Technical University of Denmark

DTU

Application of the MCNPX-McStas interface for shielding calculations and guide design at ESS

Klinkby, Esben Bryndt; Willendrup, Peter Kjær; Bergbäck Knudsen, Erik; Lauritzen, Bent; Nonbøl, Erik; Bentley, Philip; Filges, Uwe

Publication date:
2013

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

DTU Library
Technical Information Center of Denmark

Application of the MCNPX-McStas interface for shielding calculations and guide design at ESS

Esben KLINKBY[1,3],

Peter Kjær WILLENDRUP[2,3], Erik Bergbäck KNUDSEN [2,3], Bent LAURITZEN[1,3], Erik NONBØL[1,3], Phillip BENTLEY[4], Uwe FILGES[5]

1) DTU Nutech, Box 49, 4000 Roskilde, Denmark
2) DTU Physics, Fysikvej 307-312, 2800 Kgs. Lyngby, Denmark
3) ESS design update programme, Denmark
4) European Spallation Source ESS AB, Box 176,S-221 00 Lund, Sweden
5) Paul Scherrer Institut, 5232 Villigen, Switzerland. ESS design update programme, Switzerland

Motivation

➢ ESS will be a long pulse spallation source and deliver neutrons to 22 instruments located ~20-200m from the target station

$\Downarrow$

➢ Guide demands are unprecedented

BUT…

➢ Guides & shielding is expensive!

➢ Useful to have a tool that:
- can monitor where in a guide neutrons are lost
- allows to optimize reflectivity requirements along a guide
- serves as an input for dose-rate calculations along guide (n,γ)
- works within the work-flow accustomed to instrument designers (McStas)

➢ **McStas Scatterlogger** is the backbone that facilitates this usage
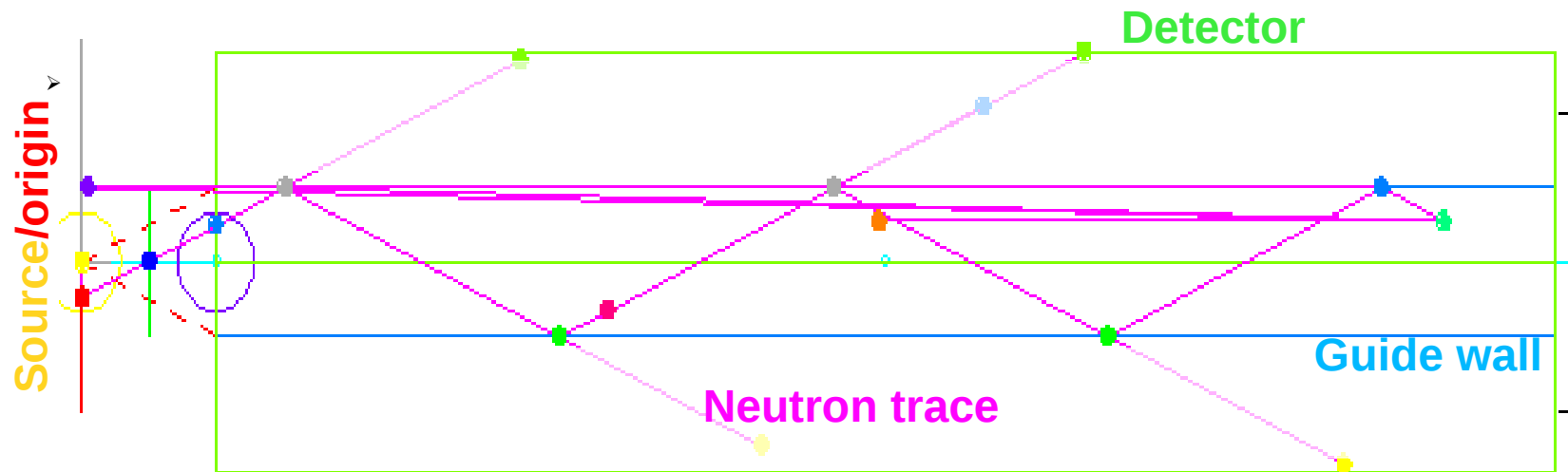
Scatterlogger: Implementation & usage

➢ At each scattering, for any McStas component (eg. a guide), the incoming and outgoing neutron state can be temporally stored & analyzed

At each scattering:

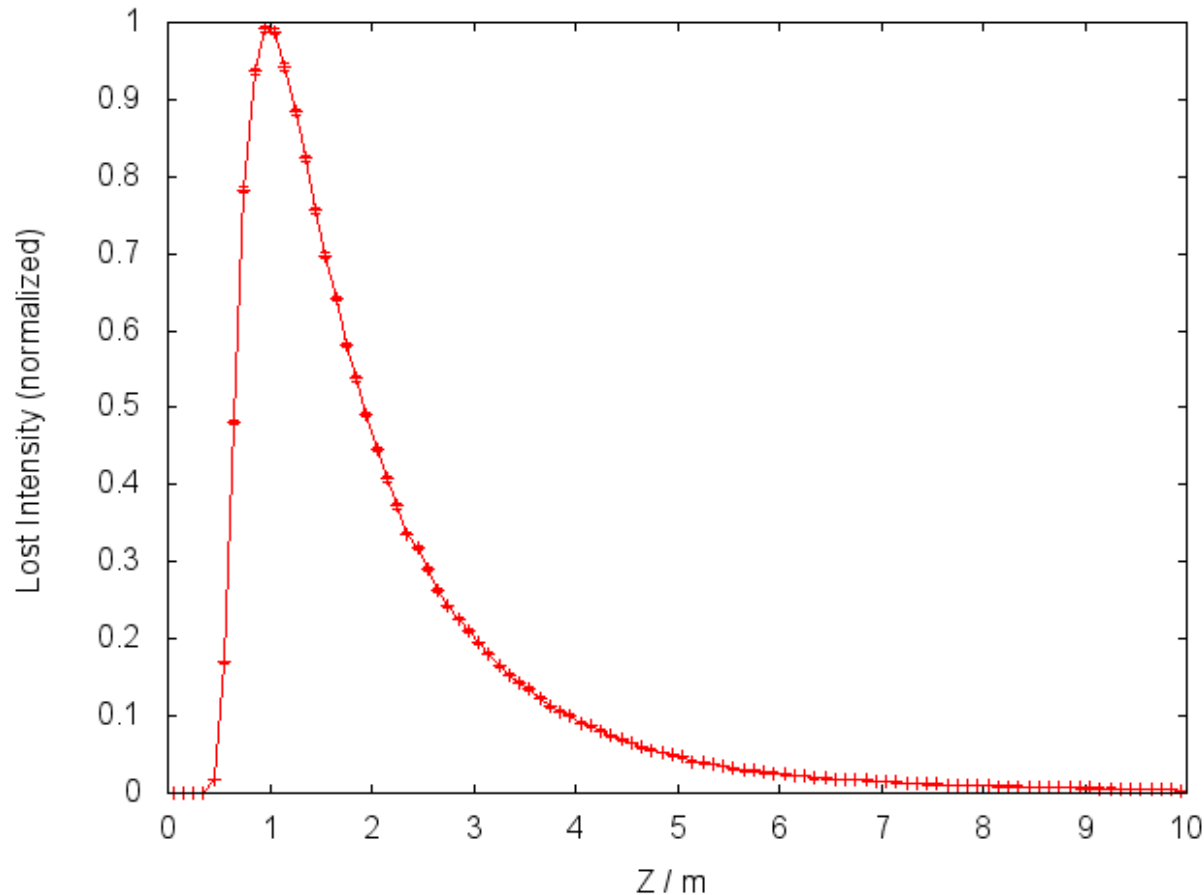Incomming state: $n_{in} = (\vec{x}, \vec{v}_{in}, t, w_{in})$

