Developing an interface between MCNPX and McStas for simulation of neutron moderators

Erik Nonbøl
Esben Klinkby
Peter Willendrup

DTU Nutech
Center for Nuclear Technologies

Motivation

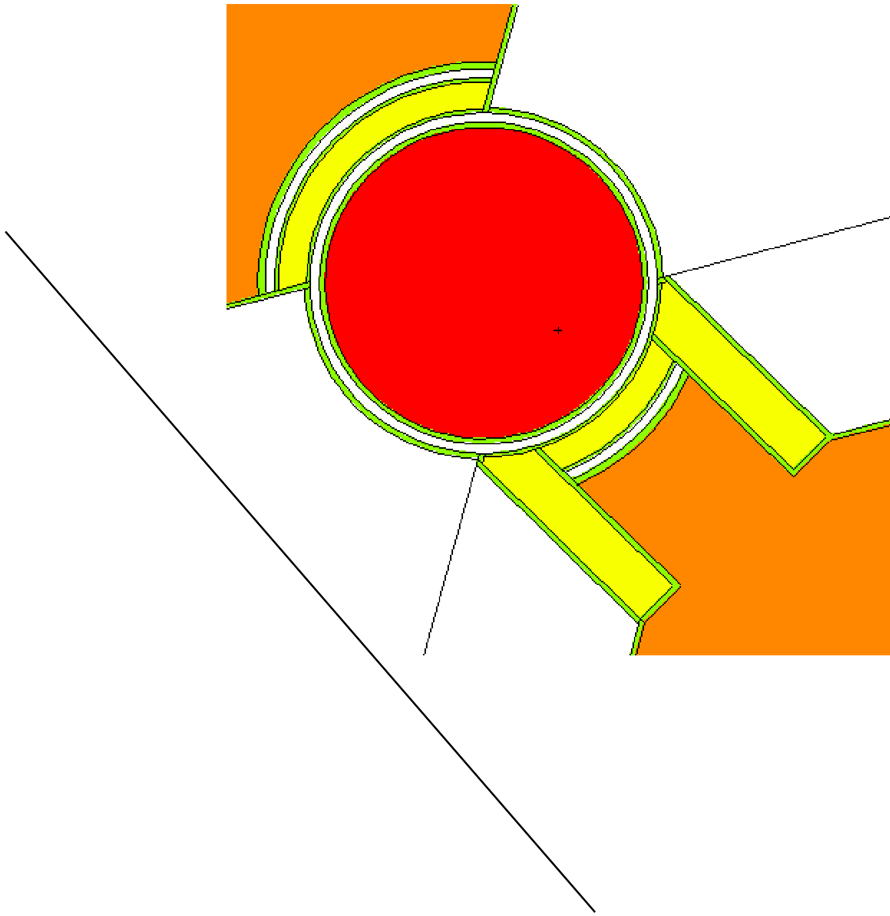
- Traditionally two decoupled Monte Carlo codes covers different needs in Neutron Scattering simulations:
  - MCNP/X used for TMS calculations
  - Neutron ray tracing code, e.g. McStas used for instrument design + data analysis

- Even more precise simulations may be possible by combining the best of the two worlds: The detailed description of incoherent scattering from MCNP/X with the coherent scattering of McStas.

- **Prospects**: usage of direct MCNP/X McStas coupling:
  - Optimization of complex moderator design
  - Shielding along neutron guide
  - Crosstalk between neutron guides
- Test case: ESS TMS and guide systems.

Outline

- **Explored interfaces:**
  - Tally fit
  - Ptrac
  - SSW/SSR
  - Compile
  - Supermirror
- **Validation**
  - First results
- **Summarizing experiences**
  - Cross comparisons



Tally fitting (present default approach)

1. Neutron spectrum calculated with MCNP/X at the moderator surface
2. Spectrum is approximated by simple functions which serves as input to McStas.

Con's

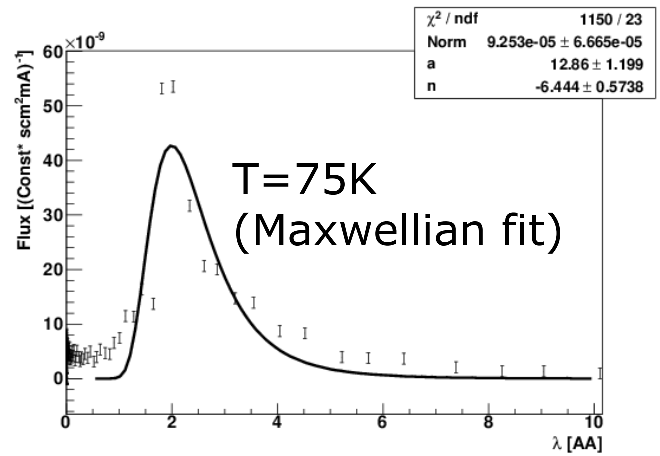
- Correlations (e.g. E, pos, angles) unaccounted for ← Discussed later
- Write out at 1 surface only
- No re-entry (format is write-only)

