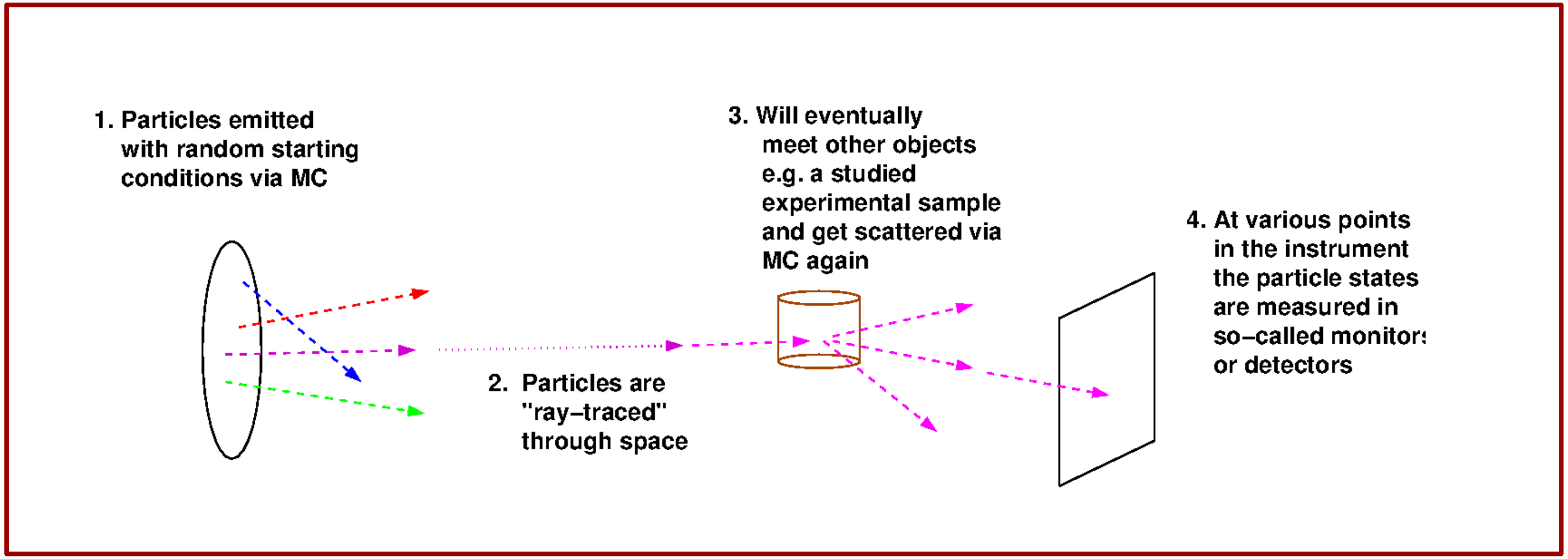
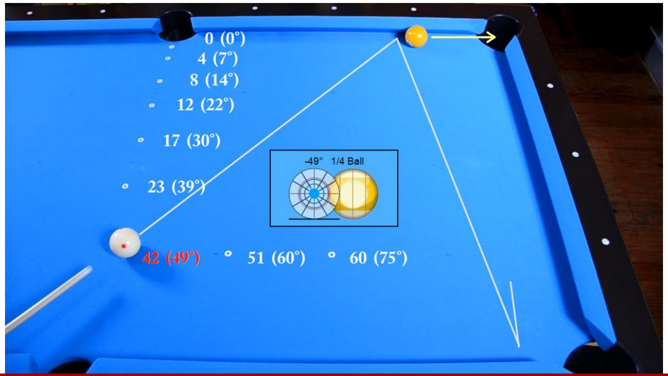
Peter Willendrup

Introduction to basic concepts of McStas

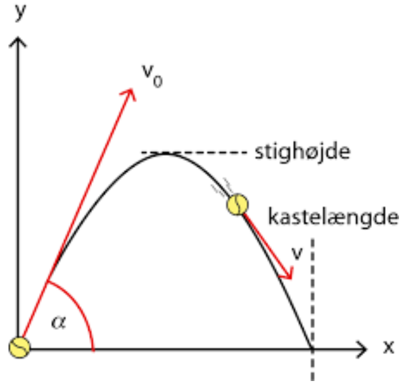
In the big picture, McStas is this…



- Classical Newtonian mechanics, i.e.

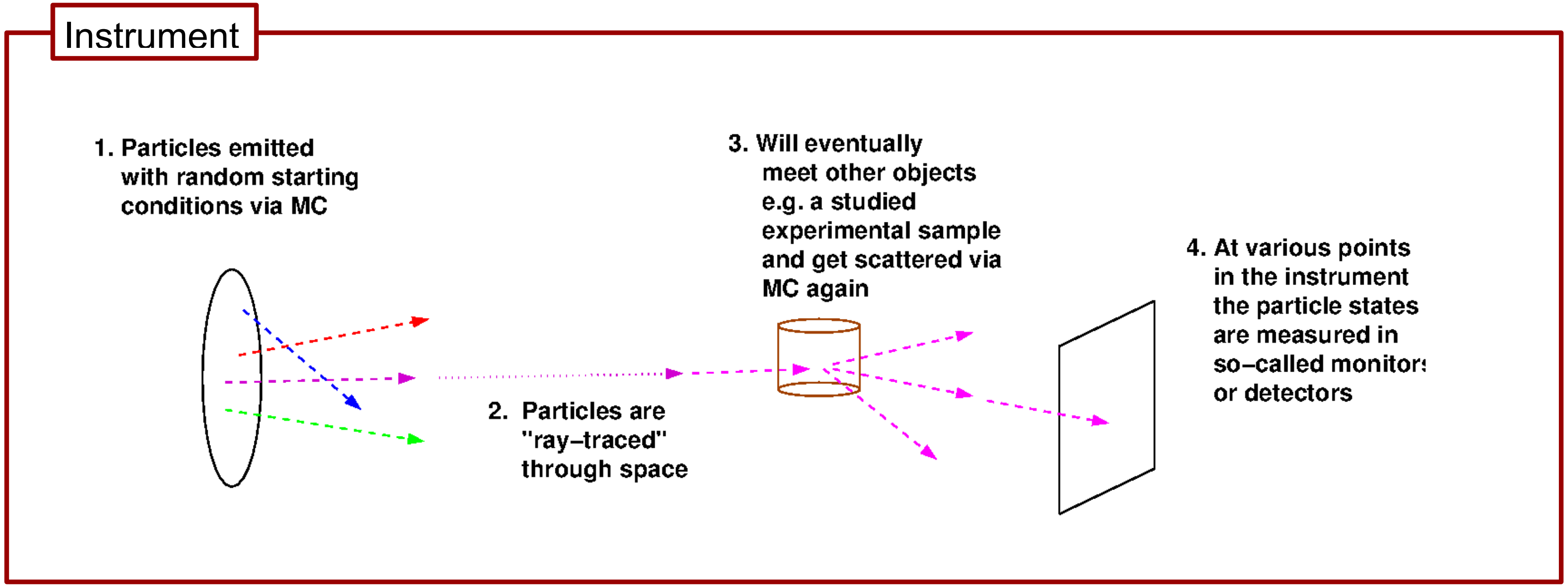


and



- (independent, particles though…)

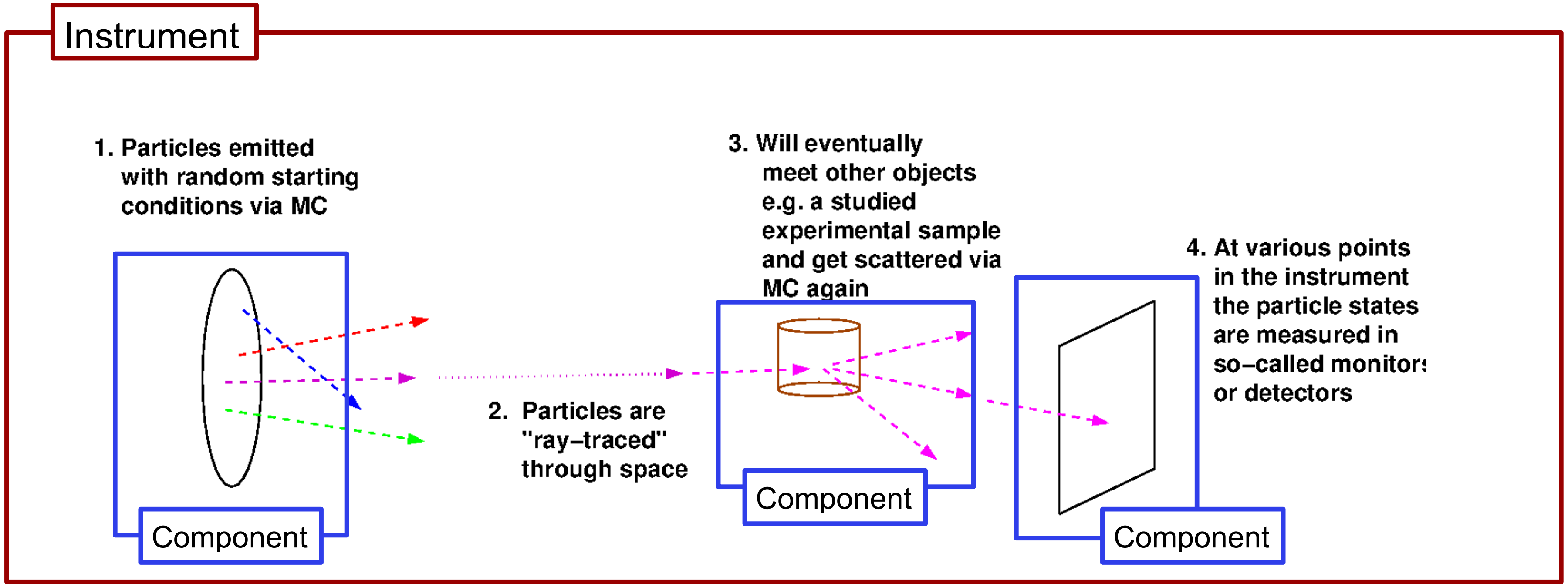
In the big picture, McStas is this…



The instrument defines our “lab coordinate system”