Transmitted state: $n_{trans} = (\vec{x}, \vec{v}_{in}, t, w_{trans})$

Reflected state: $n_{refl} = (\vec{x}, \vec{v}_{out}, t, w_{in} - w_{itrans})$

Detector

Source/origin

Guide wall

Neutron trace

Example1: Lost intensity

➢ Lost intensity along guide: w_{trans} *versus* $\vec{\mathbf{x}} \cdot \hat{\mathbf{z}}$

➢ Depends strongly on specific guide design, and incoming neutrons (i.e. source) spectrum and divergence.

Example2: Reflectivity

➢ Neutrons are reflected if the energy/incident angle is low enough
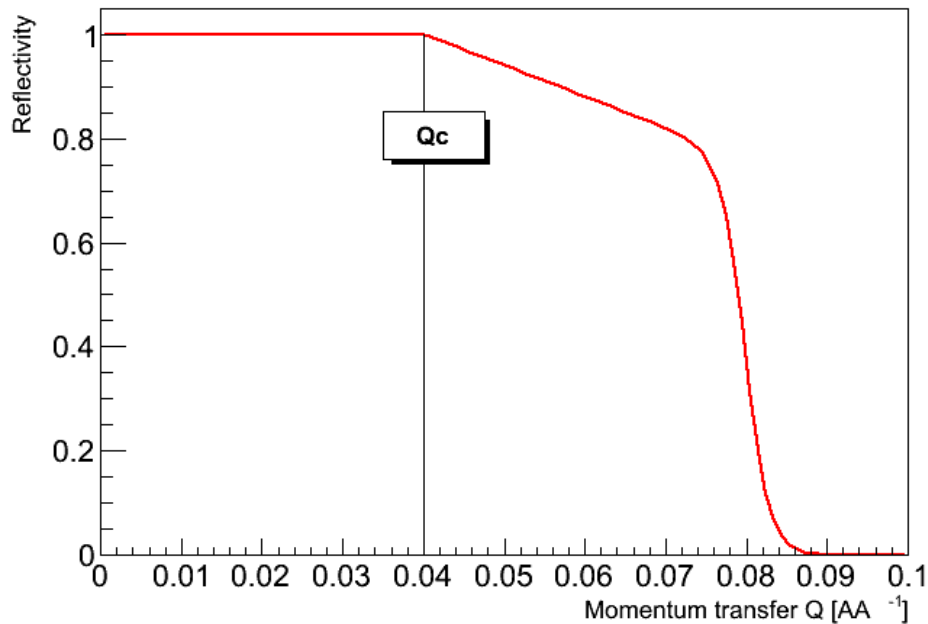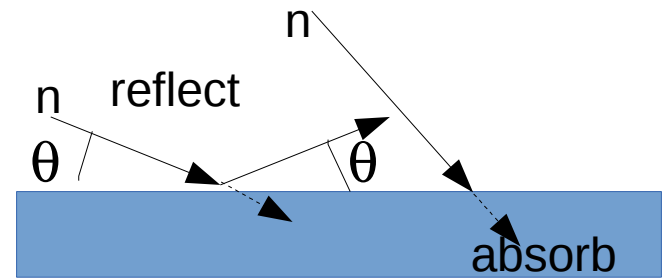
$$\cos 2\theta =(\vec{\mathbf{v}}_{in} \cdot \vec{\mathbf{v}}_{out})/| \mathbf{v}_{in}|^2$$
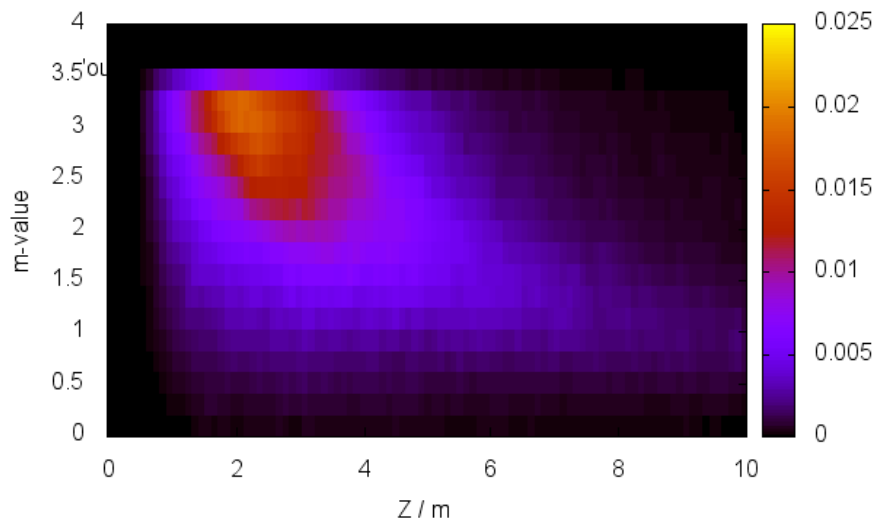$$k=| \mathbf{v}_{in}|\cdot m_n/\hbar$$
$$\Downarrow$$
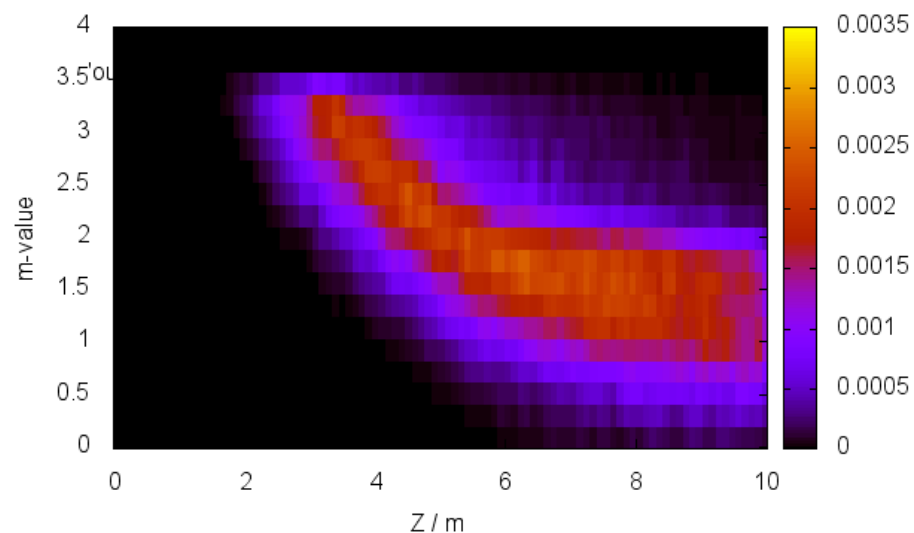$$m_{min} = 2\cdot k\cdot \sin(\theta) / 0.0219$$