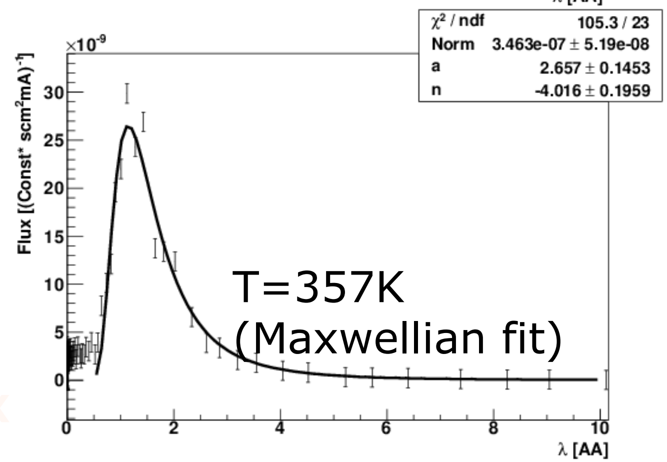
Pro's

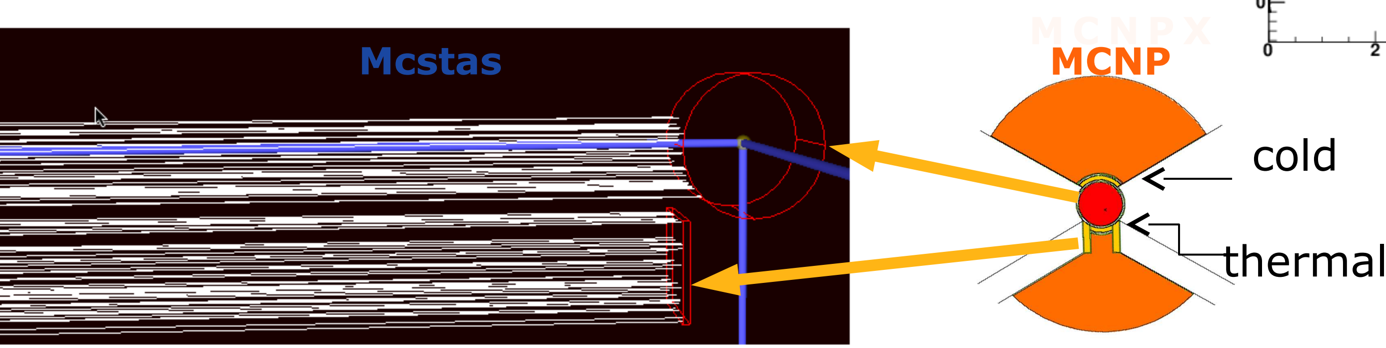
- Fast - MCNP calculation done once-and-for-all
- Avoids licensing issues

Tally fitting (update)

- Based on the latest MCNPX ESS target station (bi-spectral) geometry from ESS-Bilbao (obsolete by now, but nevermind that), we have developed a McStas component mimicking both geometry and spectra.
- The component is named: *ESS_Moderator_revised*







Ptrac

- MCNP/X can output an ascii file containing individual neutron states: pos, angles, energy, time & weight
- The McStas component: *MCNP_Virtual_Input* converts the neutron state into McStas readable and works as a source

Con's

- ascii file enormous: ~0.2kB/evt
- Write out at 1 surface only
- No re-entry (format is write-only)
- Cannot run MPI

Pro's

- Correlations conserved (e.g. E,pos)
- Fast

Ptrac format

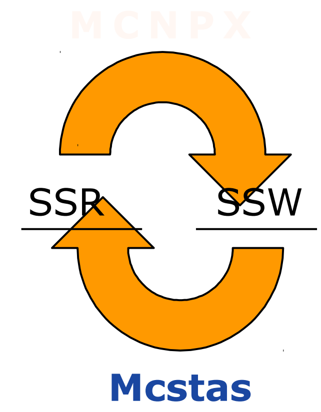
```
…………………………………
 3000        2       10       179
100         2         0
   0.00000E+00  0.28640E+00
0.43531E+00 -0.10000E+01
0.00000E+00  0.00000E+00
0.10000E+00  0.10000E+01
0.33356E-02
       3000        3       110       179
   10         2         0
  -0.20000E+00  0.28640E+00
0.43531E+00 -0.10000E+01
0.00000E+00  0.00000E+00
0.10000E+00  0.10000E+01
0.40028E-02
       3000        4       120       179
  100         2         0
  -0.40000E+00  0.28640E+00
0.43531E+00 -0.10000E+01
0.00000E+00  0.00000E+00
0.10000E+00  0.10000E+01
0.46699E-02
       3000        5       130       179
………………………………..
```

