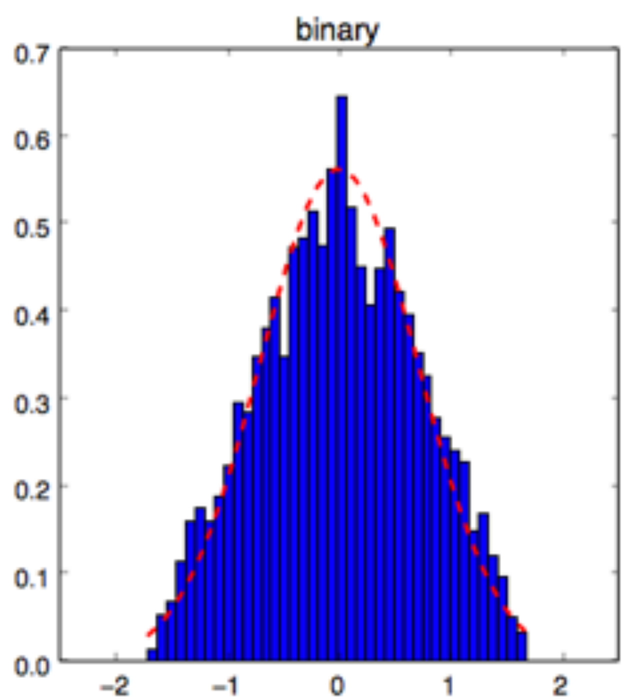
$$\langle r \rangle = a^2/12 \qquad \sigma = a/\sqrt{12}$$



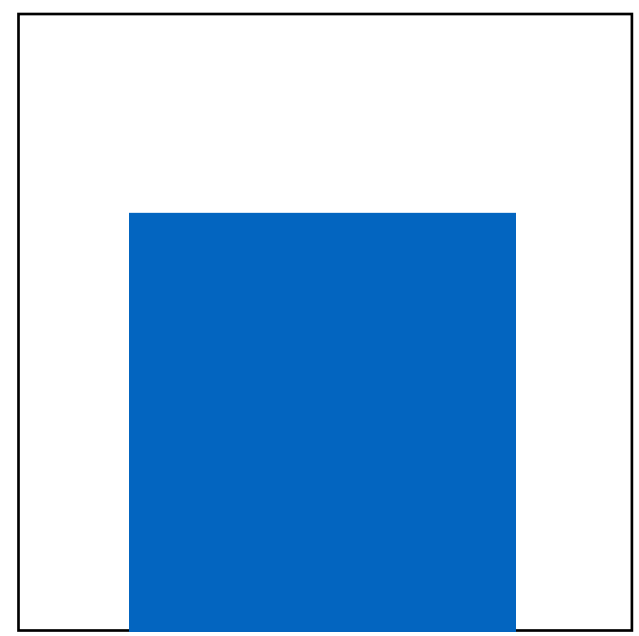
MC Toy: pixel clustering

- what's the variance of a uniform distribution between $-a/2$ and $a/2$?

$$\langle r \rangle = a^2/12 \qquad \sigma = a/\sqrt{12}$$



$p = (m - s)/\sigma$



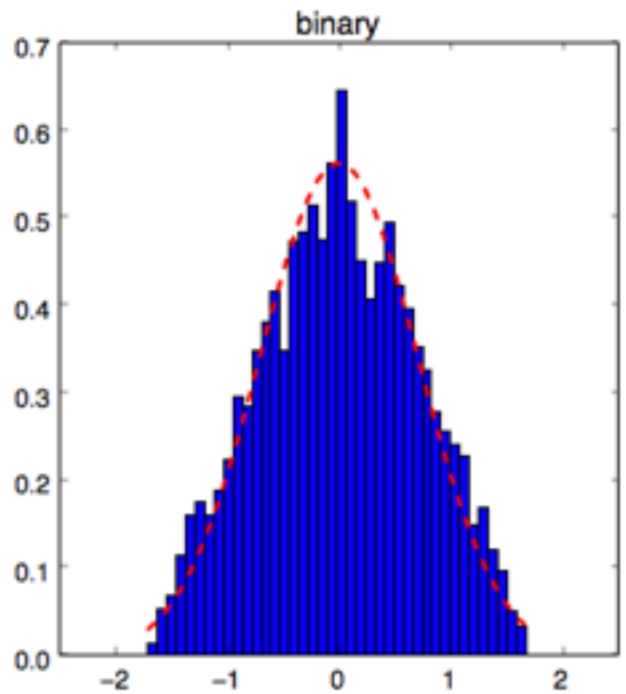
$-a/2$ $+a/2$

```
salzburg$ ipython -i --matplotlib=osx PixelClustering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 5000, pullBinary = True )
[>>] mu =  0.00814352769061   | sigma =  0.70764581917
```

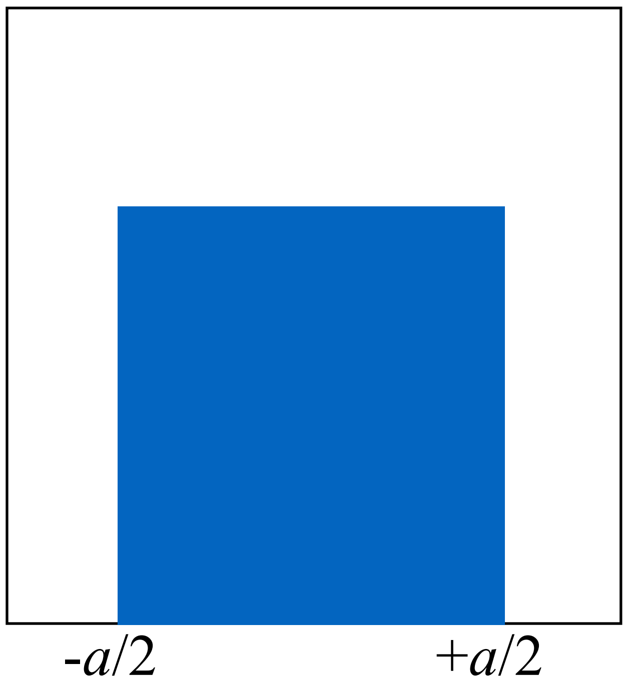
MC Toy: pixel clustering

- what's the variance of a uniform distribution between $-a/2$ and $a/2$?

$$\langle r\rangle = a^2/12 \qquad \sigma = a/\sqrt{12}$$



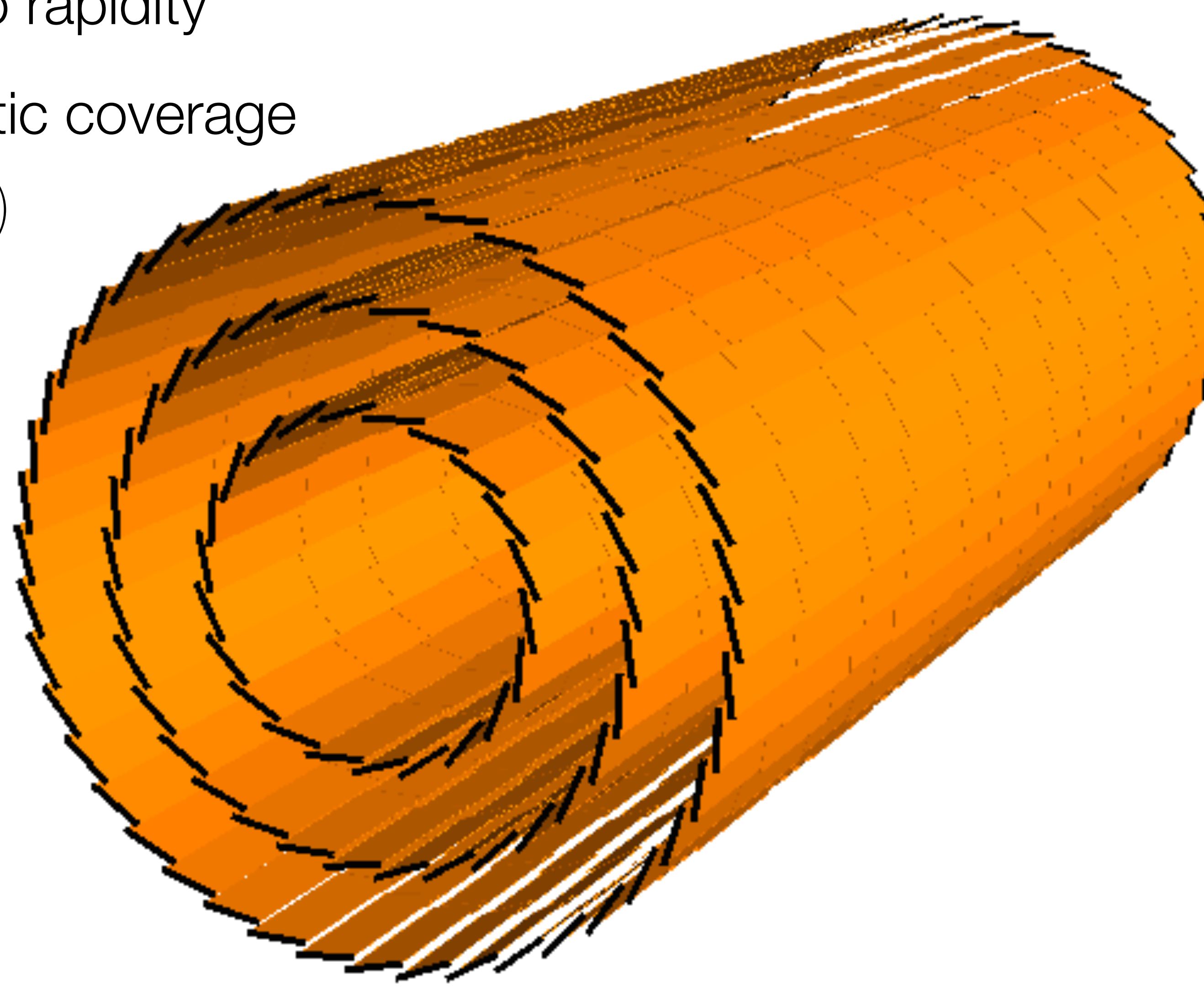
pull distribution : $\boxed{p = (m - s)/\sigma}$

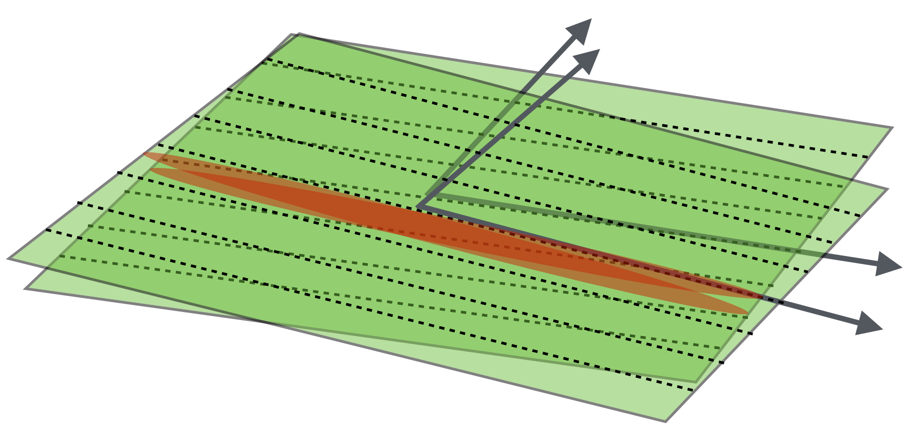


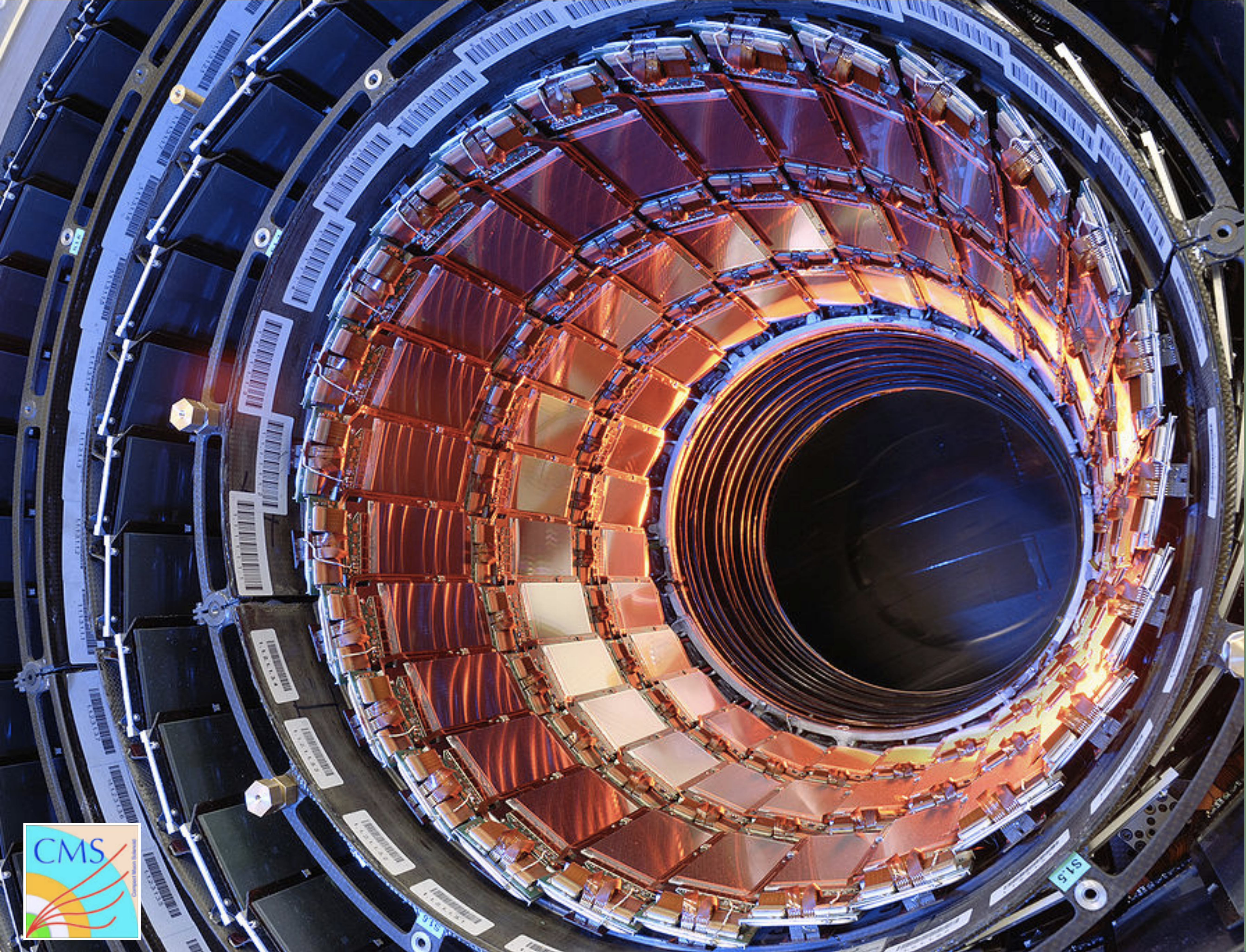

```
salzburg$ ipython -i --matplotlib=osx PixelClustering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 5000, pullBinary = True )
[>>] mu =  0.00814352769061   | sigma =  0.70764581917
```

A classic planar detector design

- Planar modules arranged in cylinders & discs
  - highest granularity in innermost layers
  - barrel structure around the interaction region
  - end-cap disk structure at higher pseudo rapidity
  - overlap of modules to guarantee hermetic coverage
    (e.g. overlaps in ϕ, and along z in general)
  - stereo angle technique for strip detectors
    (two- or double sided modules)

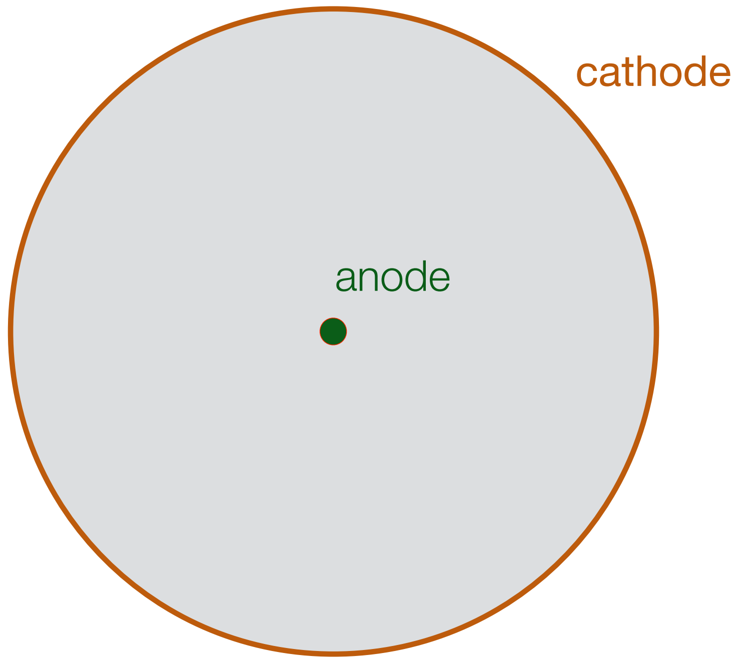






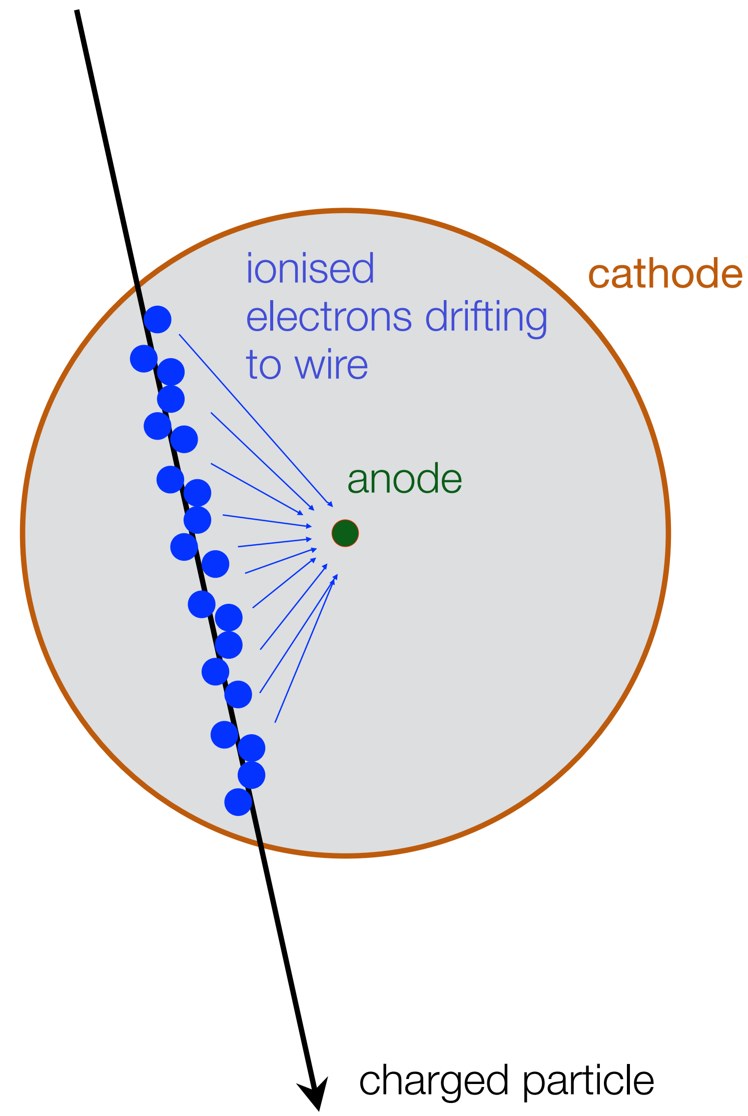
Drift tube detectors

- Gas-filled tubes with a central wire
  - inoisation of gas by traversing charged particle
  - charge drift to wire through electric field (E), in case of embedding in magnetic field also some Lorentz force drift effects
  - measurement is a drift time measurement



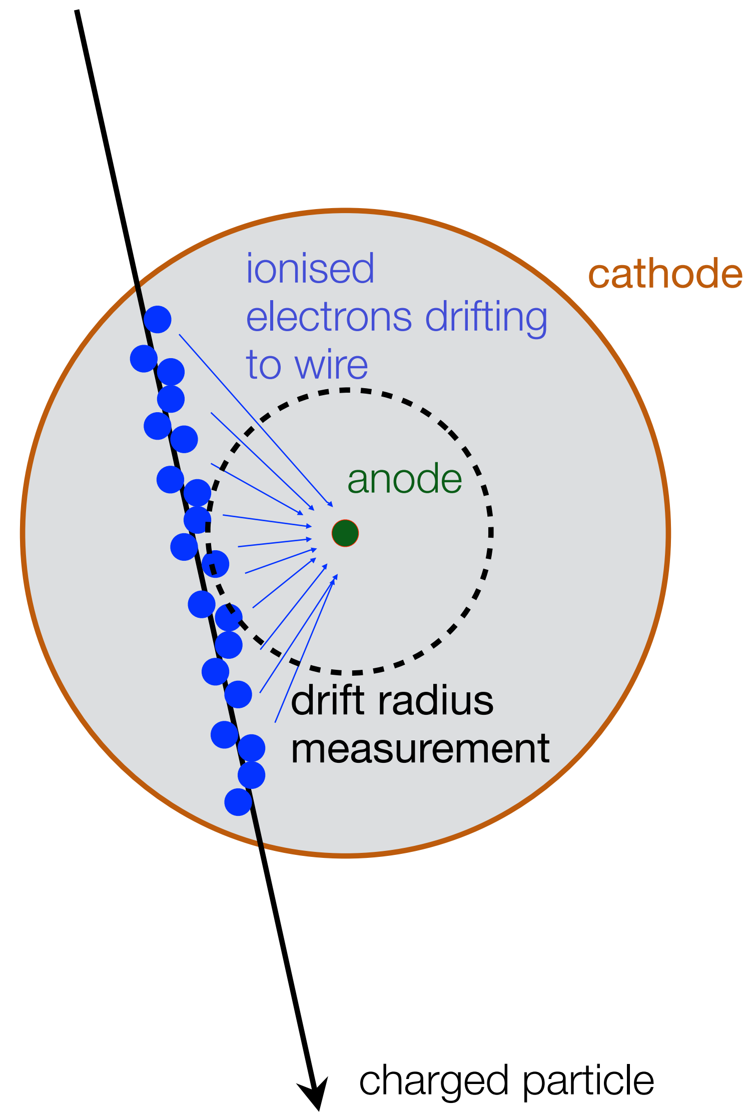
Drift tube detectors

- Gas-filled tubes with a central wire
  - inoisation of gas by traversing charged particle
  - charge drift to wire through electric field (E), in case of embedding in magnetic field also some Lorentz force drift effects
  - measurement is a drift time measurement



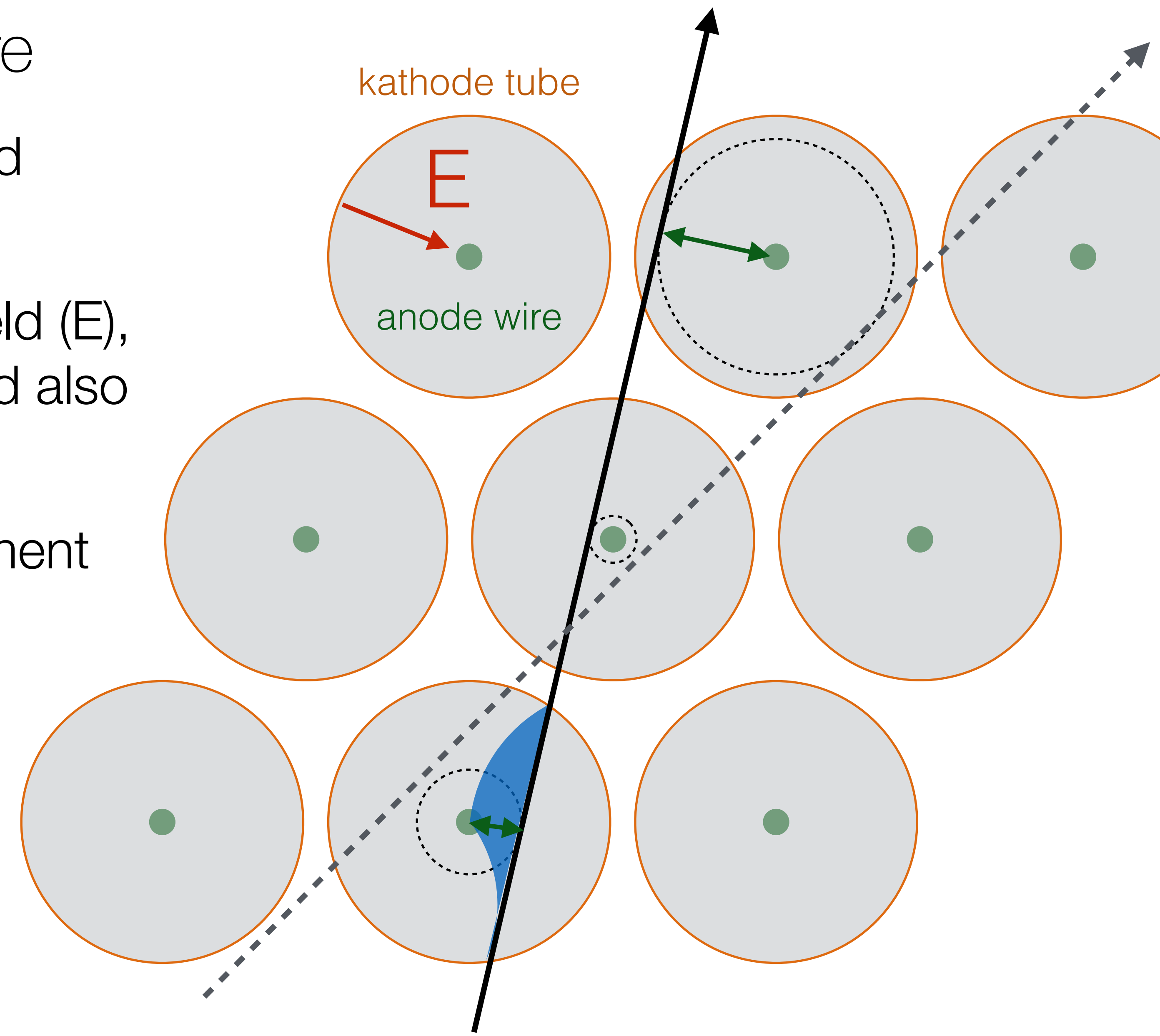
Drift tube detectors

- Gas-filled tubes with a central wire
  - inoisation of gas by traversing charged particle
  - charge drift to wire through electric field (E), in case of embedding in magnetic field also some Lorentz force drift effects
  - measurement is a drift time measurement



