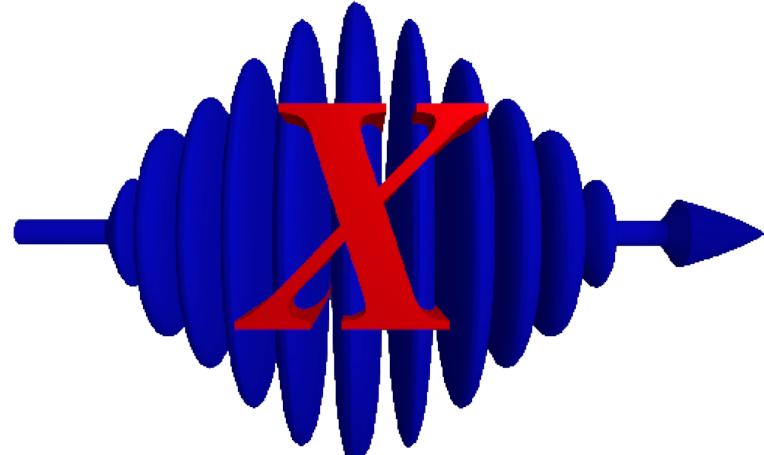
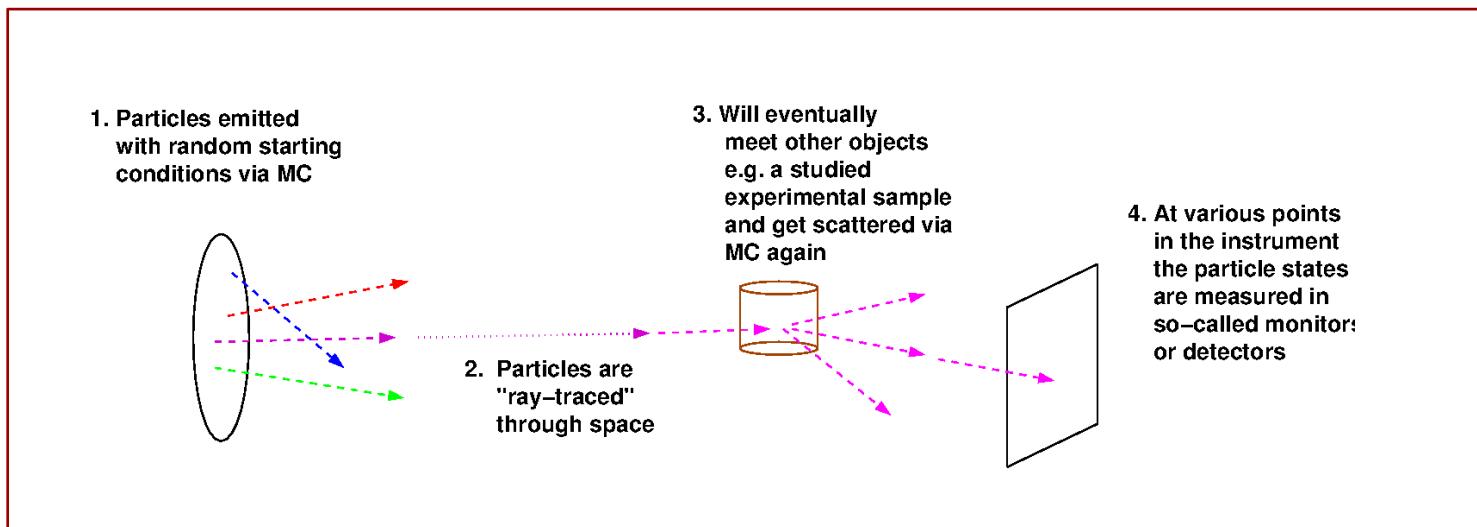