2021 Virtual ISIS McStas School

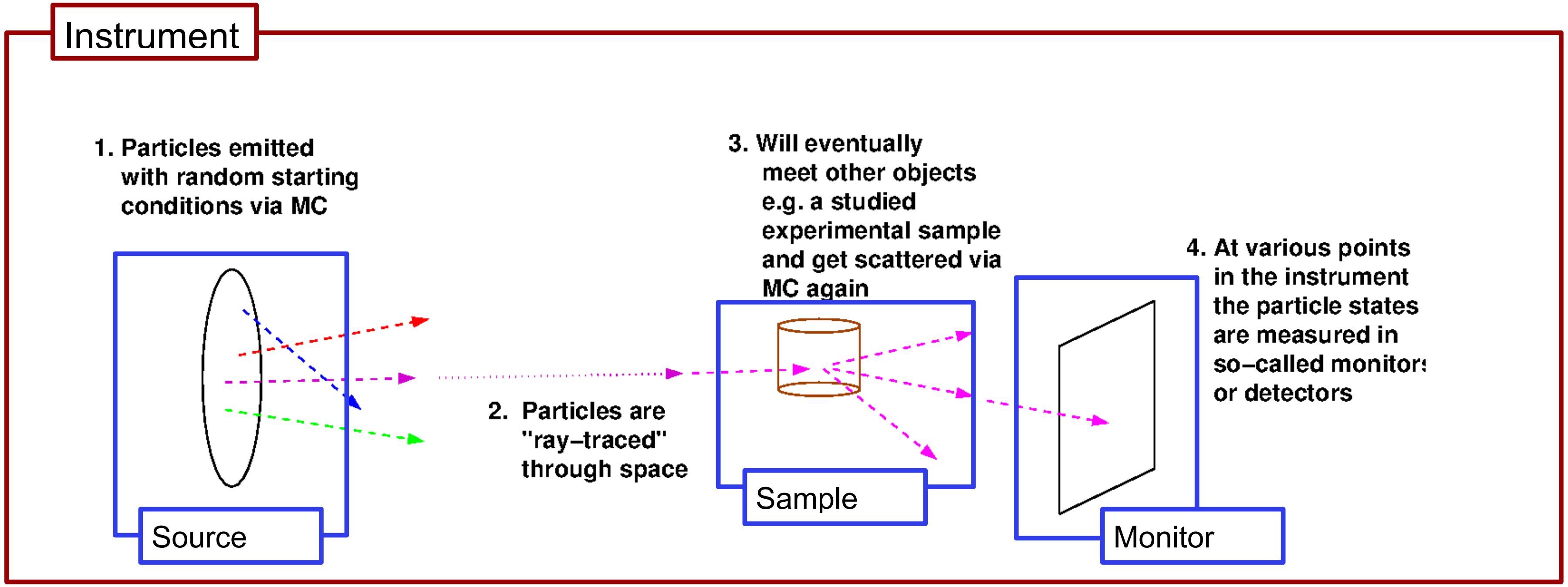
In the big picture, McStas is this…



The instrument defines our “lab coordinate system”

The components define devices or features available in our instrument

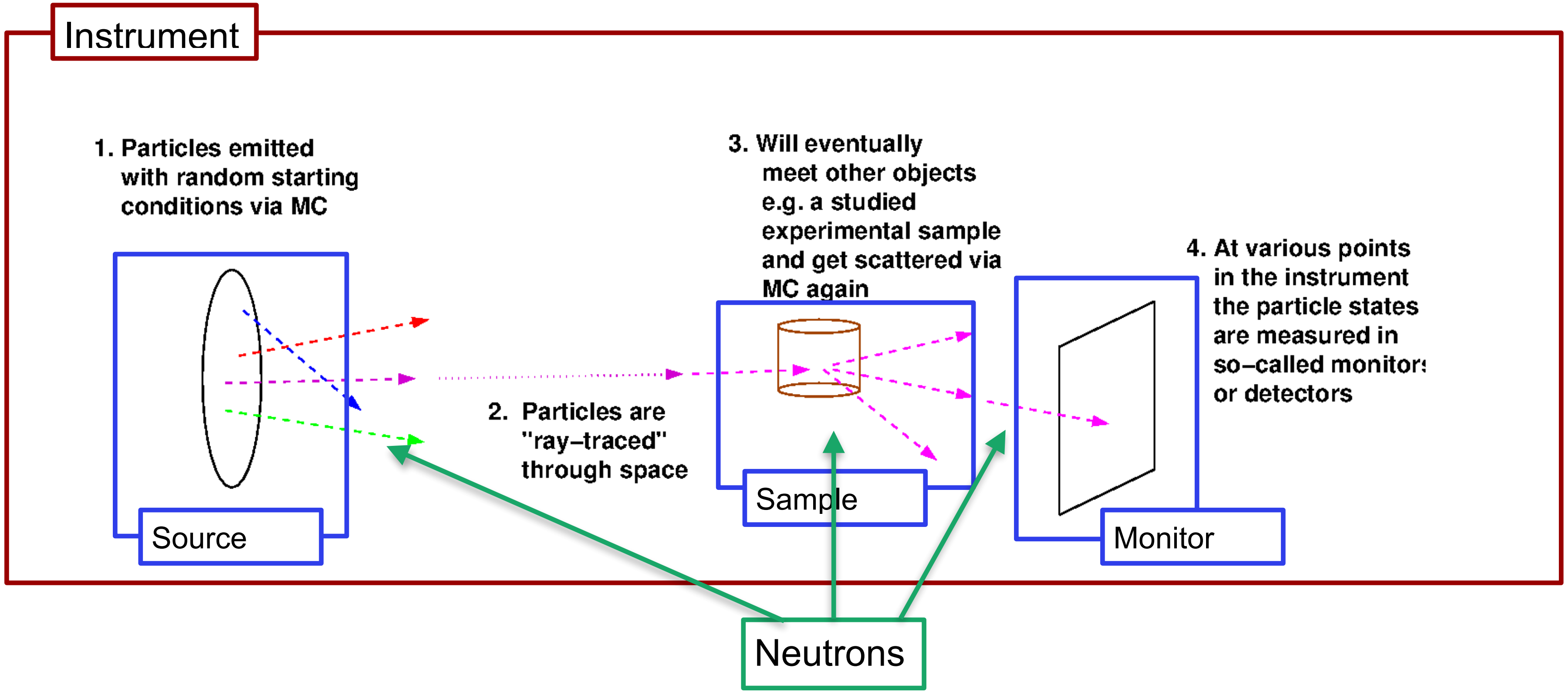
In the big picture, McStas is this…



The instrument defines our "lab coordinate system"

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

In the big picture, McStas is this…



The instrument defines our “lab coordinate system”

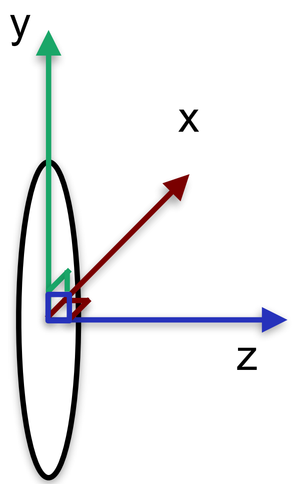
The components define devices or features available in our instrument - they have different function

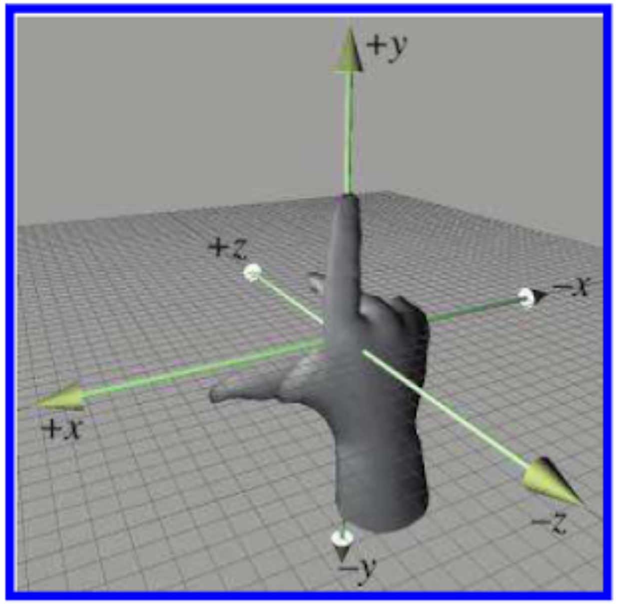
Neutron particles are passed on from one component to the next, changing state under way