Drift tube detectors

- Gas-filled tubes with a central wire
  - inoisation of gas by traversing charged particle
  - charge drift to wire through electric field (E), in case of embedding in magnetic field also some Lorentz force drift effects
  - measurement is a drift time measurement



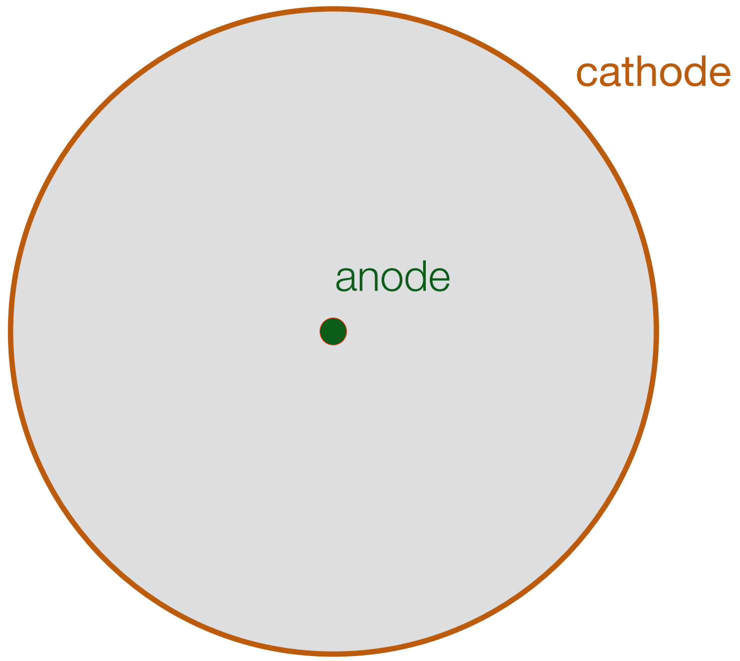
- Track reconstruction with drift measurements
  - drift time converted into drift radius
  - remaining left-right ambiguity that needs to be resolved
    usually done in the pattern recognition when already having some idea about the track direction

Drift tube detectors

- ATLAS Transition Radiation Tracker
  - used to do particle identification (PID)
  - needs a dedicated detector design:

    material with rapidly changing dielectric constant
    -> transition radiation creates additional ionisation, e.g. higher signal

    -> transition radiation is strongly dependent on Lorentz factor



ATLAS Testbeam results

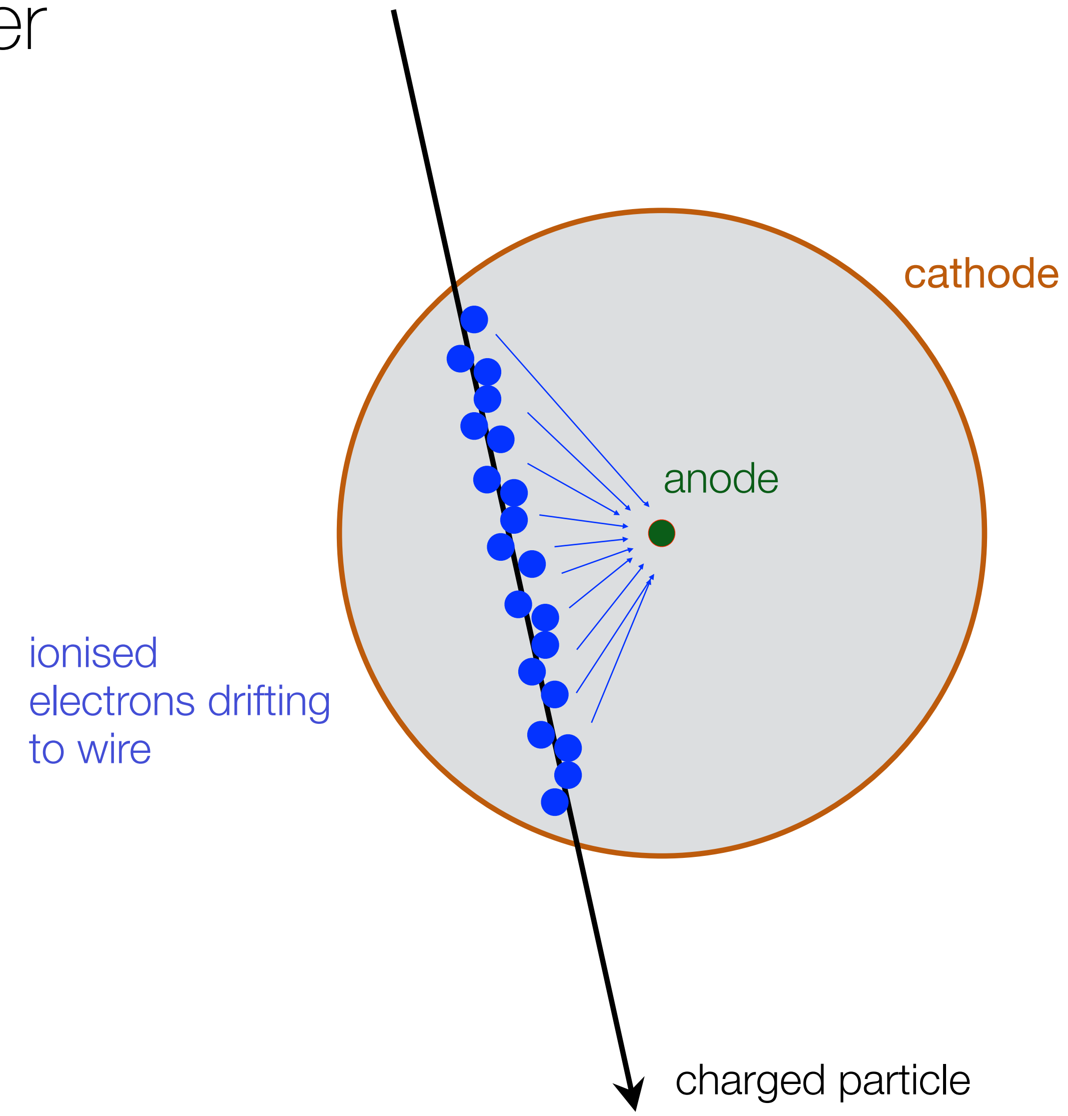
Drift tube detectors

- ATLAS Transition Radiation Tracker
  - used to do particle identification (PID)
  - needs a dedicated detector design:

    material with rapidly changing dielectric constant

    -> transition radiation creates additional ionisation, e.g. higher signal

    -> transition radiation is strongly dependent on Lorentz factor



ATLAS Testbeam results

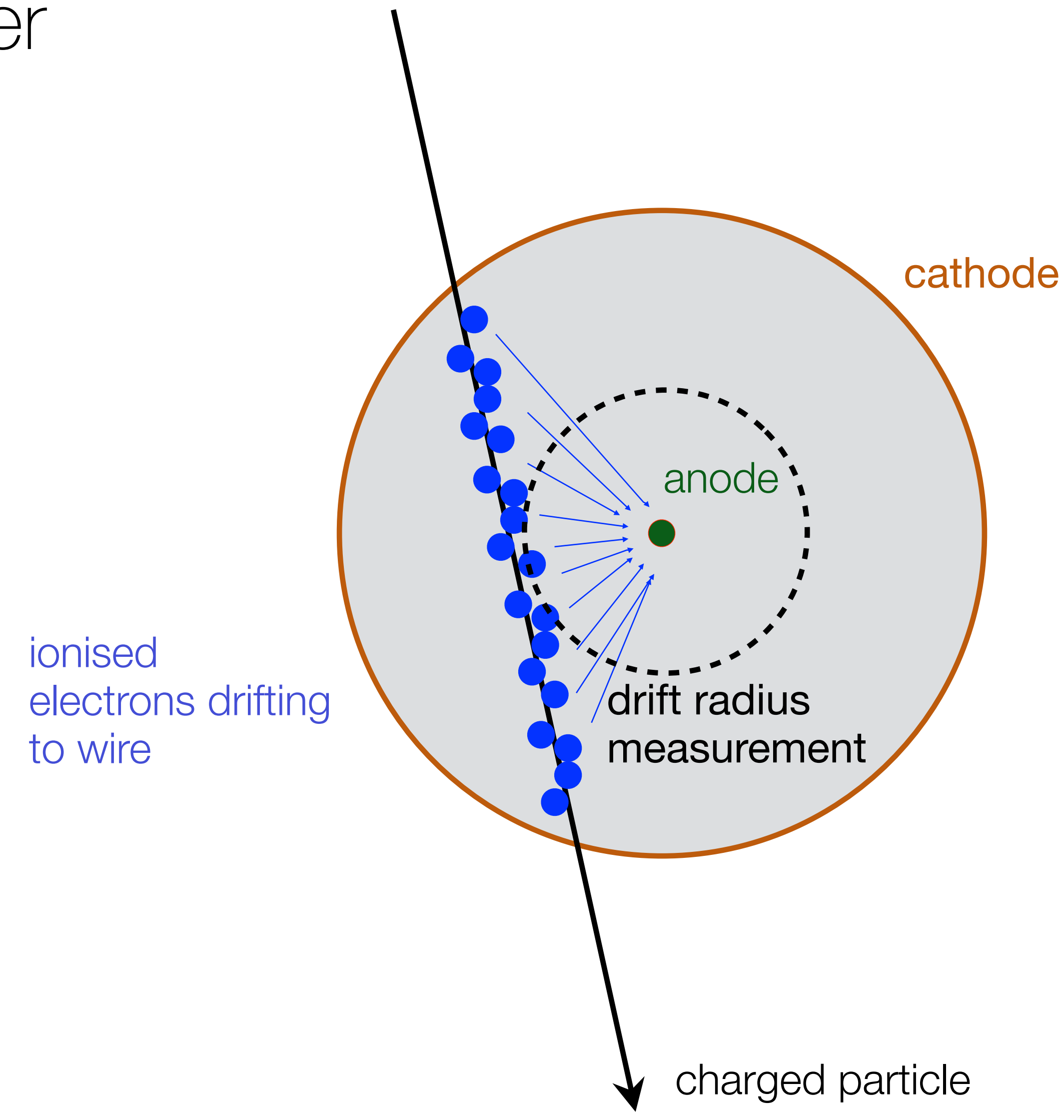
Drift tube detectors

- ATLAS Transition Radiation Tracker
  - used to do particle identification (PID)
  - needs a dedicated detector design:

    material with rapidly changing dielectric constant

    -> transition radiation creates additional ionisation, e.g. higher signal

    -> transition radiation is strongly dependent on Lorentz factor



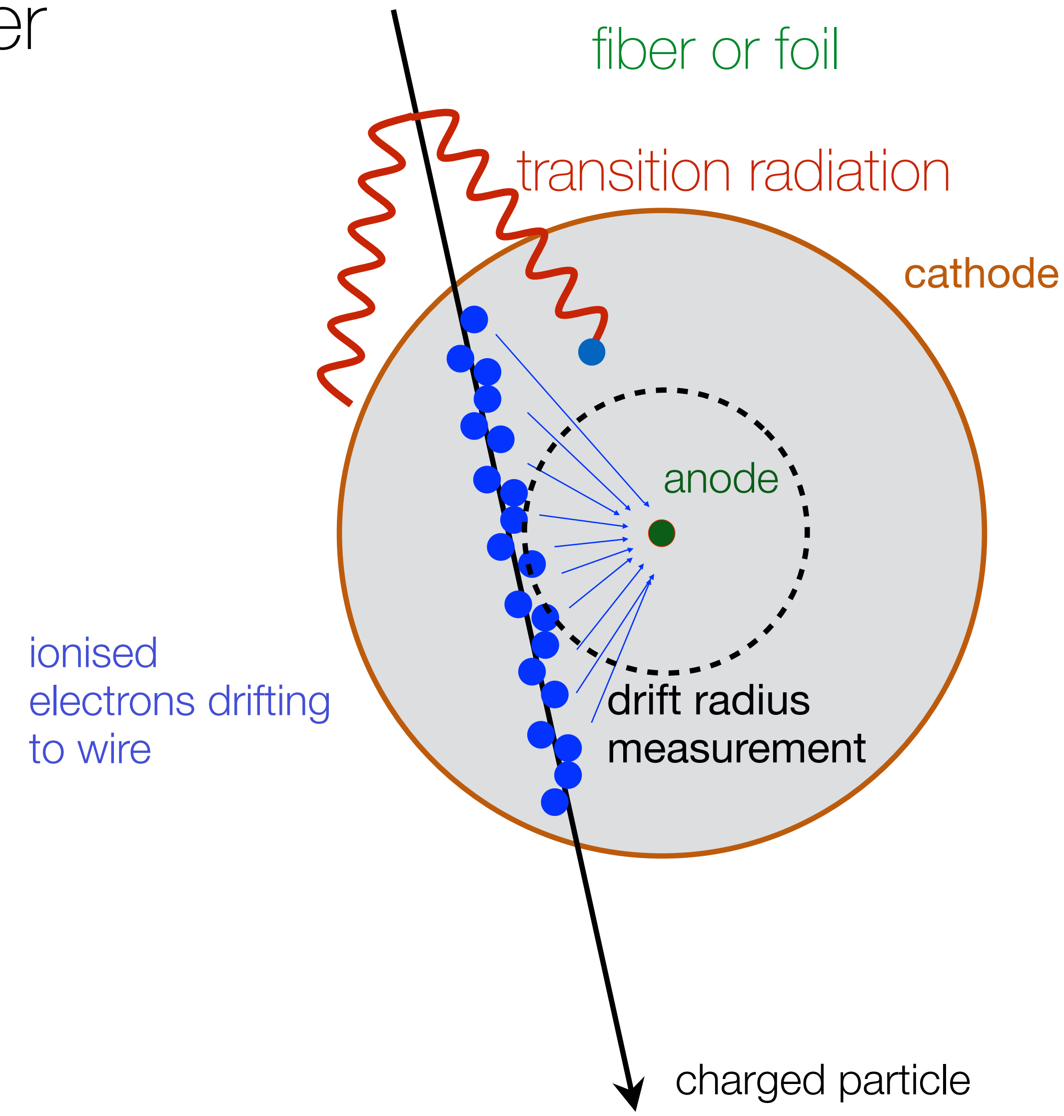
ATLAS Testbeam results

Drift tube detectors

- ATLAS Transition Radiation Tracker
  - used to do particle identification (PID)
  - needs a dedicated detector design:

    material with rapidly changing dielectric constant
    -> transition radiation creates additional ionisation, e.g. higher signal

    -> transition radiation is strongly dependent on Lorentz factor



ATLAS Testbeam results

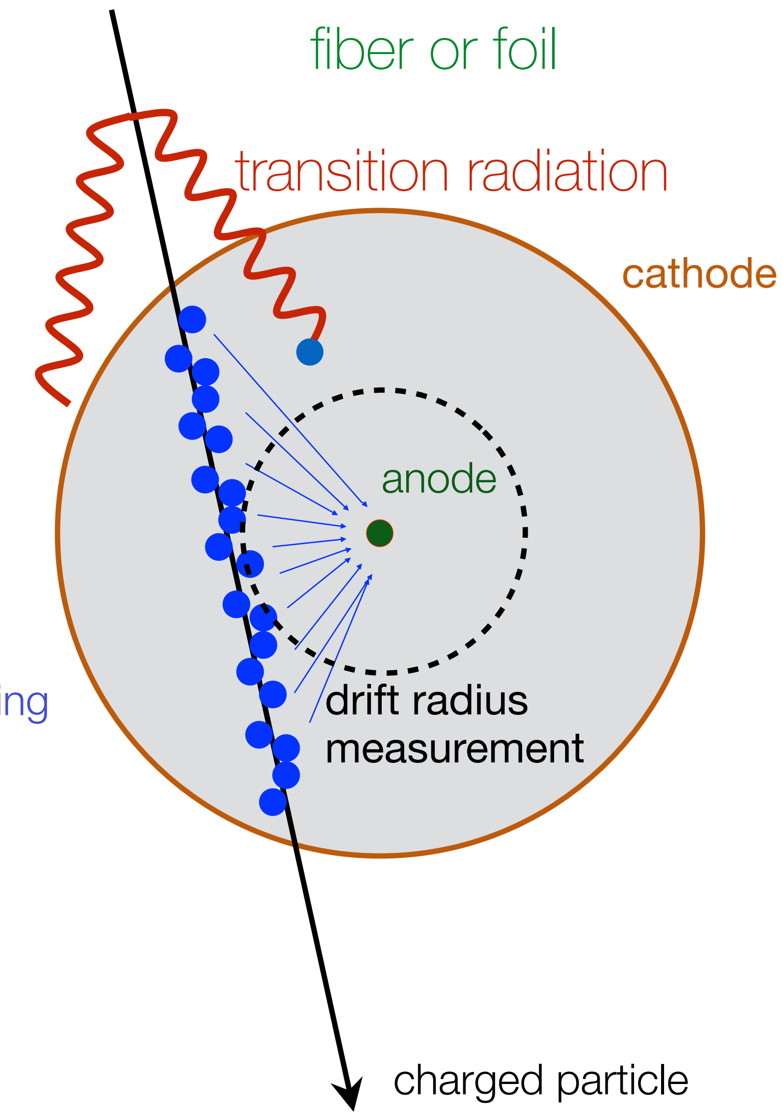
Drift tube detectors

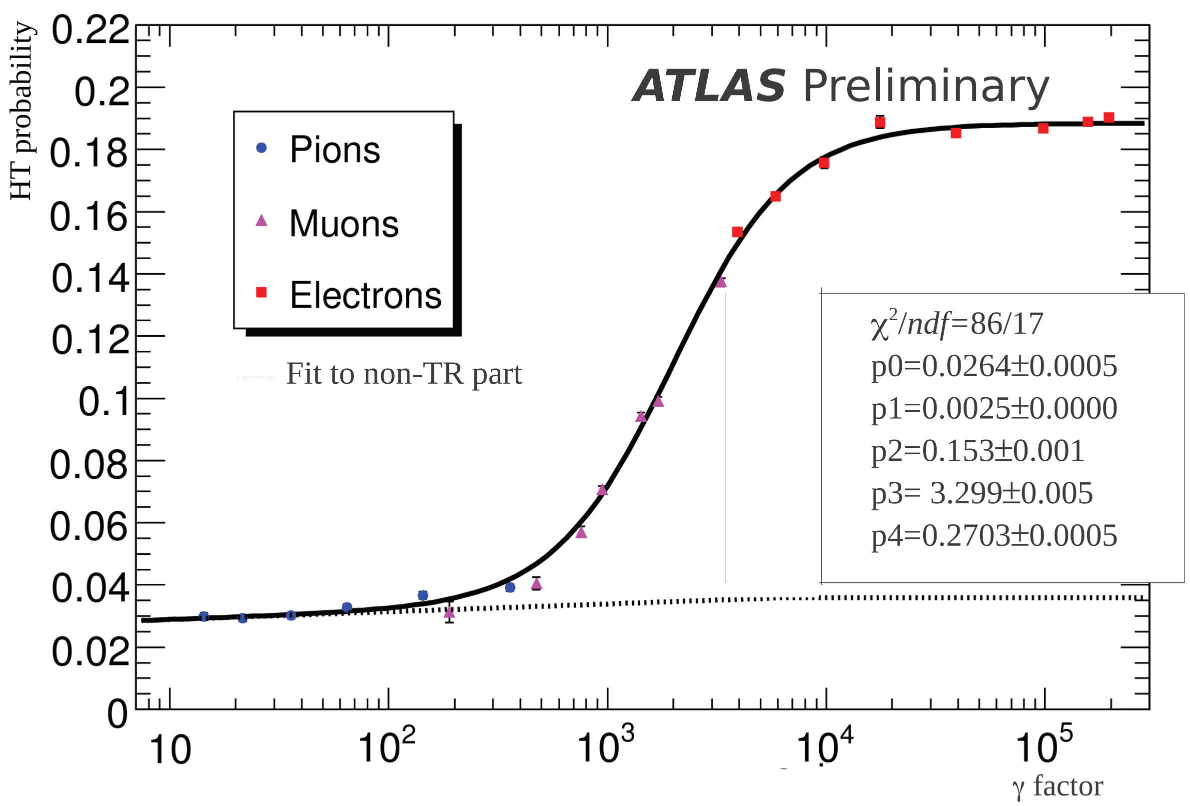
- ATLAS Transition Radiation Tracker
  - used to do particle identification (PID)
  - needs a dedicated detector design:

    material with rapidly changing dielectric constant

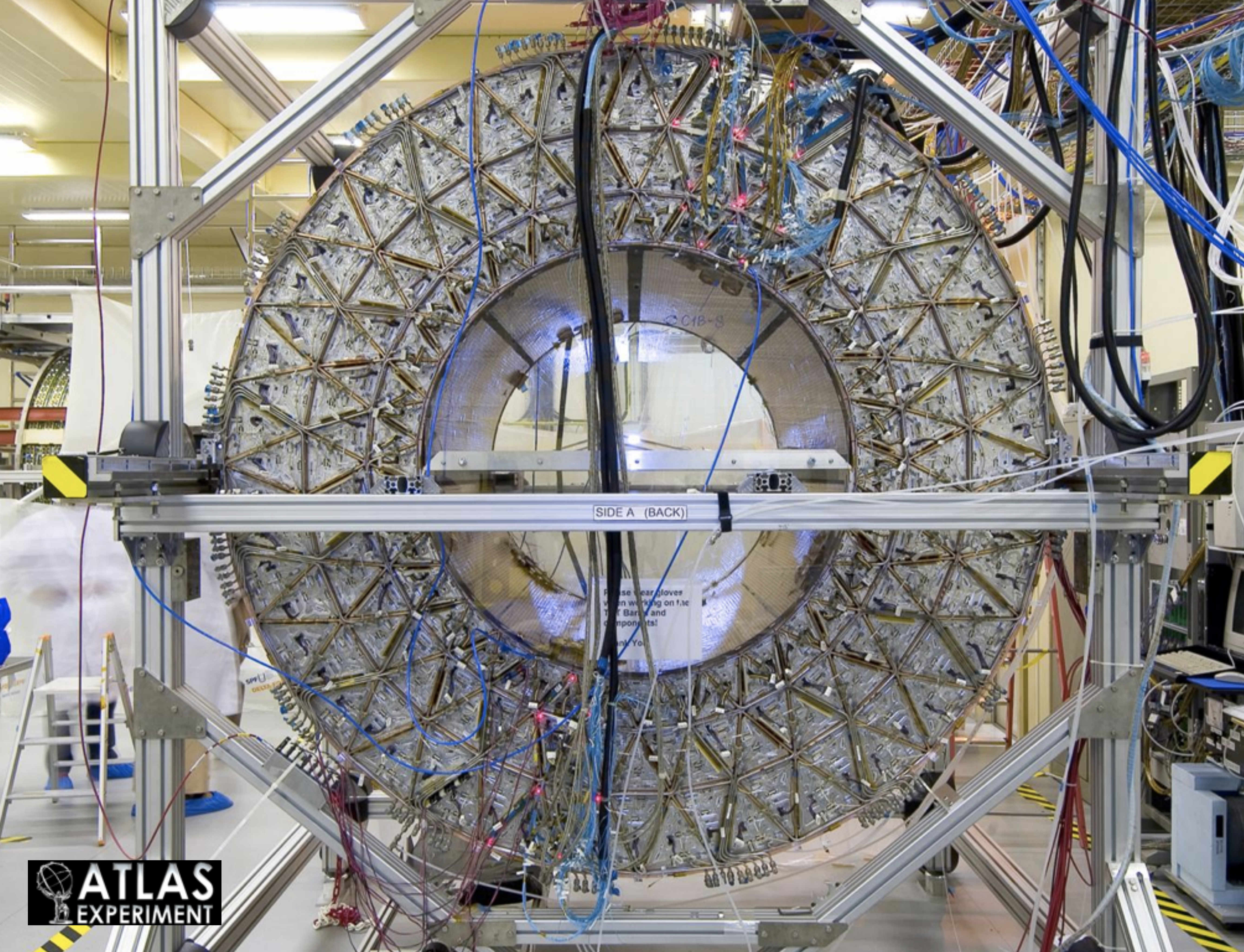
    -> transition radiation creates additional ionisation, e.g. higher signal