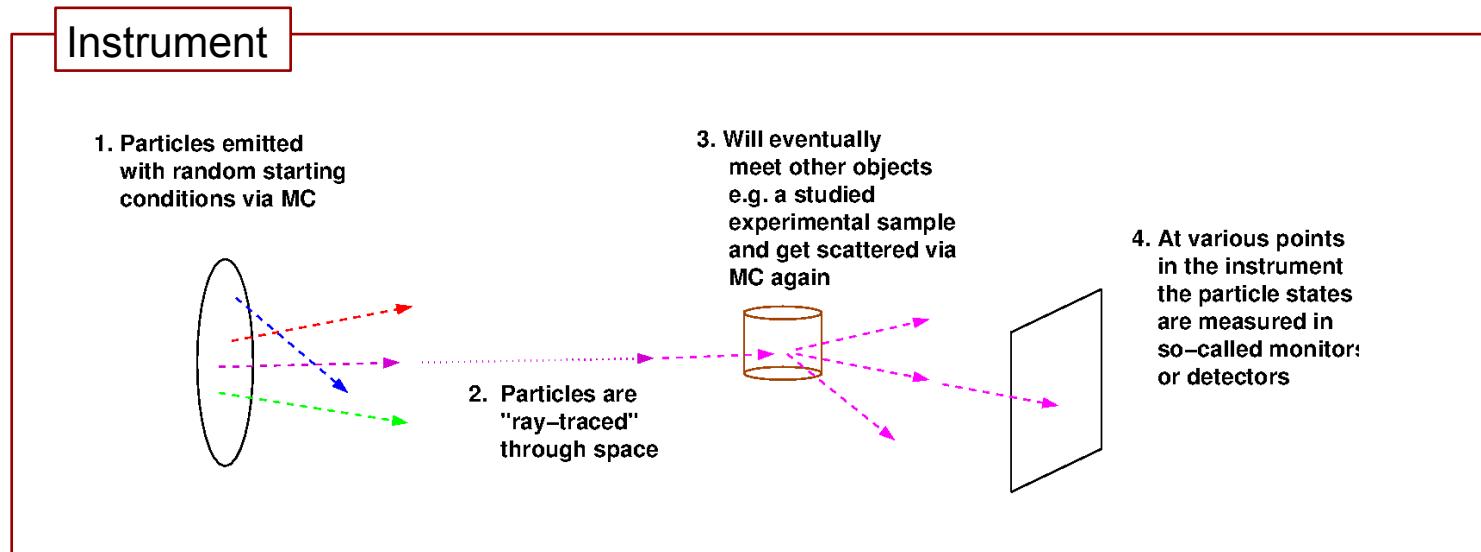
McXtrace



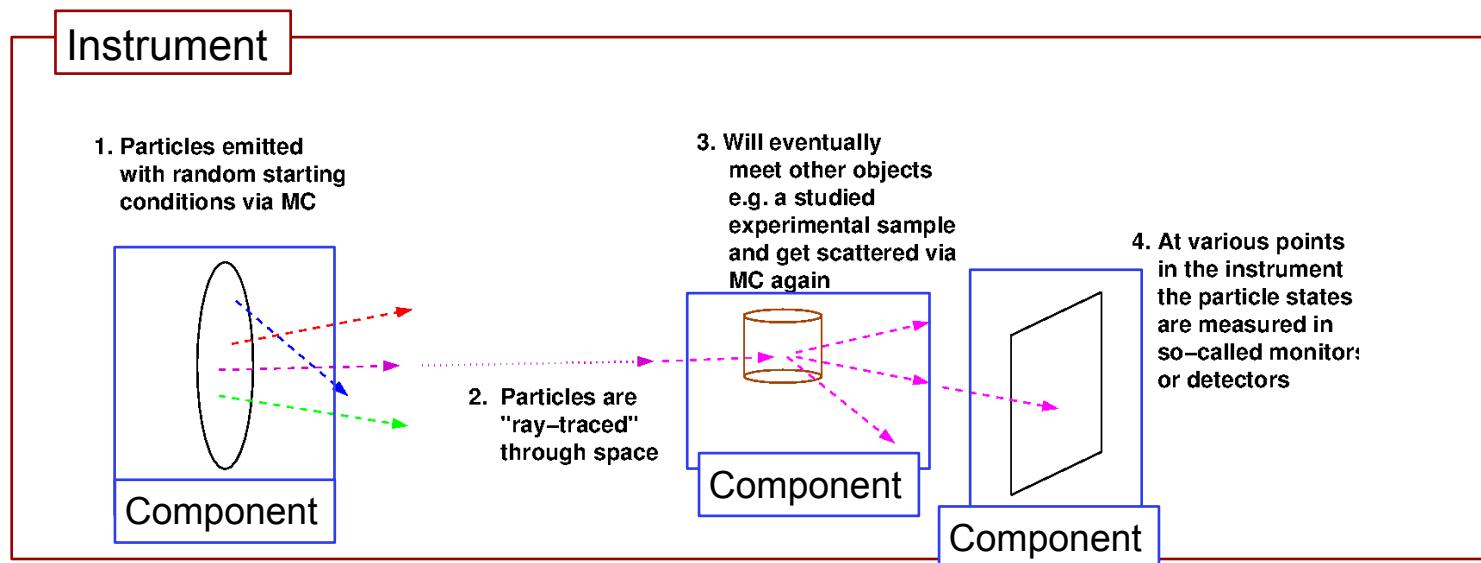
McXtrace components and instruments

Peter Willendrup (pkwi@fysik.dtu.dk)



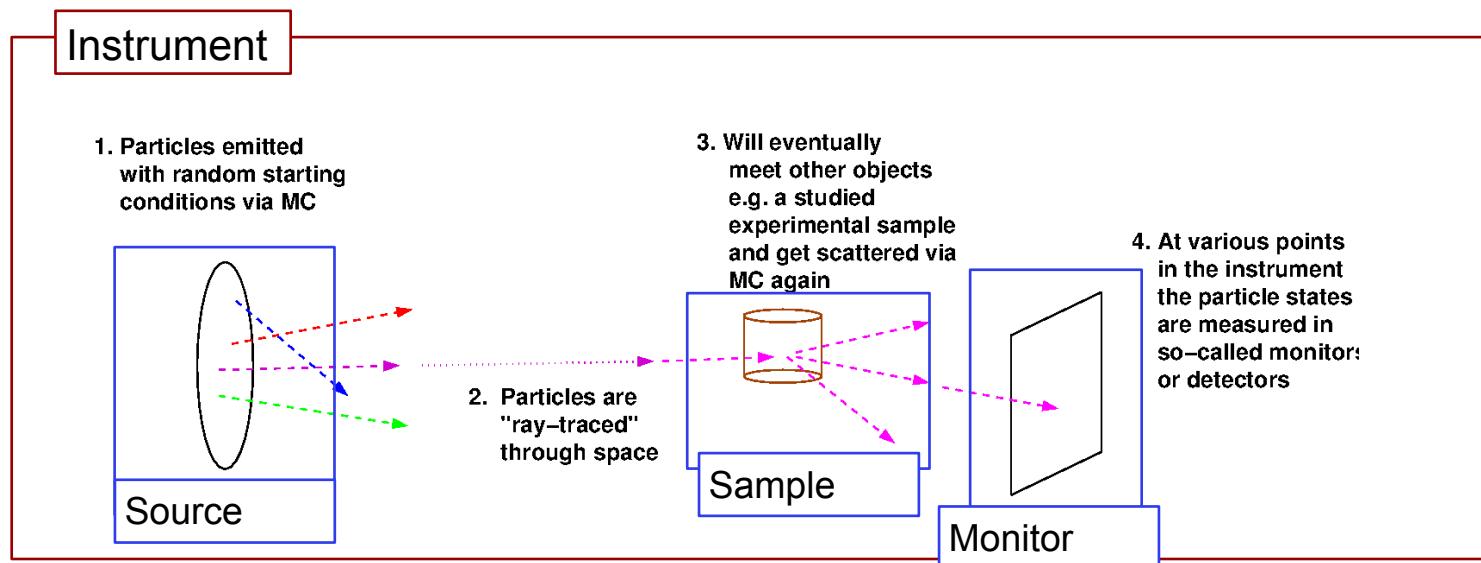


The instrument defines our “lab coordinate system”



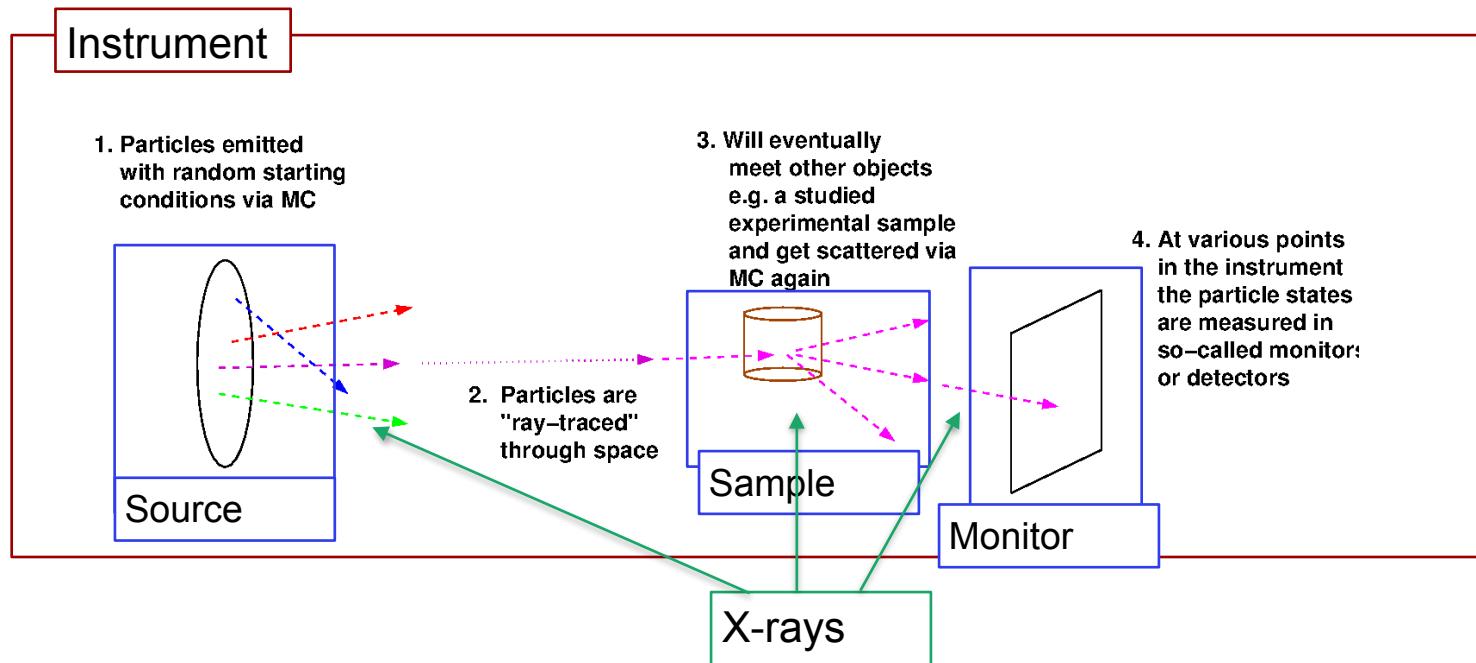
The instrument defines our “lab coordinate system”

The components define devices or features available in our instrument



The instrument defines our “lab coordinate system”

The components define devices or features available in our instrument
- they have different function