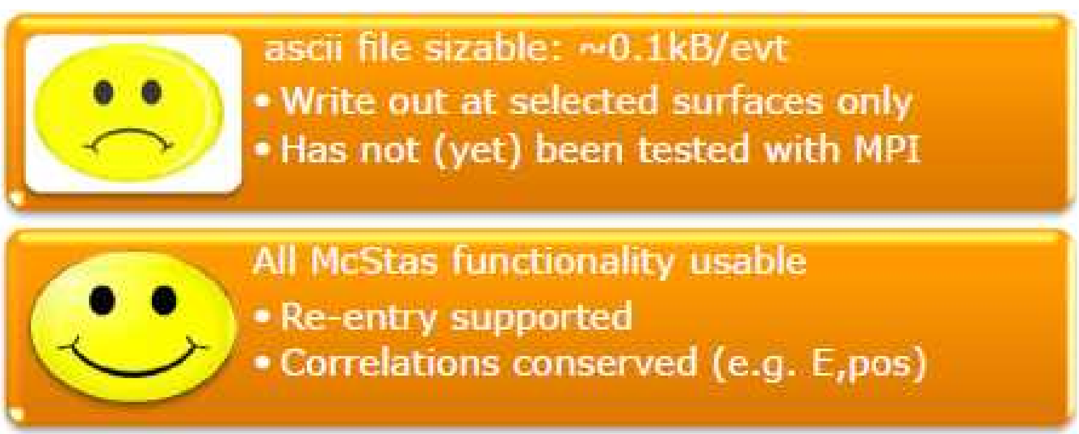
SSW/SSR

- **S**ource **S**urface **R**ead/**W**rite in MCNPX starts/stops simulations at a given (set of) surface(s)
- The neutron state written to binary file.
- New McStas components:

Virtual_mcnp_ss_input & *Virtual_mcnp_ss_output* reads MCNPX output and writes MCNPX input

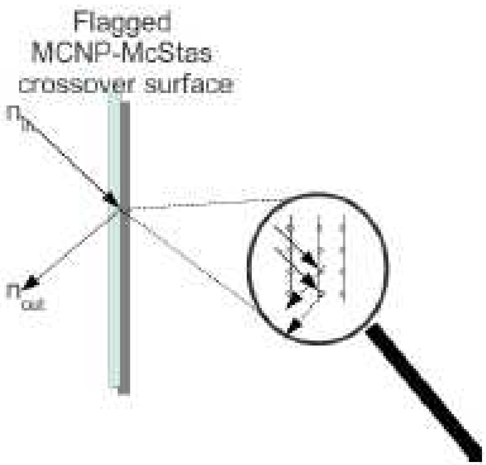
- Neutron propagation started in MCNPX, continued in McStas and finalizing in MCNP





Combined compilation

mcstas "-surface"



Method

- McStas surface flag introduced in MCNPX
- Neutron crossing McStas surface causes initiation of McStas simulation, based on neutron state.
- Updated neutron state returned to MCNPX

Technically difficult to make general

- Licensing issue
- Slow: MCNPX called for each neutron

Potentially very flexible (but not yet fully developed)

- All McStas functionality usable
- Re-entry supported
- Correlations conserved (e.g. E,pos)

<u>in MCNPX input file:</u>

.....

-110 PX -0.2

-120 PX -0.4

Supermirror

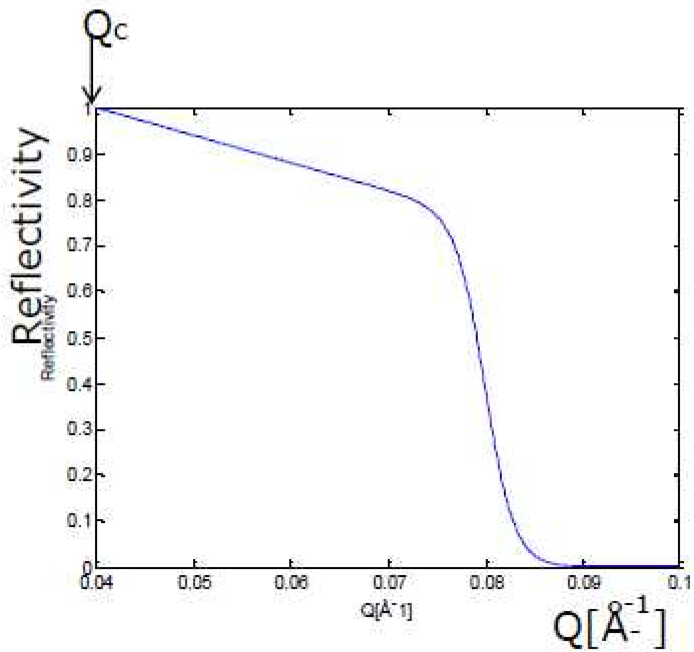
- Existing implementation, introducing McStas inspired supermirrors as a surface card in MCNPX (Gallmeier et all, Nuc.Tech. 168(3))

- Reflectivity $R=R_0$ if $Q<Q_c$
- $R=R_0/2\{1-\tanh[(Q-mQ_c)/W]\}\{1-a(Q-Q_c)\}$ if $Q>Q_c$

- Ported to MCNPX 2.7, but not yet validated

Doesn't scale: workload per functionality significant. Only McStas mirrors ported
- Licensing issue

Re-entry supported
- Correlations conserved (e.g. E,pos)
- Avoids intermediate files and multiple codes