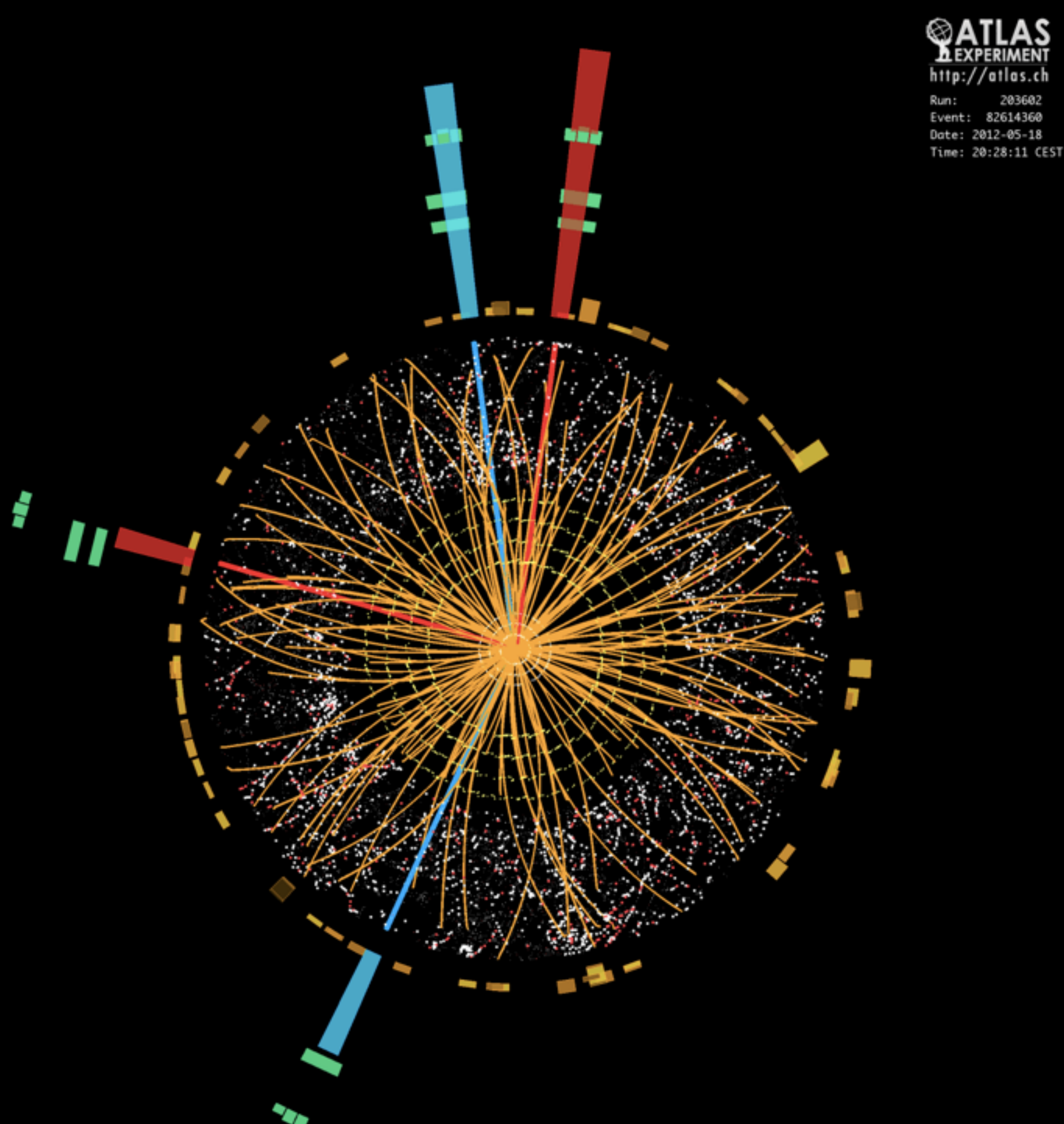
    -> transition radiation is strongly dependent on Lorentz factor





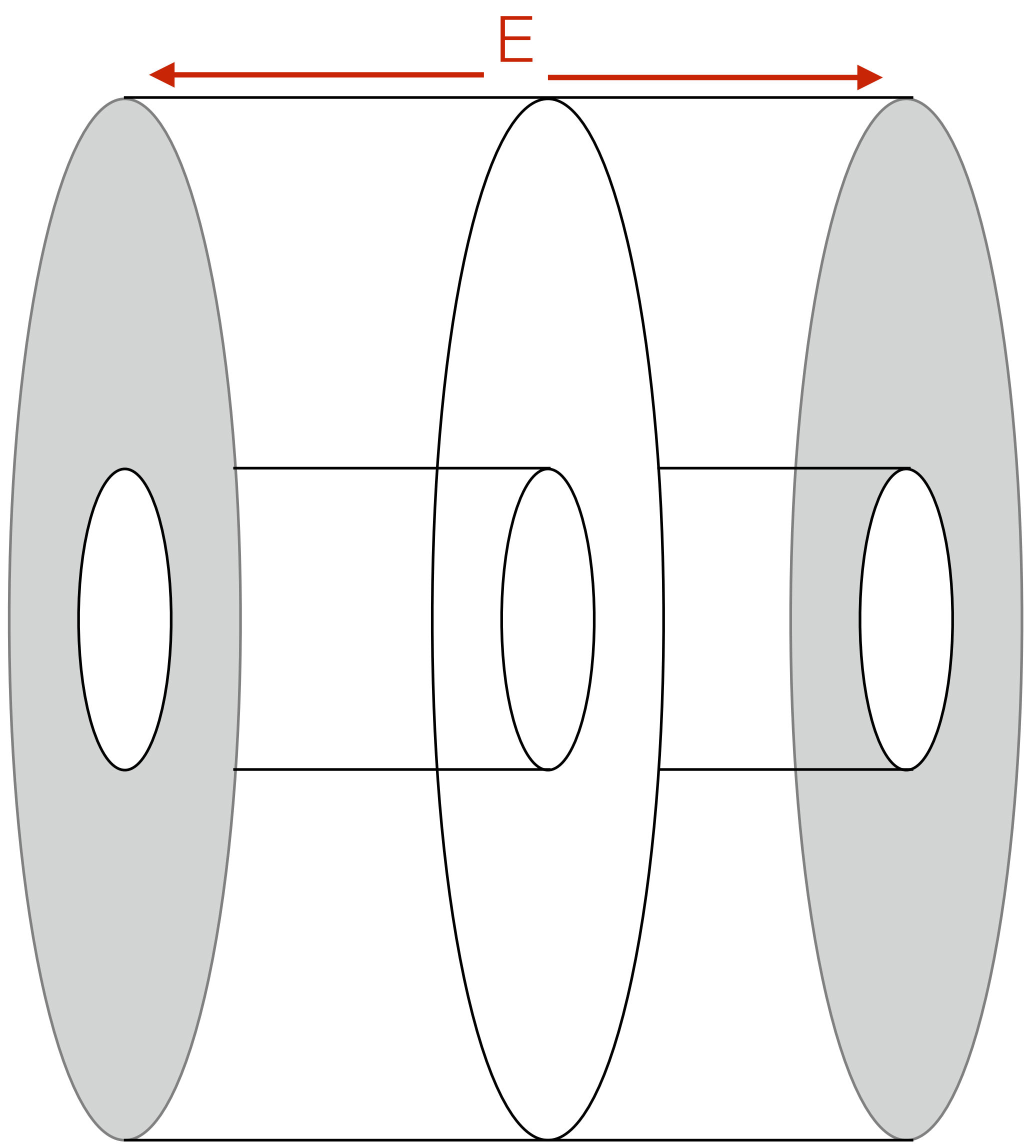
ATLAS Testbeam results


SIDE A (BACK)
ATLAS
EXPERIMENT


ATLAS
EXPERIMENT
http://atlas.ch
Run: 203602
Event: 82614360
Date: 2012-05-18
Time: 20:28:11 CEST

Time projection chamber (TPC)

- TPCs allow to build huge tracking devices to relative moderate cost
  - precise track reconstruction

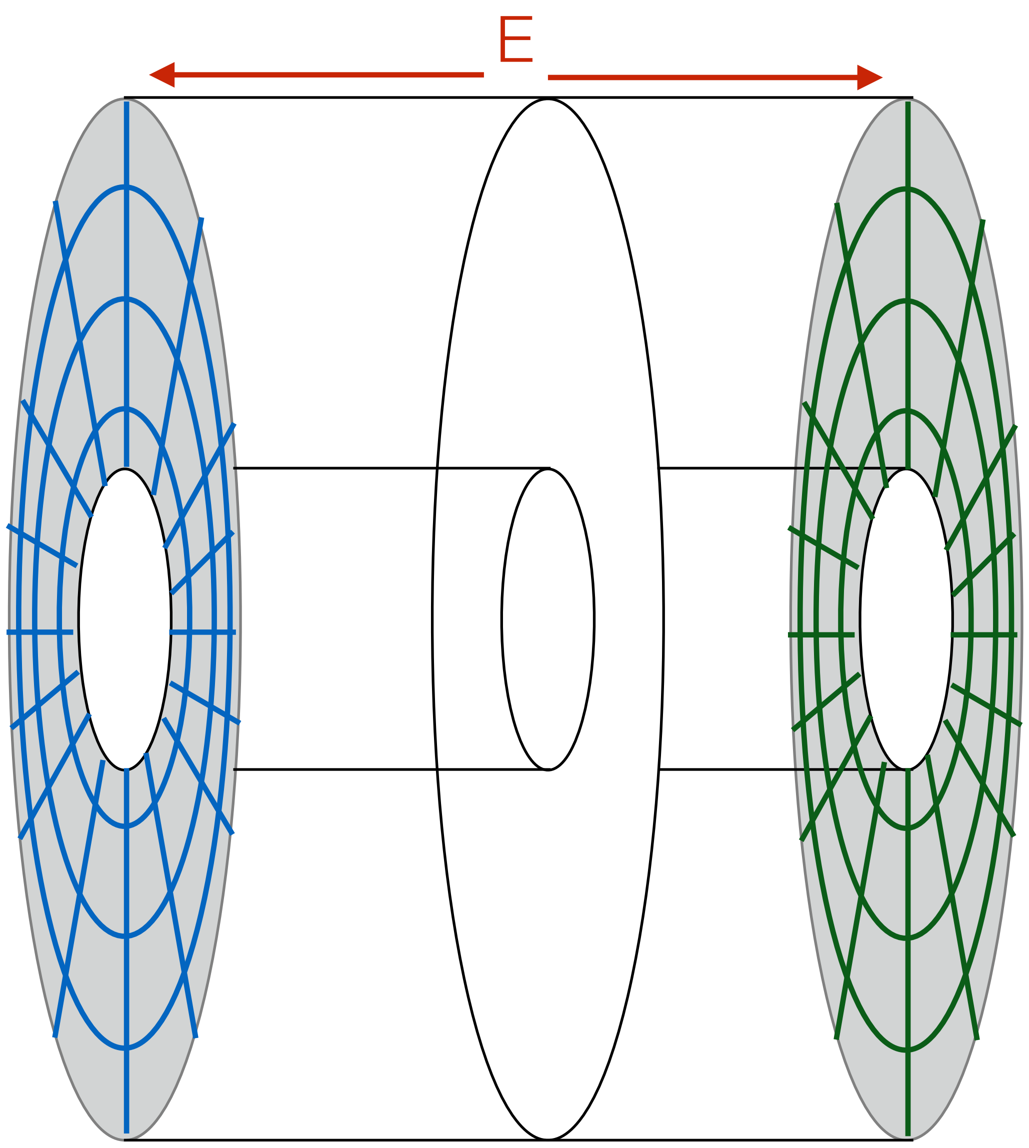


a gas filled vessel (ionisable)

electric field for the charge drift

Time projection chamber (TPC)

- TPCs allow to build huge tracking devices to relative moderate cost
  - precise track reconstruction



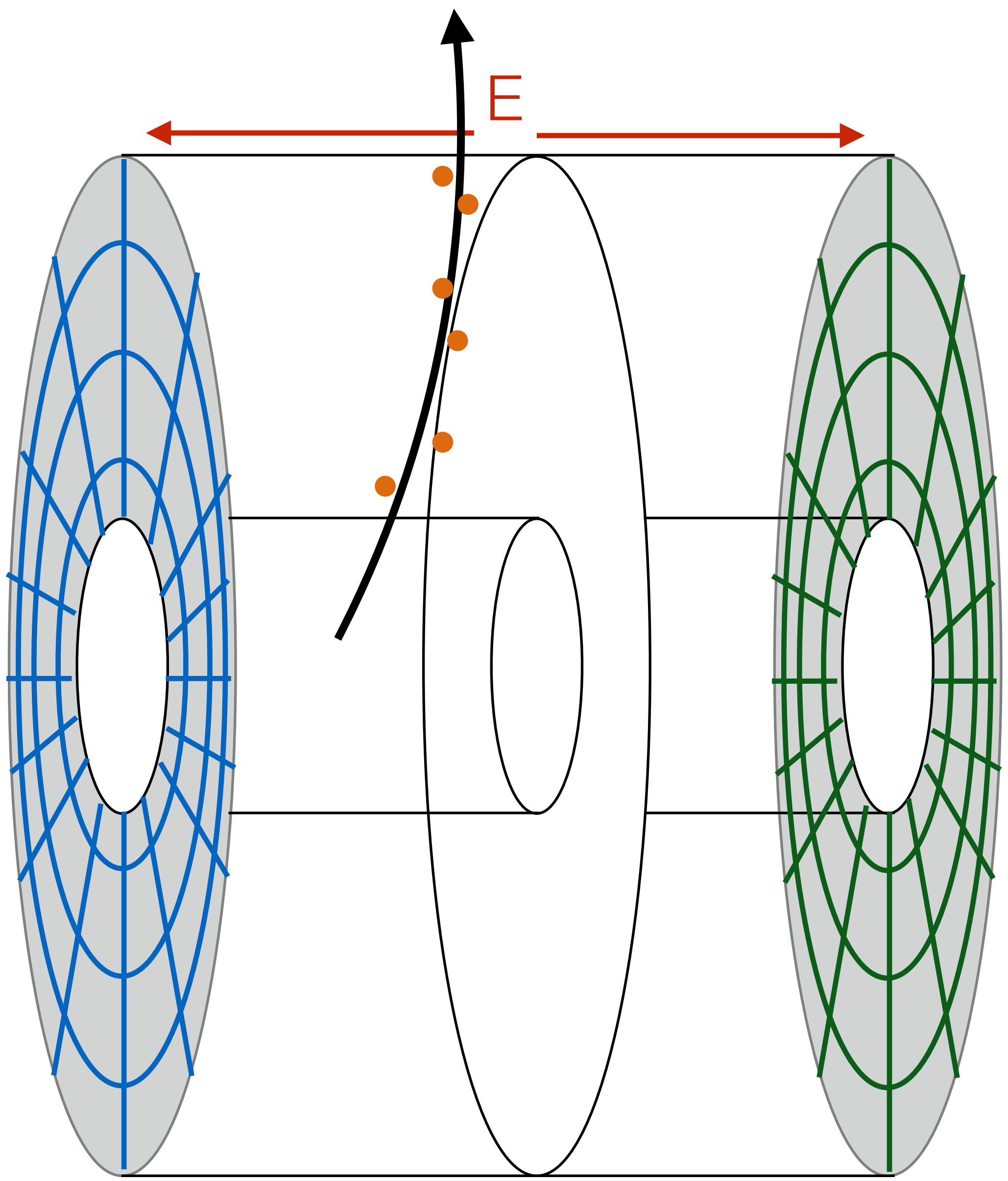
a gas filled vessel (ionisable)

electric field for the charge drift

segmented readout chambers
(different technologies possible)

Time projection chamber (TPC)

- TPCs allow to build huge tracking devices to relative moderate cost
  - precise track reconstruction



a gas filled vessel (ionisable)

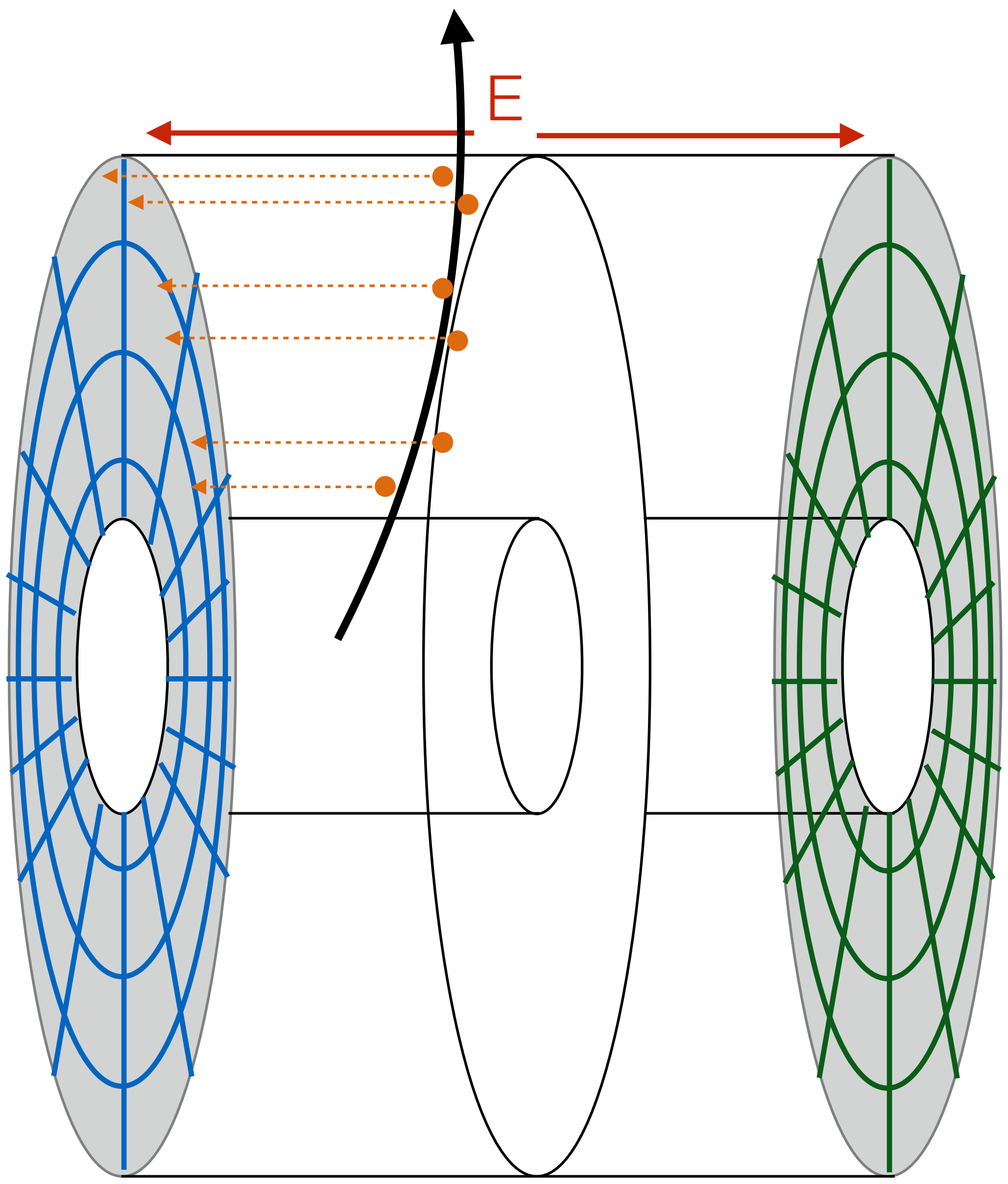
electric field for the charge drift

segmented readout chambers
(different technologies possible)

track ionises the gas

Time projection chamber (TPC)

- TPCs allow to build huge tracking devices to relative moderate cost
  - precise track reconstruction



a gas filled vessel (ionisable)

electric field for the charge drift

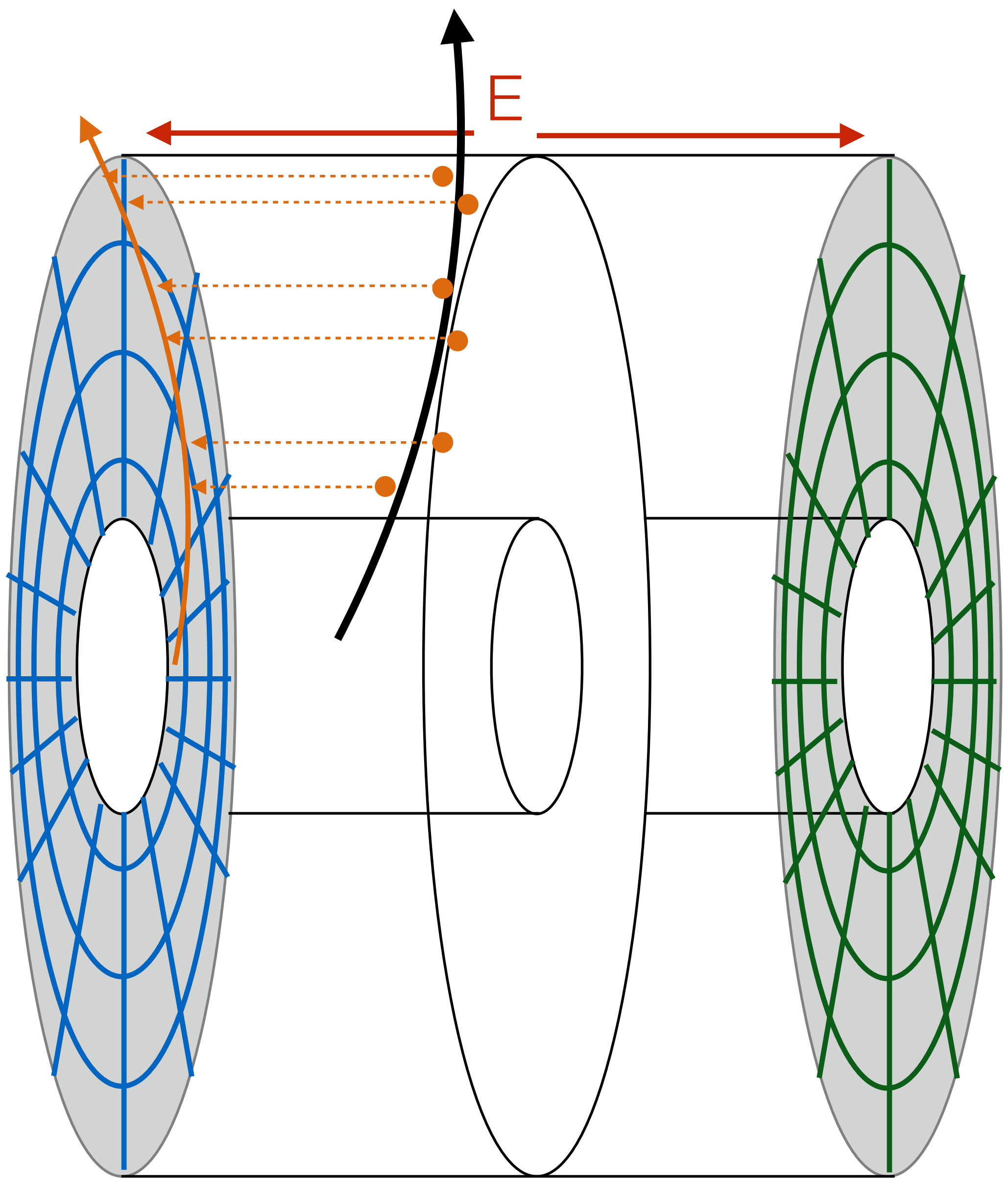
segmented readout chambers
(different technologies possible)

track ionises the gas

charge drift to the readout chambers

Time projection chamber (TPC)

- TPCs allow to build huge tracking devices to relative moderate cost
  - precise track reconstruction



a gas filled vessel (ionisable)

electric field for the charge drift

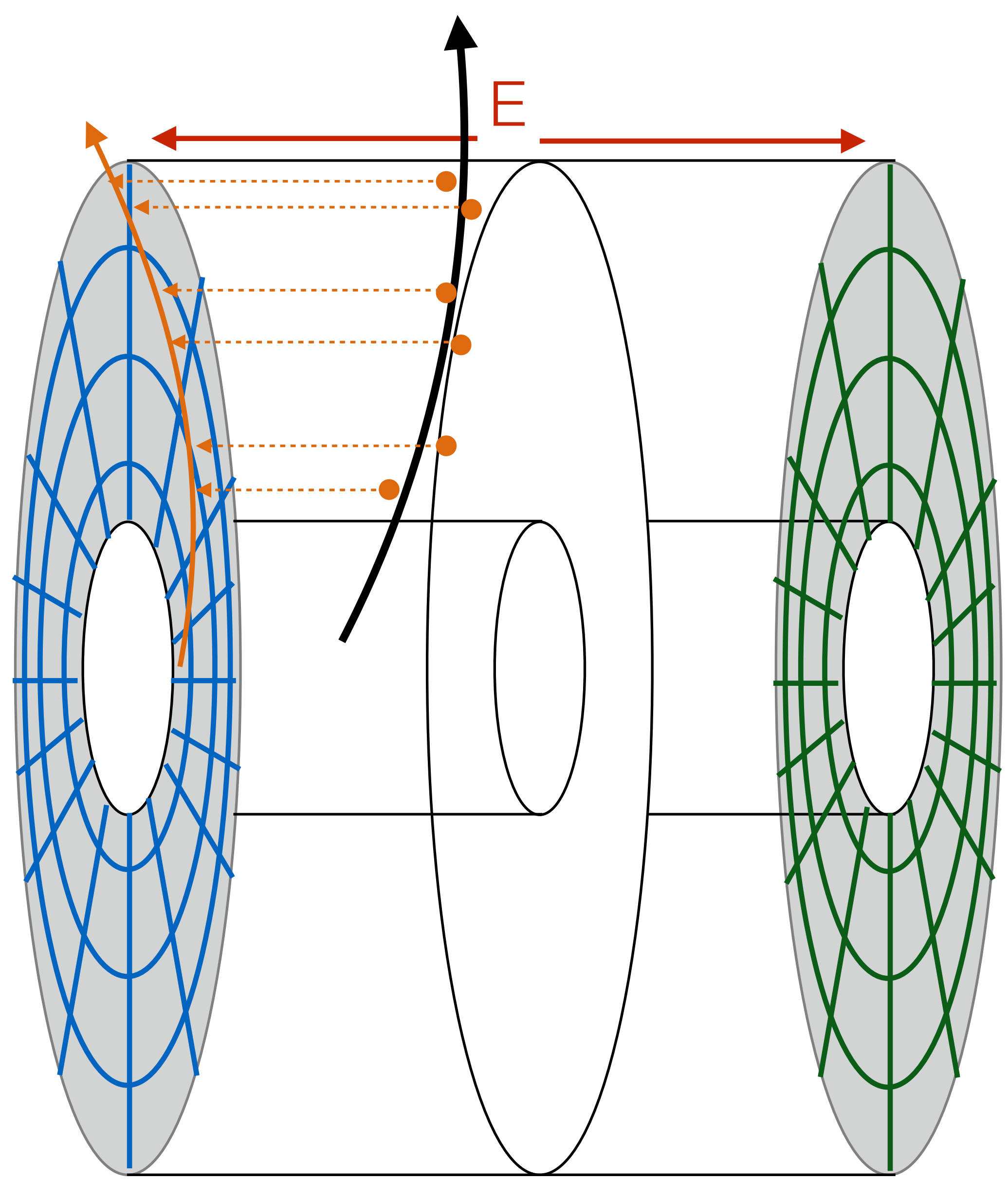
segmented readout chambers
(different technologies possible)

track ionises the gas

charge drift to the readout chambers

Time projection chamber (TPC)

- TPCs allow to build huge tracking devices to relative moderate cost
  - precise track reconstruction



a gas filled vessel (ionisable)