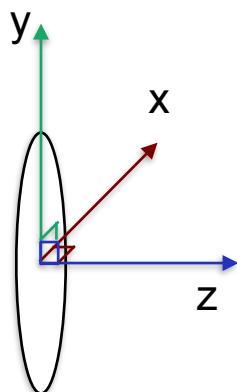


The instrument defines our “lab coordinate system”

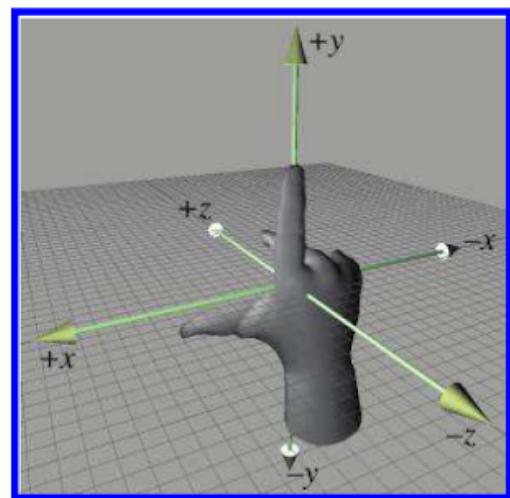
The components define devices or features available in our instrument
- they have different function

X-ray photons particles are passed on from one component to the next,
changing state under way

- One of the first components in your instrument is typically a source, which has a coordinate system like this....

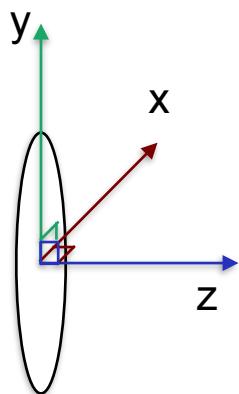


- z is along neutron beam direction
- y is vertical
- x at an angle of 90° wrt. z,y



Right-handed
coordinate system

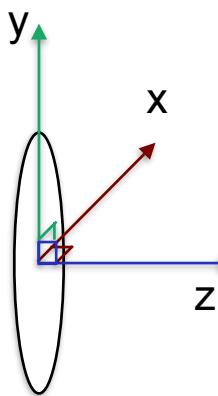
- Often the source coordinate system coincides with the “lab” coordinate system, denoted ABSOLUTE in McXtrace language, i.e.



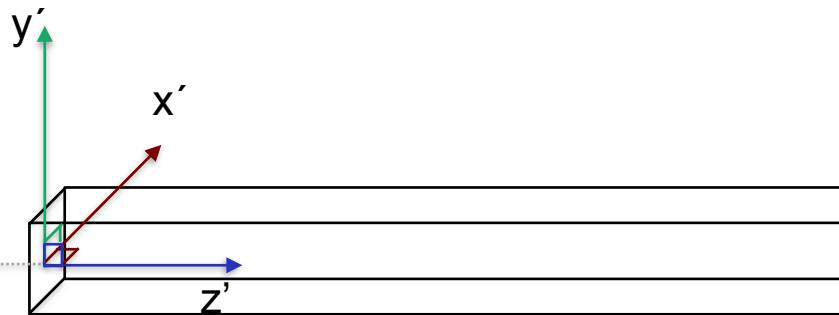
- COMPONENT Source = Source_flat(...)
AT (0,0,0) ABSOLUTE

Placing further components is done by order of

1. Location, i.e



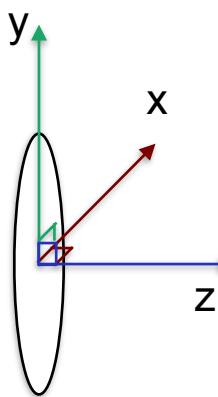
COMPONENT Source = Source_flat(...)
AT (0,0,0) ABSOLUTE



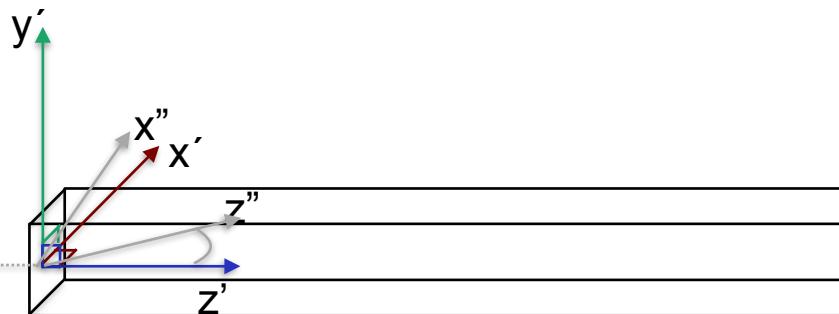
COMPONENT Capillary = Capillary(...)
AT (0,0,1) RELATIVE Source

Placing further components is done by order of

2. Rotation, i.e



COMPONENT Source = Source_flat(...)
AT (0,0,0) ABSOLUTE



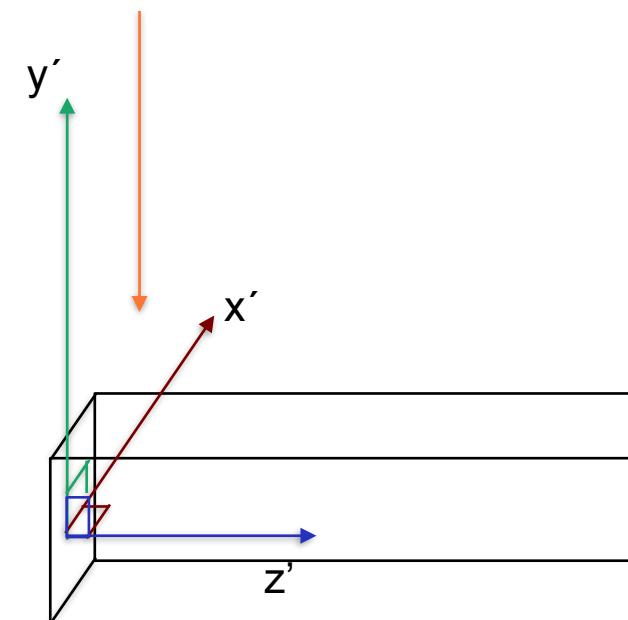
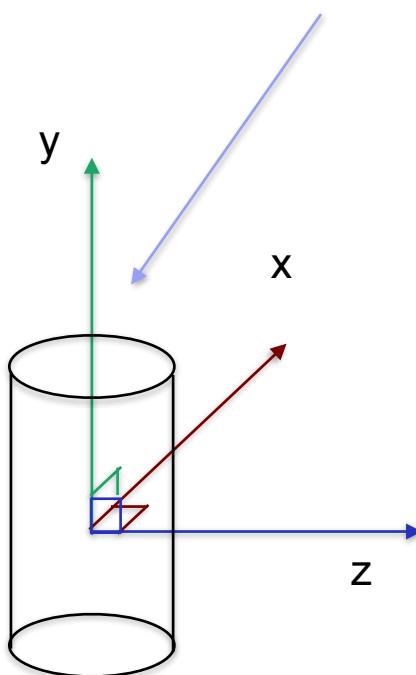
COMPONENT Capillary = Capillary(...)
AT (0,0,1) RELATIVE Source
ROTATED (0,0,1,0) RELATIVE Source

(Reference labels can also be PREVIOUS or PREVIOUS+1 etc.)