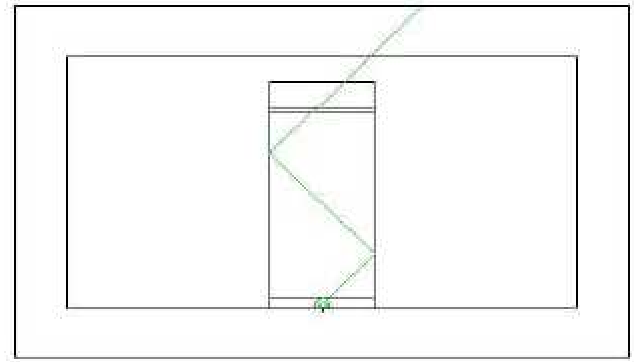


Validation of interfaces

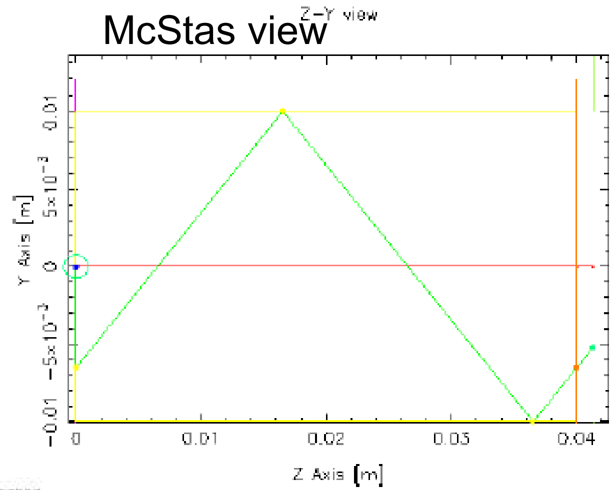
Approach: simplified geometry

- 1E6 neutrons of 20meV emitted at 45degree from a plane
- 100% reflectivity
- Void volumes

MCNP view



McStas view



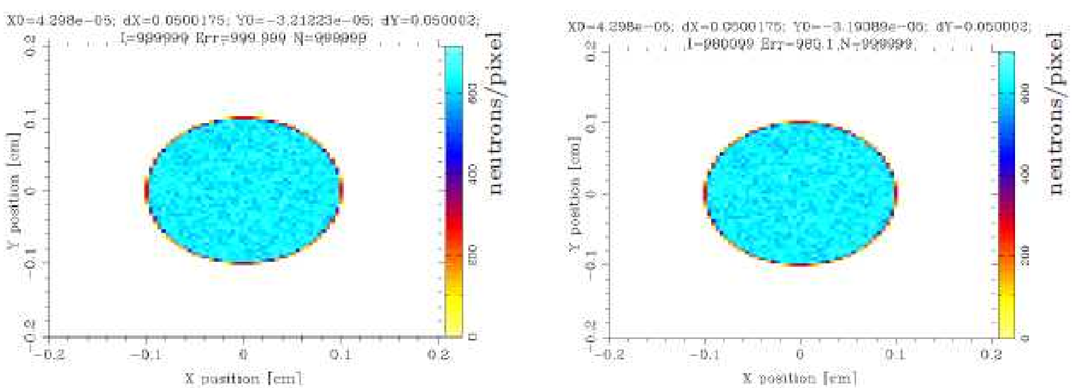
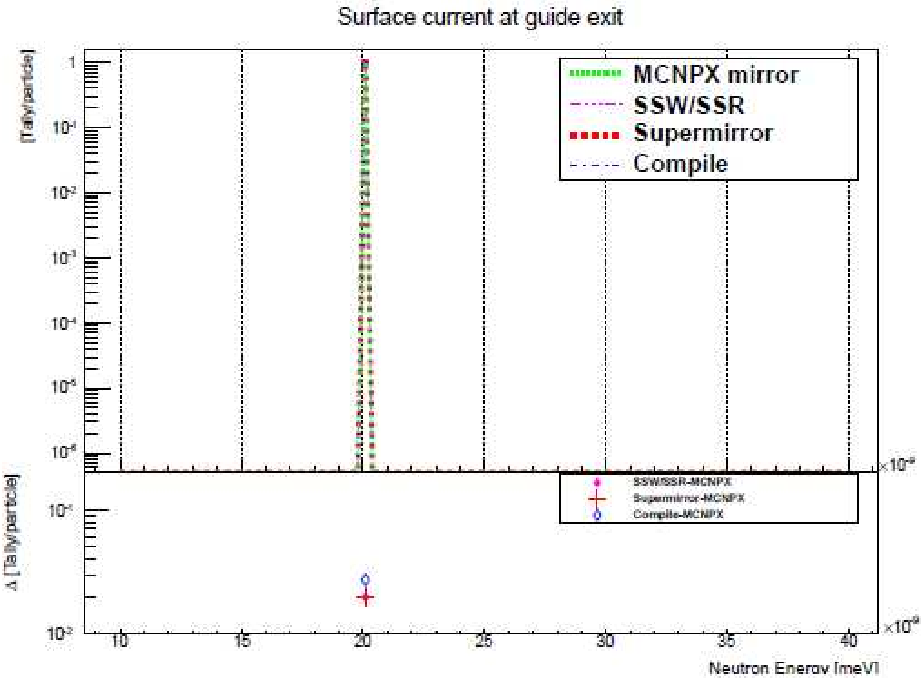
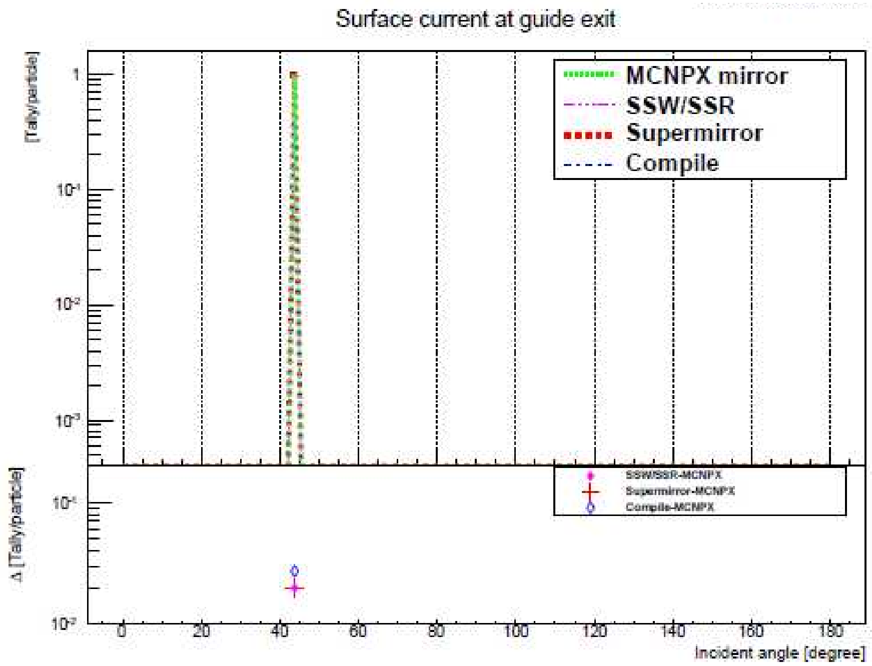