2021 Virtual ISIS McStas School

Placing components - source

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



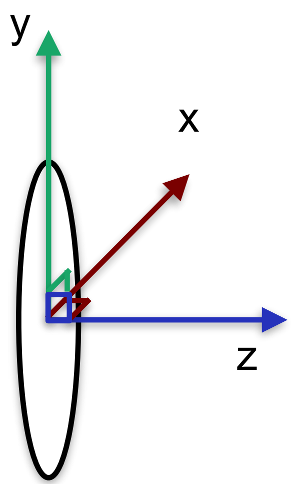


Right-handed coordinate system

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

Placing components - source

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

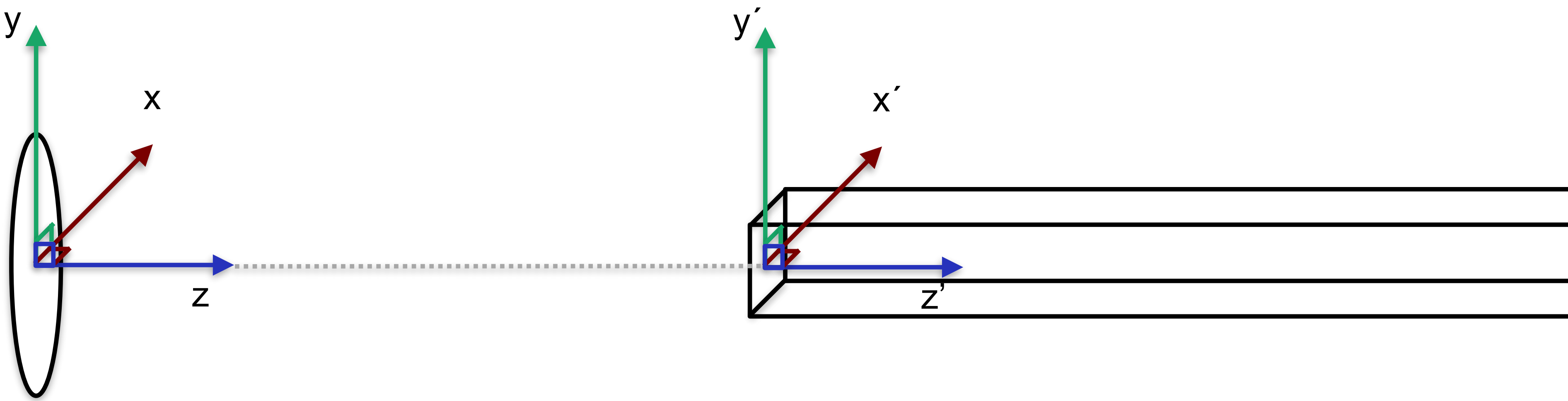


- COMPONENT Source = Source_simple(…)
  AT (0,0,0) ABSOLUTE

2021 Virtual ISIS McStas School

Placing further components - RELATIVE

Placing further components is done by order of
1. **Location, i.e**

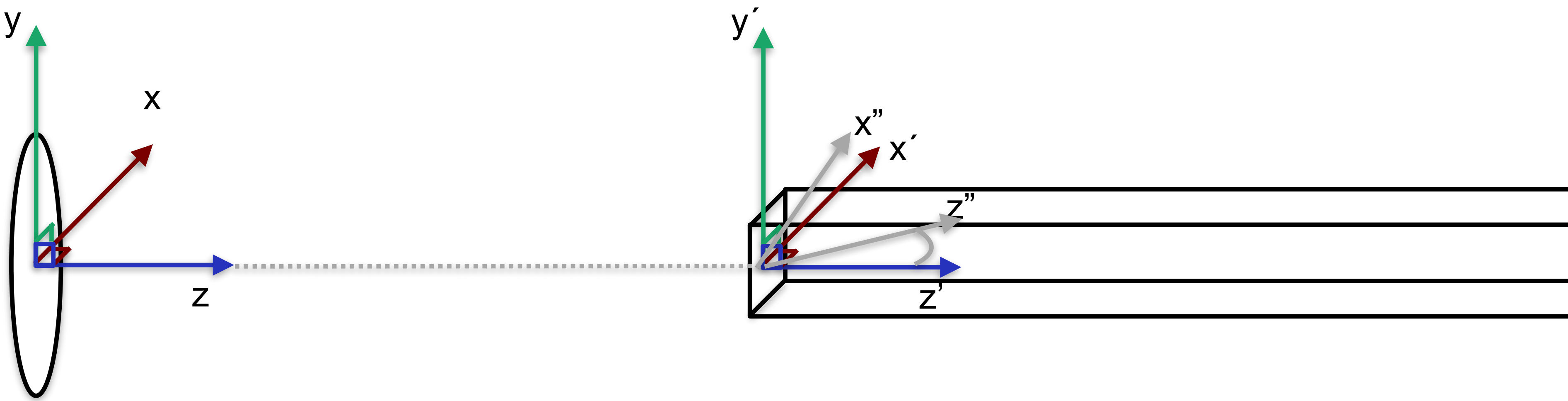


COMPONENT Source = Source_simple(…)
AT (0,0,0) ABSOLUTE

COMPONENT Guide = Guide(…)
AT (0,0,1) RELATIVE Source

Placing further components - RELATIVE

Placing further components is done by order of
2. **Rotation, i.e**

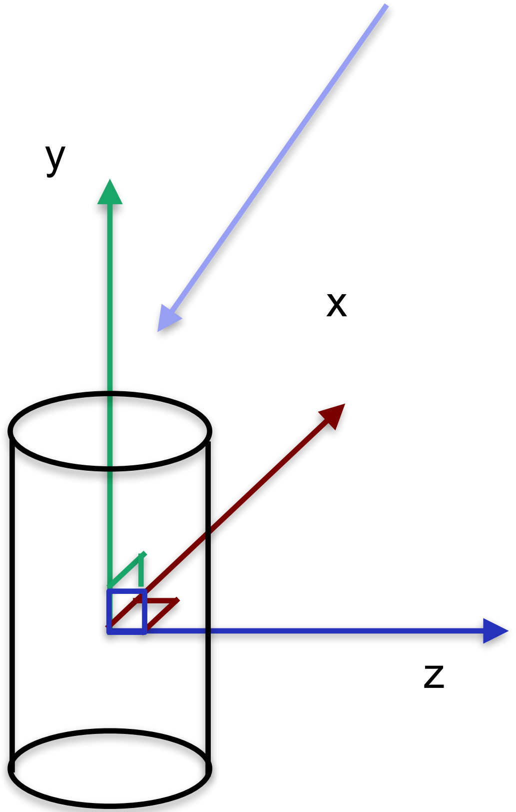


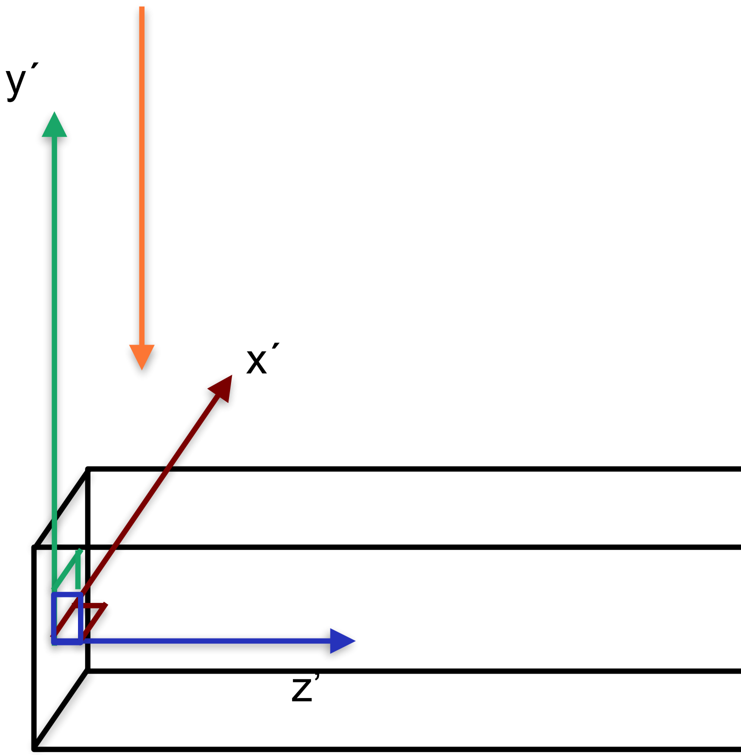
COMPONENT Source = Source_simple(…)
AT (0,0,0) ABSOLUTE

COMPONENT Guide = Guide(…)
AT (0,0,1) RELATIVE Source
ROTATED (0,30,0) 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 neutron guides

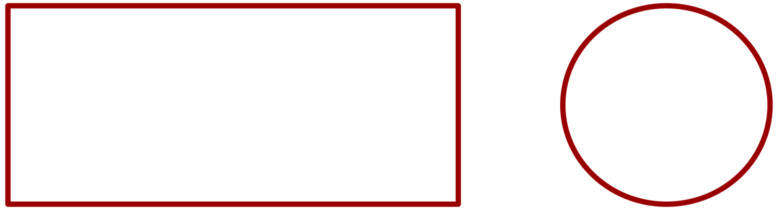




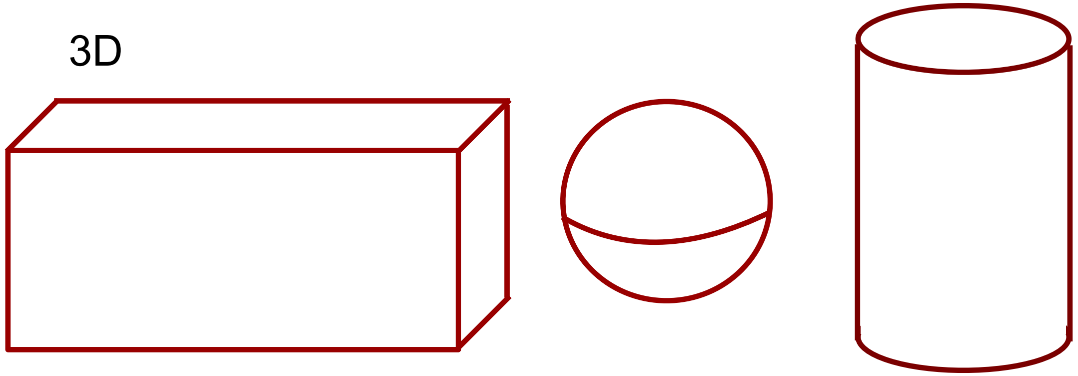
Generally speaking, the component author can choose
the meaningful coordinate system for the given problem!
- The McStas system takes care of the transformation between them….