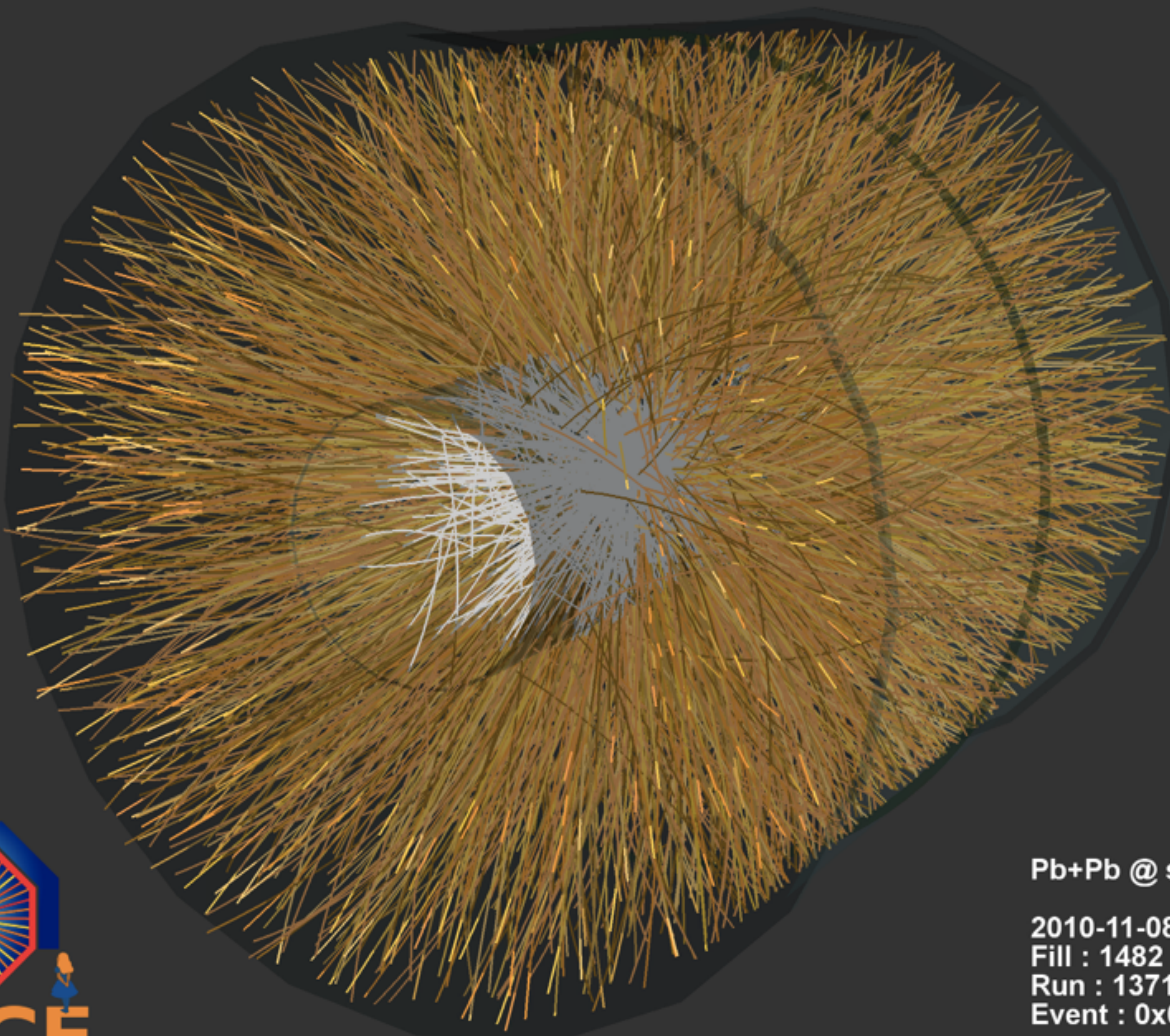
electric field for the charge drift

segmented readout chambers
(different technologies possible)

track ionises the gas

charge drift to the readout chambers

measurements:
(x,y) from readout segmentation
(z) from drift time



Pb+Pb @ sqrt(s) = 2.76 ATeV

2010-11-08 11:30:46
Fill : 1482
Run : 137124
Event : 0x00000000D3BBE693

ALICE


ALICE

Enemy No. 1: material

- Unfortunately there a difference between how we'd like an ideal detector to be and the reality

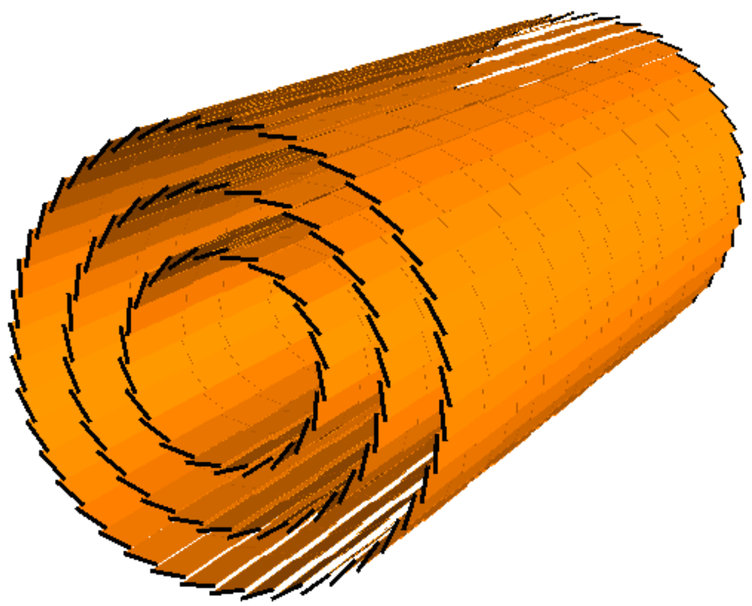
- Let's face it: the reality is always more messy …
- General aim in the construction of tracking detectors:
  - build them as light as possible

    material interactions disturb the measurement in the tracker itself

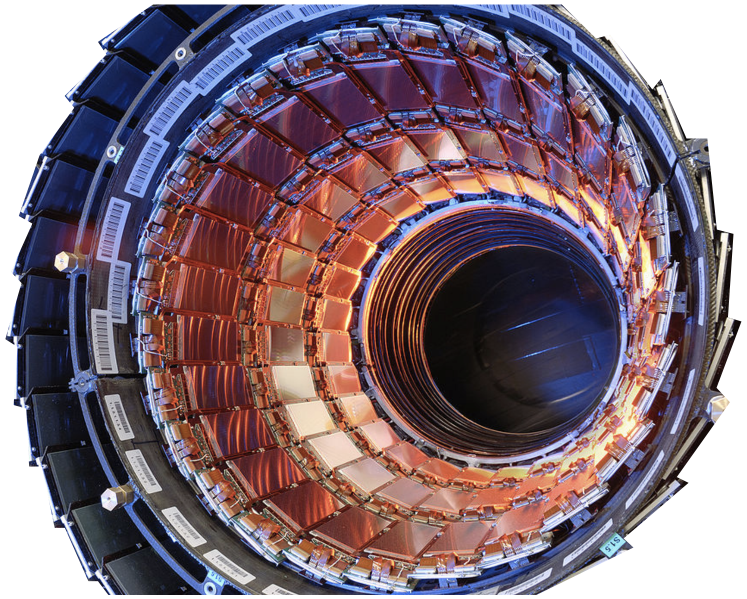
    tracker is usually before the calorimeter (material disturbs the calorimeter measurement)

Enemy No. 1: **material**

- Unfortunately there a difference between how we'd like an ideal detector to be and the reality



ideal



real

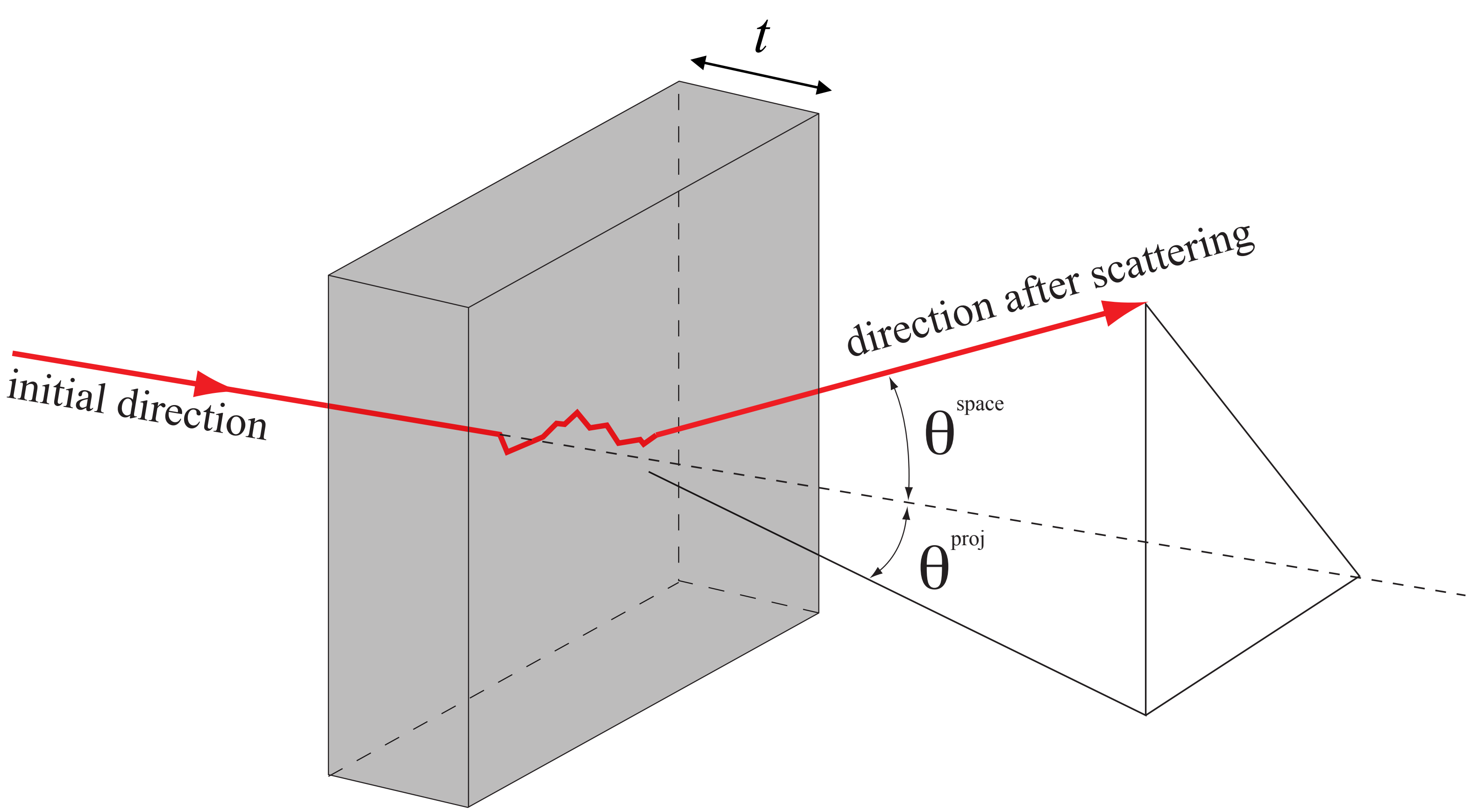
- Let's face it: the reality is always more messy …
- General aim in the construction of tracking detectors:
  - build them as light as possible

    material interactions disturb the measurement in the tracker itself

    tracker is usually before the calorimeter (material disturbs the calorimeter measurement)

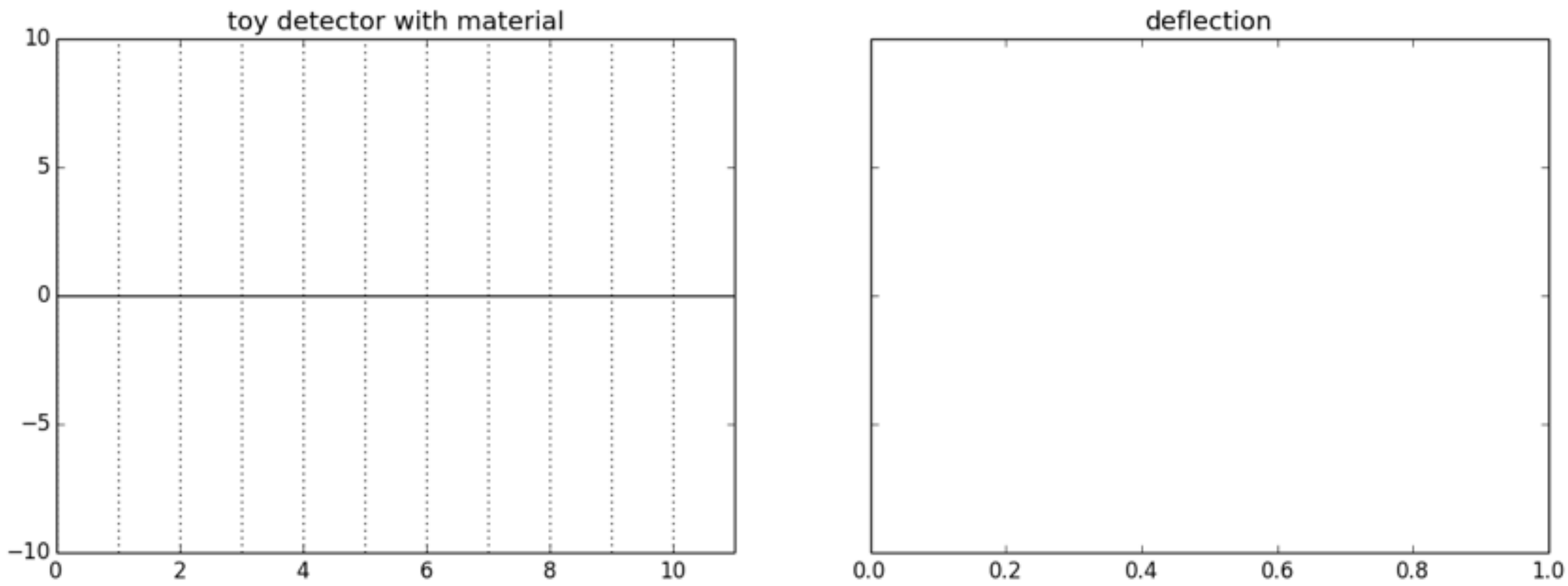
Multiple (Coulomb) scattering

- A charged particle undergoes random deflection
  - mainly caused by multiple (Coulomb) scattering off the core of atoms
  - additional component from single large (Rutherford) scattering



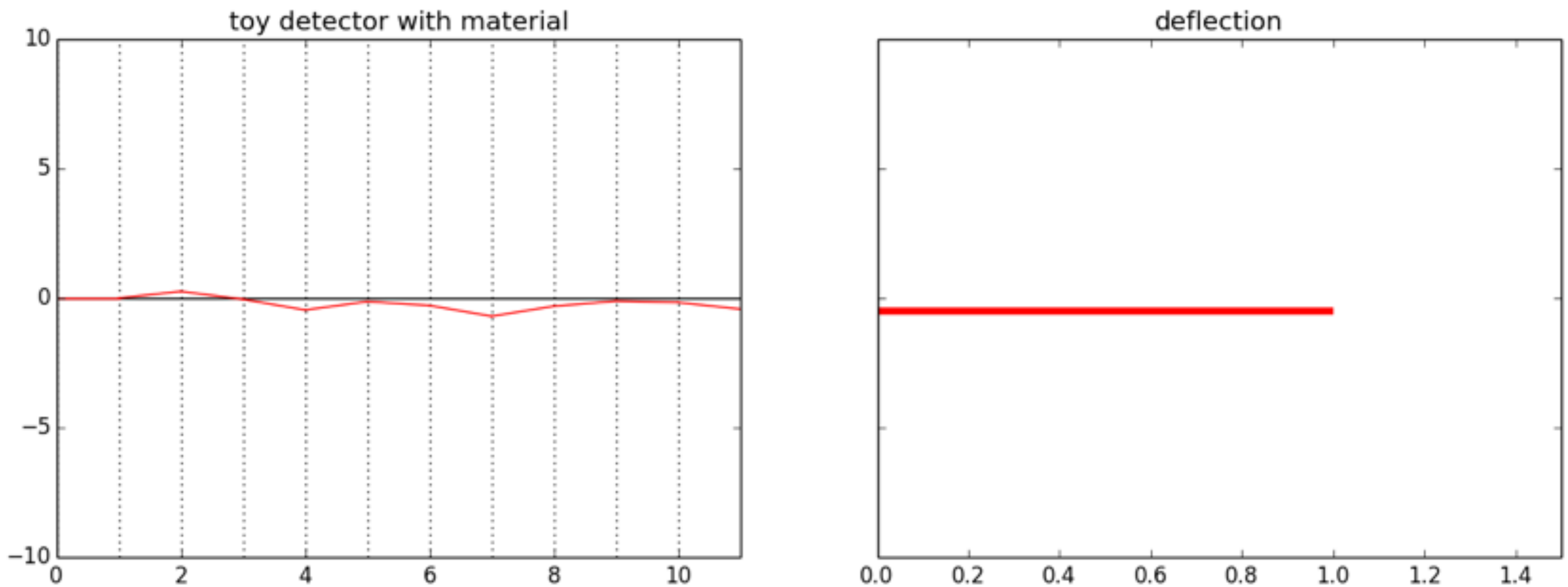
$$\sigma_{ms}^{proj} = \frac{13.6 \text{ MeV}}{\beta c p} Z \sqrt{t/X_0} \left[1 + 0.038 \ln\left(t/X_0\right)\right]$$

MC Toy: multiple scattering



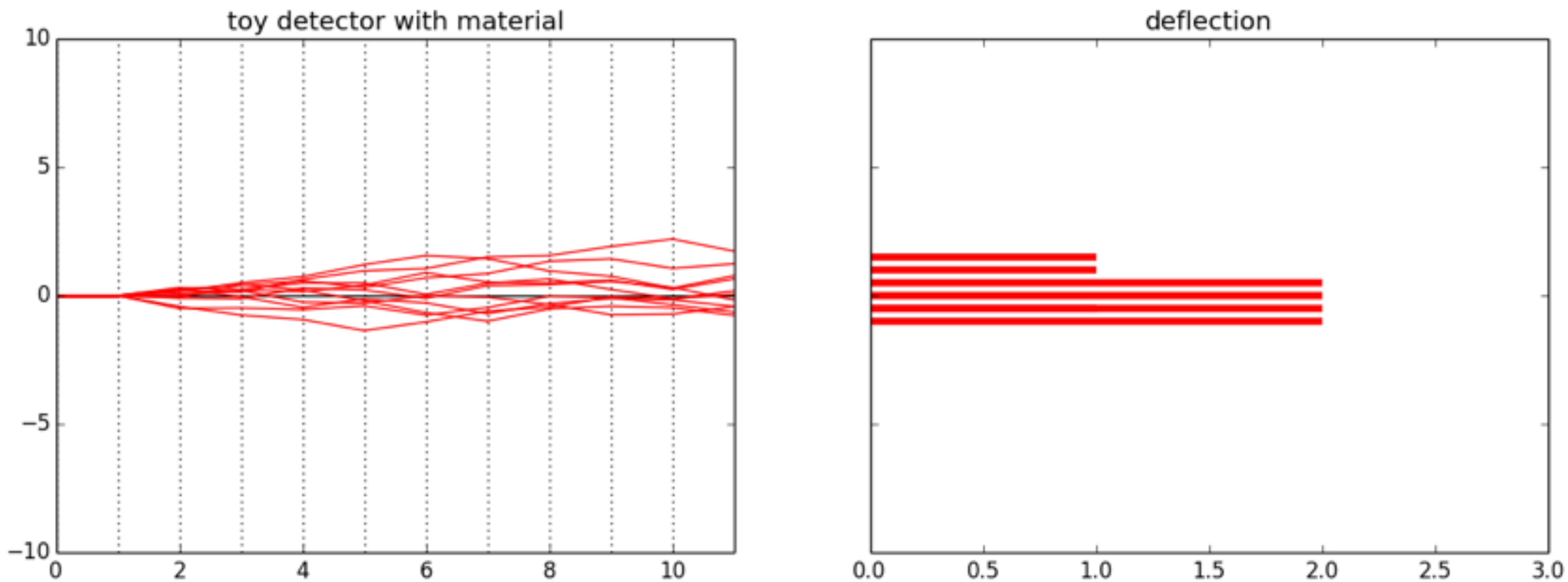

```
salzburg$ ipython -i --matplotlib=osx MultipleScattering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 1000 )
```

MC Toy: multiple scattering



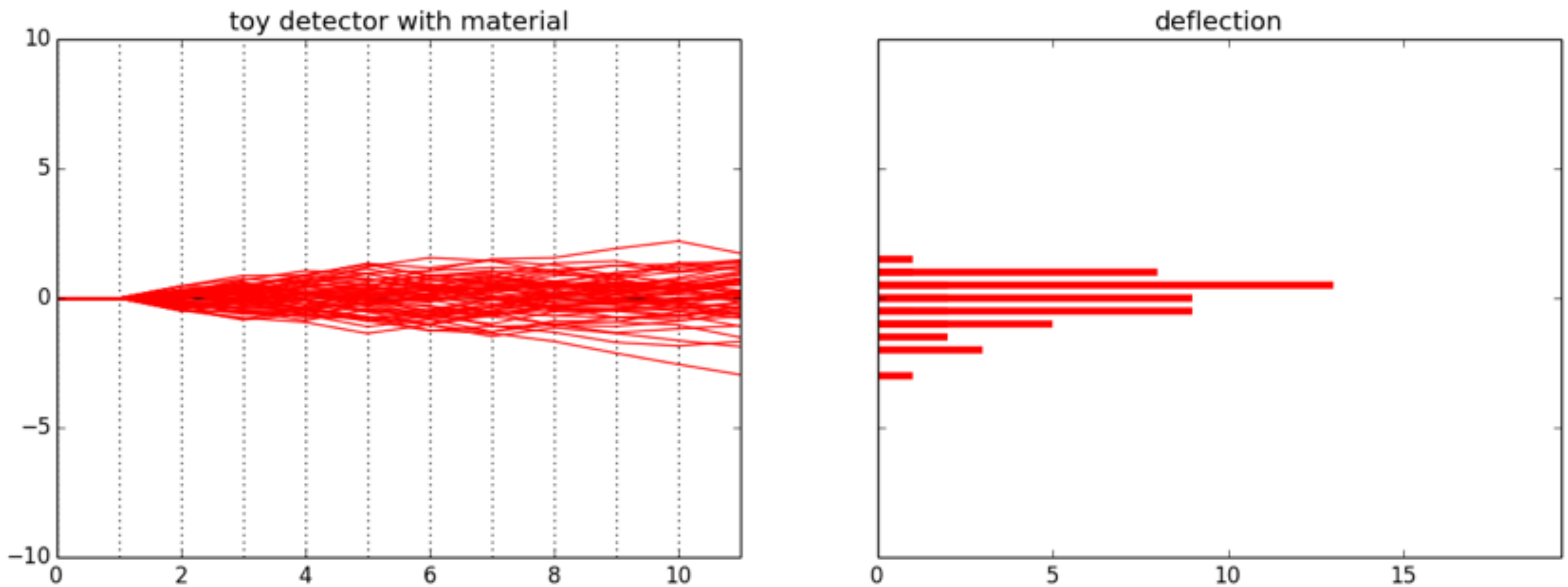

```
salzburg$ ipython -i --matplotlib=osx MultipleScattering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 1000 )
```

MC Toy: multiple scattering



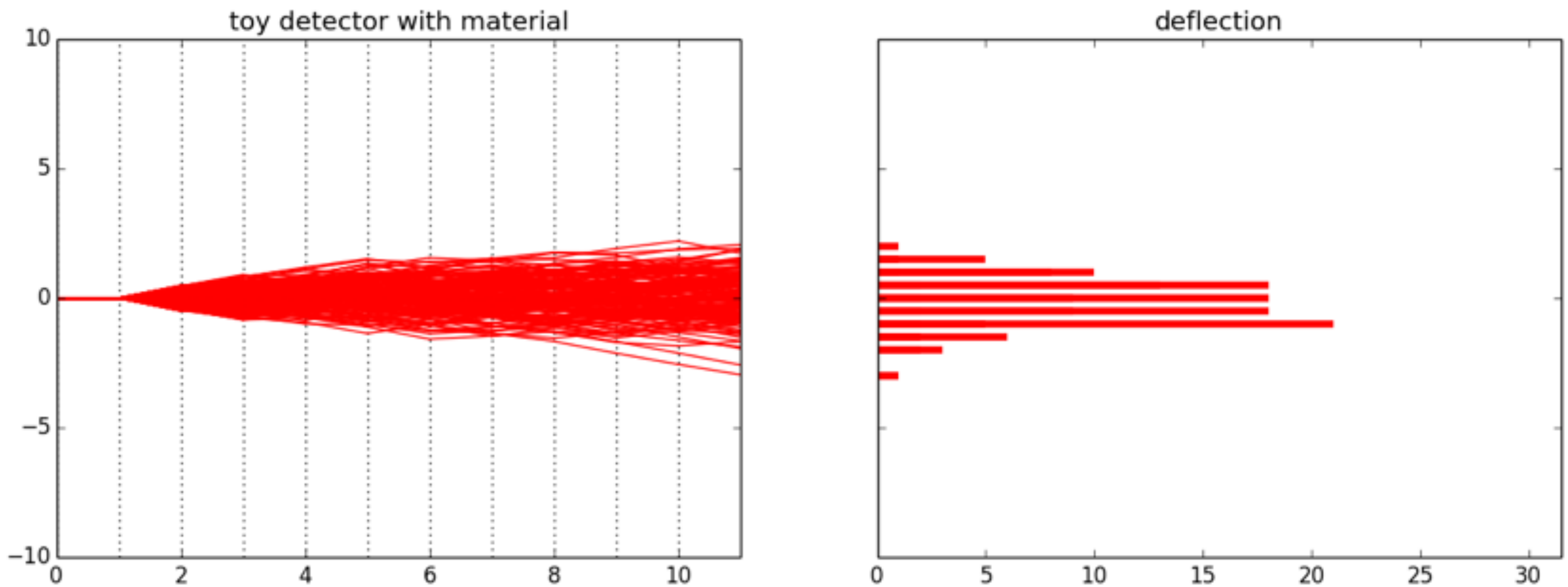

```
salzburg$ ipython -i --matplotlib=osx MultipleScattering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 1000 )
```