Figure 5: McStas PSD at the guide entrance (left) and exit (right).

Nucl. Instr. & Meth. A 700, 106-110

Validation of interfaces





→ Interfaces work. But are they useful?

Cross comparison

	Re-entry neutrons	Speed	Single neutron trace	Require License	Comments
Tally	No	Fast*	No	Yes/No	Should try to determine validity at least once
Ptrac	No	Fast*	Yes	Yes	Somewhat outdated by SSW/SSR
SSW/SSR	Yes	Fast*	Yes	Yes	Promising…
Compile	Yes	Slow	Yes	Yes	Generalizes poorly (auto gen c-code hacks)
Supermirror	Yes	Slow	yes	yes	Generalizes poorly (but who cares?)

*) The computational heavy MCNP/X calculation can be performed once-and-for-all

Interfaces work. But are they useful?

Intermezzo:

coffee & questions

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]

1) DTU Nutech, Box 49, 4000 Roskilde, Denmark
2) DTU Physics, Fysikvej 307-312, 2800 Kgs. Lyngby, Denmark
3) ESS design update programme, Denmark

Motivation

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

⇓

- 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

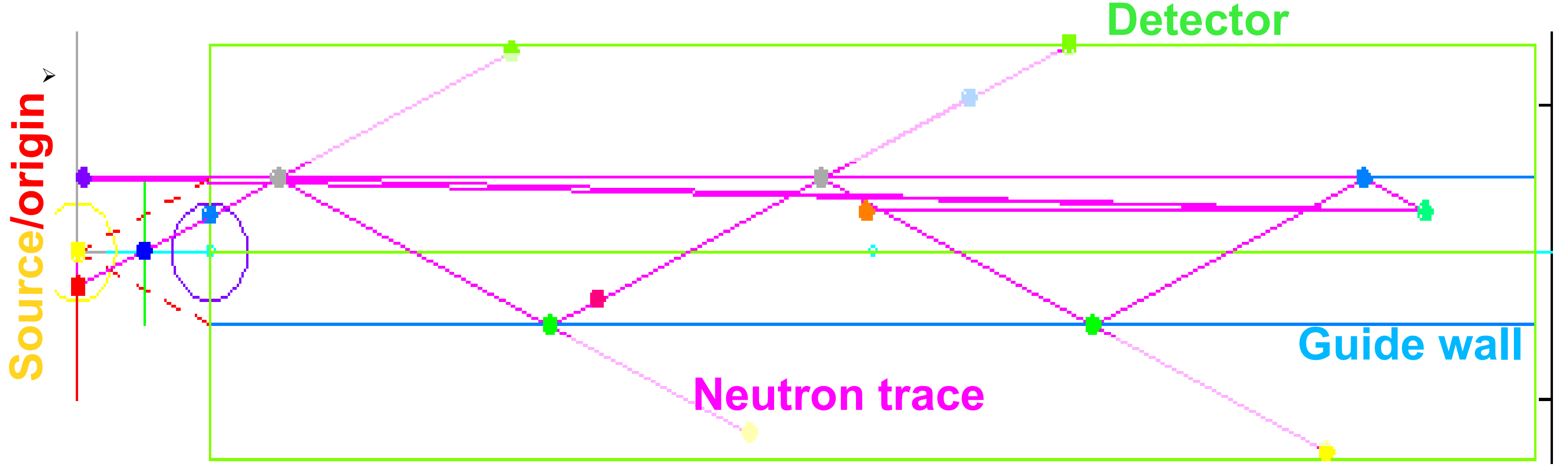
- At each scattering, for any McStas component (eg. a guide), the incoming and outgoing neutron state can be temporally stored & analyzed