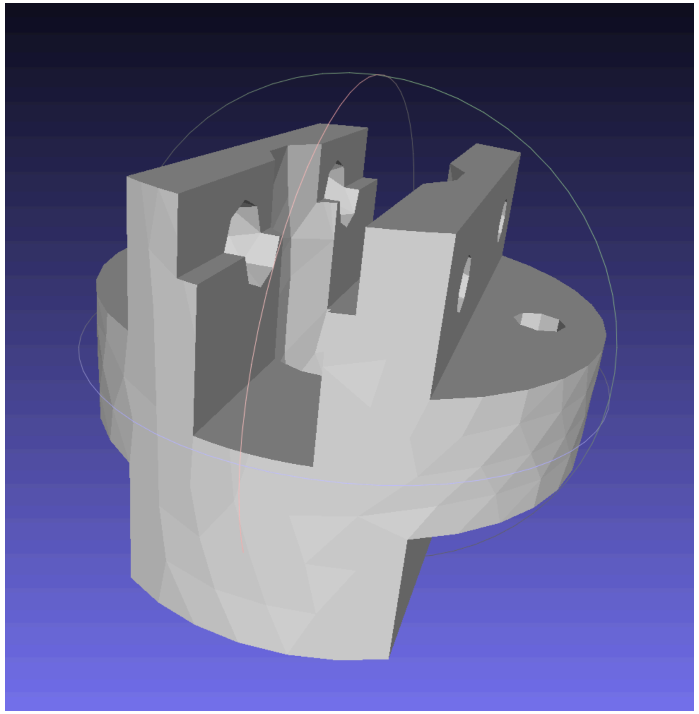
Component geometries are typically simple objects… But some have polygon-description of the surface

2D

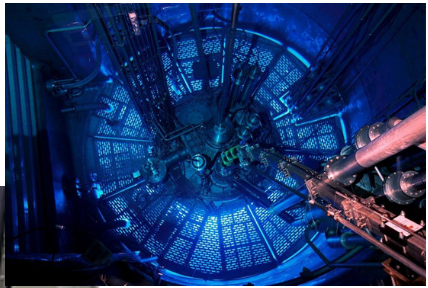


3D





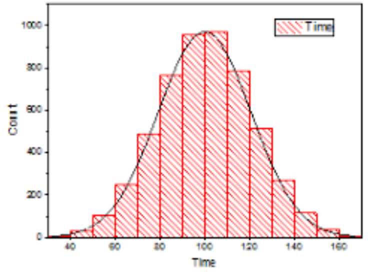
Component classes



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

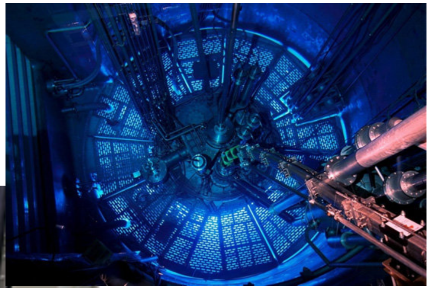
- Optics - used to tailor properties of the neutron beam
  - Examples are mirrors, guides, choppers, collimators, slits, …

- Samples - “matter” of some form
  - Powders, single crystals, liquids, micelles in solution, reflecting surfaces…
- Monitors - may probe the state of the neutron beam and store histograms / event lists



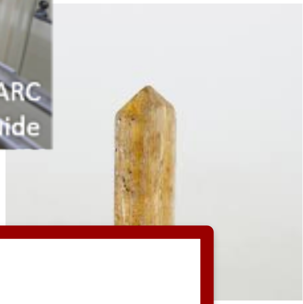
- Misc, obsolete
  - “Other stuff” and “Old stuff”

Component classes



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

- Optics - used to tailor properties of the neutron beam
  - Examples are mirrors, guides, choppers, collimators, slits, …



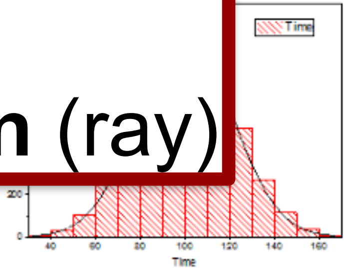
- Samples - “matter” of some form
  - Powders, single crystals, liquids, micelles in solution, reflecting surfaces

Common to all components:

They set, manipulate/interact with

or measure the **state of the neutron** (ray)

- Monitors - may probe

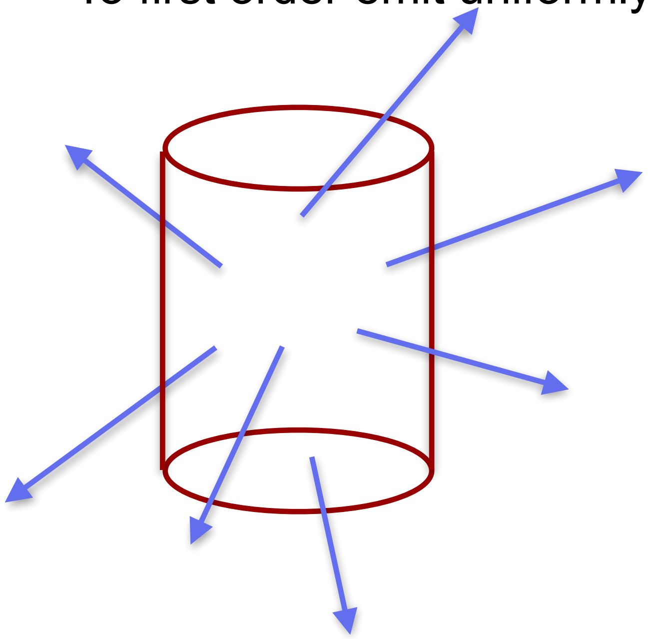


- Misc, obsolete
  - “Other stuff” and “Old stuff”