Components often have their origin at the centre of mass,
i.e. for samples ... but not for e.g. Capillary

Placing further components is done by order of

2. Rotation, i.e.

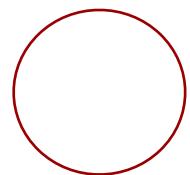


COMPONENT Sample = Some_sample(...)
AT (0,0,0) [RELATIVE] ABSOLUTE

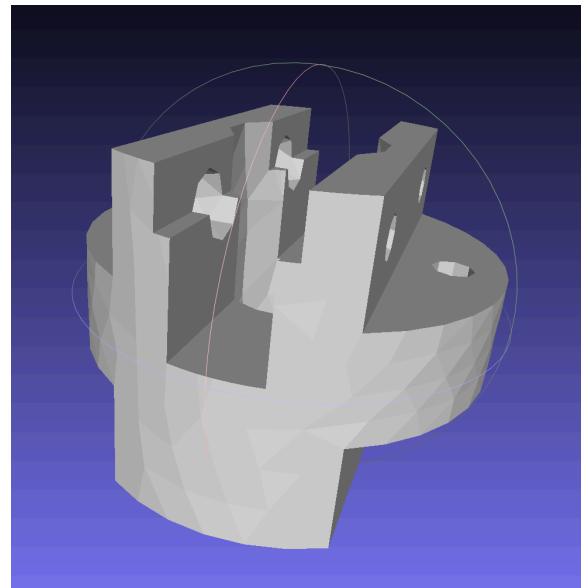
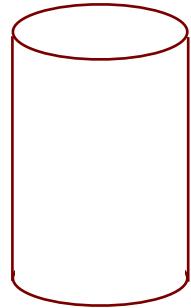
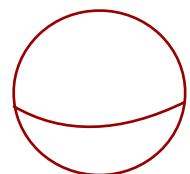
Generally speaking, the component author can choose
the meaningful coordinate system for the given problem!
- The McXtrace system takes care of the transformation between
them....

Component geometries are typically simple objects... But
some have polygon-description of the surface

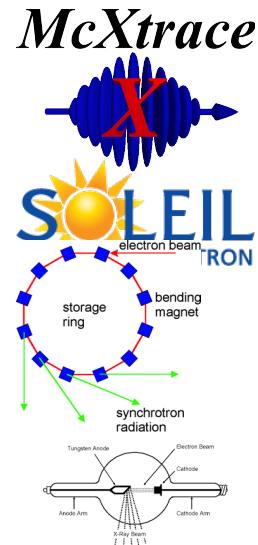
2D



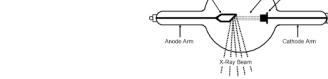
3D



Component classes

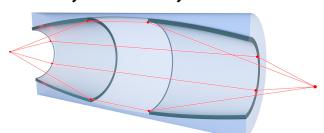


- Sources - these define MC starting conditions / “inject” rays to our simulation



- Optics - used to tailor properties of the X-ray beam

- Examples are mirrors, capillaries, lenses, zone-plates, choppers, collimators, slits, ...

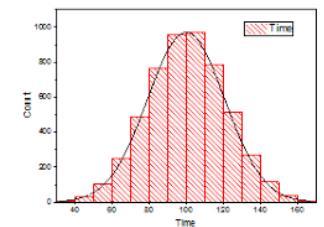


- Samples - “matter” of some form

- Powders, single crystals, liquids, micelles in solution, reflecting surfaces...



- Monitors - may probe the state of the X-ray beam and store histograms / event lists

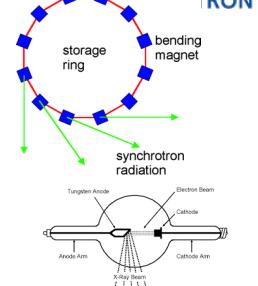


- Misc, obsolete

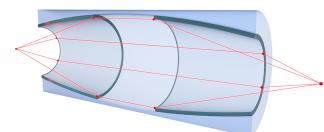
- “Other stuff” and “Old stuff”

*Other
Stuff*





- Sources - these define MC starting conditions / “inject” neutrons to our simulation



- Optics - used to tailor properties of the X-ray beam

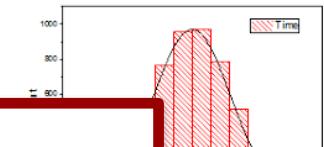
- Examples are mirrors, capillaries, lenses, zone-plates, choppers, collimators, slits, ...



- Samples - “matter” of some form

- Powders, single crystals, liquids, micelles in solution, reflecting surfaces...

- Monitors - may probe the state of the X-ray beam and store histograms / event lists



- Misc, obsolete

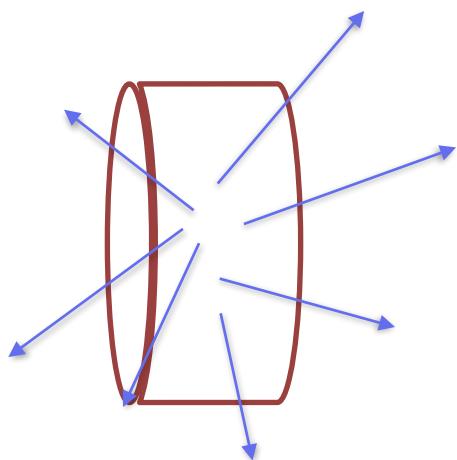
- “Other”

Common to all components:

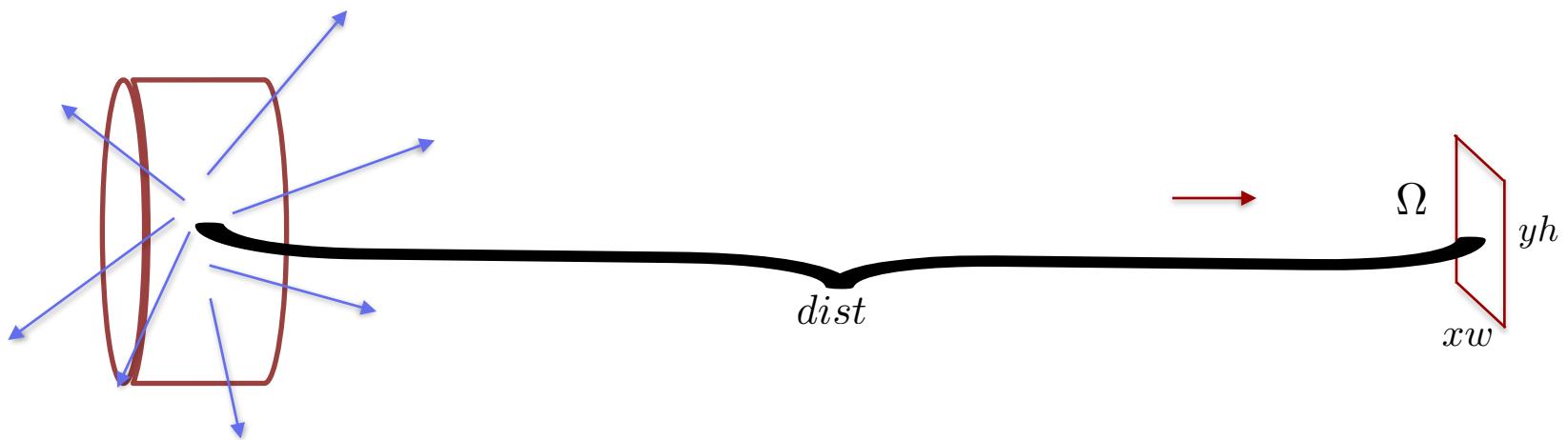
**They set, manipulate/interact with
or measure the **state of the X-ray (ray)****