At each scattering:

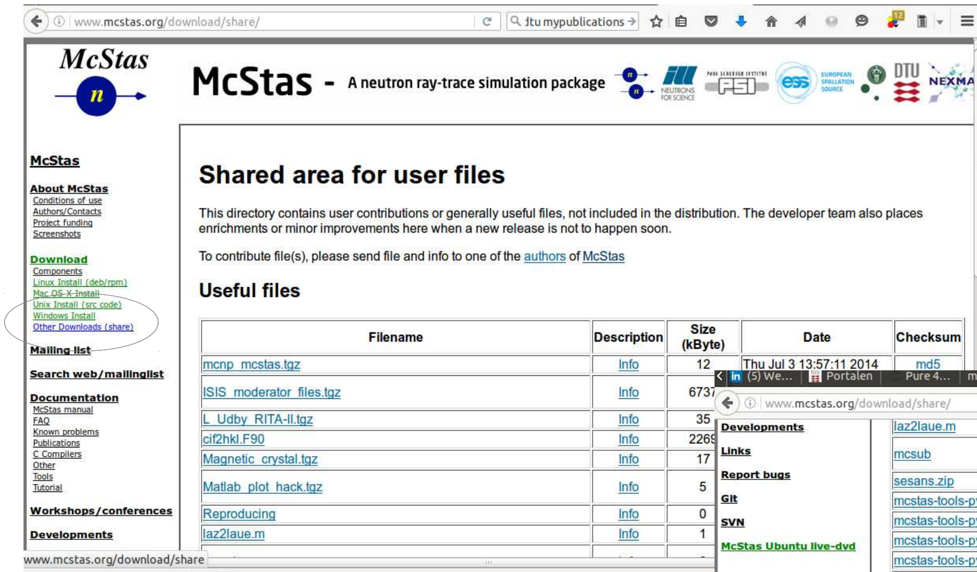
Incomming state: $n_{in} = (\vec{\mathbf{x}}, \vec{\mathbf{v}}_{in}, t, w_{in})$

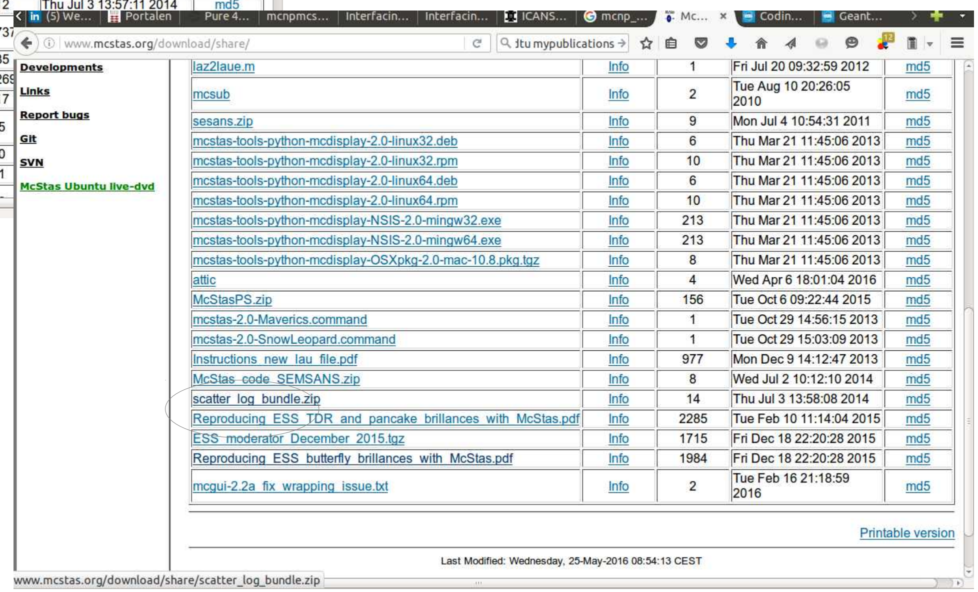
Transmitted state: $n_{trans} = (\vec{\mathbf{x}}, \vec{\mathbf{v}}_{in}, t, w_{trans})$