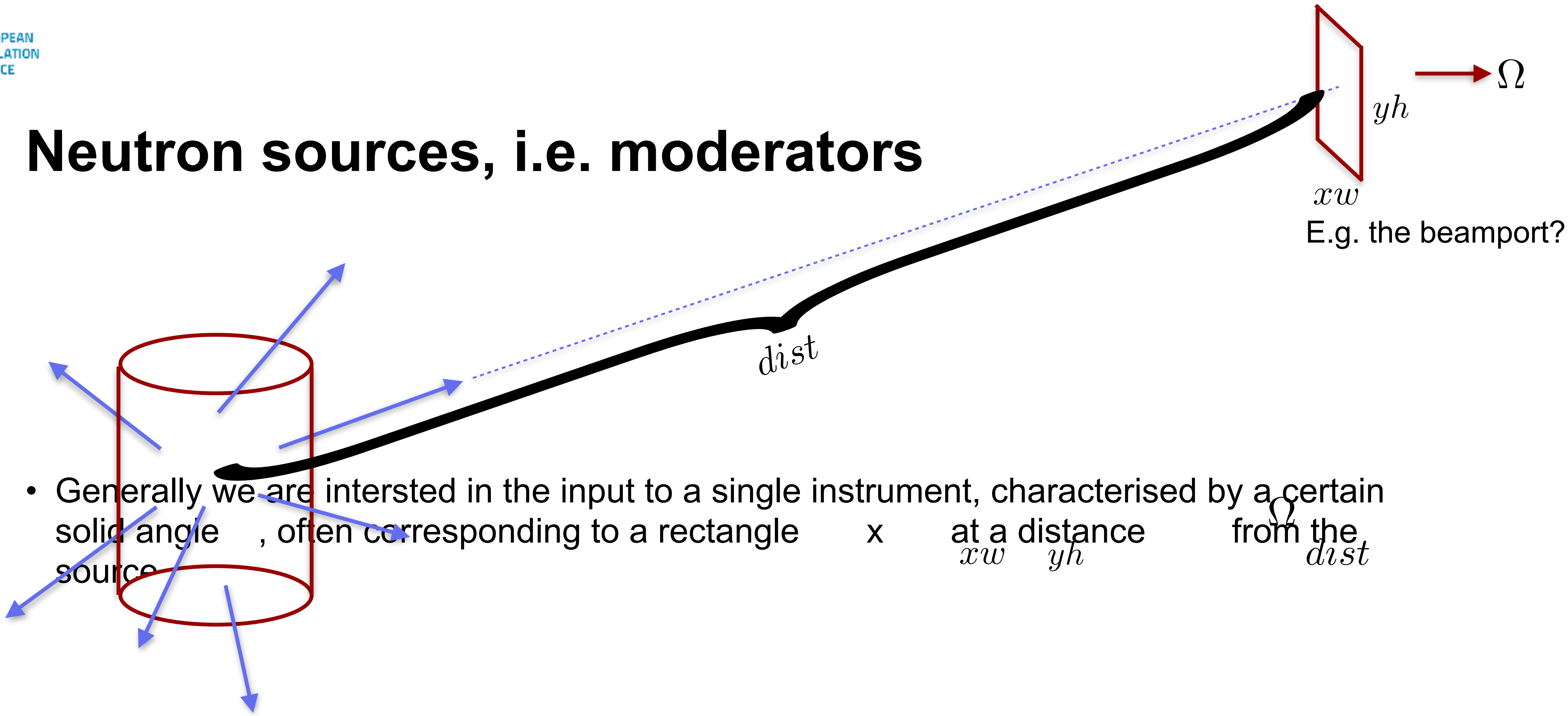
Other Stuff

Neutron sources, i.e. moderators

- To first order emit uniformly into 4π steradian

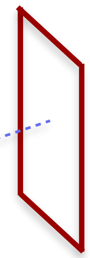


Neutron sources, i.e. moderators

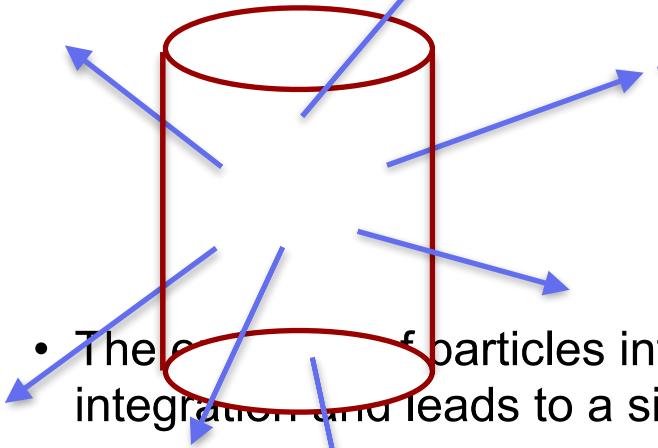


- Generally we are intersted in the input to a single instrument, characterised by a certain solid angle Ω, often corresponding to a rectangle xw x yh at a distance $dist$ from the source

Neutron sources, i.e. moderators



- The emmission 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



$I(\lambda)$ $[n/s/str]$

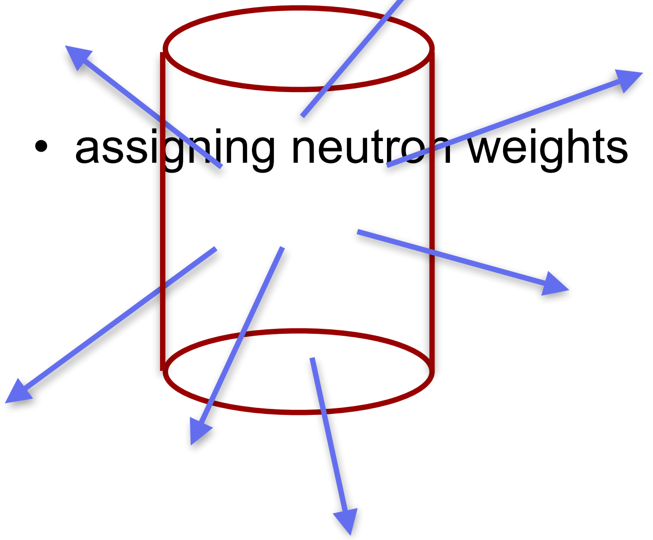
$I(\lambda, t)$ $[n/s/str]$

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

- 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 McStas, 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

Neutron sources, i.e. moderators

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



- assigning neutron weights such that

Ω

$$I_\Omega(\lambda, t, \vec{r})\ [n/s]$$

$$p$$

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

2021 Virtual ISIS McStas School

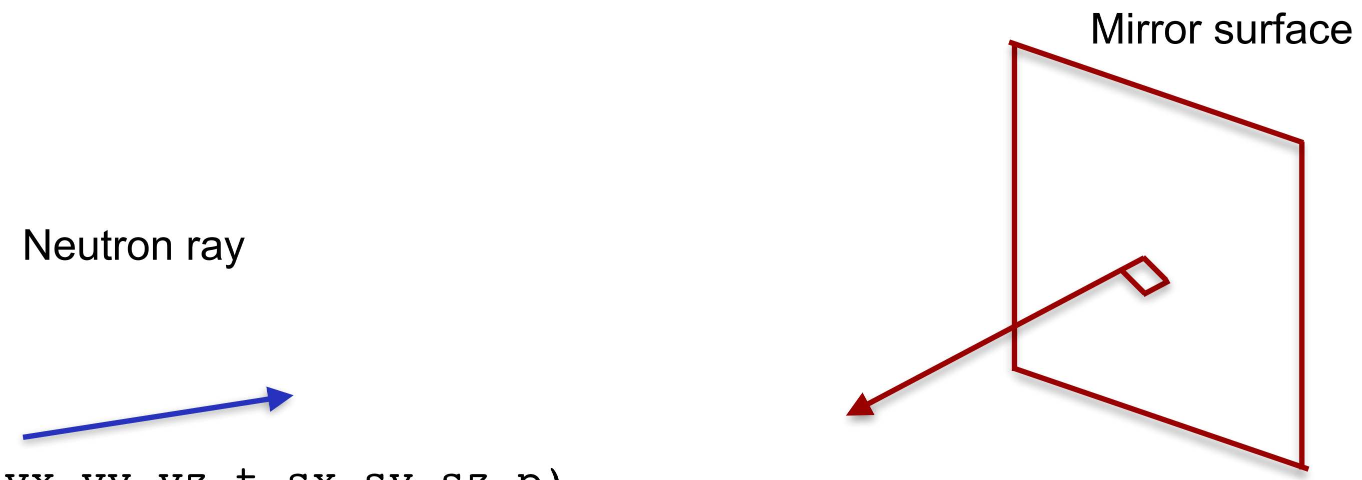
Neutron rays in McStas - what are they?

- Defining the neutron 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 E_{kin} (in the code variables `vx,vy,vz`)
  - 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 **polarisation** (in the code the variables `sx,sy,sz`)

Neutron **ray** in McStas:	
Location	`x,y,z`
Velocity	`vx,vy,vz`
Time	`t`
Polarisation.	`sx,sy,sz`
Intensity	`p`