MC Toy: multiple scattering



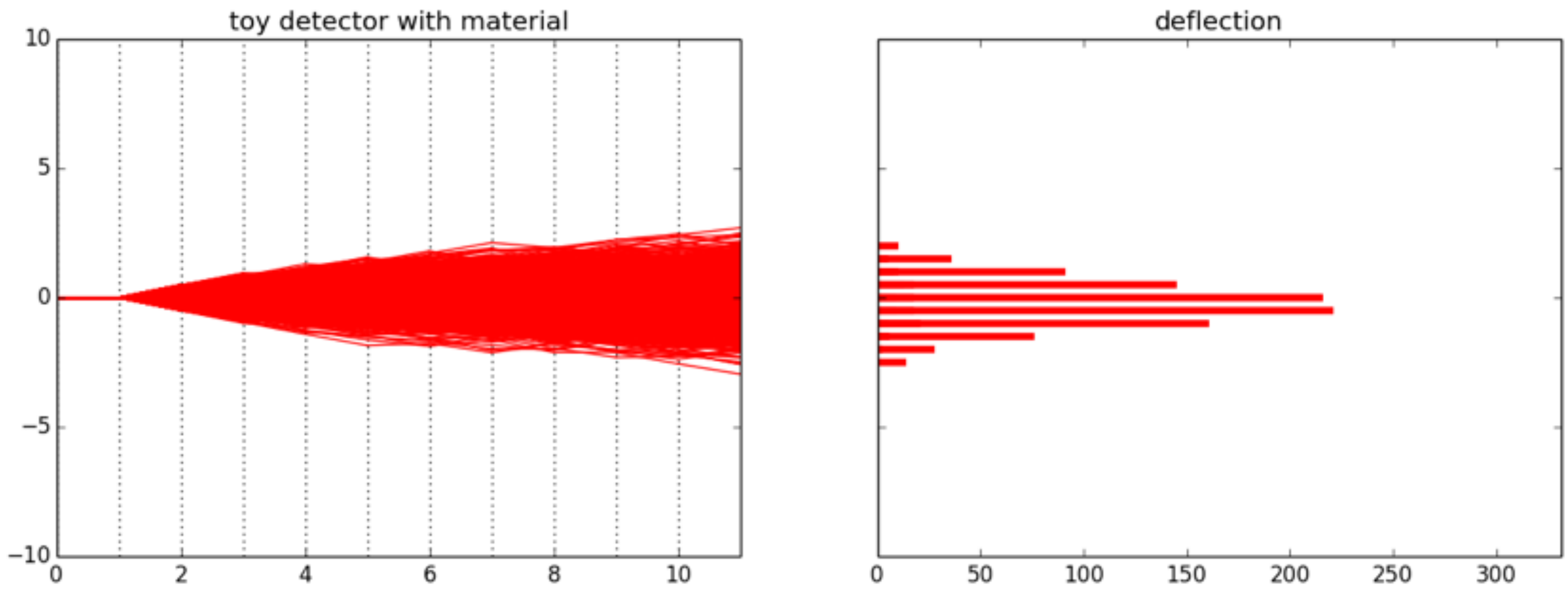

```
salzburg$ ipython -i --matplotlib=osx MultipleScattering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 1000 )
```

MC Toy: multiple scattering



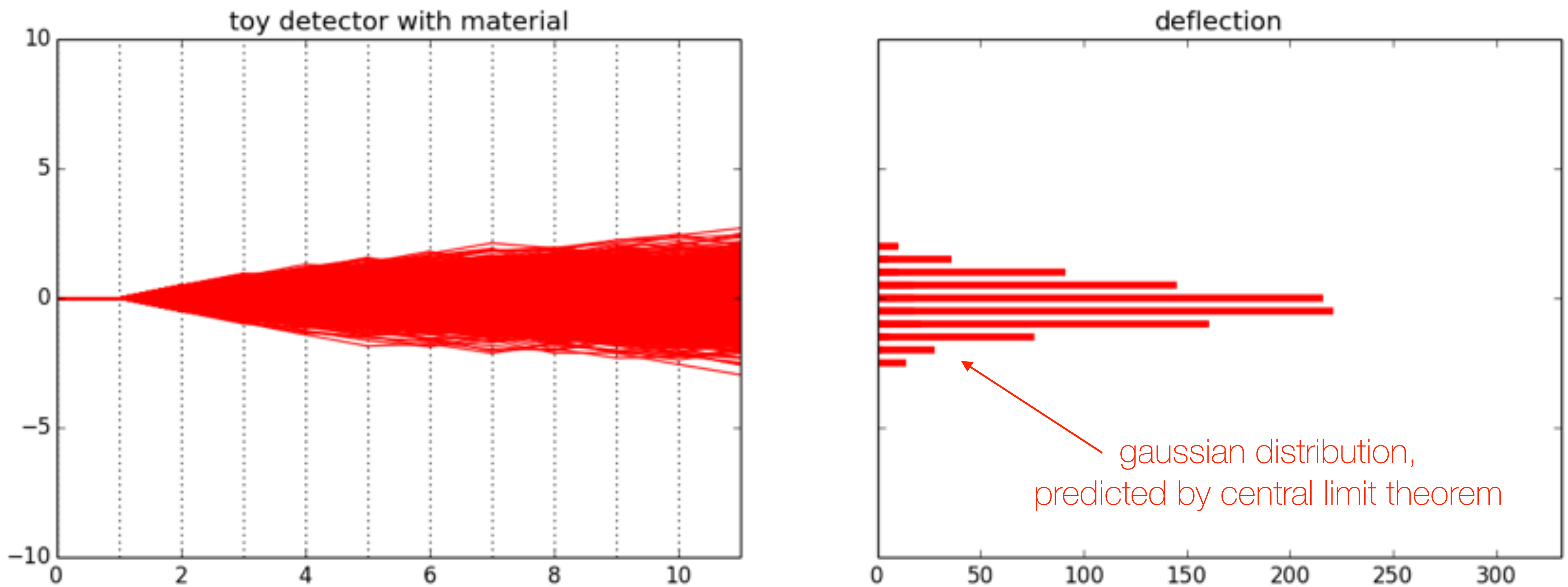

```
salzburg$ ipython -i --matplotlib=osx MultipleScattering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 1000 )
```

MC Toy: multiple scattering




```
salzburg$ ipython -i --matplotlib=osx MultipleScattering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 1000 )
```

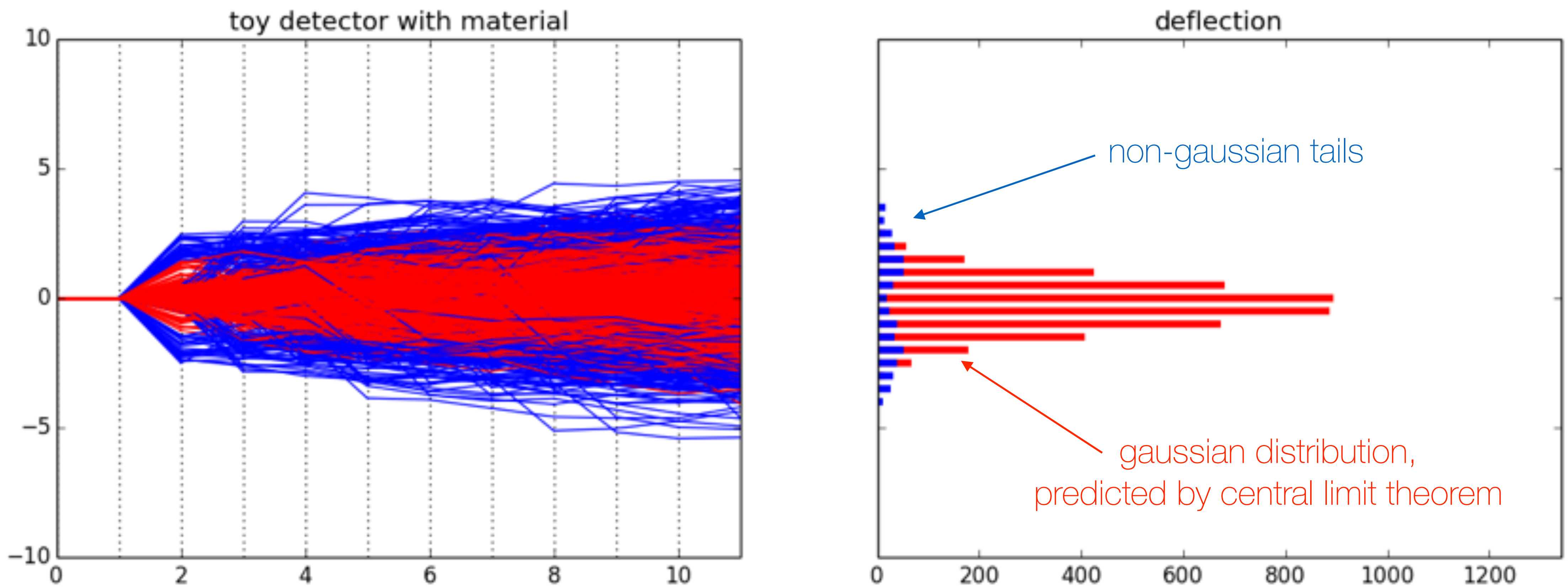
MC Toy: multiple scattering




```
salzburg$ ipython -i --matplotlib=osx MultipleScattering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 1000 )
```

MC Toy: multiple scattering

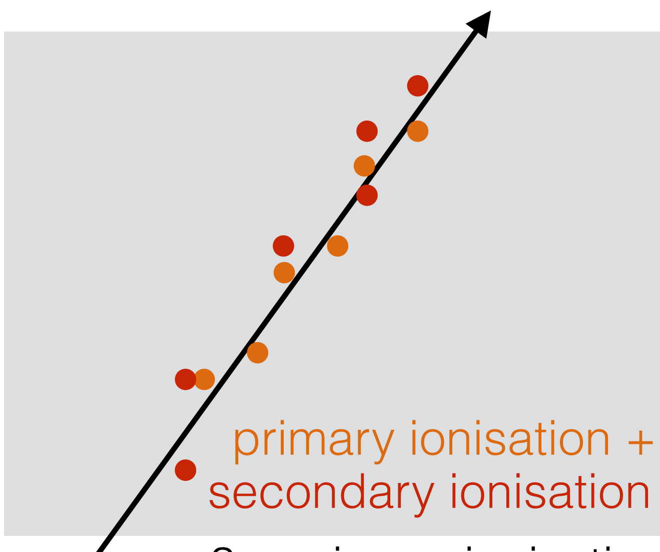
in the presence of multiple coulomb scattering and single large Rutherford scattering




```
salzburg$ ipython -i --matplotlib=osx MultipleScattering.py

In [1]: fig, plots = buildPixels()

In [2]: shoot(fig, plots, 5000, sfraction = 0.01 )
```

Energy loss

- charged particle loses energy when traversing material
  - ionisation loss : Bethe-Bloch

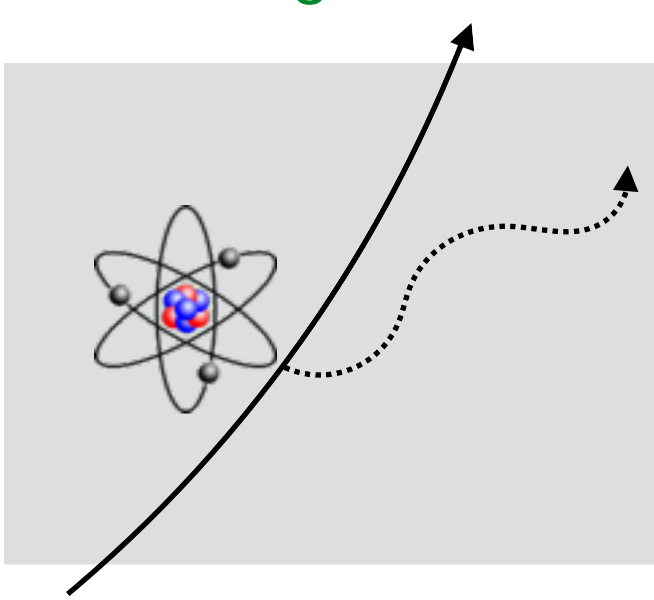


$\approx$ 3 x primary ionisation

$$(dE/dx)_{ion} = \alpha^2 2\pi N_a \lambda_e^2 \frac{Z m_e}{A\beta^2}\left[\ln\frac{2m_e\beta^2\gamma^2 E'_m}{I^2(Z)} - 2\beta^2 + 1/4\frac{E'^2_m}{E^2} - \delta\right]$$

N_a	$=$	$6.023 \cdot 10^{23}$, Avogadro's number
Z, A		atomic number and weight of the traversed medium
m, m_e		rest masses of the particle and the electron
β	$=$	p/E, where p is the particle momentum
γ	$=$	E/m
λ_e	$=$	$3.8616 \cdot 10^{-11}$ cm is the Compton wavelength of the electron
$I(Z)$		the mean ionisation potential of the medium,
E'_m		the maximum energy transferable to the electrons of the medium with $E'_m = 2m_e \frac{p^2}{m_e^2 + m^2 + 2m_e\sqrt{p^2+m^2}}$
δ		density correction.

  - bremsstrahlung: Bethe-Heitler



$$(dE/dx)_{rad} = 4\alpha N_A \frac{z^2 Z^2}{A}\left(\frac{1}{4\pi\epsilon_0}\frac{e^2}{mc^2}\right)^2 E \ \ln\frac{183}{Z^{\frac{1}{3}}} \propto \frac{E}{m^2}$$

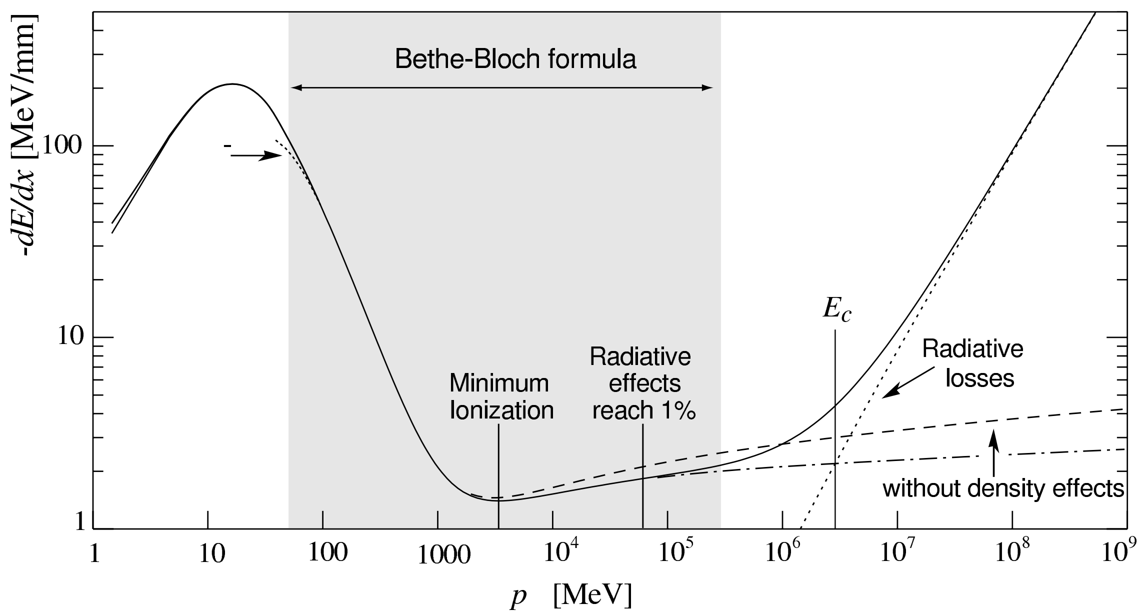
$$(dE/dx)_{rad} = -E_i/X_0$$

$$\boxed{X_0 = \frac{A}{4\alpha N_A \ Z^2 r_e^2 \ \ln\frac{183}{Z^{\frac{1}{3}}}}}$$

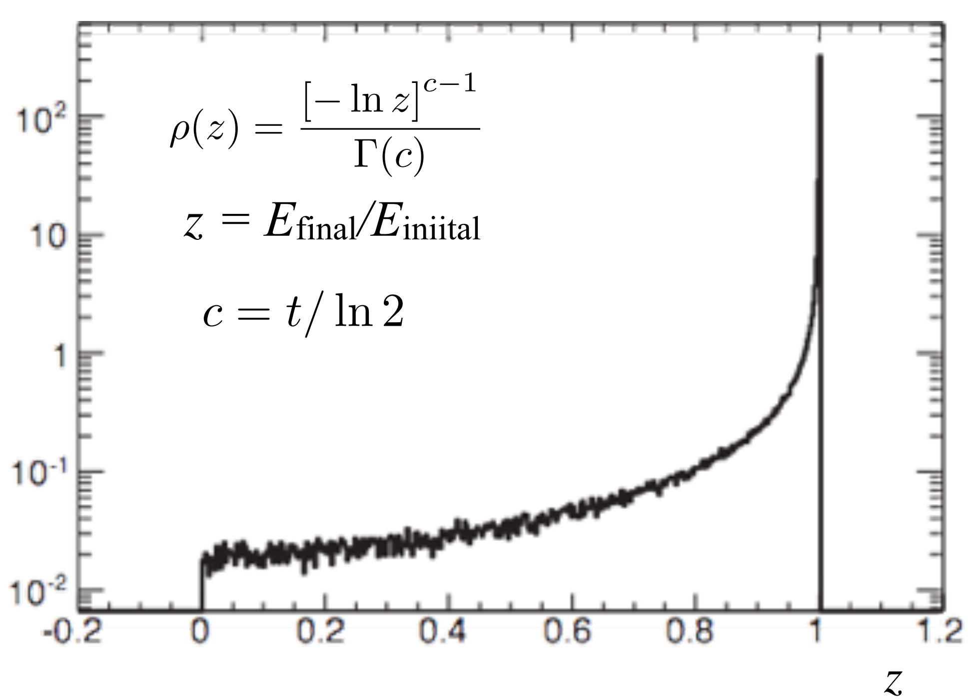
radiation length

Energy loss

- A charged particle loses energy when traversing material
  - ionisation loss: Bethe-Heitler

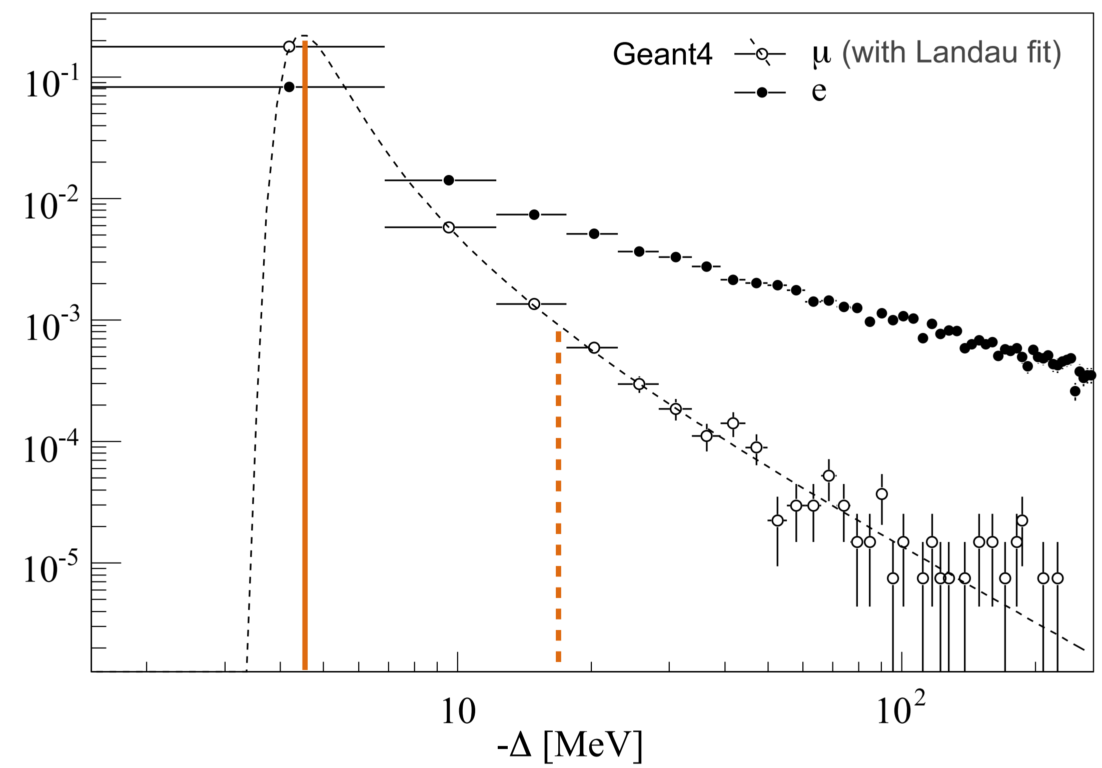


  - bremsstrahlung: Bethe-Bloch



Landau distribution with most probable value, mean value and Landau tail

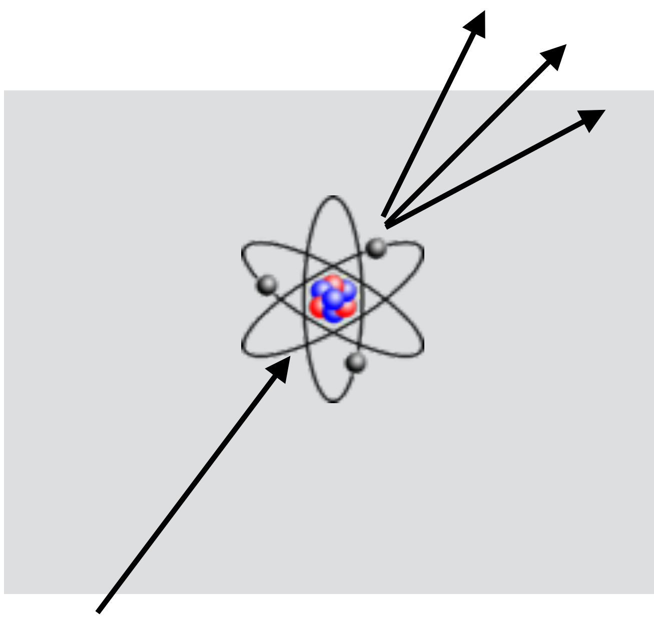
For Tracking detectors with rather little material: $\Delta E \ll E$ in



Very long tail with high probability to lose significant fraction of the particle energy

Hadronic interaction

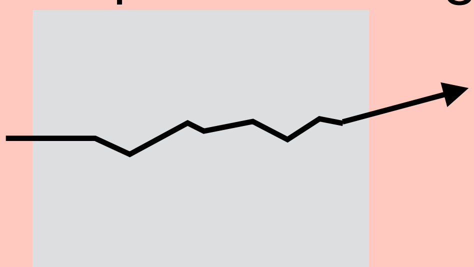
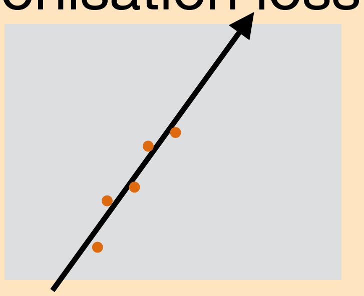
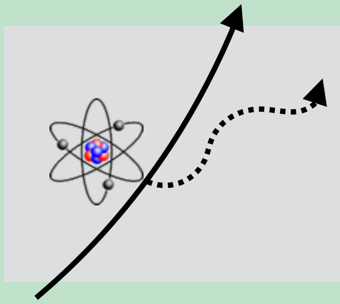
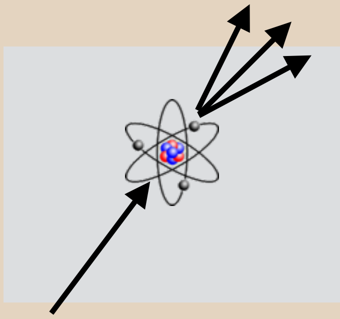
- Hadrons can undergo nuclear interaction with the detector material
  - leads usually to the destruction of the particle (as much as it concern us)



- there are many different processes that can happen in hadron-nucleus interactions
- resulting shower has hadronic, but also EM shower components
- nuclear interaction length defined as the mean path length Λ_0 by which the number of charged particles is traversing through matter is reduced by $1/e$

- Unfortunately most our charged particles are hadrons
  - this is the main source of track reconstruction inefficiency (if you wrote you algorithms correctly)

Summary - particle interaction with matter

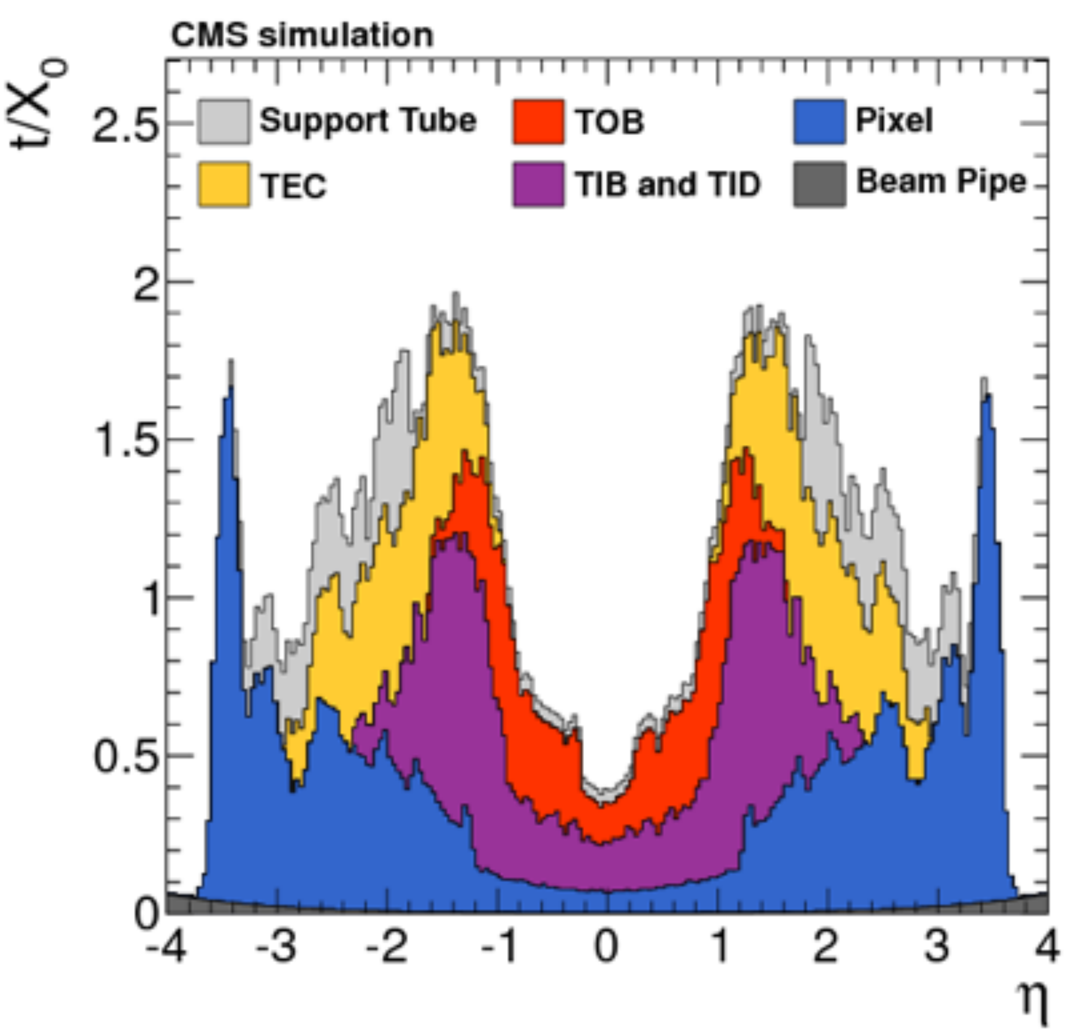
Type	particles	fund. parameter	characteristics	effect
Multiple Scattering 	all charged particle	radiation length X	almost gaussian average effect 0 depends ~ $1/\mathrm{p}$	deflects particles, increases measurement uncertainty
Ionisation loss 	all charged particle	effective density $A/Z * \rho$	small effect in tracker, small dependence on p	increases momentum uncertainty
Bremsstrahlung 	all charged particle, dominant for e	radiation length X	highly non-gaussian, depends	introduces measurement bias
Hadronic Int. 	all hadronic particles	nuclear interaction length Λ	destroys particle, rather constant effect in p	main source of track reconstruction inefficiency