- To first order emit uniformly into 4π steradian

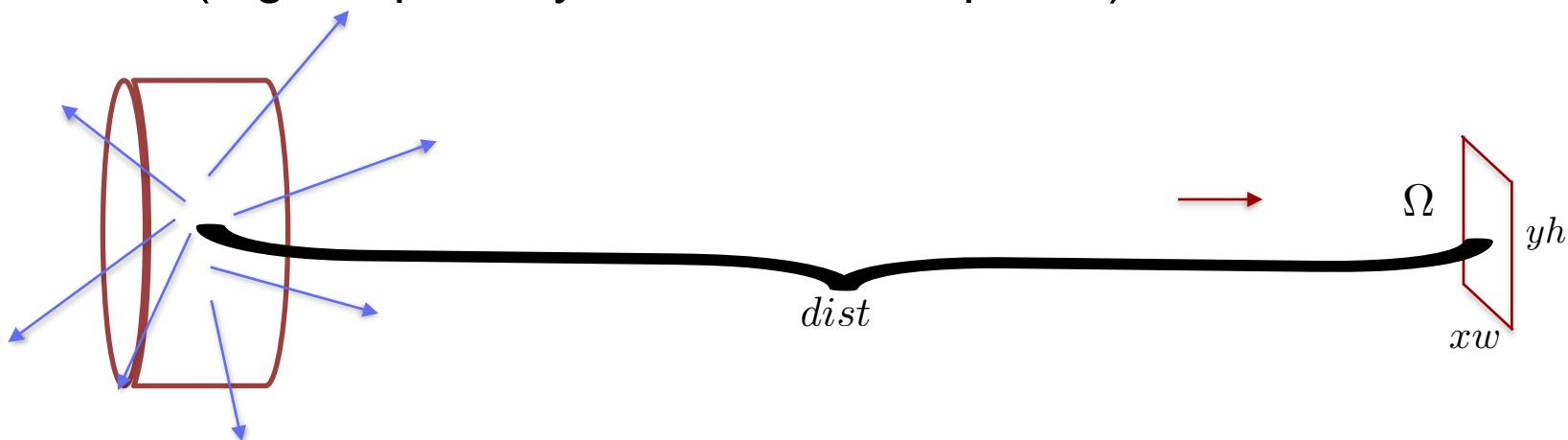


- To first order emit uniformly into 4π steradian



- Generally we are interested in sending the input to an aperture, characterised by a certain solid angle Ω , often corresponding to a rectangle $xw \times yh$ at a distance $dist$ from the source

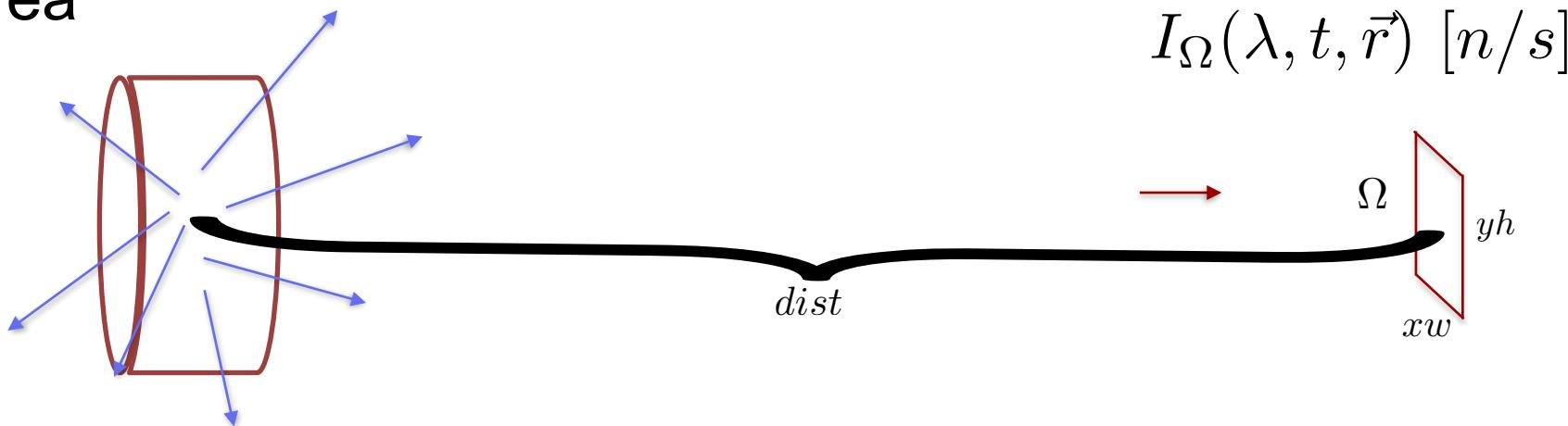
- The emission intensity into our chosen solid angle Ω can be a function of wavelength, time (pulsed sources) and possibly point of origin on the source surface (e.g. for partially coherent descriptions).



- The emission of particles into the solid angle Ω is in fact an integration and leads to a simulated “intensity” of I_Ω [n/s].
- In McXtrace, that integrated intensity is partitioned over a given set of particle rays referred to as **ncount**, **-n** or **--ncount**
- The default **ncount** is 1e6 rays

$$\begin{aligned} I(\lambda) & [n/s/str] \\ I(\lambda, t) & [n/s/str] \\ I(\lambda, t, \vec{r}) & [n/s/str] \end{aligned}$$

- Our rays are emitted randomly, sampling Ω and all variables of the source “spectrum”, i.e. wavelength, time and area



- assigning ray weights p such that

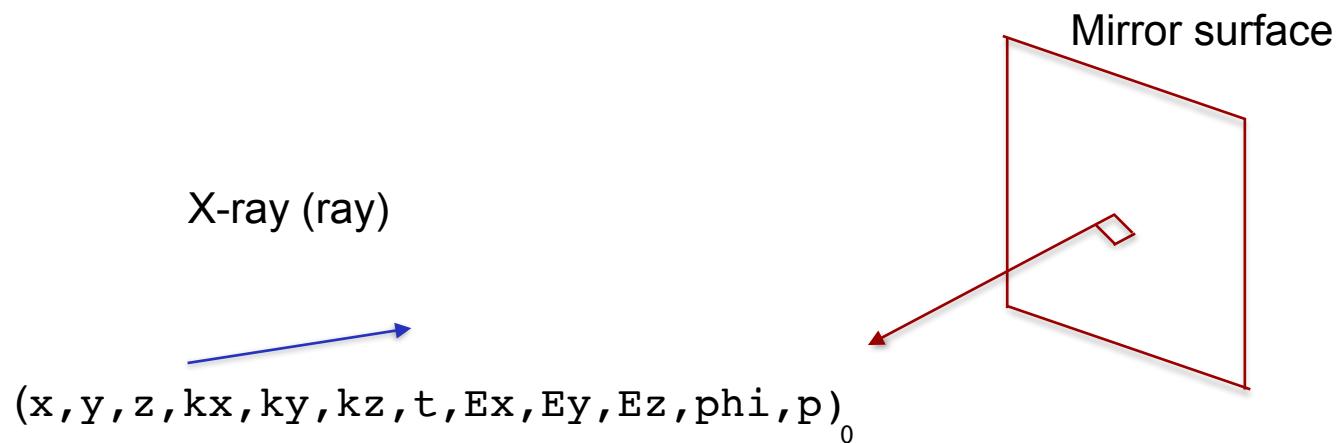
$$\sum_{j=1}^{\text{ncount}} p_j = \int_{d\lambda, dt, d\vec{r}} I_{\Omega}(\lambda, t, \vec{r})$$

- Defining the ray starting conditions imply setting:
 - The **starting point** on the surface, i.e. \vec{r} (in the code variables x, y, z)
 - The **direction** into Ω and our λ / k (in the code variables k_x, k_y, k_z)
 - The **starting time** (in the code the variable t)
 - The initial **intensity** / weight of the neutron ray (in the code the variable p)
 - If needed the initial
 - **Electric field polarisation** (in the code the variables E_x, E_y, E_z)
 - X-ray phase (in the code the variable ϕ)