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



Hands on



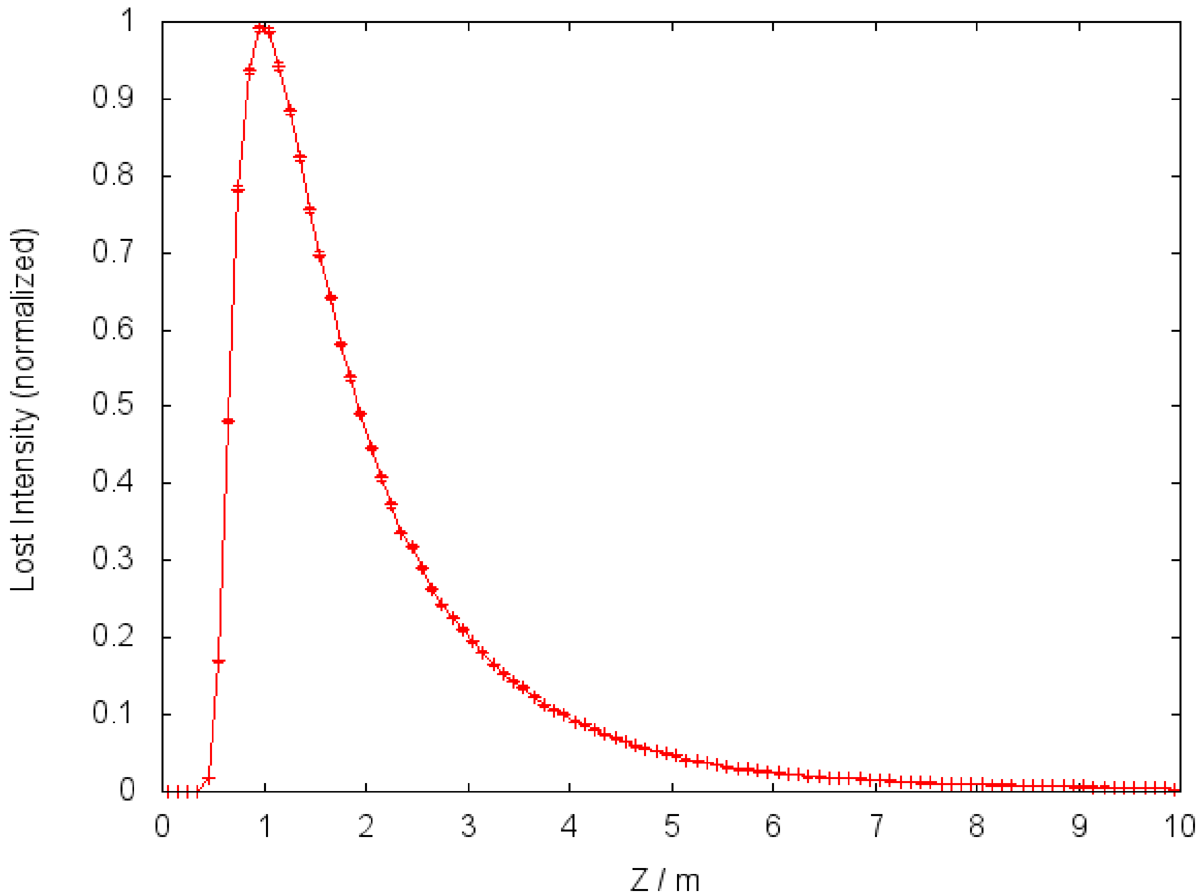


Download:
scatterlog_bundle.zip from *share*

Unpack & look at & run the instr.
Test_Scatter_log_Mon_nD.instr

Example1: Lost intensity

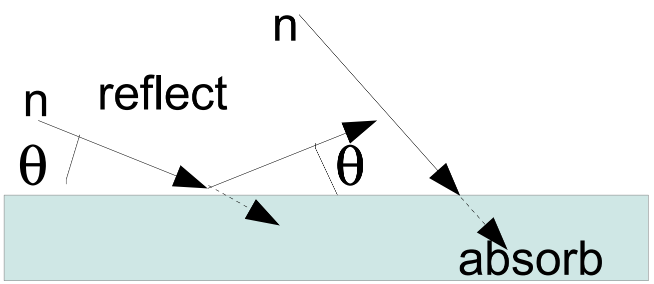
- Lost intensity along guide: w_{trans} *versus* $\vec{\mathbf{x}} \cdot \hat{\mathbf{z}}$