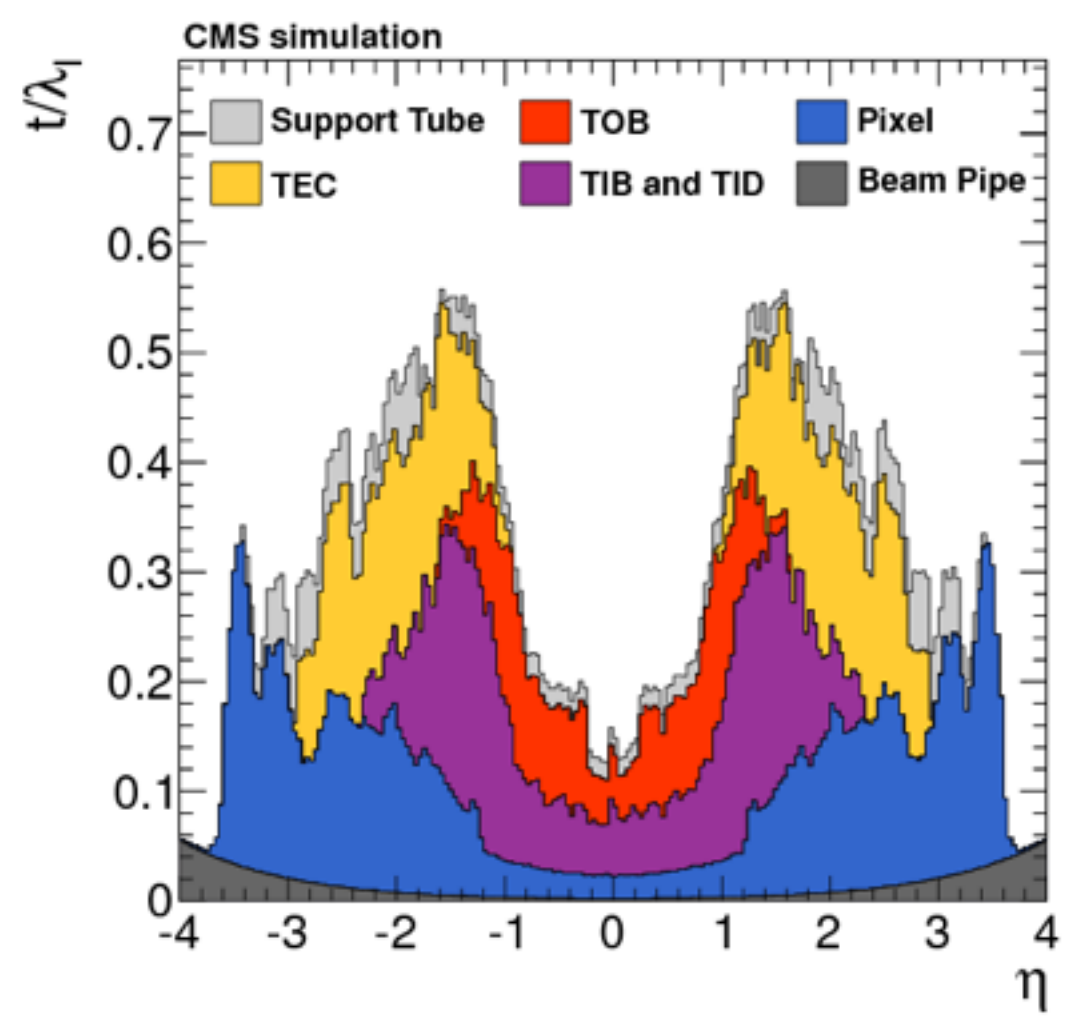
Detector material

- general aim in the construction of tracking detectors:
  - build them as light as possible

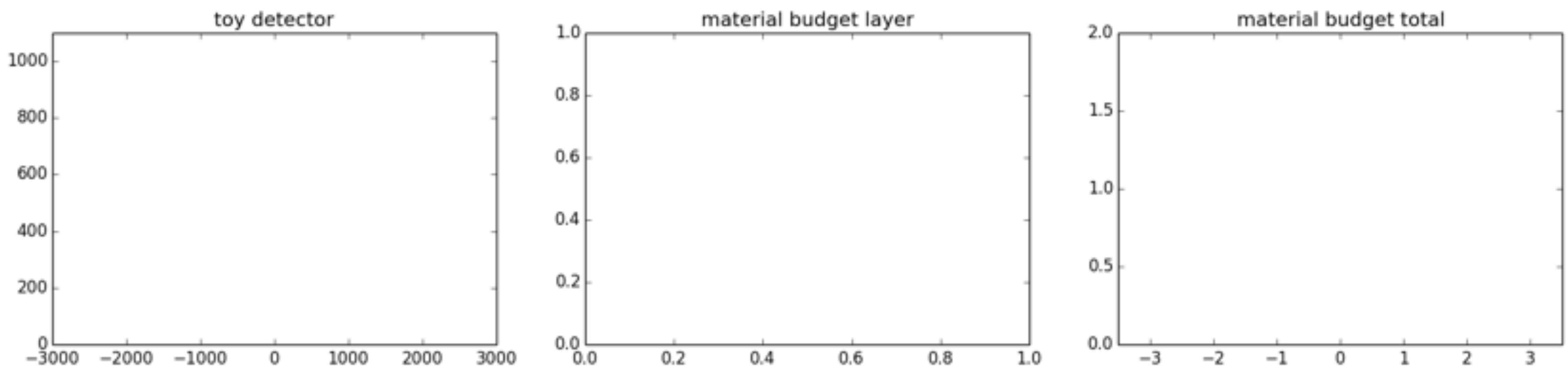
    material interactions disturb the measurement in the tracker itself

    tracker is usually before the calorimeter (material disturbs the calorimeter measurement)
  - two fundamental measures: radiation length X_0 and nuclear interaction length Λ_0



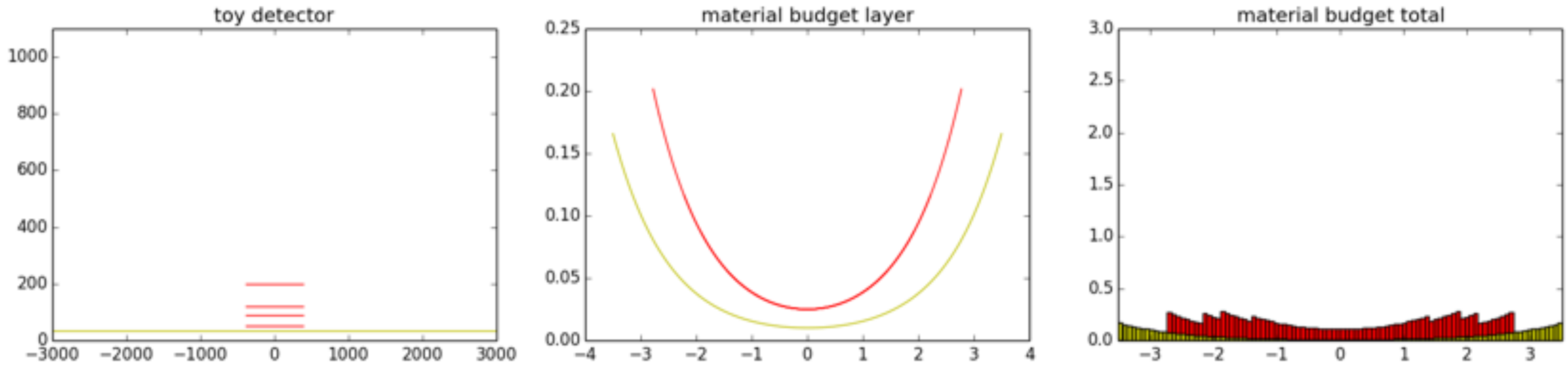


MC Toy: detector material



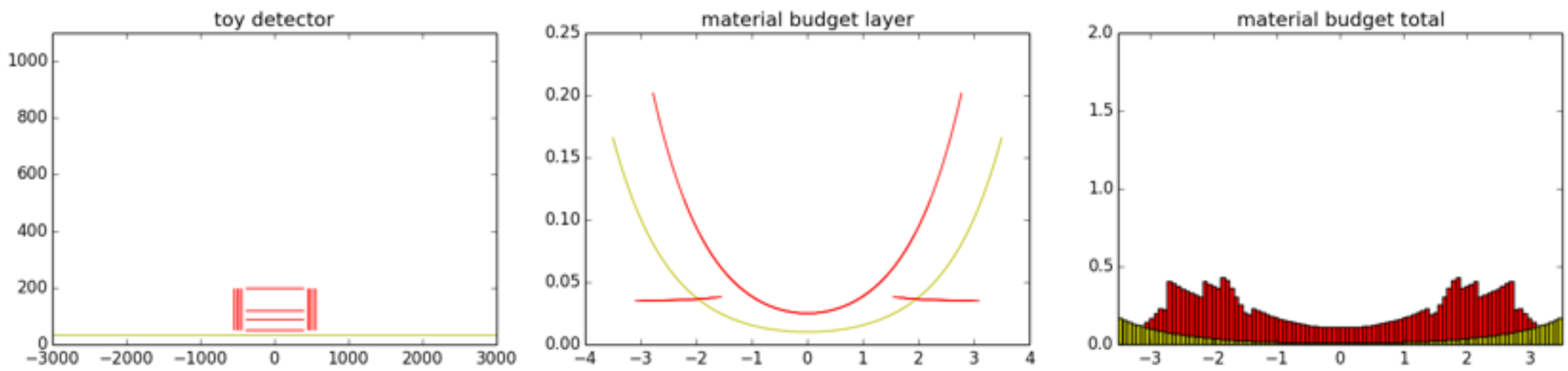

```
salzburg$ ipython -i --matplotlib=osx DetectorMaterial.py

In [1]: fig, plots = buildFrame()

In [2]: buildDetector(fig, plots)
```

MC Toy: detector material



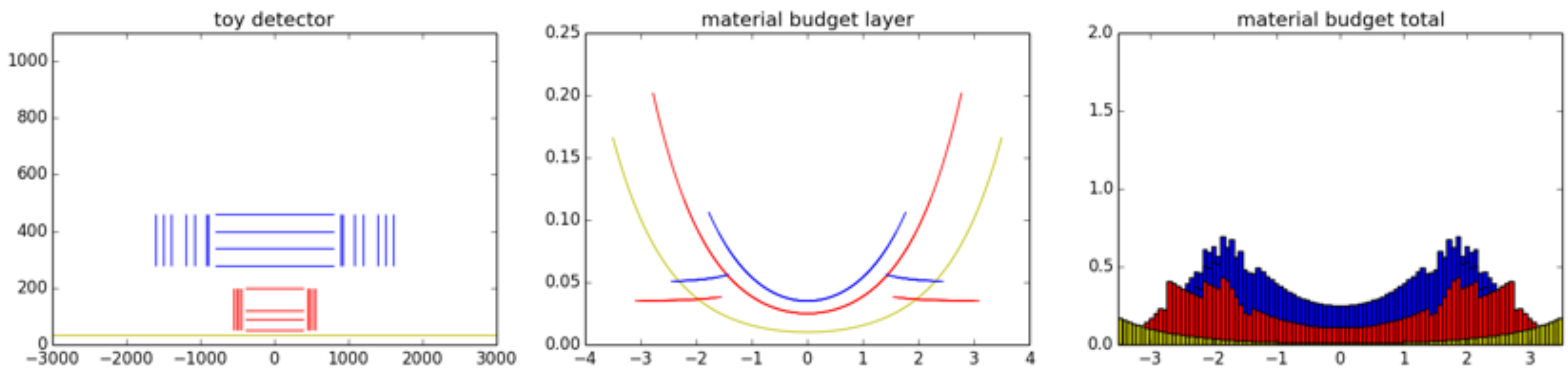

```
salzburg$ ipython -i --matplotlib=osx DetectorMaterial.py

In [1]: fig, plots = buildFrame()

In [2]: buildDetector(fig, plots)
```

MC Toy: detector material



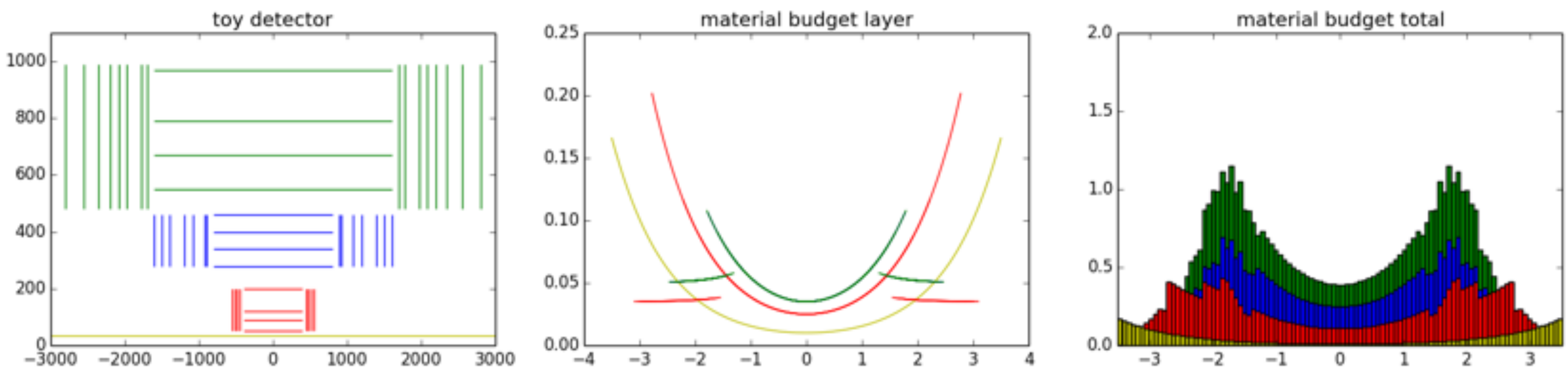

```
salzburg$ ipython -i --matplotlib=osx DetectorMaterial.py

In [1]: fig, plots = buildFrame()

In [2]: buildDetector(fig, plots)
```

MC Toy: detector material



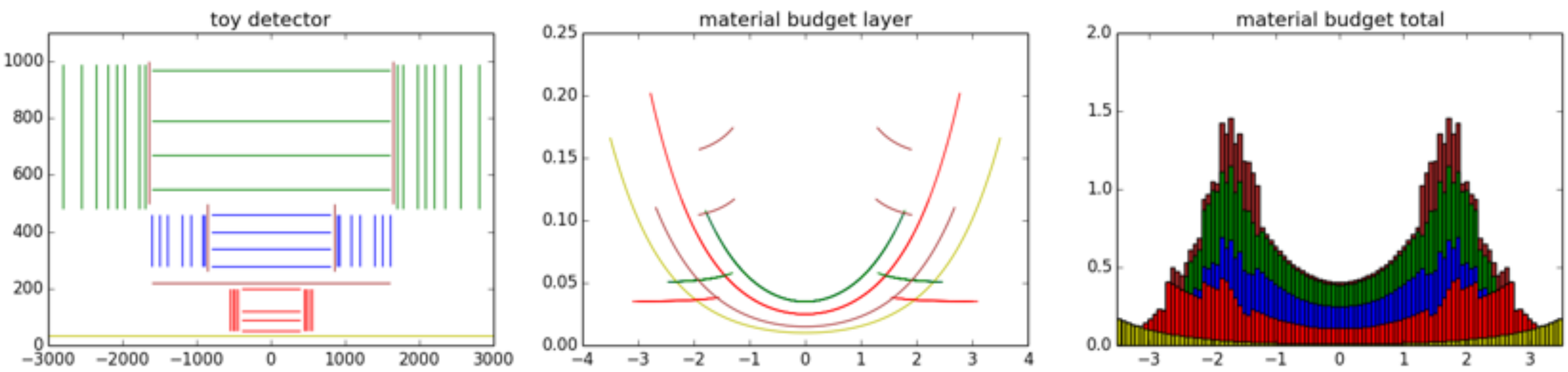

```
salzburg$ ipython -i --matplotlib=osx DetectorMaterial.py

In [1]: fig, plots = buildFrame()

In [2]: buildDetector(fig, plots)
```

MC Toy: detector material




```
salzburg$ ipython -i --matplotlib=osx DetectorMaterial.py

In [1]: fig, plots = buildFrame()

In [2]: buildDetector(fig, plots)
```

MC Toy: detector material

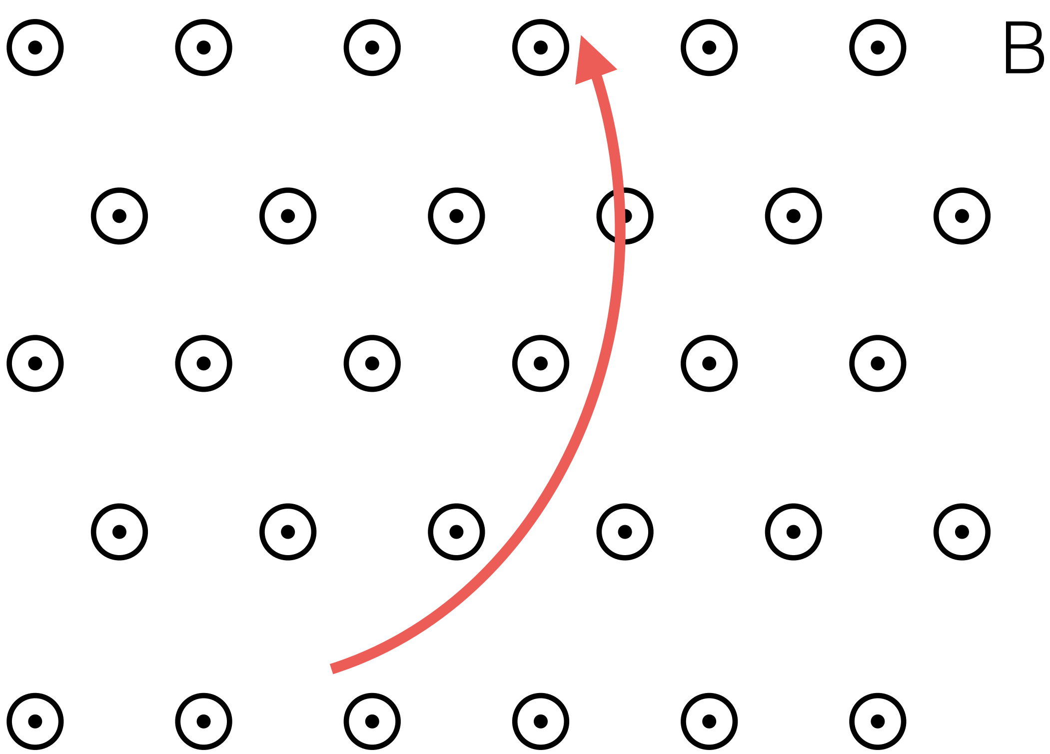



```
salzburg$ ipython -i --matplotlib=osx DetectorMaterial.py

In [1]: fig, plots = buildFrame()

In [2]: buildDetector(fig, plots)
```

The magnetic field

- A magnetic field is essential to bend the charged particles in order to measure their momenta
  - in a perfect homogenous field : circle in transverse direction
  - yields a helical track in a solenoidal field

    keep transverse & longitudinal components independent

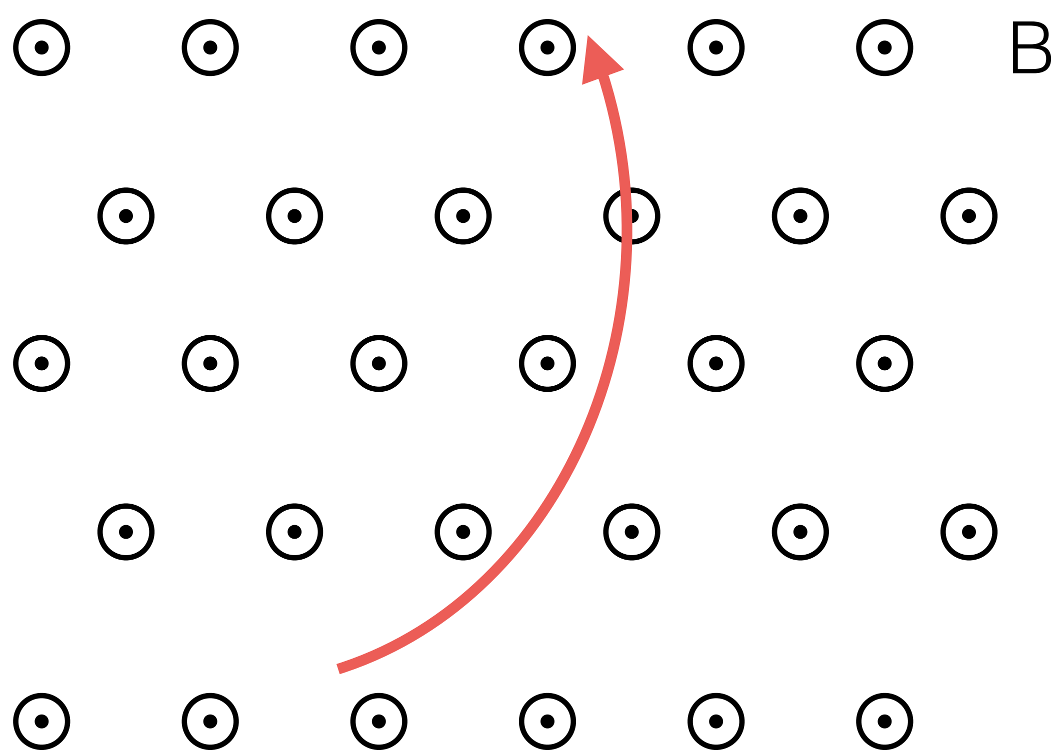


$$\frac{d^2\mathbf{r}}{ds^2} = \frac{q}{p}\left[\frac{d\mathbf{r}}{ds} \times \mathbf{B}(\mathbf{r})\right]$$

The magnetic field

- A magnetic field is essential to bend the charged particles in order to measure their momenta
  - in a perfect homogenous field : circle in transverse direction
  - yields a helical track in a solenoidal field

    keep transverse & longitudinal components independent



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

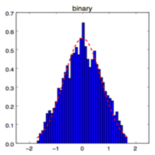
CDF $\mathbf{q}'' = (l_1, l_2, \phi, \cot(\theta), C)$

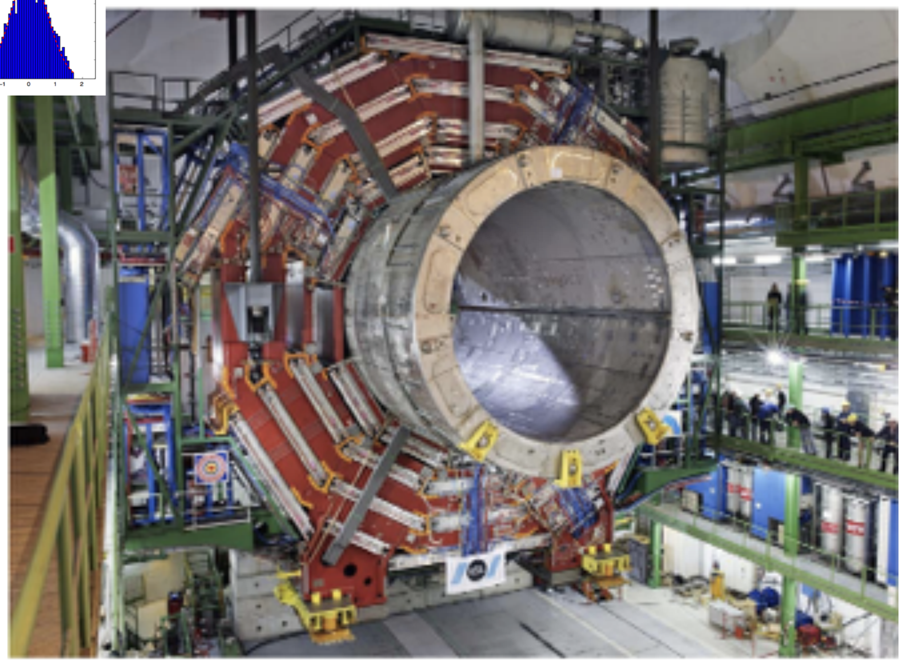
CMS $\mathbf{q}' = (l_1, l_2, \phi, \lambda, q/p)$