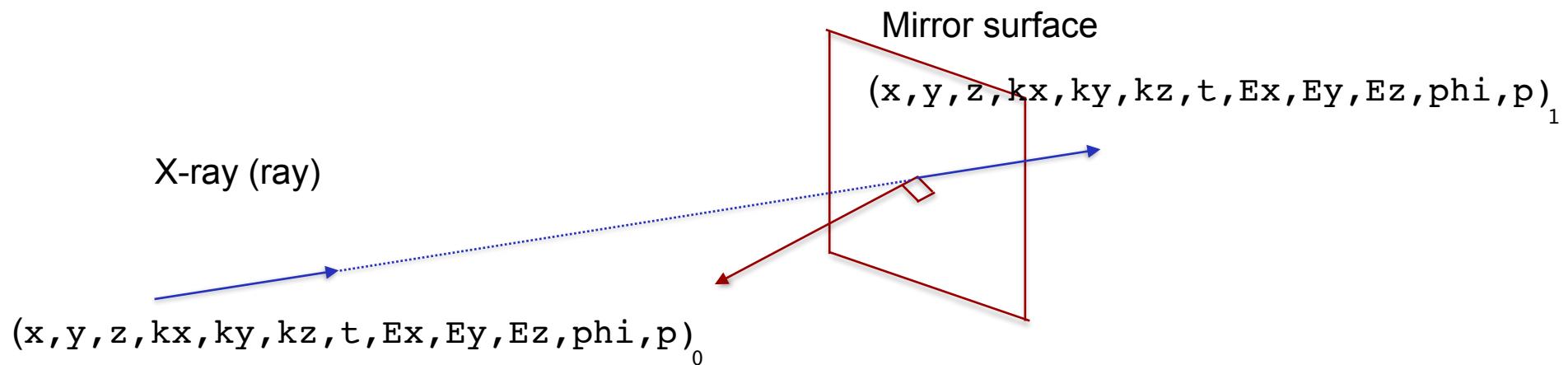
X-ray package:

Weight (p), # photons (left) in the package
Coordinates (x, y, z)
Wavevector (k_x, k_y, k_z)
Polarization (E_x, E_y, E_z)
Phase (ϕ)
Time(t)

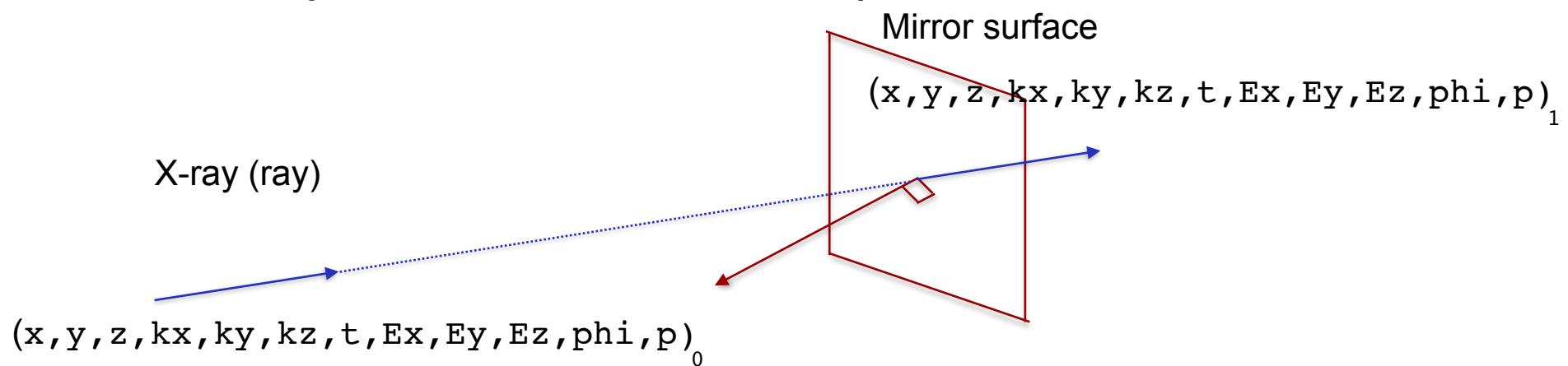
- 1 starting situation



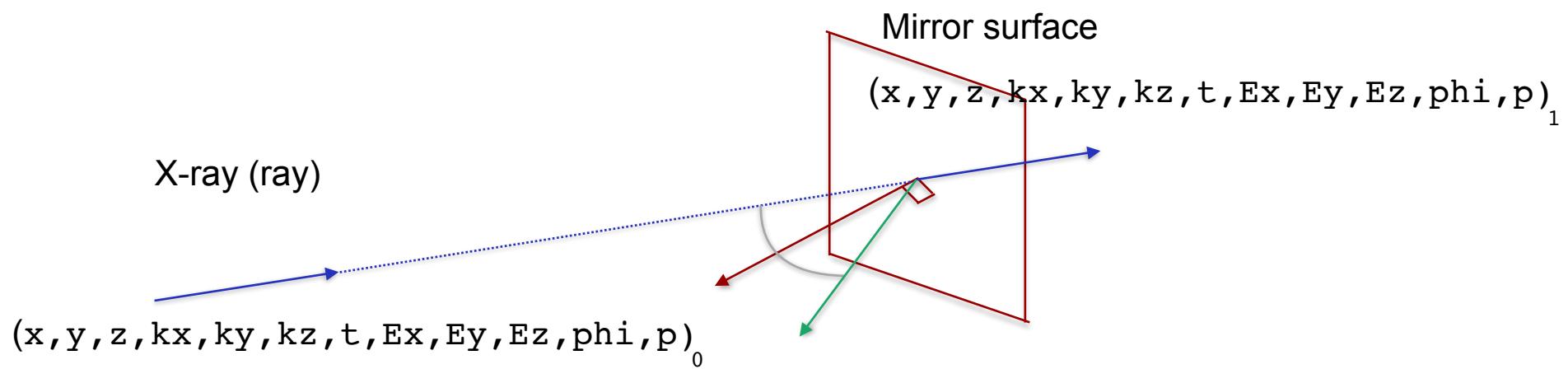
- 2. Propagate to the mirror surface



- 3. Checks (are we on surface, what is probability of reflection etc.)