➢ Given a neutron state and a guide geometry, m_{min} can be calculated at a scattering: The minimum mirror reflectivity requirement which would reflect the neutron without loss
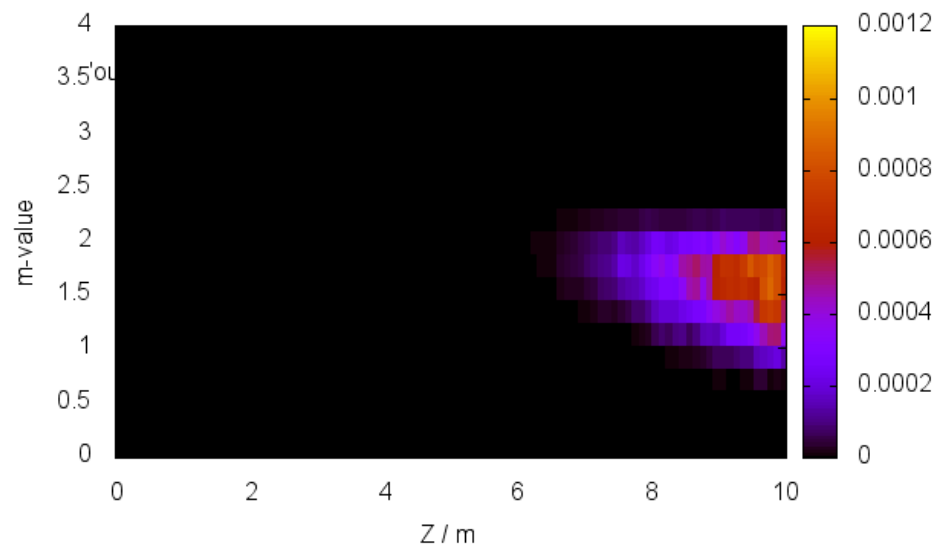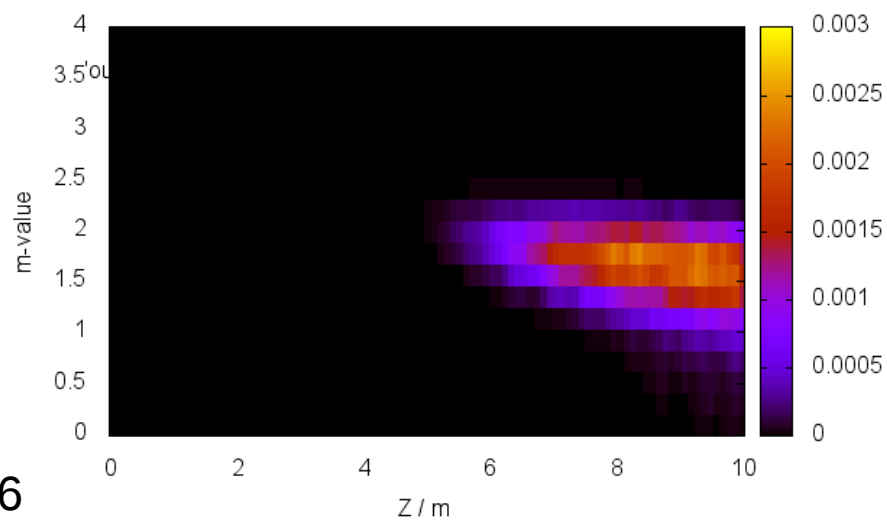
5

Example2: Reflectivity

Impinging intensity - 1st reflection

Impinging intensity - 2nd reflection

Impinging intensity - 3rd reflection

Example3: Background along guide
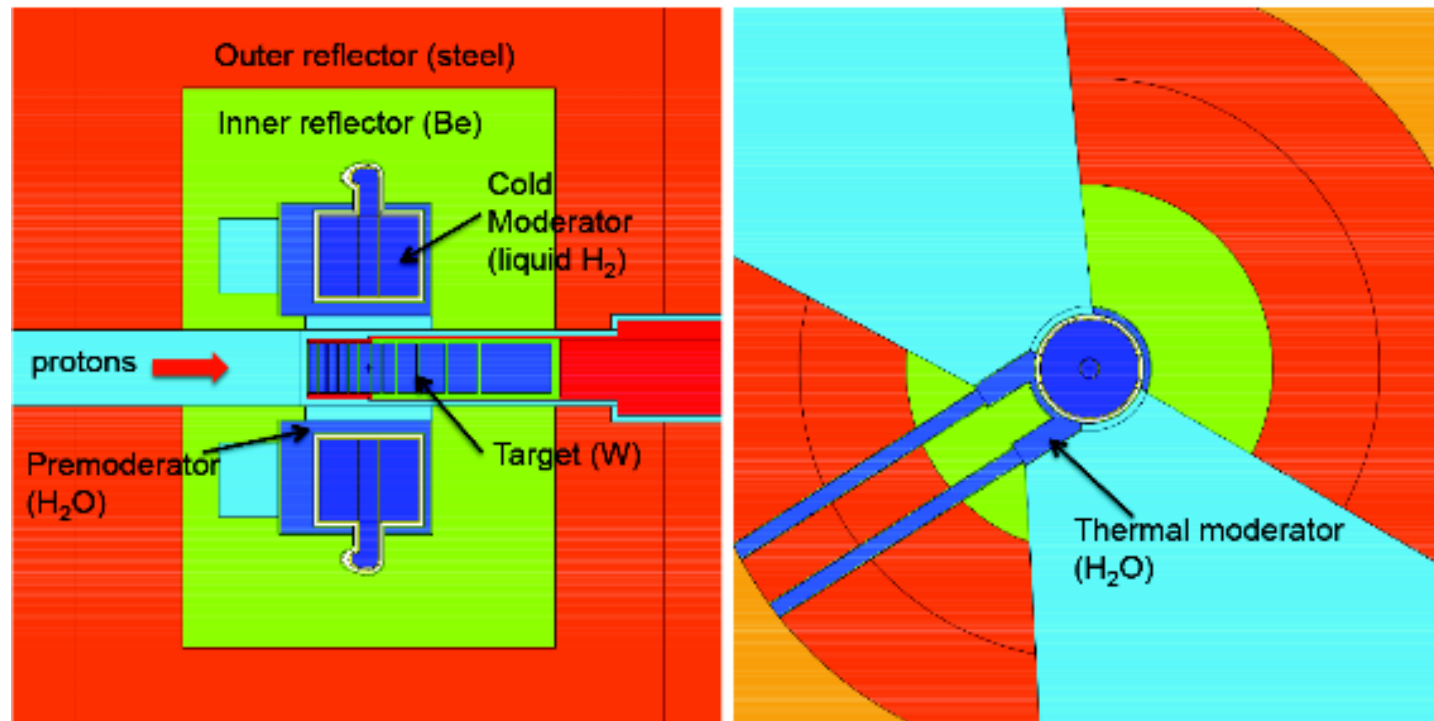
→ **I. Neutrons generated with MCNPX**
   II. Handed to McStas through SSW interface [1]
   III. Unreflected neutrons returned to MCNPX for dose-rate calculation

Example3: Background along guide

I. Neutrons generated with MCNPX

→ **II. Handed to McStas through SSW interface [1]**

III. Unreflected neutrons returned to MCNPX for dose-rate calculation

Guide cross-section

2cm

50cm

steel

MCNPX

source

2m

12cm

12cm

5°

50m

MCNPX

steel

McStas

SSW/SSR

Detectors

8 [1] E. Klinkby et al. 'Interfacing MCNPX and McStas for simulation of neutron transport.' Nucl. Instr & Meth A , 700: p106, 2013.