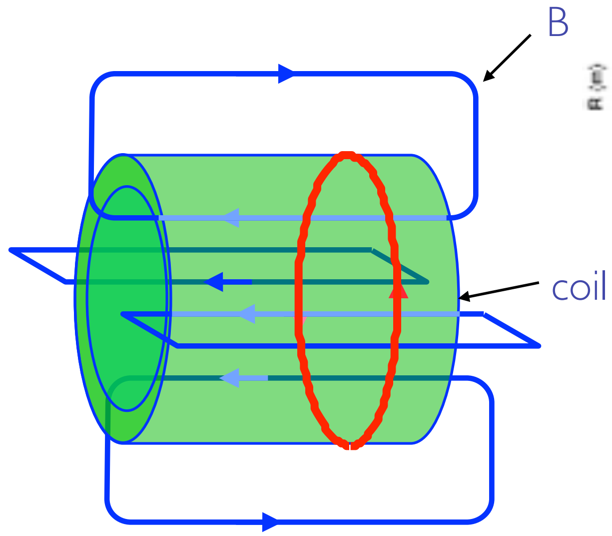
ATLAS $\mathbf{q} = (l_1, l_2, \phi, \theta, q/p)$

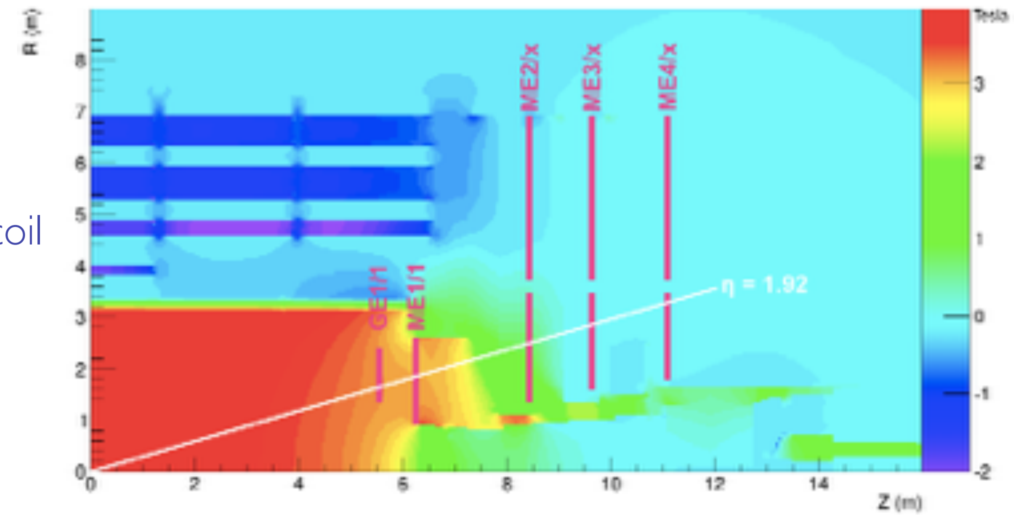
Realistic magnetic fields: CMS & ATLAS

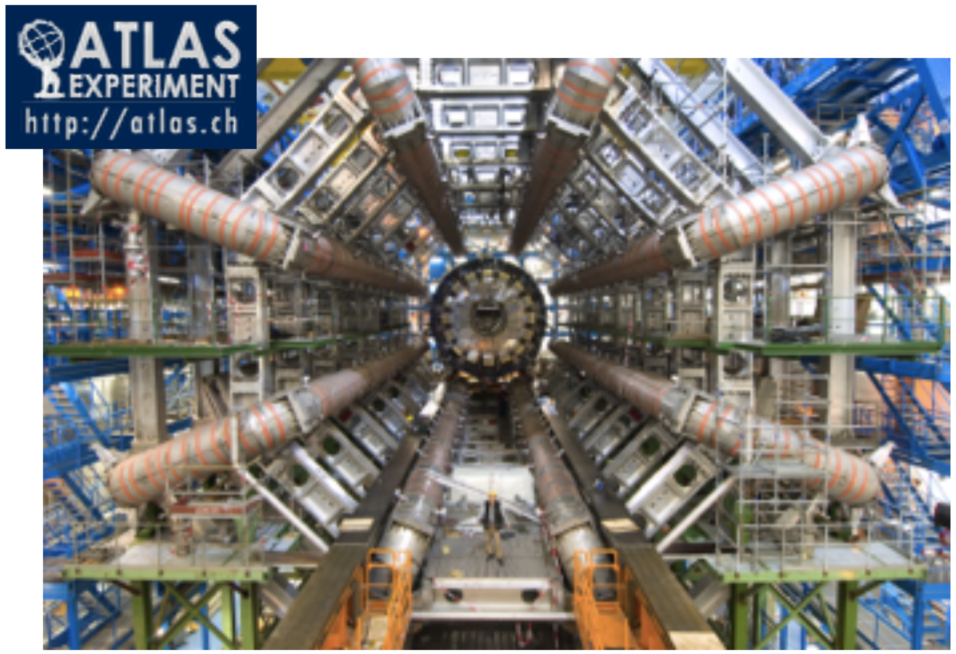
these are not homogeneous magnetic fields !

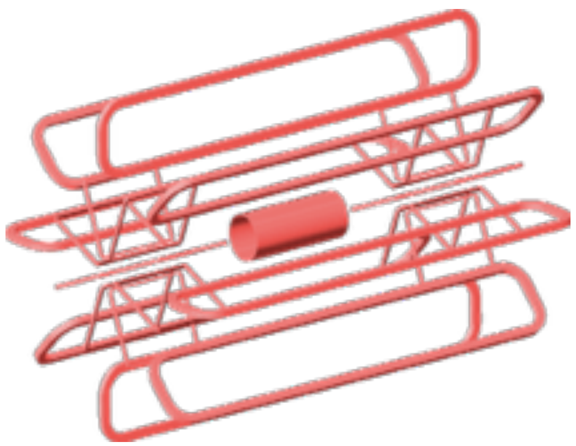


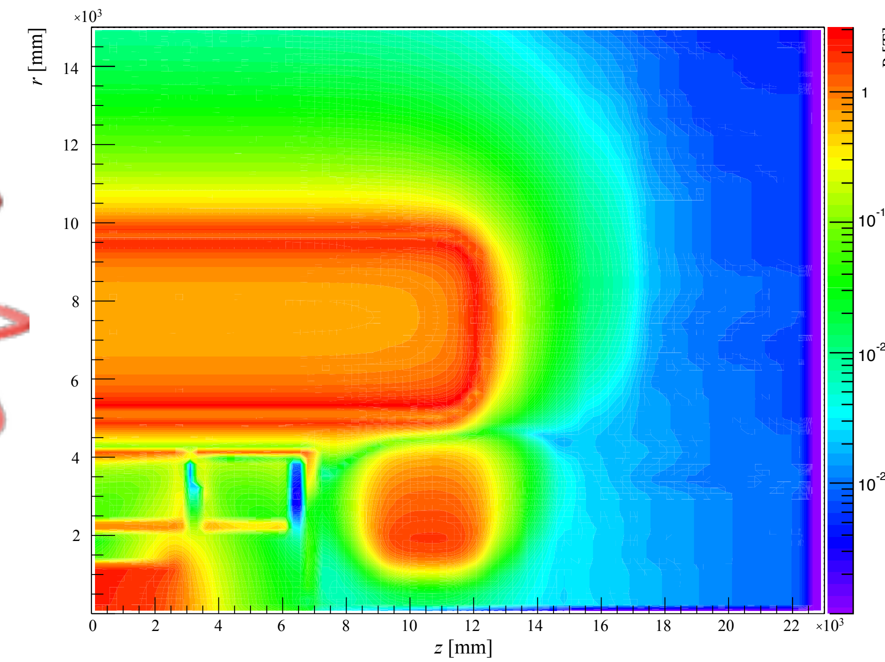












Track propagation

- problems to solve
  - transport of track parameters through the magnetic field

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

  - application of material effects according to the detector material

$$\underbrace{\frac{d^2\mathbf{r}}{ds^2} = \frac{q}{p}\left[\frac{d\mathbf{r}}{ds} \times \mathbf{B}(\mathbf{r})\right] + \boxed{g(p,\mathbf{r})\frac{d\mathbf{r}}{ds}}}_{\text{solve this for any } \mathbf{B}(\mathbf{r})}$$

deterministic energy loss treatment

we need a numerical integration method !

Numerical integration

- Re-formulate the equation of motion as a movement along *z*

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

$$\frac{d^2x}{dz^2} = \frac{q}{p}R\left[\frac{dx}{dz}\frac{dy}{dz}B_x - \left(1+\left(\frac{dx}{dz}\right)^2\right)B_y + \frac{dy}{dz}B_z\right]$$

$$\frac{d^2y}{dz^2} = \frac{q}{p}R\left[\left(1+\left(\frac{dy}{dz}\right)^2\right)B_x - \frac{dx}{dz}\frac{dy}{dz}B_y - \frac{dx}{dz}B_z\right]$$

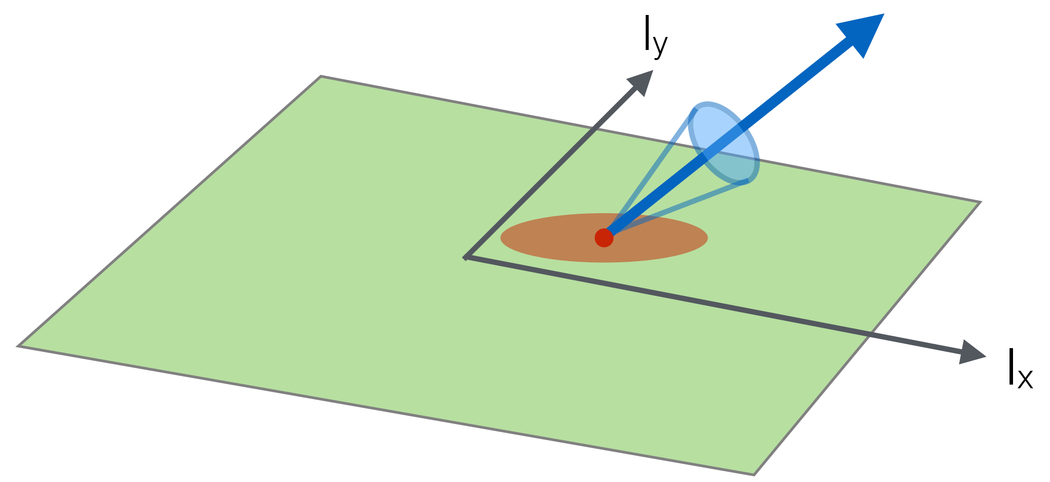
$$R = \frac{ds}{dz} = \sqrt{1+\left(\frac{dx}{dz}\right)^2+\left(\frac{dy}{dz}\right)^2}$$

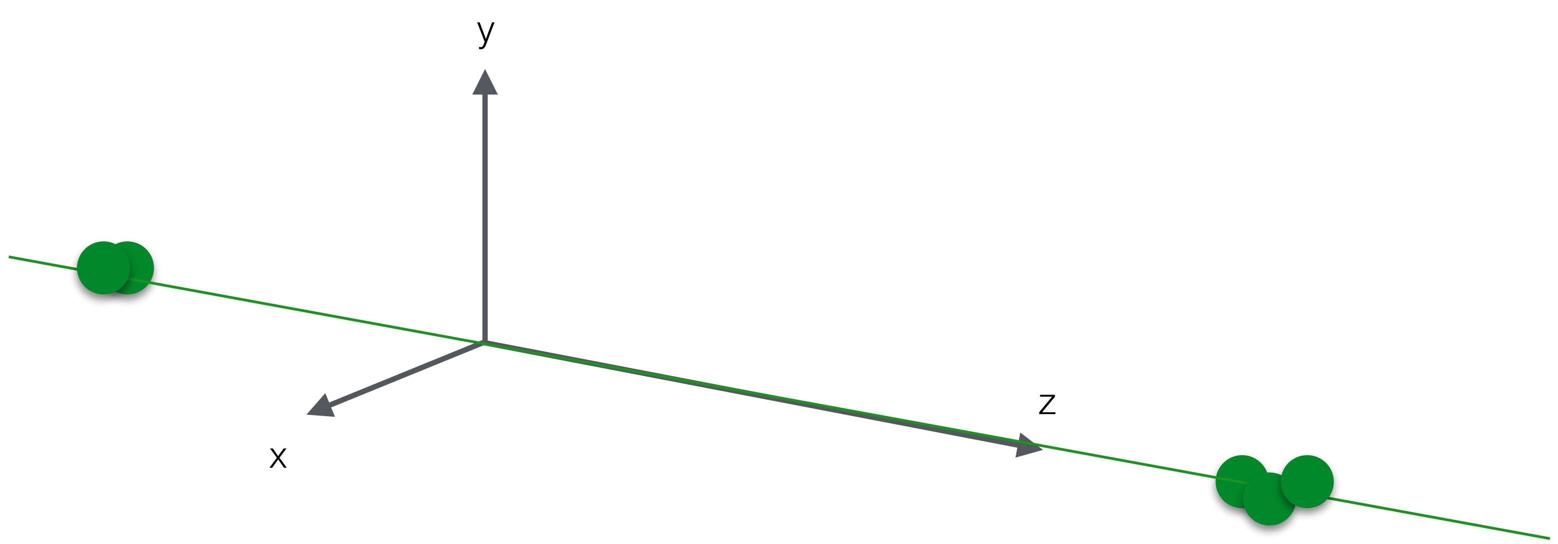
- Integrate to solve for *x(z)* and *y(z)* :

- Numerical integration methods:
  - Euler's method
  - Midpoint method
  - Runge-Kutta integration

Track propagation

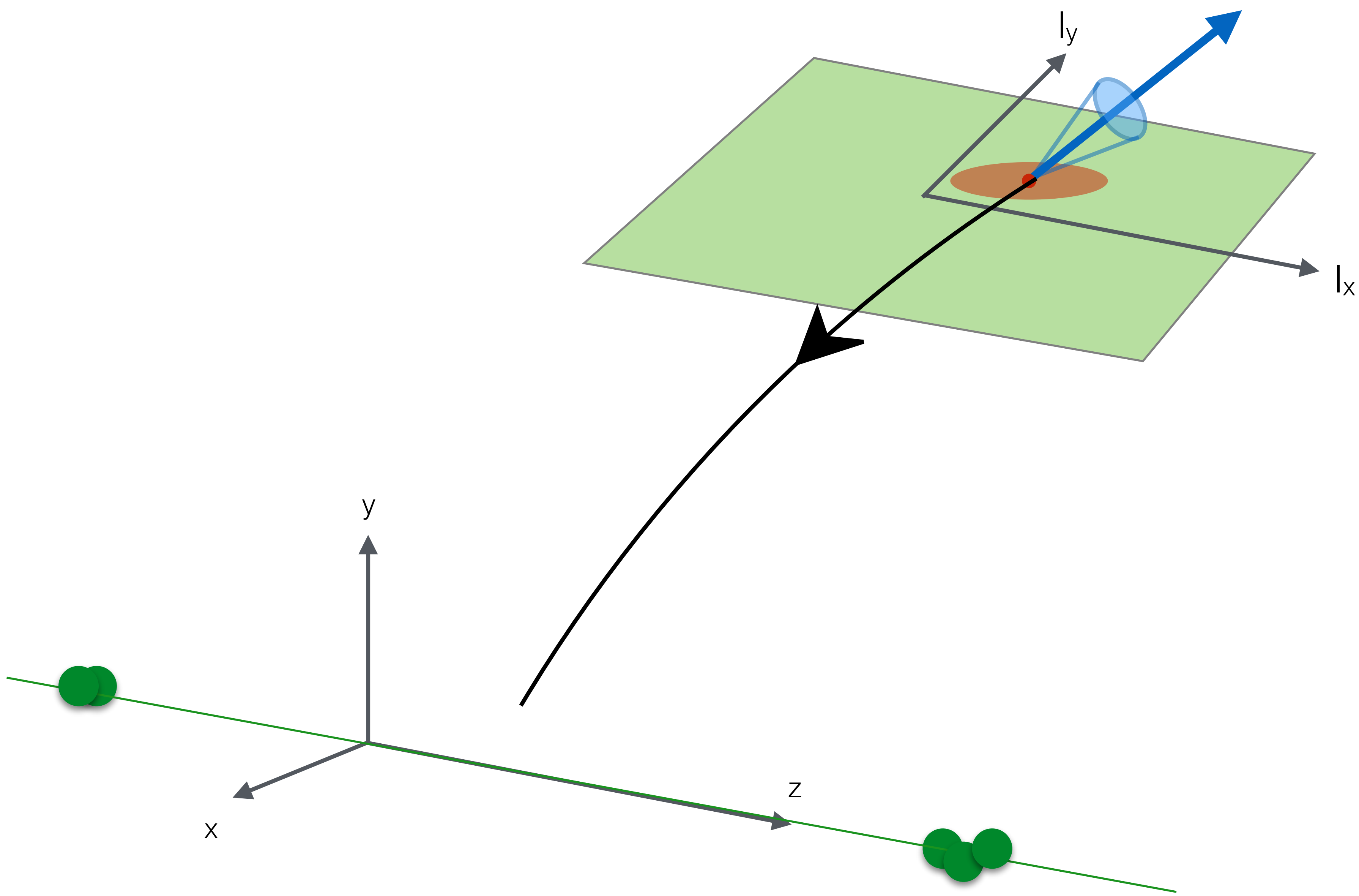
- Many components of the track reconstruction need the expression of the track at different places (i.e. surfaces) in the detector





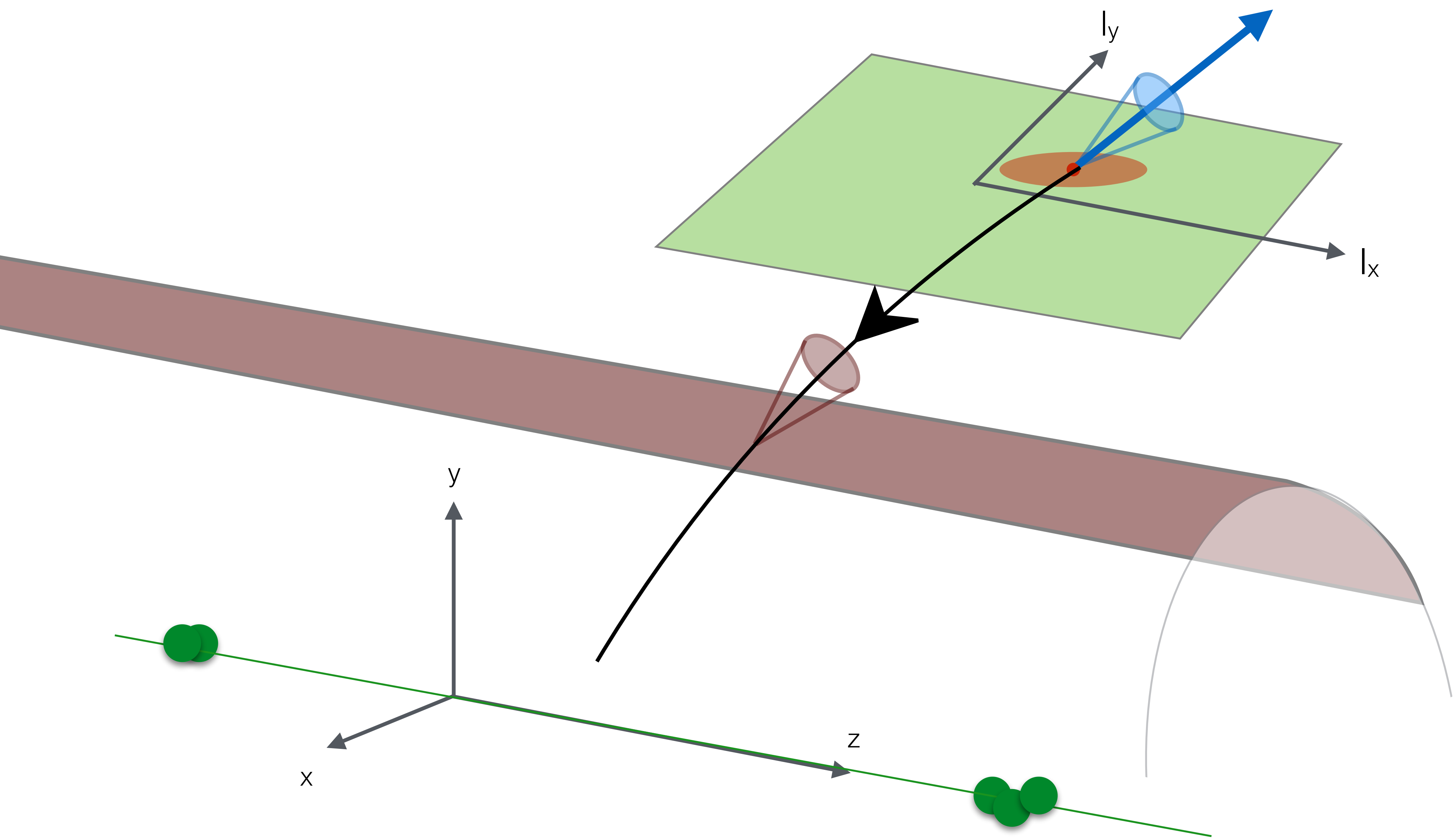
Track propagation

- Many components of the track reconstruction need the expression of the track at different places (i.e. surfaces) in the detector



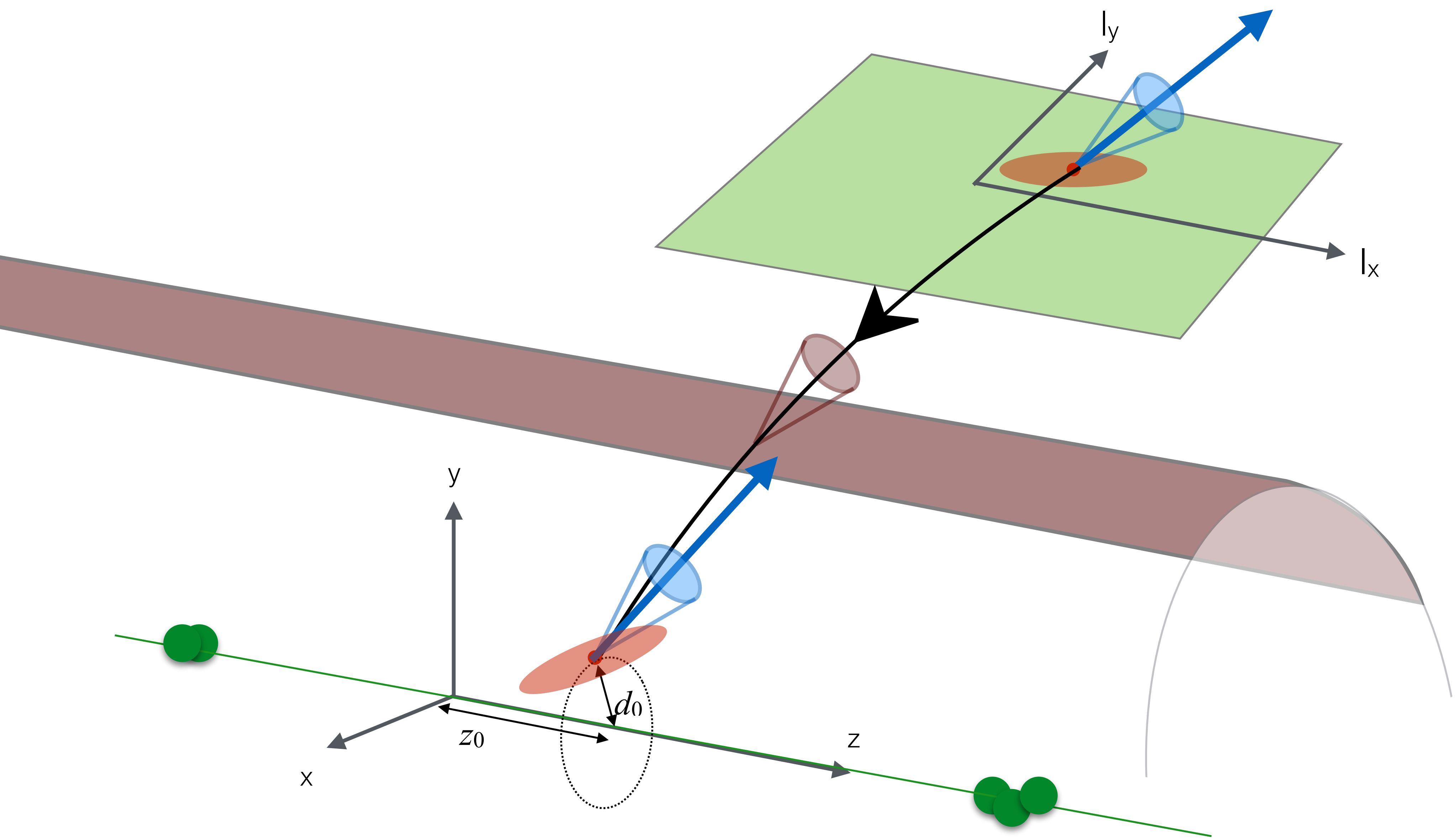
Track propagation

- Many components of the track reconstruction need the expression of the track at different places (i.e. surfaces) in the detector