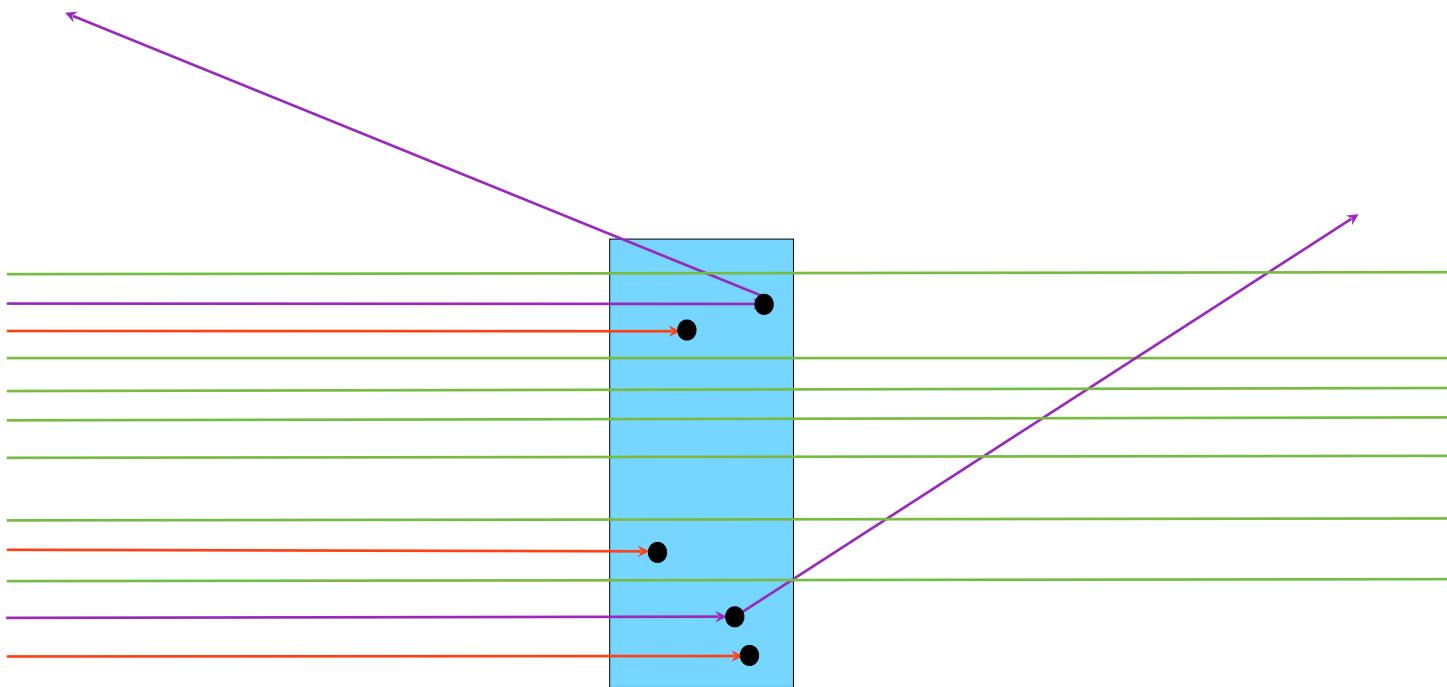


- 4. Reflect



Weight of final ray is adjusted according to reflectivity, see next slide

$(x, y, z, kx, ky, kz, t, Ex, Ey, Ez, \phi, p)_2$



A photon hitting a sample can be:
absorbed, **transmitted**, or **scattered**

Samples

For a **non-thin** sample the probabilities for **absorption**, **transmission** or **scattering** are given by

$$p_A = (1 - e^{-\Sigma_T t})(\Sigma_A/\Sigma_T)$$

$$p_S = (1 - e^{-\Sigma_T t})(\Sigma_S/\Sigma_T)$$

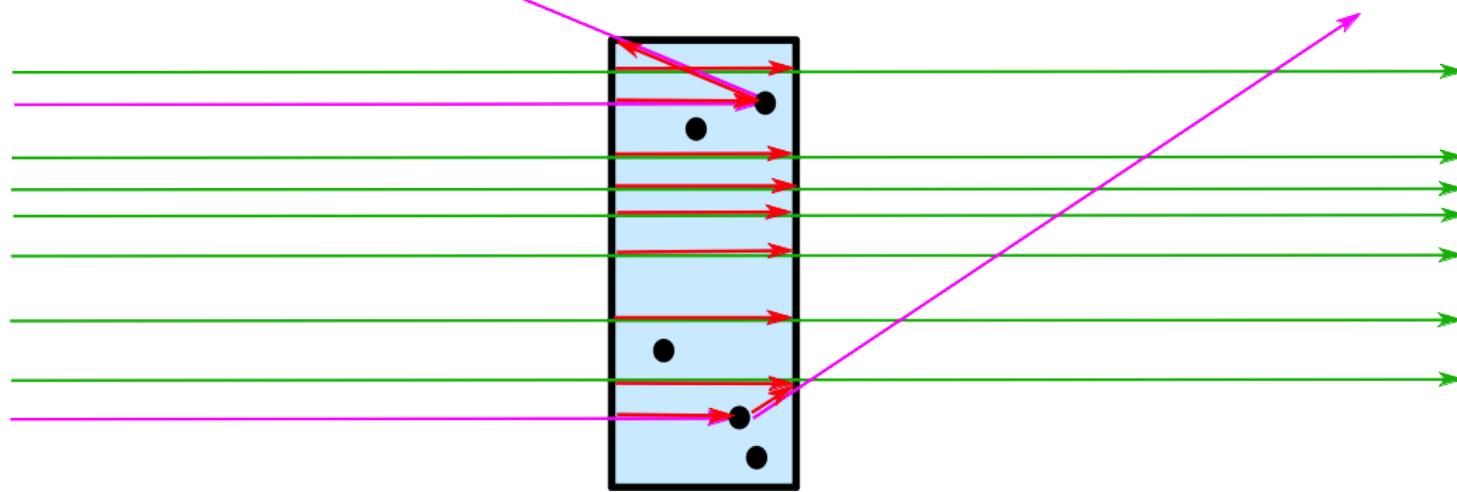
$$p_T = 1 - p_S - p_A = e^{-\Sigma_T t}$$

t = sample thickness

$$\Sigma_* = \rho \sigma_*$$

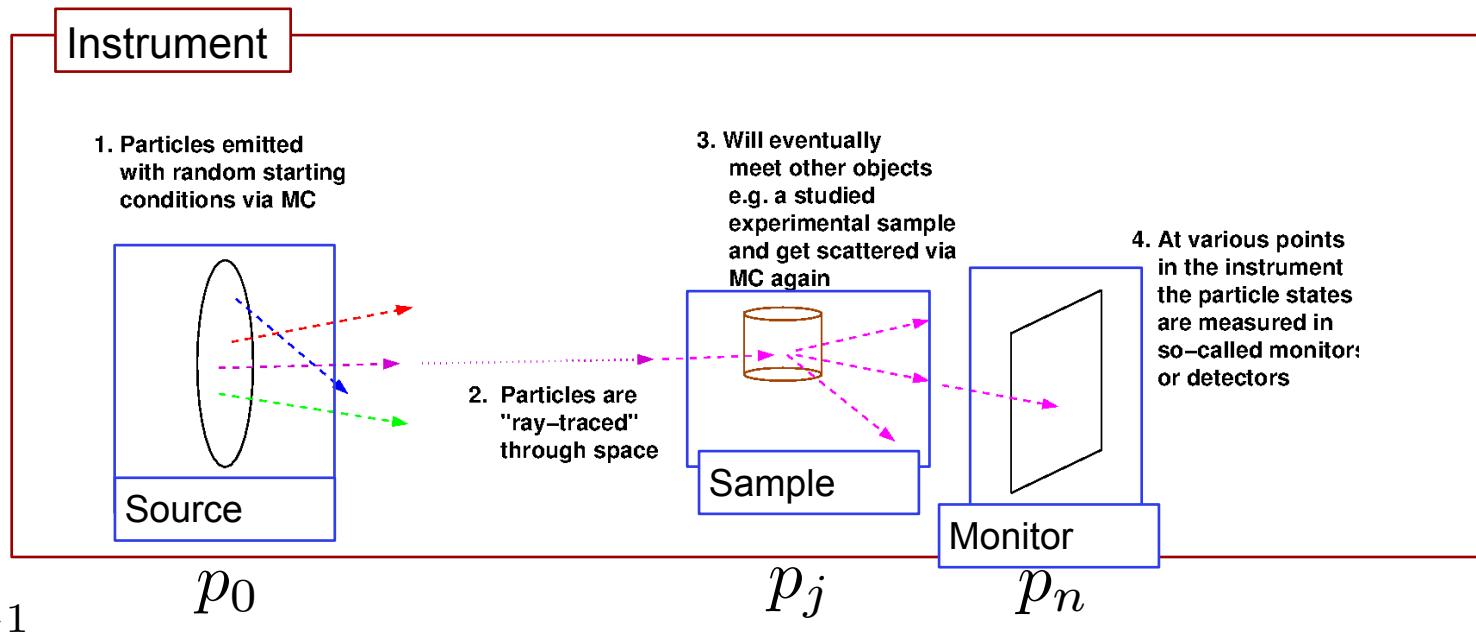
macroscopic cross section [cm⁻¹] microscopic cross section [barn/fm²]
number density [atoms/cm³]

Samples/Matter interaction in General in McXtrace



A photon ray hitting a sample can be:
transmitted+absorption, or scattered+absorption