Neutron (ray)-matter interaction 1: reflecting surface

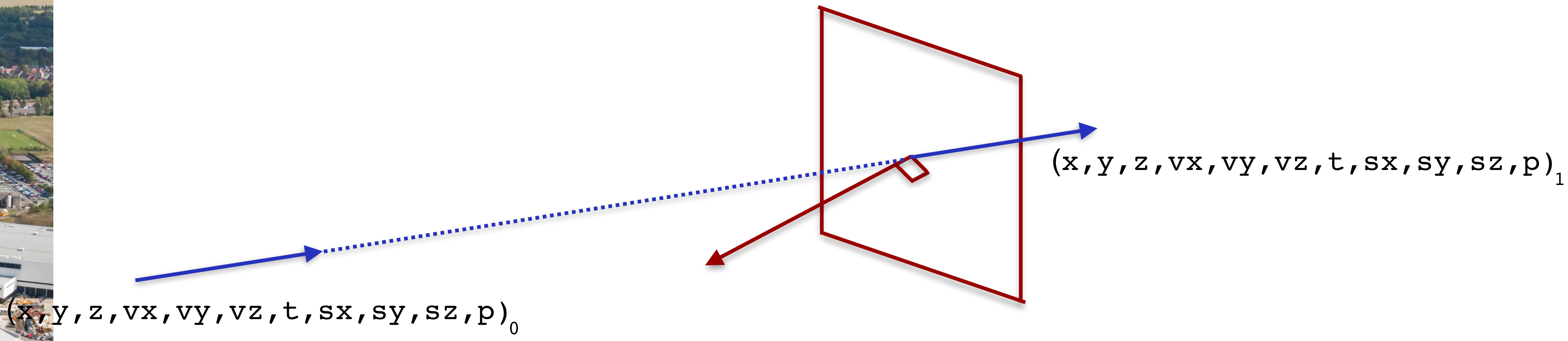
- 1 starting situation



$(x, y, z, vx, vy, vz, t, sx, sy, sz, p)_0$

Neutron (ray)-matter interaction 1: reflecting surface

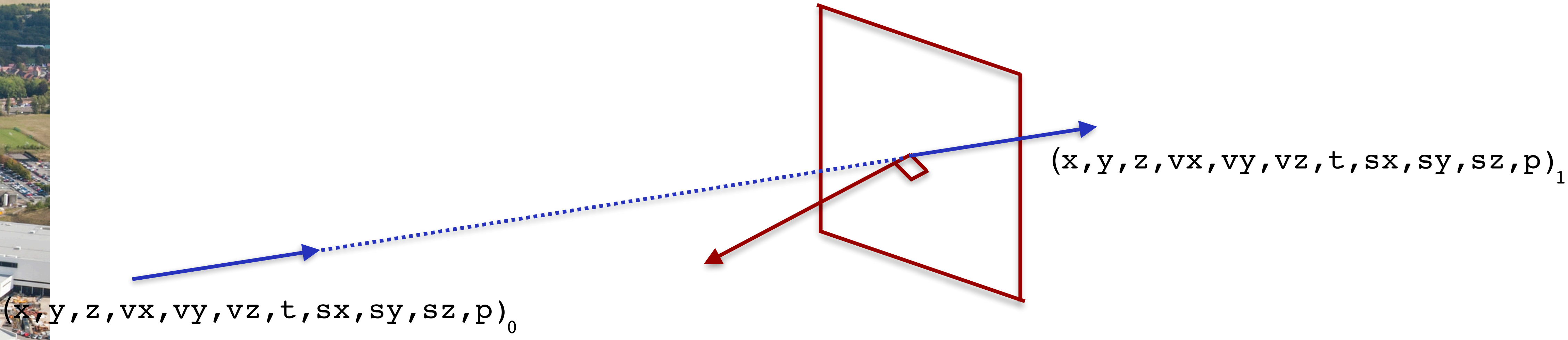
- 2. Propagate to the mirror surface



2021 Virtual ISIS McStas School

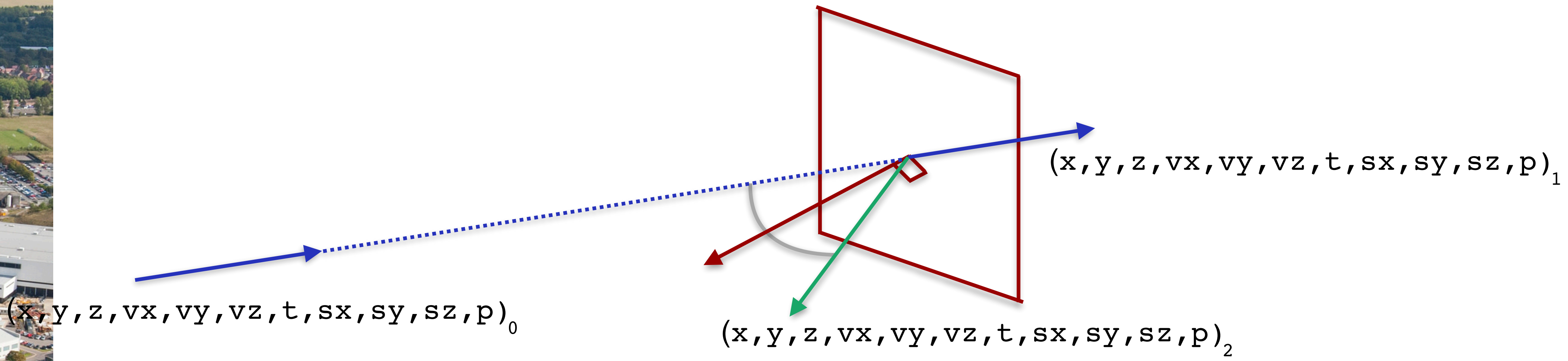
Neutron (ray)-matter interaction 1: reflecting surface

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



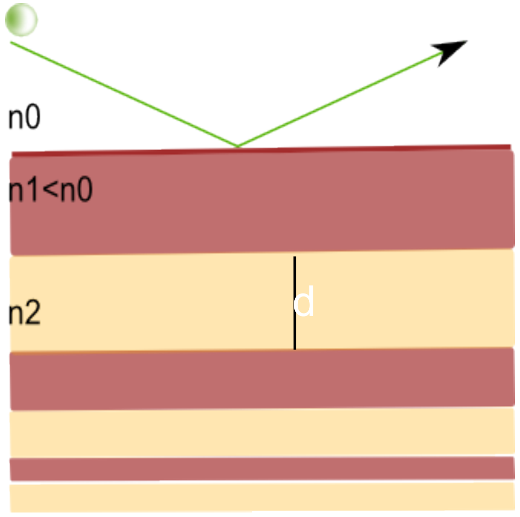
Neutron (ray)-matter interaction 1: reflecting surface

- 4. Reflect



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

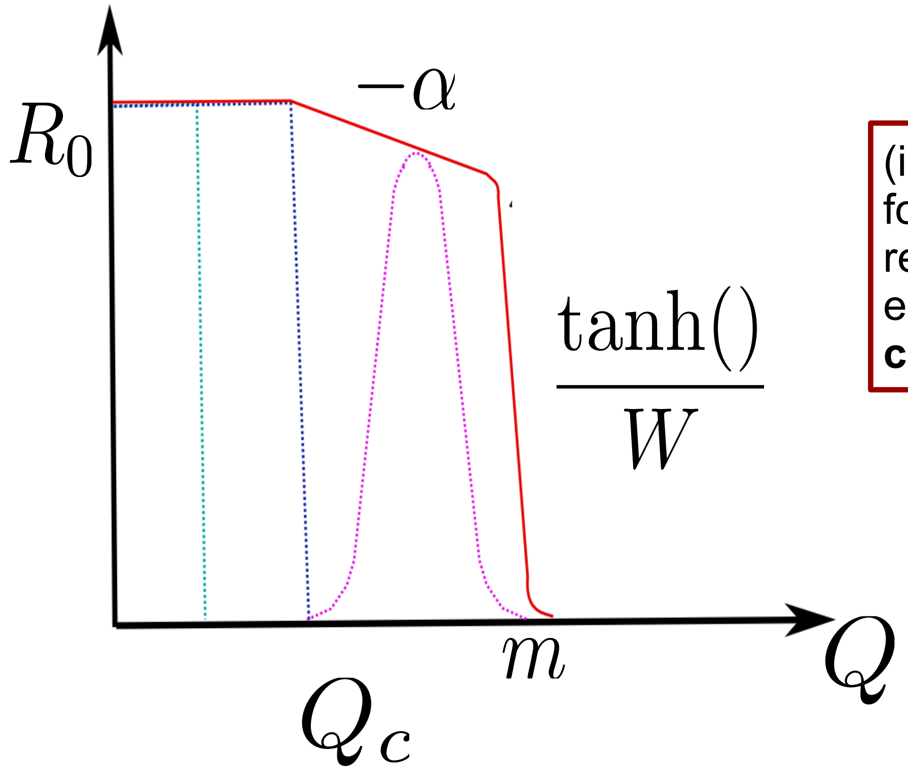
Parametrisation of reflectivity on mirrors etc.



$$V = \frac{2\pi\hbar^2}{m} bN \quad \sin\theta < \sqrt{\frac{mV}{2\pi^2\hbar^2}}\lambda$$

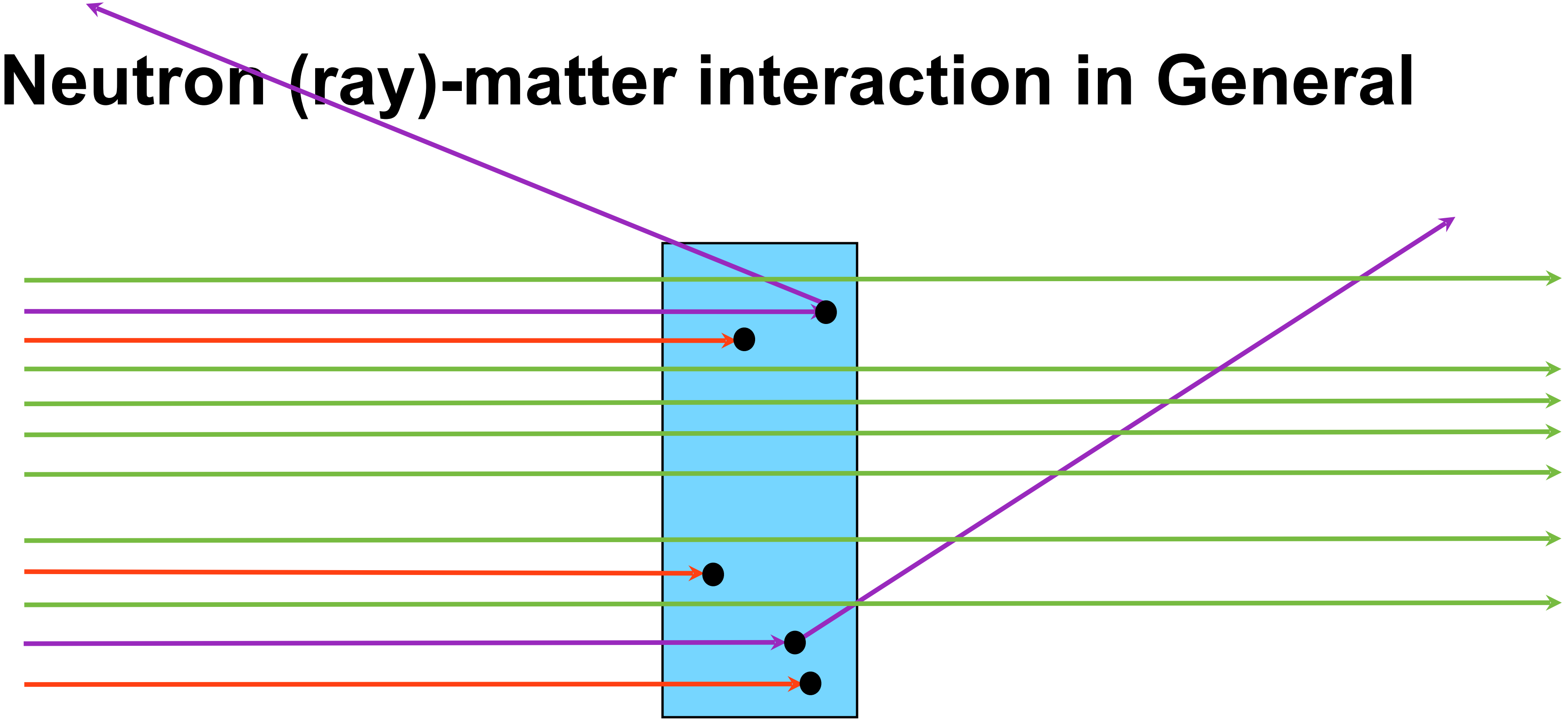
$$m = \frac{\theta_{mirror}}{\theta_{Ni}}$$

$$R_0 \cdot \left(1 - \frac{\tanh(Q - mQ_c)}{W}\right) \cdot \left(1 - \alpha(Q - Q_c)\right)$$



(i.e. Q is calculated for given neutron, reflectivity encoded in **changed *p* value**)