Example3: Background along guide

I. Neutrons generated with MCNPX

II. Handed to McStas through SSW interface [1]

→ **III. Unreflected neutrons returned to MCNPX for dose-rate calculation**

At each scattering:

⟹

Incomming state: $n_{in} = (\mathbf{x}, \mathbf{v}_{in}, t, w_{in})$

Transmitted state: $n_{trans} = (\vec{\mathbf{x}}, \vec{\mathbf{v}}_{in}, t, w_{trans})$

Reflected state: $n_{refl} = (\vec{\mathbf{x}}, \vec{\mathbf{v}}_{out}, t, w_{in} - w_{itrans})$

➢

Example3: Background along guide

I. Neutrons generated with MCNPX

II. Handed to McStas through SSW interface [1]

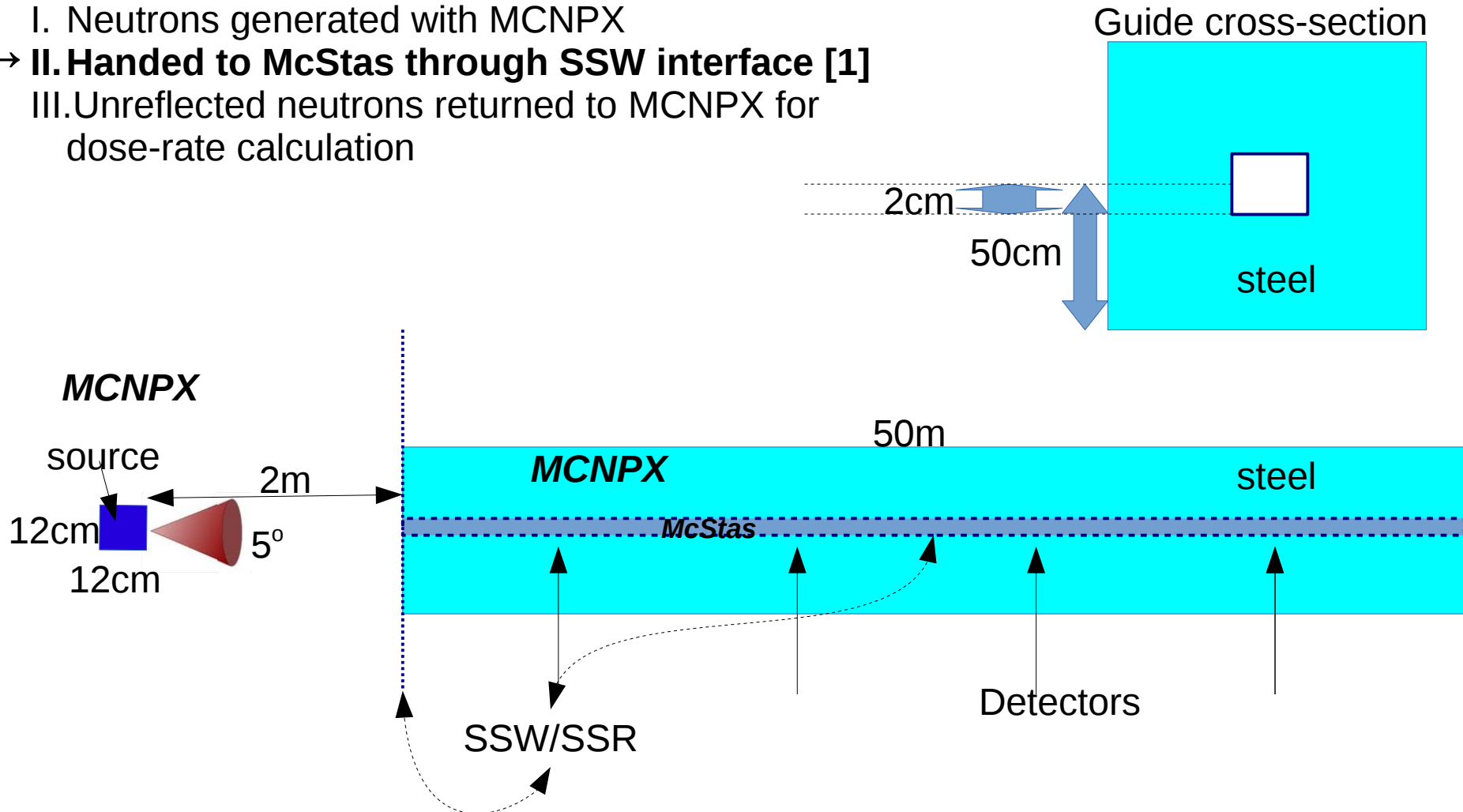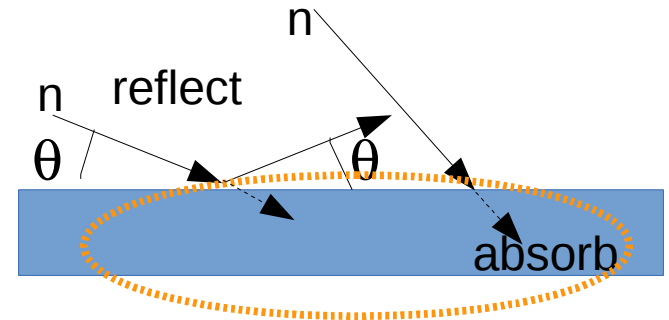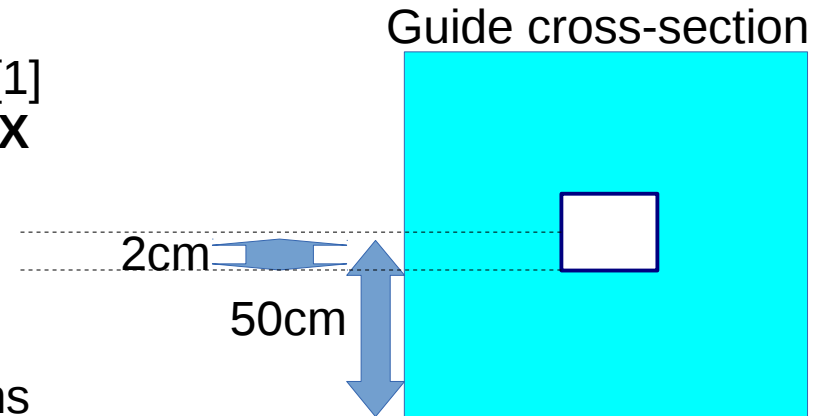→ **III. Unreflected neutrons returned to MCNPX for dose-rate calculation**