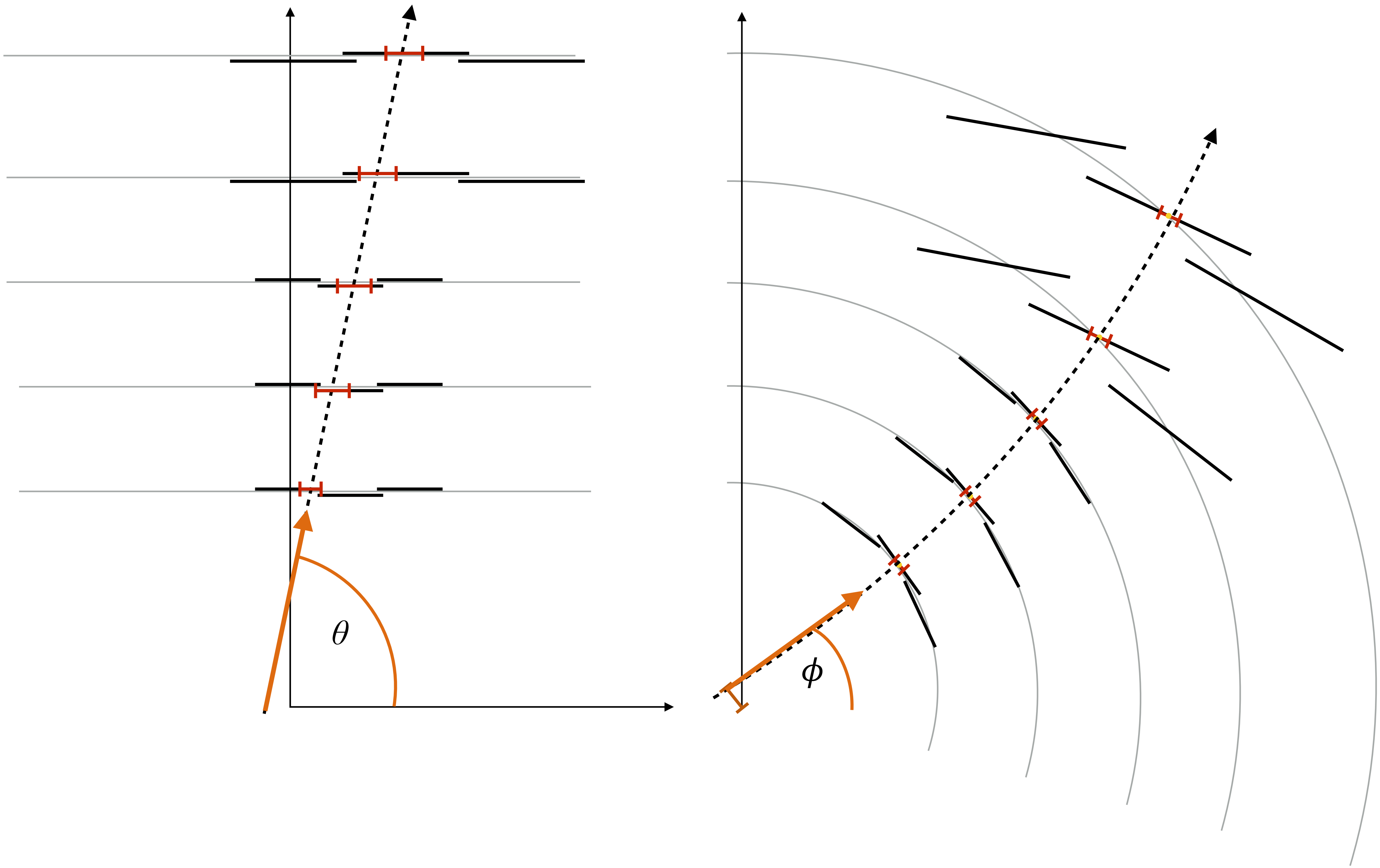


Track propagation

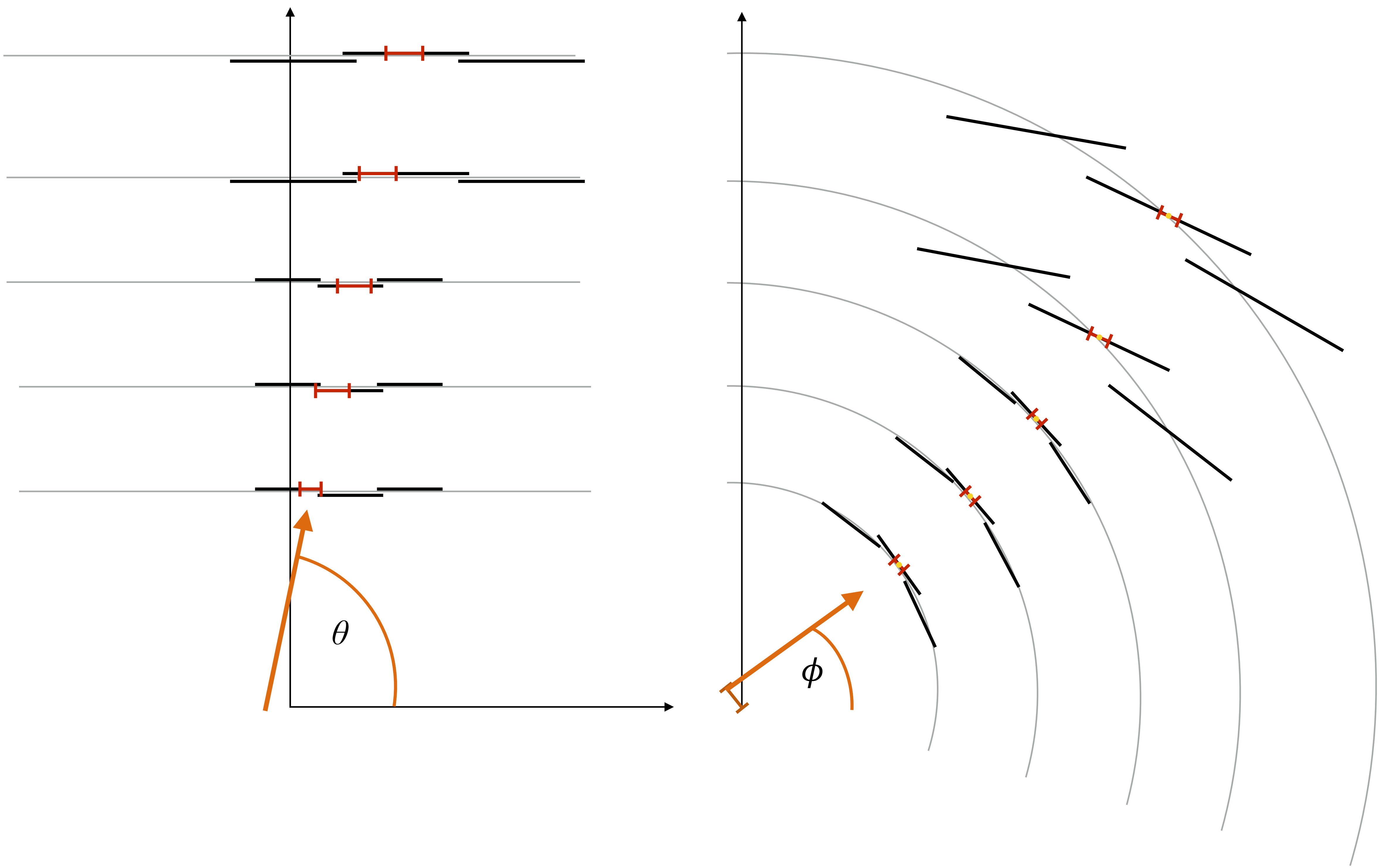
- Many components of the track reconstruction need the expression of the track at different places (i.e. surfaces) in the detector



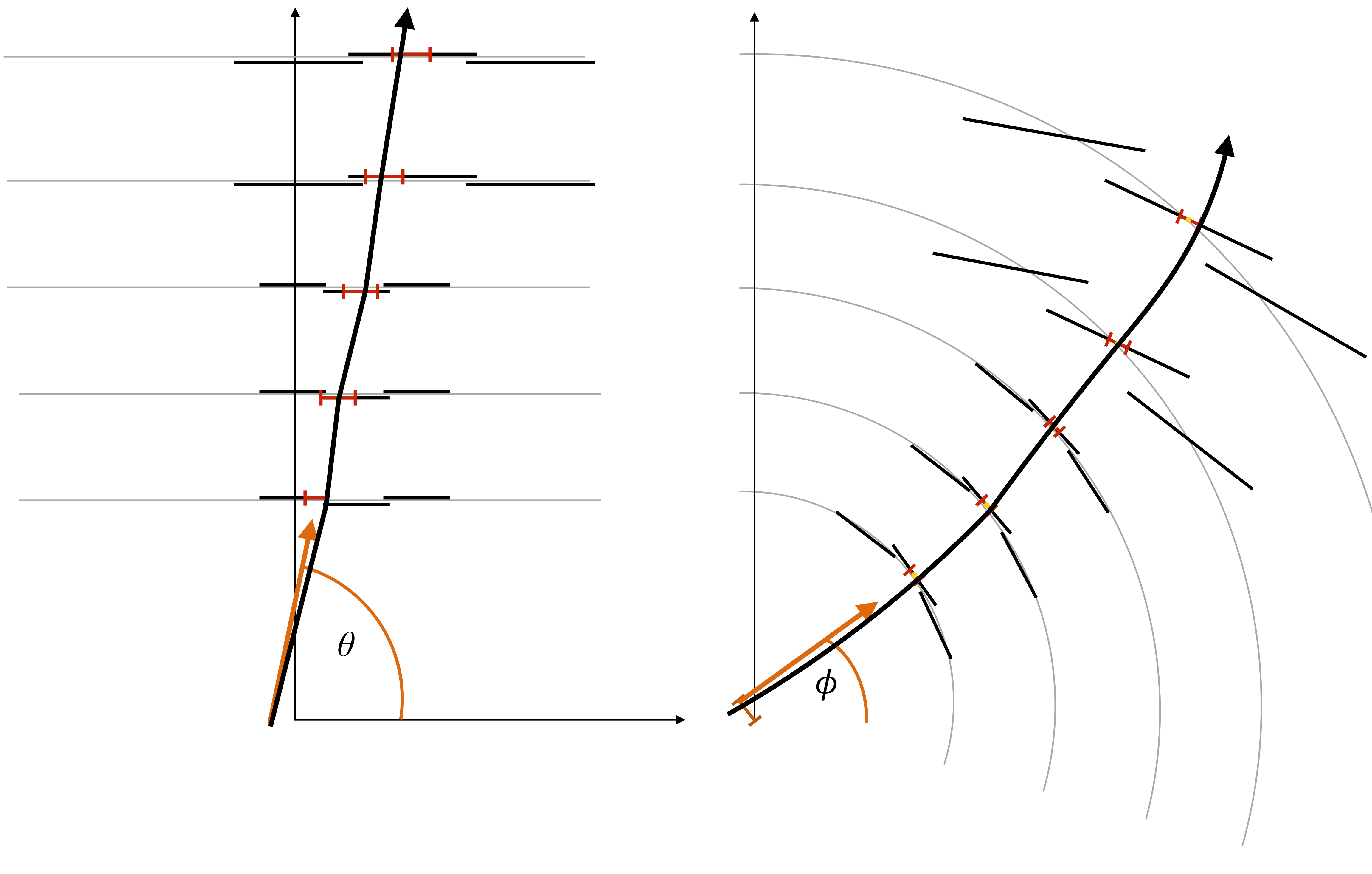
Recap for today



Recap for today



Recap for today



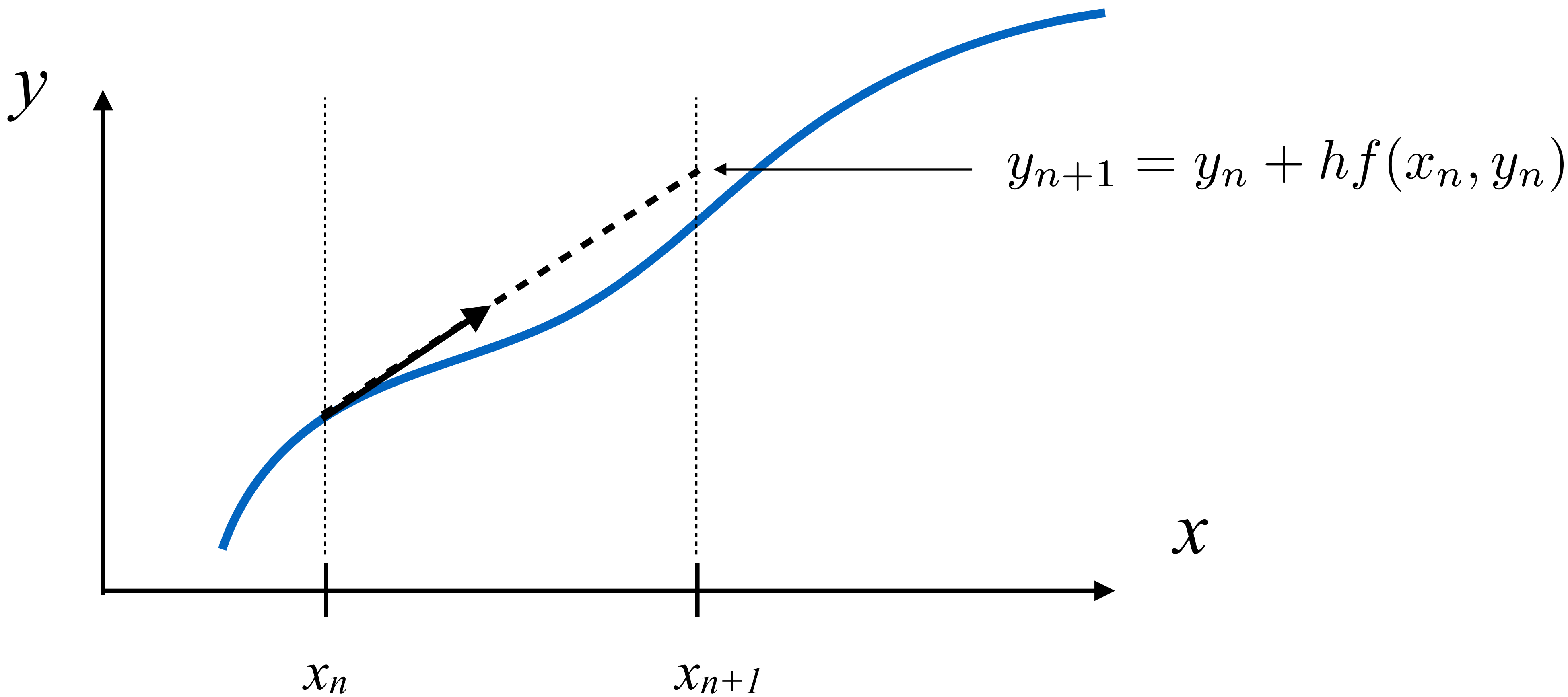
Summary - Tracking detectors

	ALICE	ATLAS	CMS
R inner	3.9 cm	5.0 cm	4.4 cm
R outer	3.7 m	1.1 m	1.1 m
Length	5 m	5.4 m	5.8 m
$\|\eta\|$ range	0.9	2.5	2.5
B field	0.5 T	2 T	4 T
Total X_0 near η=0	0.08 (ITS) + 0.035 (TPC) + 0.234 (TRD)	0.3	0.4
Power	6 kW (ITS)	70 kW	60 kW
rϕ resolution near outer radius	~ 800 μm TPC ~ 500 μm TRD	130 μm per TRT straw	35 μm per strip layer
p_T resolution at 1GeV and at 100 GeV	0.7% 3% (in pp)	1.3% 3.8%	0.7% 1.5%

Numerical integration in a nutshell

$\partial y/\partial x = f(x,y)$

- Euler method with start values x_n, y_n
  - what is the function value at y_{n+1} at $x_{n+1} = x_n + h$?

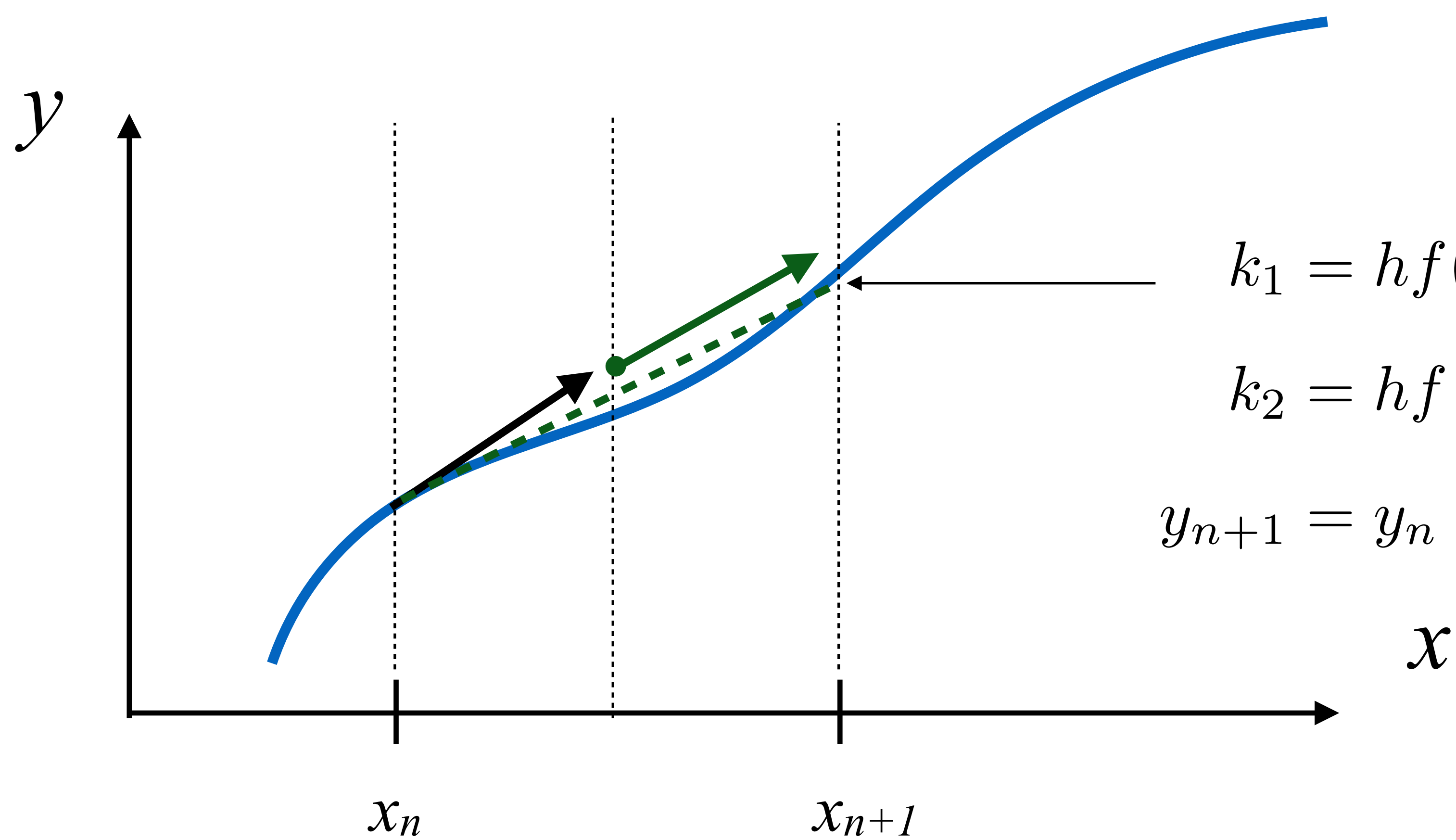


Accuracy: 1st order

Numerical integration in a nutshell

$\partial y/\partial x = f(x,y)$

- Midpoint method with start values x_n, y_n
  - what is the function value at y_{n+1} at $x_{n+1} = x_n + h$?



$$k_1 = hf(x_n, y_n)$$
$$k_2 = hf\left(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}k_1\right)$$
$$y_{n+1} = y_n + k_2 + O(h^3)$$

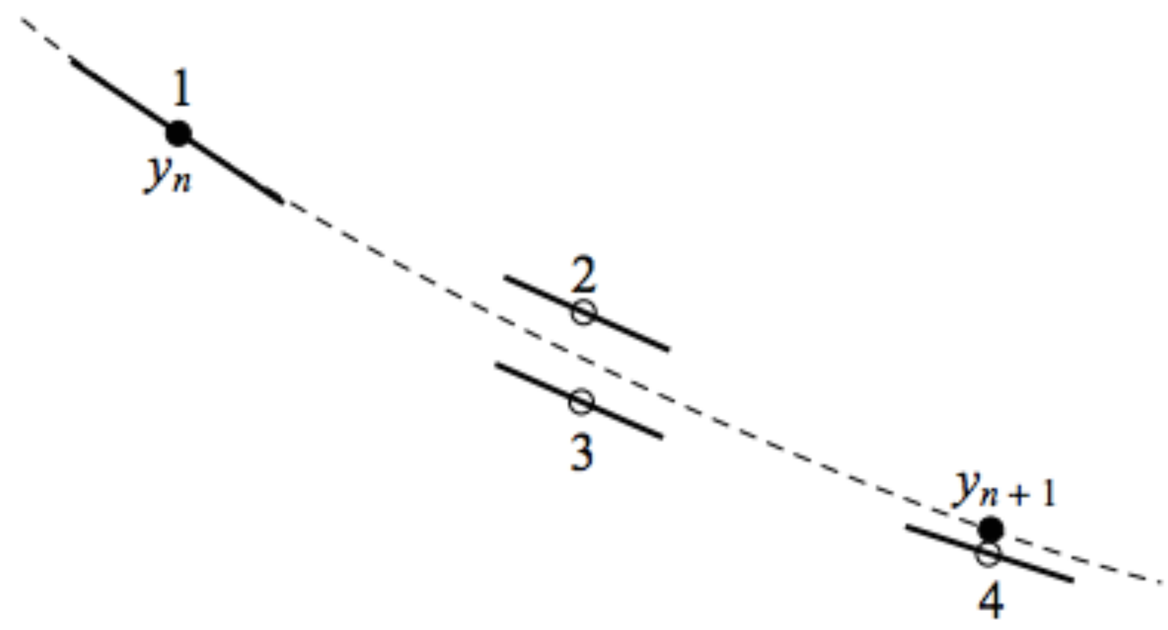
  - on the step to $x_{n+1} = x_n + h$ you stop at the midpoint and take this derivate for the evaluation of your final value from the full step

Accuracy: 2nd order

ADDON

Numerical integration in a nutshell

- Runge-Kutta method with start values

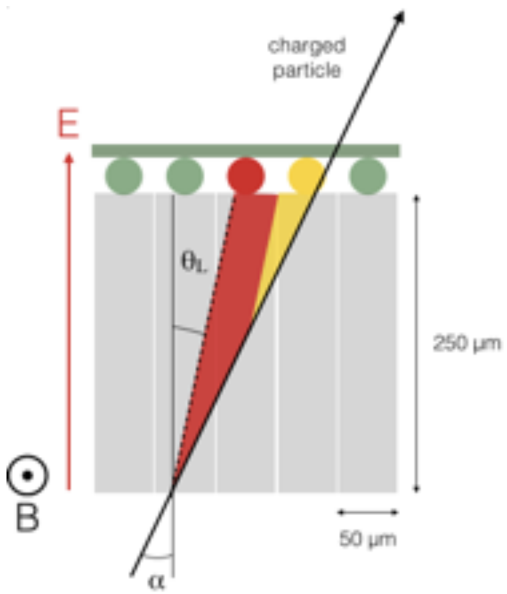


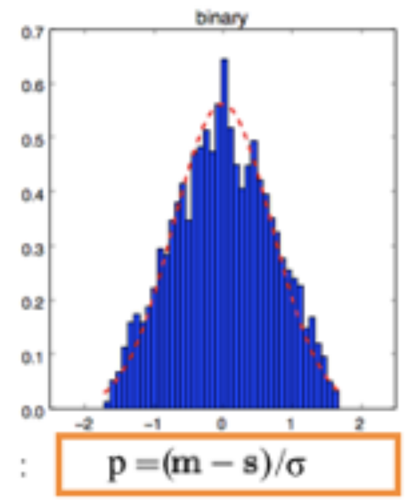
Fourth-order Runge-Kutta method. In each step the derivative is evaluated four times: once at the initial point, twice at trial midpoints, and once at a trial endpoint. From these derivatives the final function value (shown as a filled dot) is calculated.

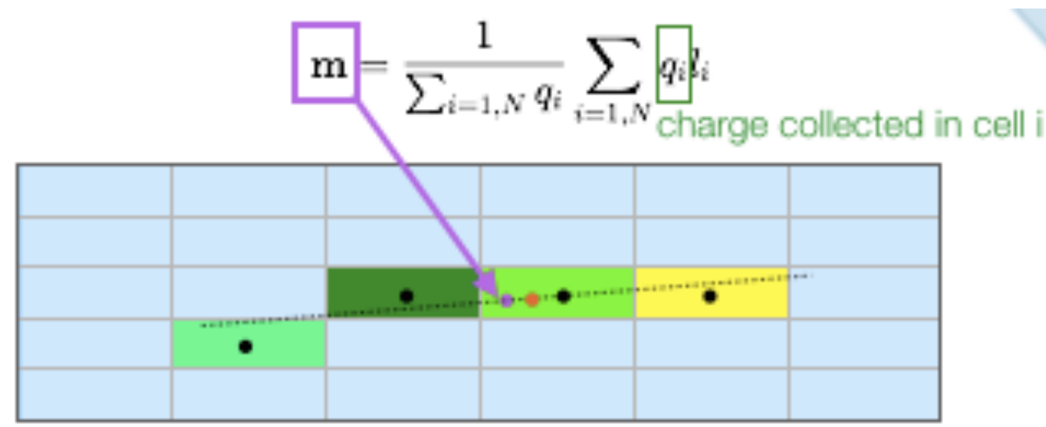
$$k_1 = hf(x_n, y_n)$$
$$k_2 = hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2})$$
$$k_3 = hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2})$$
$$k_4 = hf(x_n + h, y_n + k_3)$$
$$y_{n+1} = y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + O(h^5)$$

image from: Numerical Recipes in C++

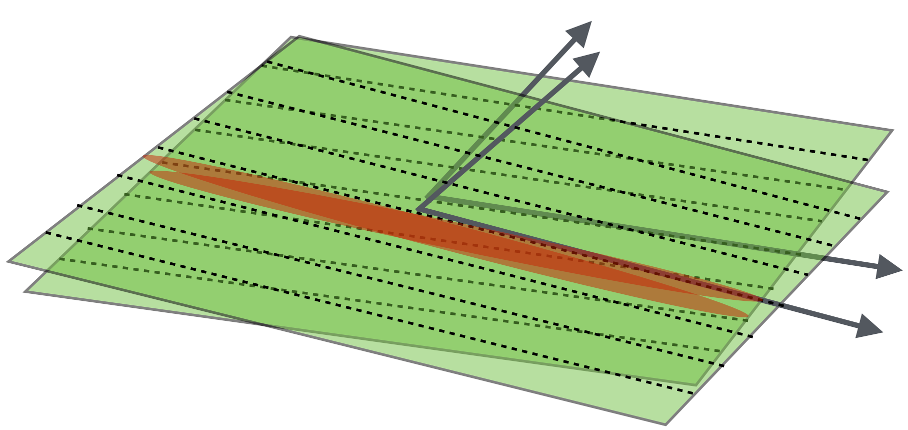
Some food for thoughts







- How can a non-binary readout work ?
- How would you “measure” the Lorentz angle ?
- Why were if off with our pull distribution ?



- Think of a great positive feature of such double sided modules

- Can we do PID with the silicon detector/TPC ?