Neutron (ray)-matter interaction in General



absorbed, transmitted, or scattered

2021 Virtual ISIS McStas School

absorption transmission

scattering

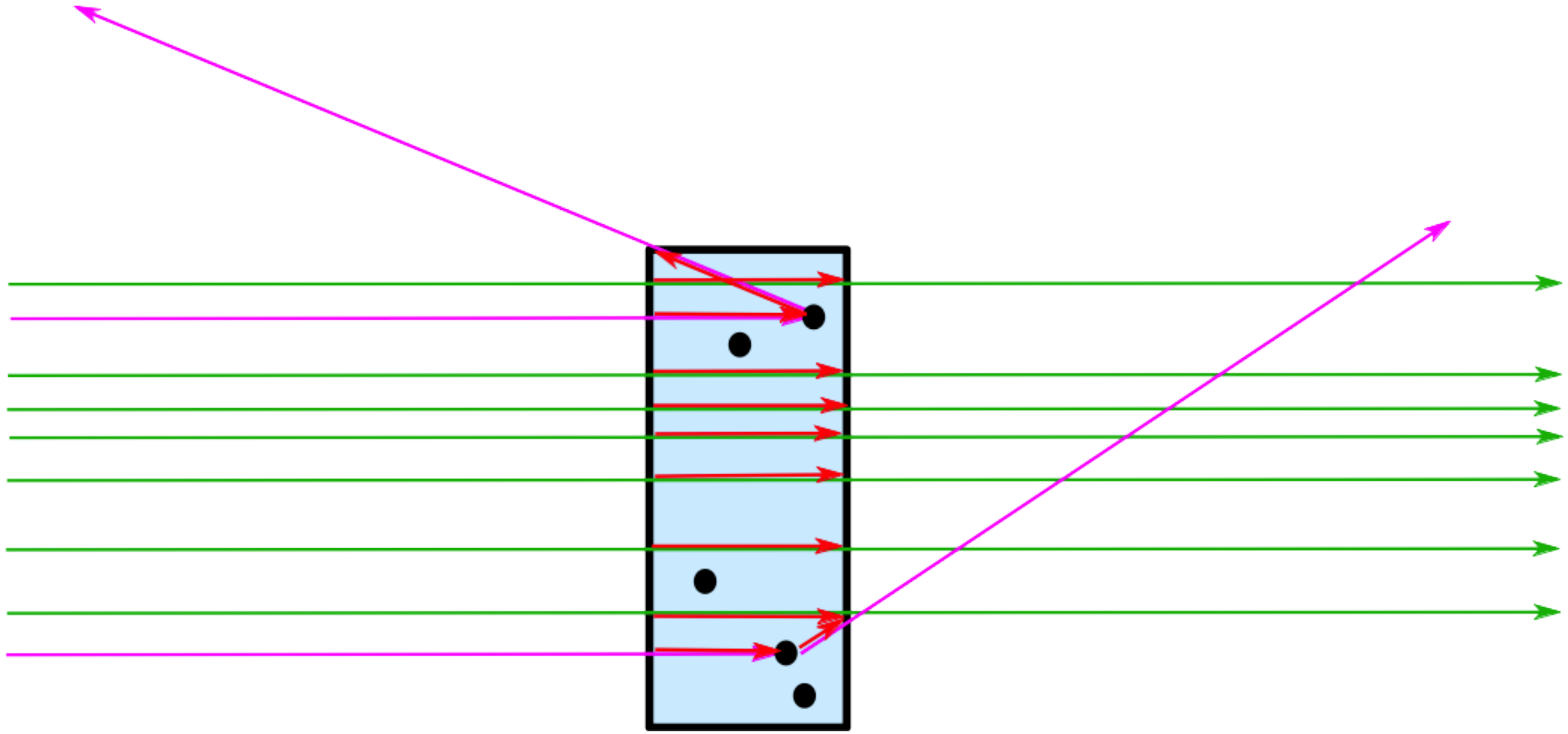
$$\boxed{p_A = (1 - e^{-\Sigma_T t})(\Sigma_A/\Sigma_T)}$$