Guide cross-section

2cm

50cm

Guide end overilluminated by energetic neutrons
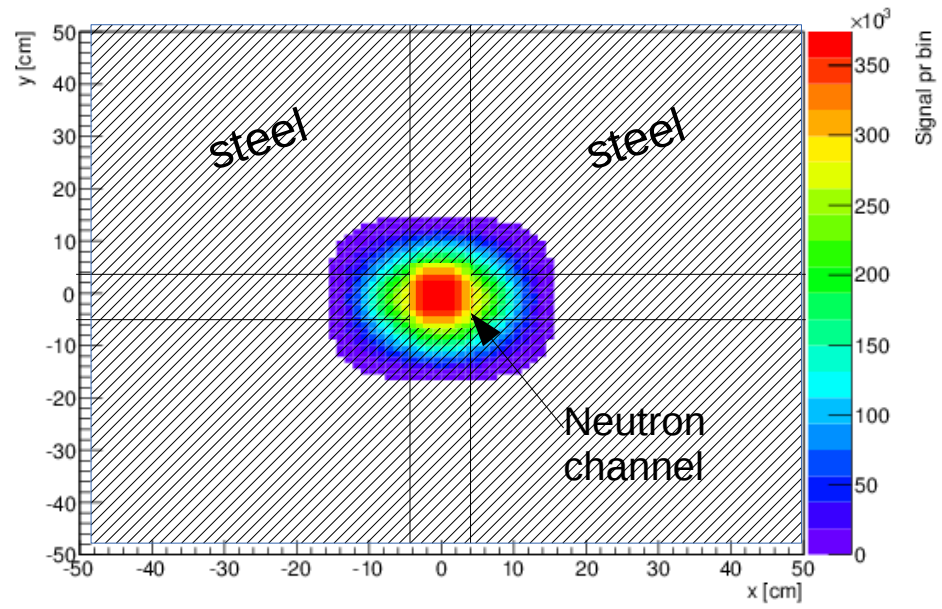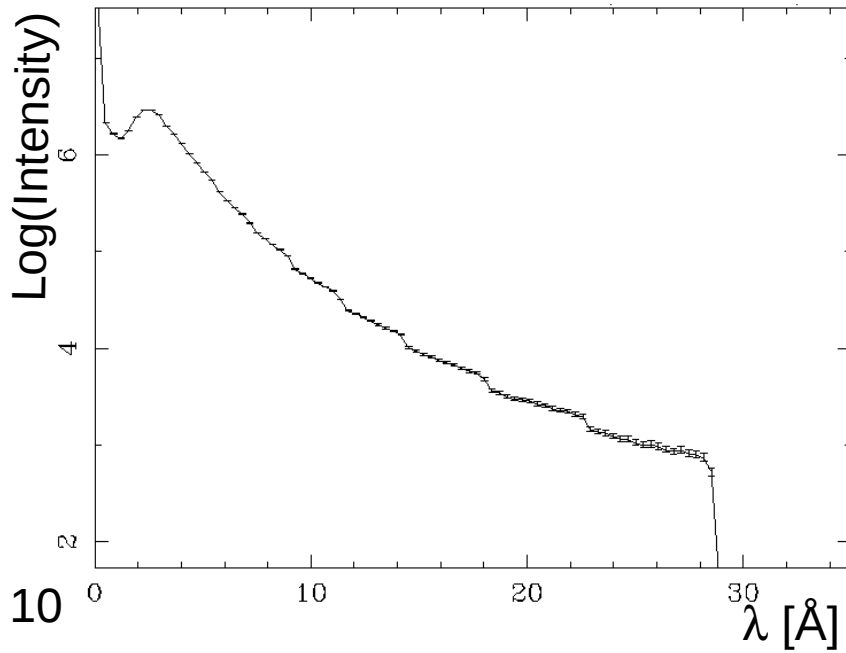
Example3: Background along guide
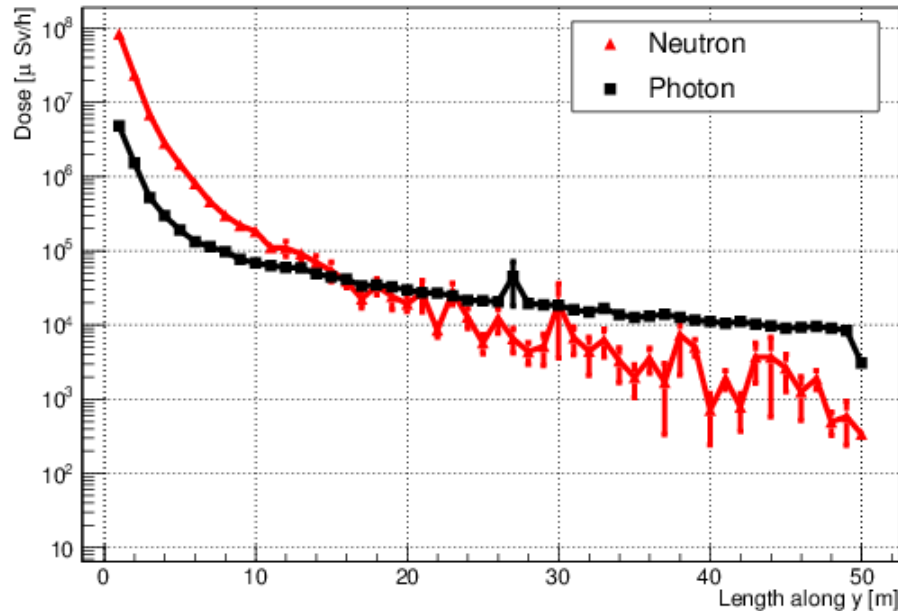
Straight guide

Curved guide ($r_{curvature}$=1500m)
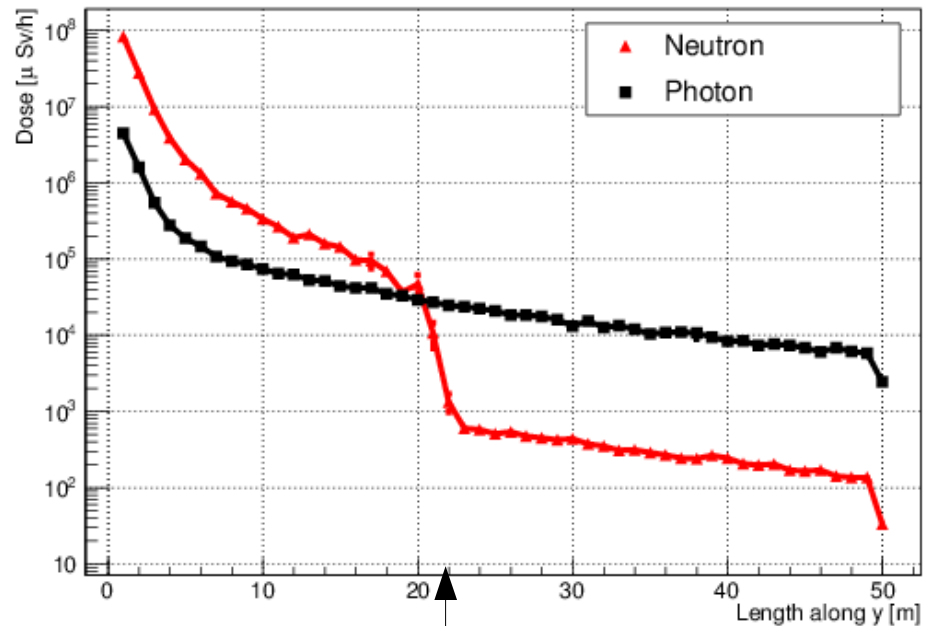
Line-of-sight lost