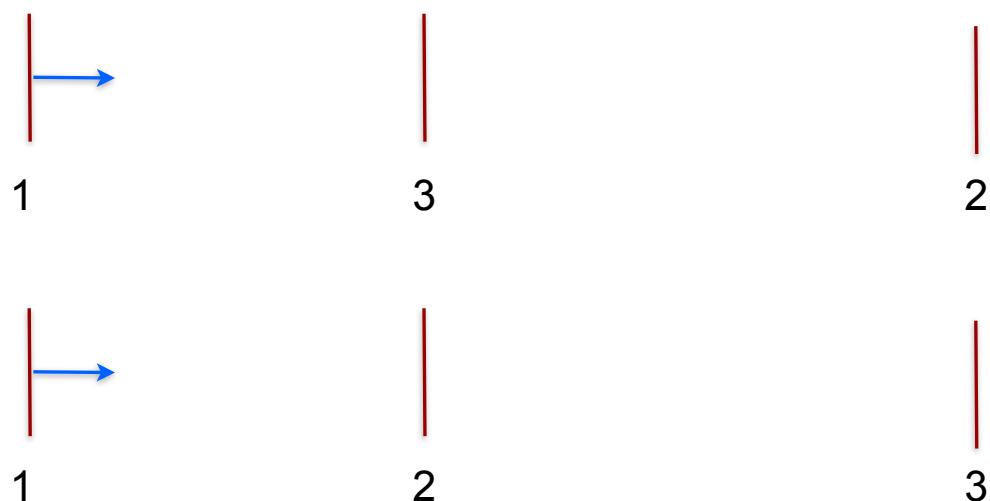
In a given component, the neutron intensity is adjusted by a multiplicative factor (probability)



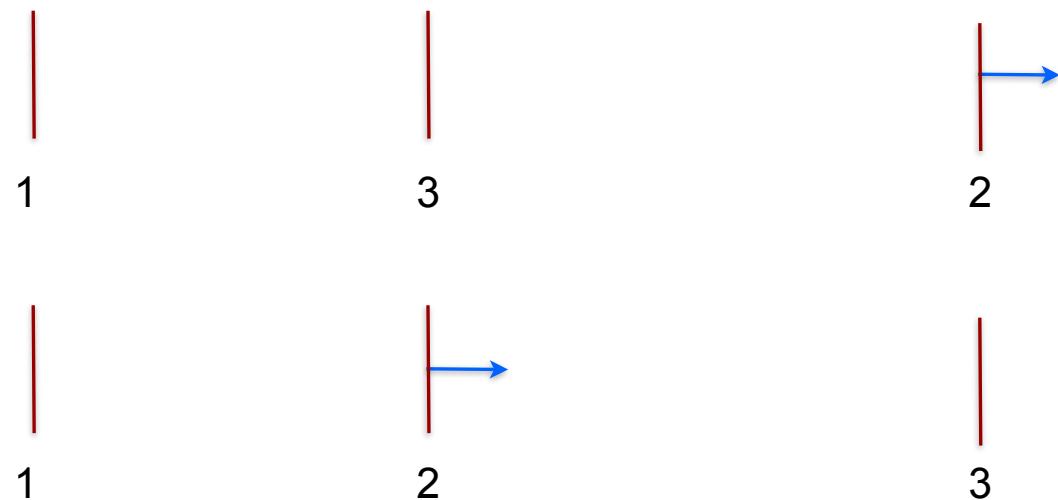
The weight multiplier of the j 'th component, w_j , is calculated by the probability rule $f_{MC,b}w_j = P_b$ where P_b is the physical probability for the event "b", and $f_{MC,b}$ is the probability that the Monte Carlo simulation selects this event.

In case of "branching", i.e. multiple outcomes, it is clear that $\sum_b f_{MC,b} = 1$

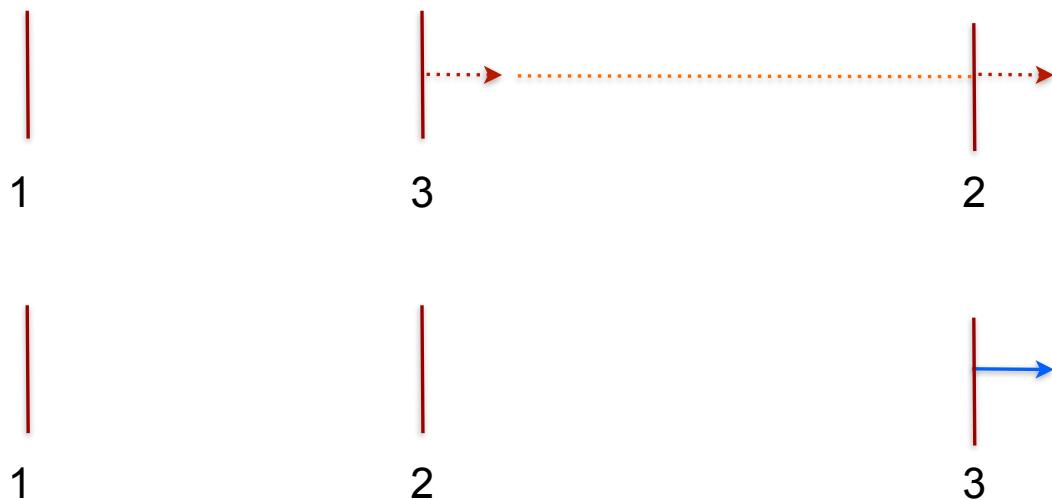
$$p_j = p_0 \prod_{k=1}^j w_k$$



Starting at the source



Moving to first comp in the list

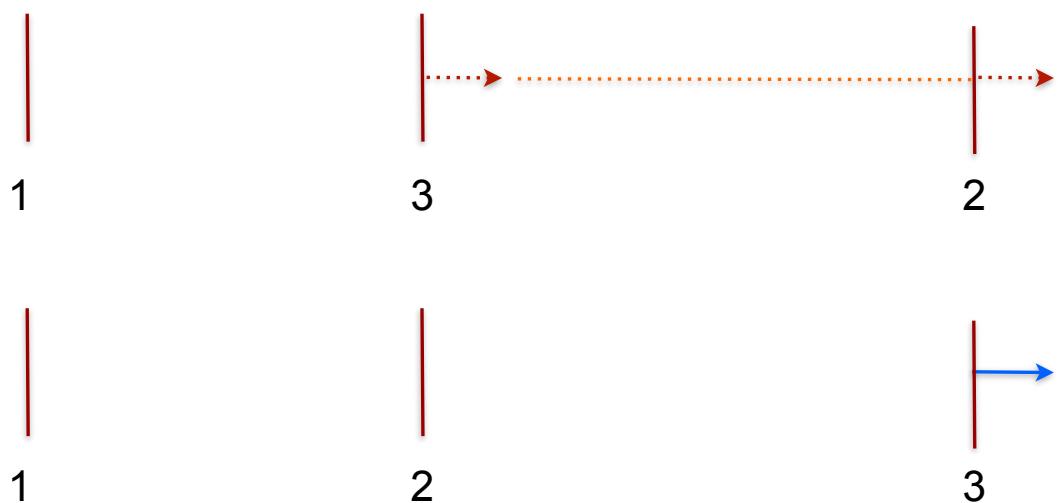


Moving to 3rd comp in list requires “moving back in time”.

Default behavior is to ABSORB this type of neutron.

For monitors use `restore_xray=1` in this case.

For homegrown comps use `ALLOW_BACKPROP` macro.



Moving to 3rd comp in list requires “moving back in time”.

Default behavior is to ABSORB this type of neutron.

For monitors use `restore_xray=1` in this case.

For homegrown comps use `ALLOW_BACKPROP` macro.

The order of components is important, and in general overlaps should be avoided!