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

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

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

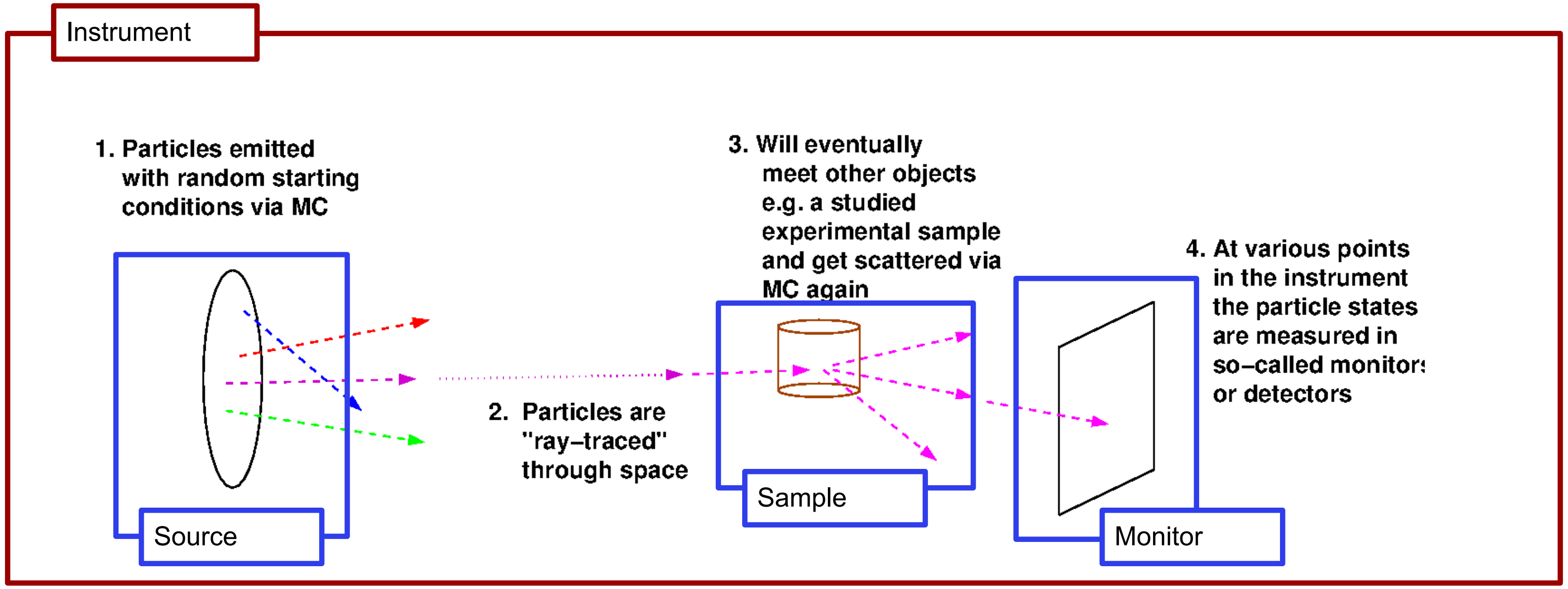
1 barn = 10^{-24}cm^2



transmitted+absorption, or scattered+absorption

Transport of weight through the instrument…

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



p_0 $\qquad$ p_j $\qquad$ p_n

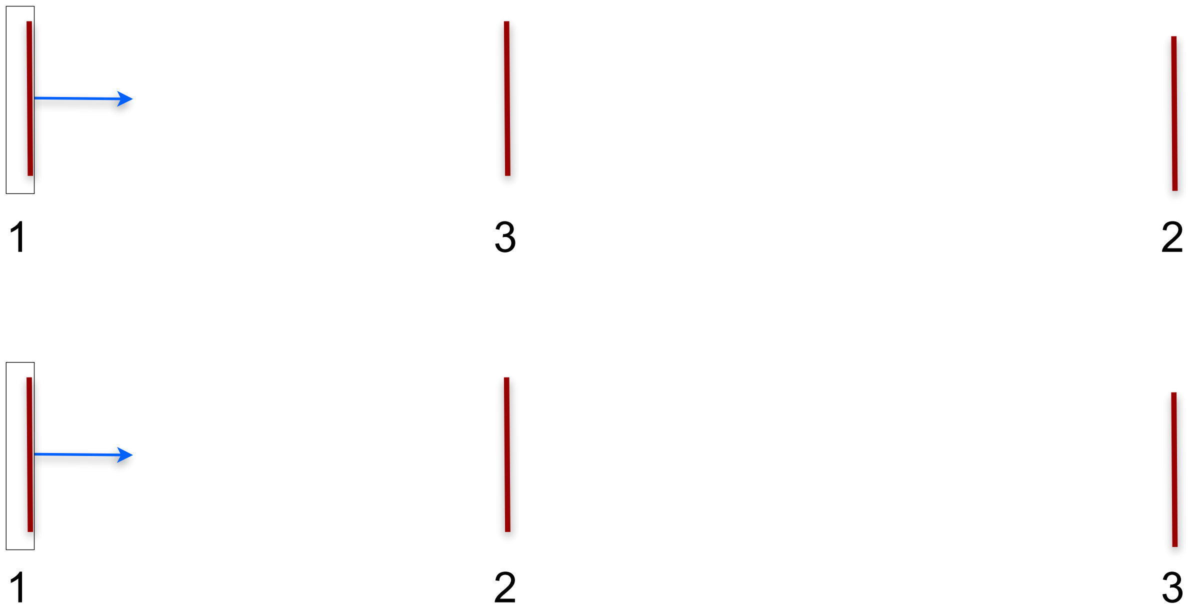
$$p_j = w_j p_{j-1}$$

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

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

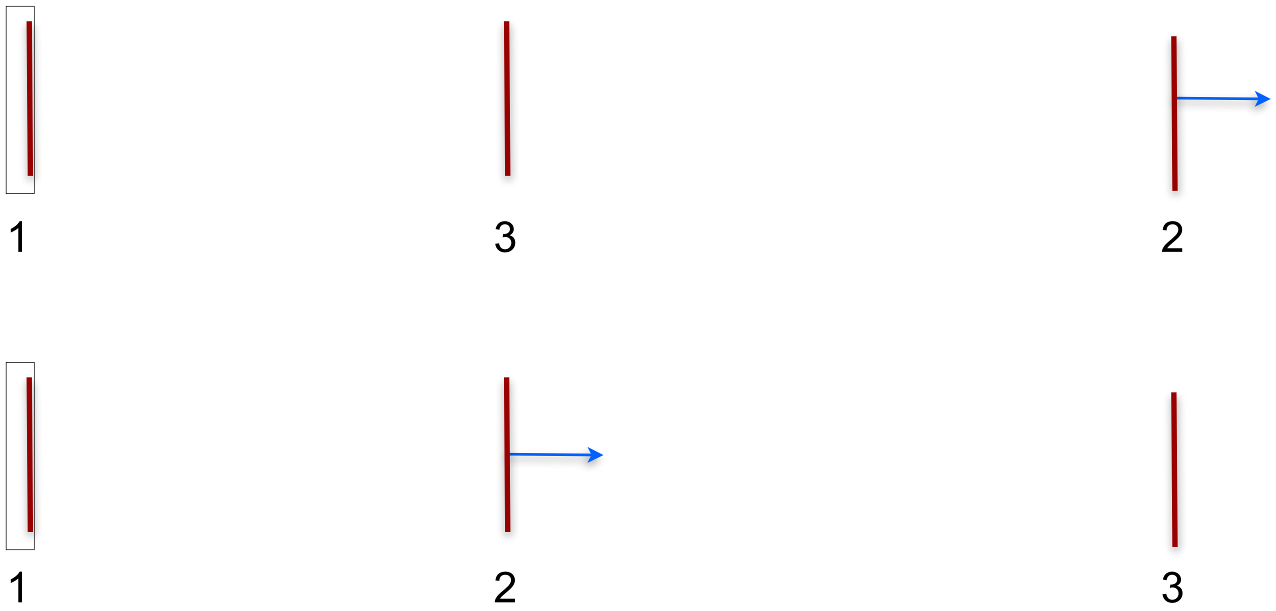
In case of "branching", i.e. multiple outcomes, it is clear that $\sum_b f_{\mathrm{MC,b}} = 1$

To first order, McStas is linear and follows sequence of components in your file…



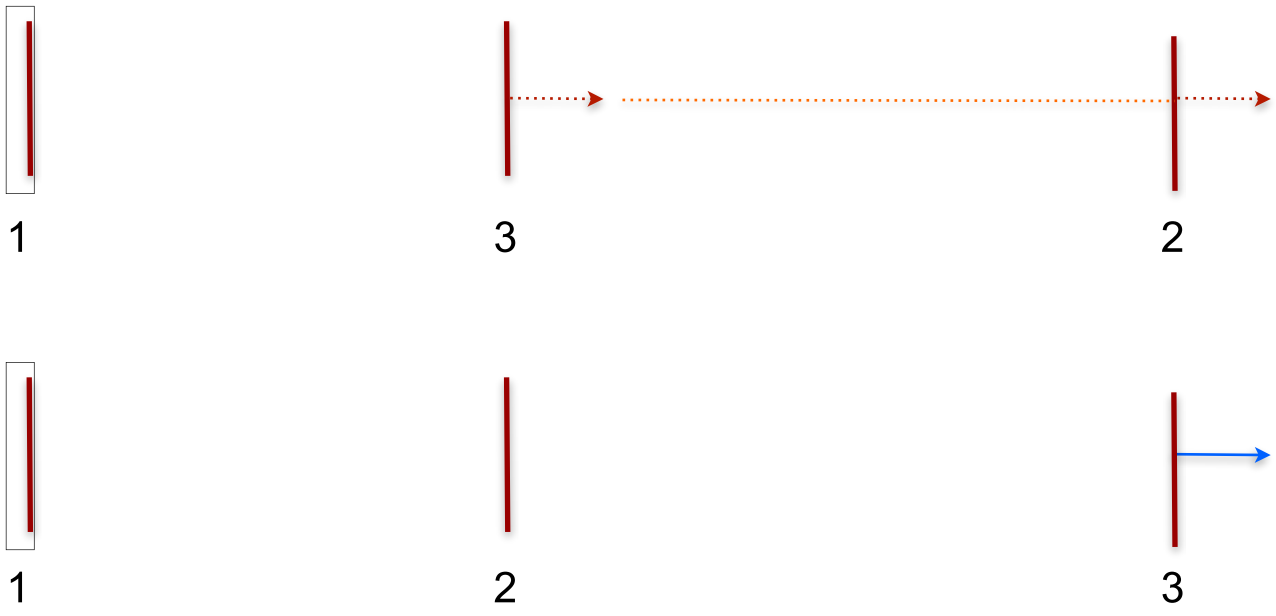
Starting at the source

To first order, McStas is linear and follows sequence of components in your file…



Moving to first comp in the list

To first order, McStas is linear and follows sequence of components in your file…



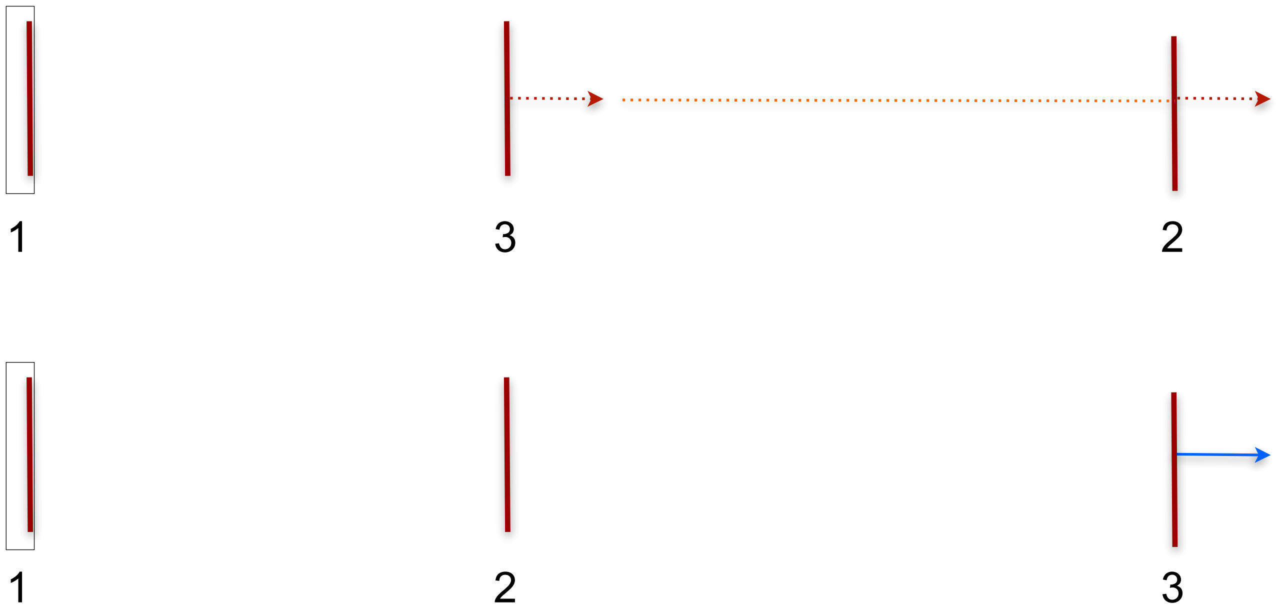
Moving to 3rd comp in list requires “moving back in time”.

Default behavior is to ABSORB this type of neutron.

For monitors use restore_neutron=1 in this case.

For homegrown comps use ALLOW_BACKPROP macro.

To first order, McStas is linear and follows sequence of components in your file…



The order of components is important,

and in general overlaps should be avoided!

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

Default behavior is to ABSORB this type of neutron.

For monitors use restore_neutron=1 in this case.

For homegrown comps use ALLOW_BACKPROP macro.

Units used - and differences to Vitess

- Generally SI-Units, e.g. **meters** and **seconds** etc.
- Added neutron-scattering meaningful quantities of **E[meV]**, **λ[Å]** and cross sections in **σ [barns]**

Main differences / difficulties translating between	McStas	⟷	Vitess
• Different length-units for placement / sizes	**[m]**		[cm]
• Different propagation-coordinate system	z ∥ beam y vertical x ⊥ beam		x ∥ beam z vertical y ⊥ beam

- McStas explicitly and automatically contains “free space”, Vitess has this “inside” the modules or by adding “spacewindow”
- Sources in McStas always propagates “by virtual window”, i.e. does not itself propagate

McStas ⟷ Vitess compatibility features

- MCPL particle list is supported from McStas 2.3- (2016) and Vitess 3.4 (2018)
- McStas includes Vitess_output and Vitess_input, but better use MCPL
- mcstas2vitess can be used to port McStas components to Vitess
- Vitess_ChopperFermi is the Vitess Fermi chopper ported to McStas

Friendly competition and collaboration for 1.5 decades! - e.g. Klaus Lieutenant was part of McStas team for 2 years