➢ Dose-rates, measured 5cm in the steel converted from flux according to official Swedish radiation protection procedures

11

Example3: Background along guide

Straight guide
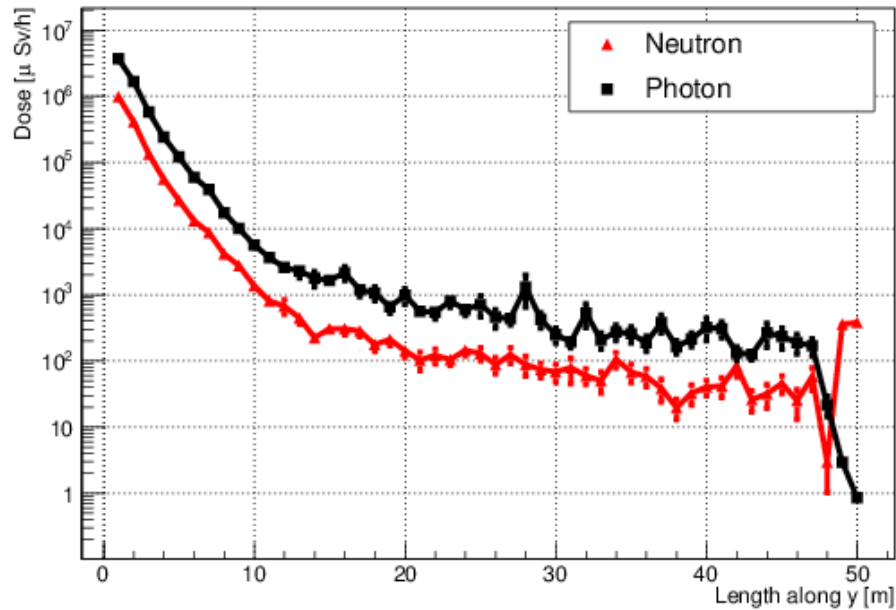
Curved guide ($r_{curvature}$=1500m)
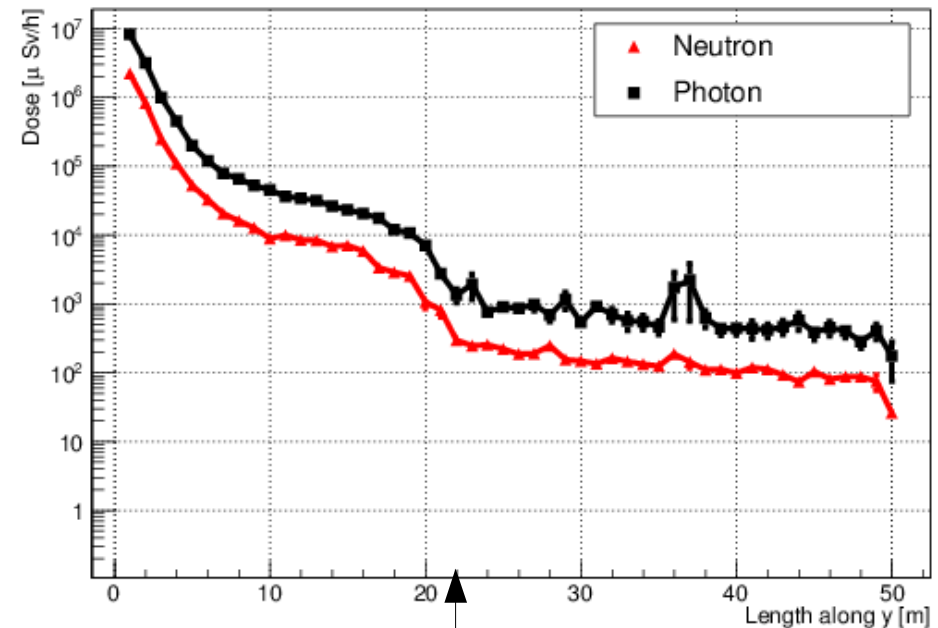
Line-of-sight lost

➢ Restricting to $\lambda \in \left\{ 0.5\,\text{Å} - 1.0\,\text{Å} \right\}$
➢ Photon dose-rate follows neutron dose-rate ✓

Conclusions

Lost Intensity

➢ Logging mechanism useful for guide design:

    ➢ Monitor **intensity loss**
    ➢ Optimizing use of high/low **reflectivity** mirrors
    ➢ Calculate **dose-rates** along guide

➢ Works in instrument designers accustomed work-flow (**McStas**)

Impinging intensity - 1st reflection
Reflectivity

Prospects

➢ Could be used to evaluate gamma and (energetic) neutron background at the sample position / instrument / detectors.

➢ Must be utilized on an instrument to instrument basis

Dose-rate

13
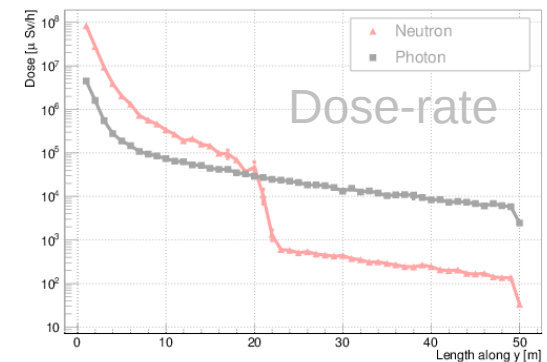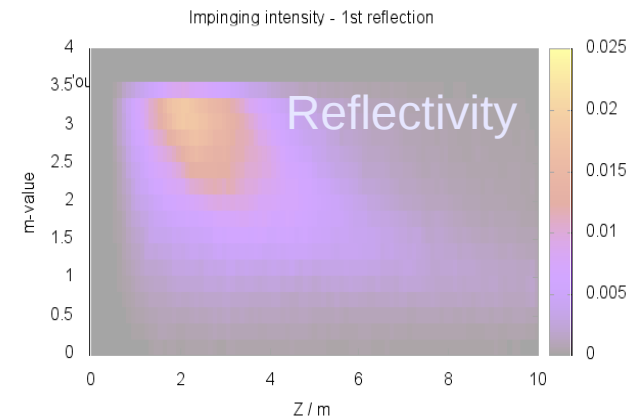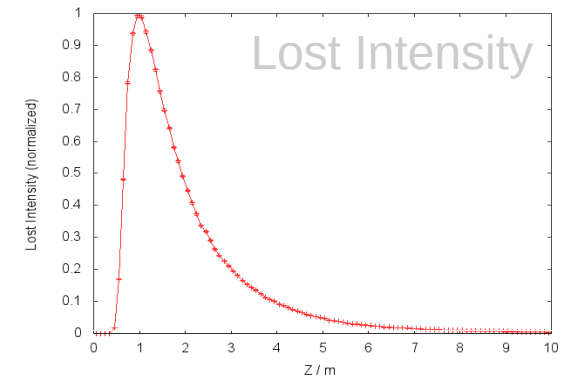
Backup slides

Reflectivity curve

$$
\begin{aligned}
R &= \frac{R_0}{2}\left(1 - tanh\frac{Q - m \cdot Q_c}{W}\right) \times (1 - \alpha(Q - Q_c)) && for \quad Q > Q_c \\
R &= R_0 && otherwise \qquad (1)
\end{aligned}
$$

where Q is the scattering vector, Q_c is the critical scattering vector, R_0 is the low angle reflectivity constant, W is the width of supermirror cut-off, α is the reflectivity slope, and m is the m-value of the material.

Taken from measurements

Front-end: logger

COMPONENT src = Source_simple(
     radius = 0.1, dist = 1, focus_xw = 0.1, focus_yh = 0.1, lambda0=5, dlambda=4.9)
AT (0, 0, 0) RELATIVE Origin

COMPONENT psd0=PSD_monitor(
     xwidth=0.1, yheight=0.1, filename="psd0")
AT(0,0,0.5) RELATIVE PREVIOUS

COMPONENT s1=**Scatter_logger()**
AT(0,0,1) RELATIVE src

COMPONENT guide_simple = Guide(
     w1 = 0.1, h1 = 0.1, w2 = 0.1, h2 = 0.1, l = 10, R0 = 0.99,
     Qc = 0.0219, alpha = 6.07, m = 2, W = 0.003)
AT (0, 0, 1) RELATIVE src

COMPONENT s2=**Scatter_logger_stop(logger=s1)**
AT(0,0,0) RELATIVE PREVIOUS

Back-end: logger iterator

COMPONENT a0=Arm()
AT(0,0,0) ABSOLUTE

COMPONENT iter1 = **Scatter_log_iterator()**
AT(0,0,0) ABSOLUTE

COMPONENT mnd=Monitor_nD (
     restore_neutron=1, yheight=10, radius=M_SQRT2*0.1,
     options="previous no slit y bins=100", filename="mnd1.dat")
AT(0,0,5) RELATIVE guide_simple
ROTATED (90,0,0) RELATIVE guide_simple

COMPONENT iter2 = **Scatter_log_iterator_stop(iterator=iter1)**
AT(0,0,0) RELATIVE iter1

COMPONENT a1 = Arm()
AT (0,0,0) ABSOLUTE
JUMP a0 WHEN(MC_GETPAR**(iter2,loop)**)

10m setup
- guide & source details

COMPONENT Origin = Progress_bar()
AT (0,0,0) ABSOLUTE

COMPONENT src = Source_simple(
    radius = 0.1, dist = 1, focus_xw = 0.1, focus_yh = 0.1, lambda0=5, dlambda=4.9)
AT (0, 0, 0) RELATIVE Origin

COMPONENT psd0=PSD_monitor(
    xwidth=0.1, yheight=0.1, filename="psd0")
AT(0,0,0.5) RELATIVE PREVIOUS

COMPONENT s1=Scatter_logger()
AT(0,0,1) RELATIVE src

COMPONENT guide_simple = Guide(
    w1 = 0.1, h1 = 0.1, w2 = 0.1, h2 = 0.1, l = 10, R0 = 0.99,
    Qc = 0.0219, alpha = 6.07, m = 2, W = 0.003)
AT (0, 0, 1) RELATIVE src

m-value calculation

```
double mvalue;
int reflc;
int reflect_m-value(double *ns_tilde, struct Generalized_State_t *S0, struct Generalized_State_t *S1){
    /*position comes from "new" state*/
    ns_tilde[0]=S1->_x;ns_tilde[1]=S1->_y;ns_tilde[2]=S1->_z;
    /*velocity is the "old" state*/
    ns_tilde[3]=S0->_vx;ns_tilde[4]=S0->_vy;ns_tilde[5]=S0->_vz;
    /*time from new*/
    ns_tilde[6]=S1->_t;
    /*weight is impinging weight - old state*/
    ns_tilde[10]=S0->_p;

    double v = sqrt(S0->_vx*S0->_vx+S0->_vy*S0->_vy+S0->_vz*S0->_vz);
    double k = v*V2K;
    double theta  = 0.5*acos(scalar_prod(S0->_vx,S0->_vy,S0->_vz,S1->_vx,S1->_vy,S1->_vz)/(v*v));
    mvalue = 2*k*sin(theta)/0.0219;
    reflc=S1->_idx;
    return 0;
}
```

Another example: Specialized pseudo neutron state function → background along guide

```
/*position comes from "new" state*/
ns_tilde[0]=S1->_x;ns_tilde[1]=S1->_y;ns_tilde[2]=S1->_z;

/*velocity is the "old" state*/
ns_tilde[3]=S0->_vx;ns_tilde[4]=S0->_vy;ns_tilde[5]=S0->_vz;

/*time from new*/
ns_tilde[6]=S1->_t;
```

Same as before

```
/*weight is difference old-new to mean the neutrons "deposited" in the guide wall*/
ns_tilde[10]=S0->_p-S1->_p;
```

I.e.: The temporarily stored state is the ***un-reflected neutrons*** - normally discarded

Using the MCNPX-McStas interface: *Virtual_MCNP_ss_output.comp* (McStas 2.0), the simulation of absorbed neutrons proceeds:


Neutron and gamma trajectories (a few example events from MCNPX → Vised)

20 **Next**: use to calculate shielding requirements for realistic ESS guide geometry and source

New source – cold pencil

- ***Guide starts at z=200cm***

- c Source definition: cold, in guide opening

- sdef x=d2 y=d3 z=0.0 dir=d4 vec=0 0 1 erg=1e-8 par=n

- si2   h -1.0 1.0

- sp2   0   1

- si3   h -1.0 1.0

- sp3   0   1

- si4   h   0.99999 1 $~0.25grad
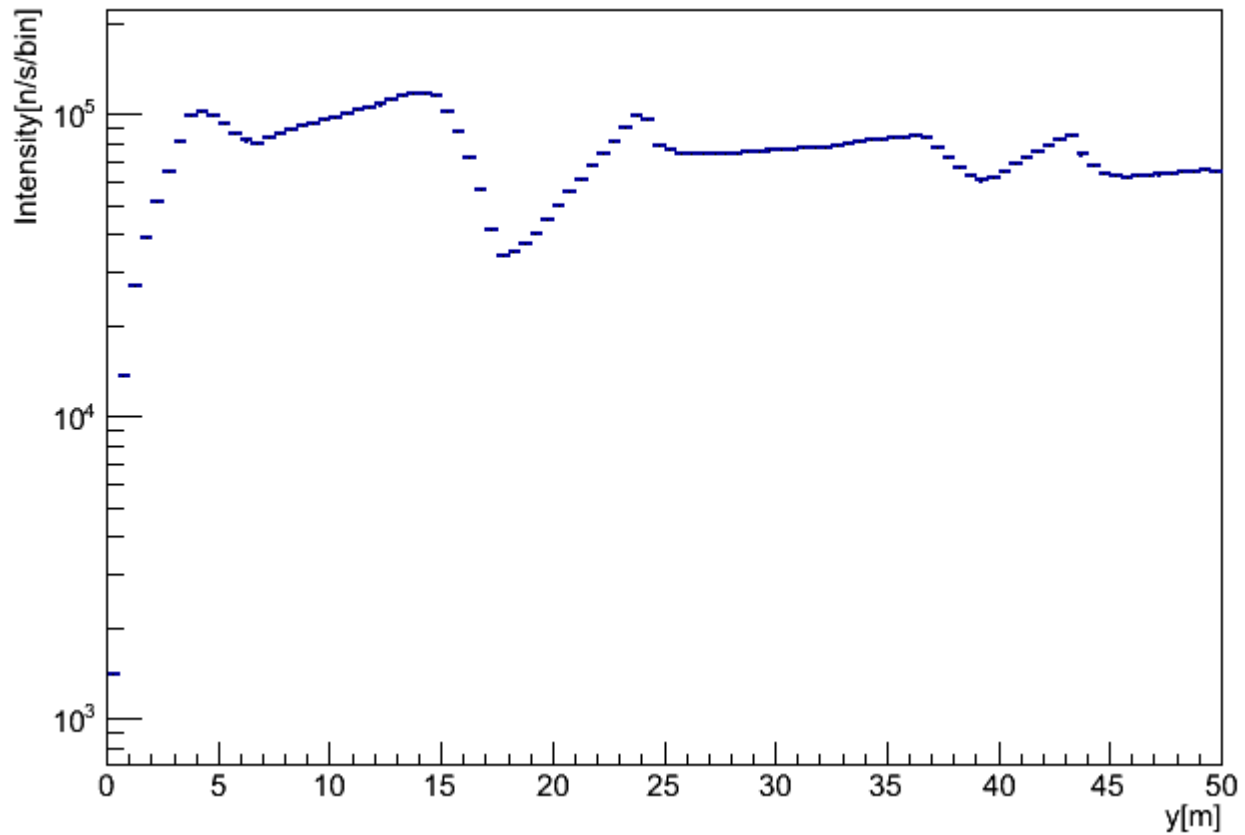
- sp4   0   1

Dose rates: cold-pencil-beam
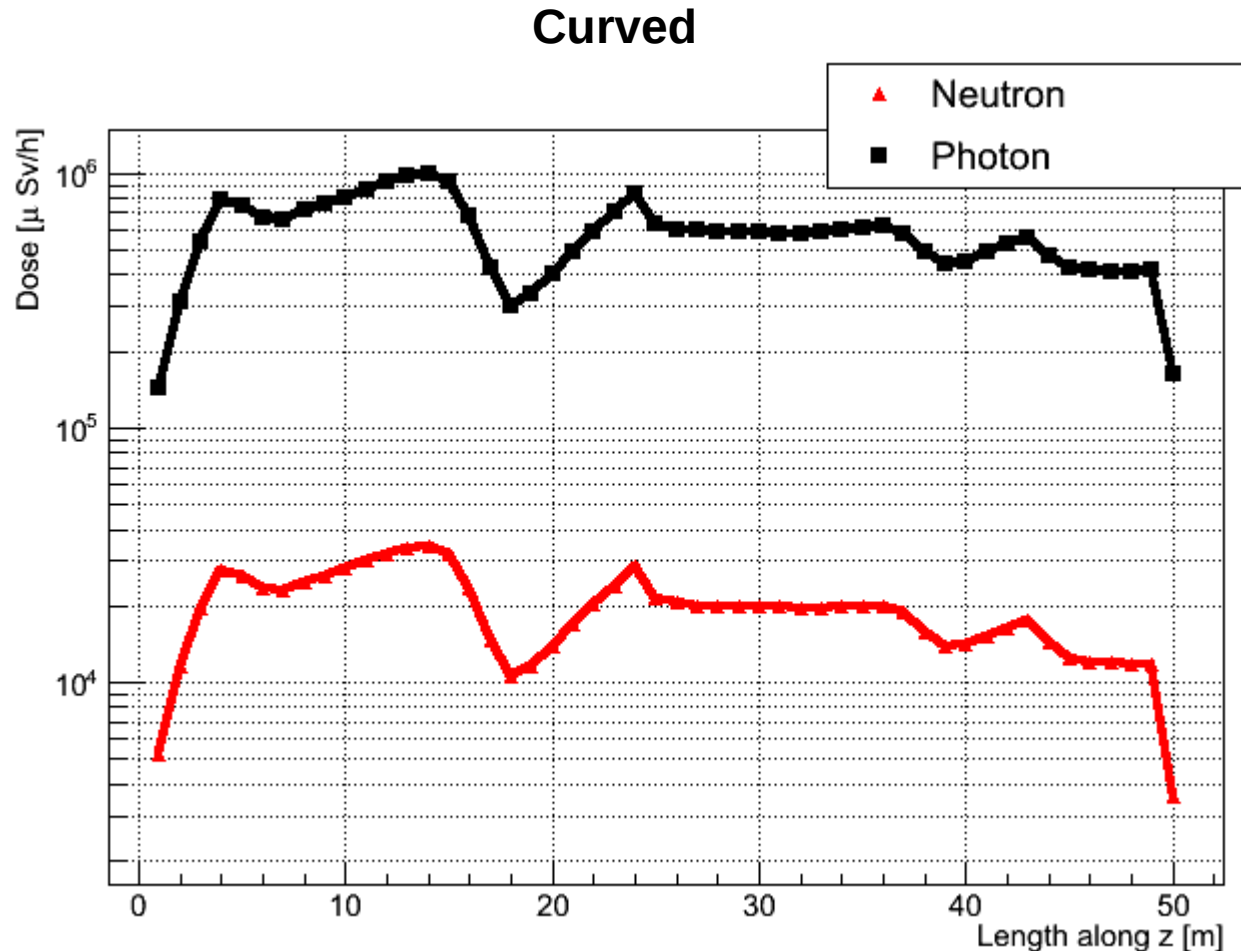
Curved

i.e. perfect match between curve structure

Q: how thus this match McStas-only treatment showing intensity loss?

McStas-only treatment: intensity loss
(pencil-cold-beam)

Better validation example

- Using same geometry, but limit MCNPX source to [0.5-1]Å

- λ range chosen such that

  - ~half the neutrons should be absorbed.

  - Avoid high energy, high intensity peak (showers!), not drawn by exp and $1/r^2$ fall of, of energetic gammas.