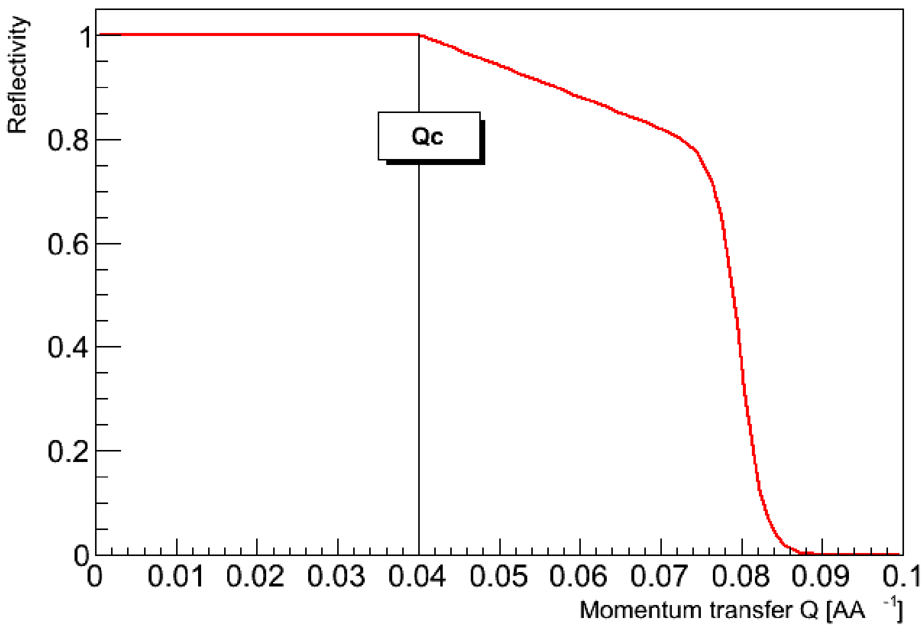


- Depends strongly on specific guide design, and incoming neutrons (i.e. source) spectrum and divergence.

Example2: Reflectivity

- Neutrons are reflected if the energy/incident angle is low enough





$\cos 2\theta = (\vec{v}_{in} \cdot \vec{v}_{out}) / |\mathbf{v}_{in}|^2$

$k = |\mathbf{v}_{in}| \cdot m_n / \hbar$

$\Downarrow$

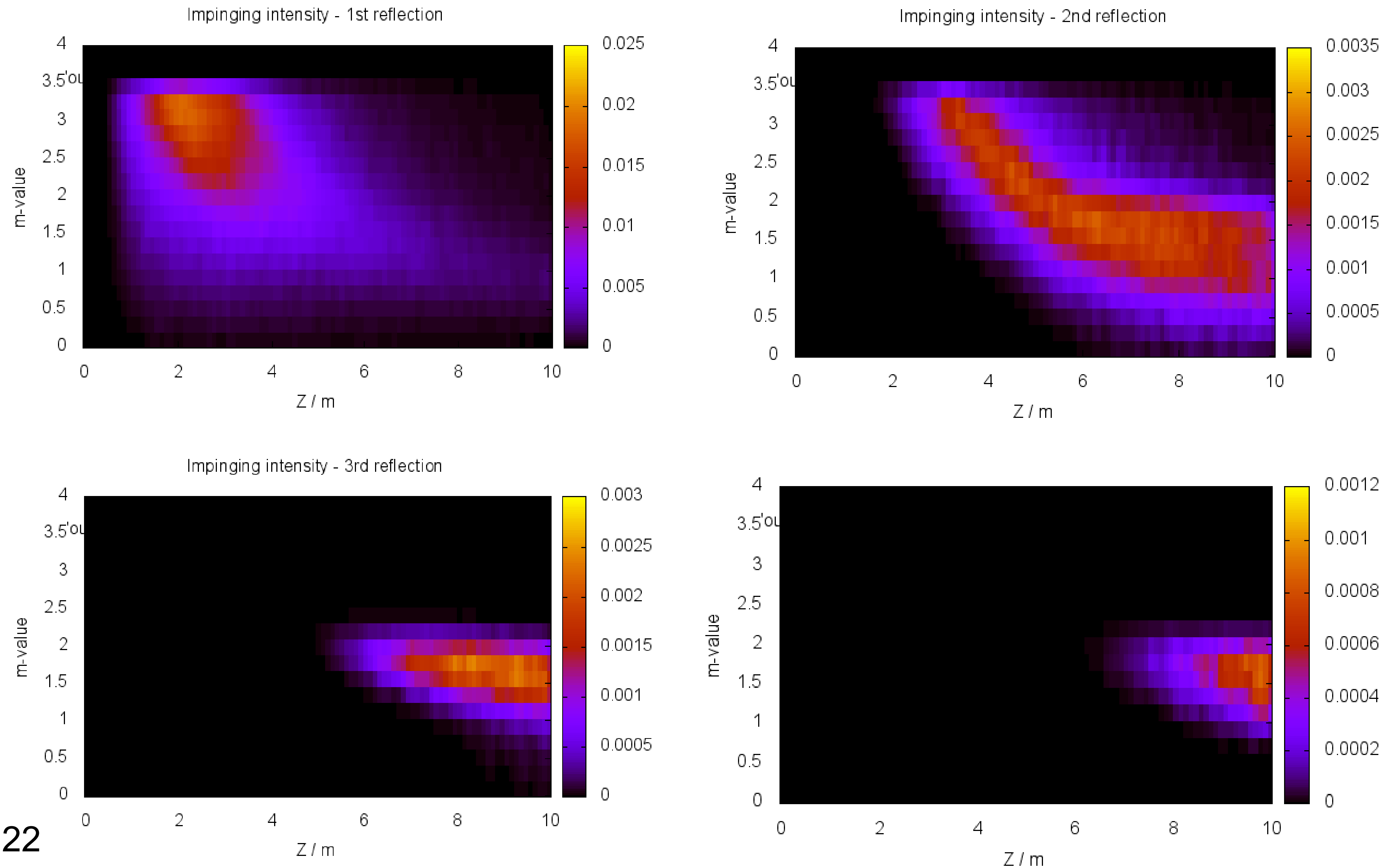
$m_{min} = 2 \cdot k \cdot \sin(\theta) / 0.0219$

- 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

Hands on #2

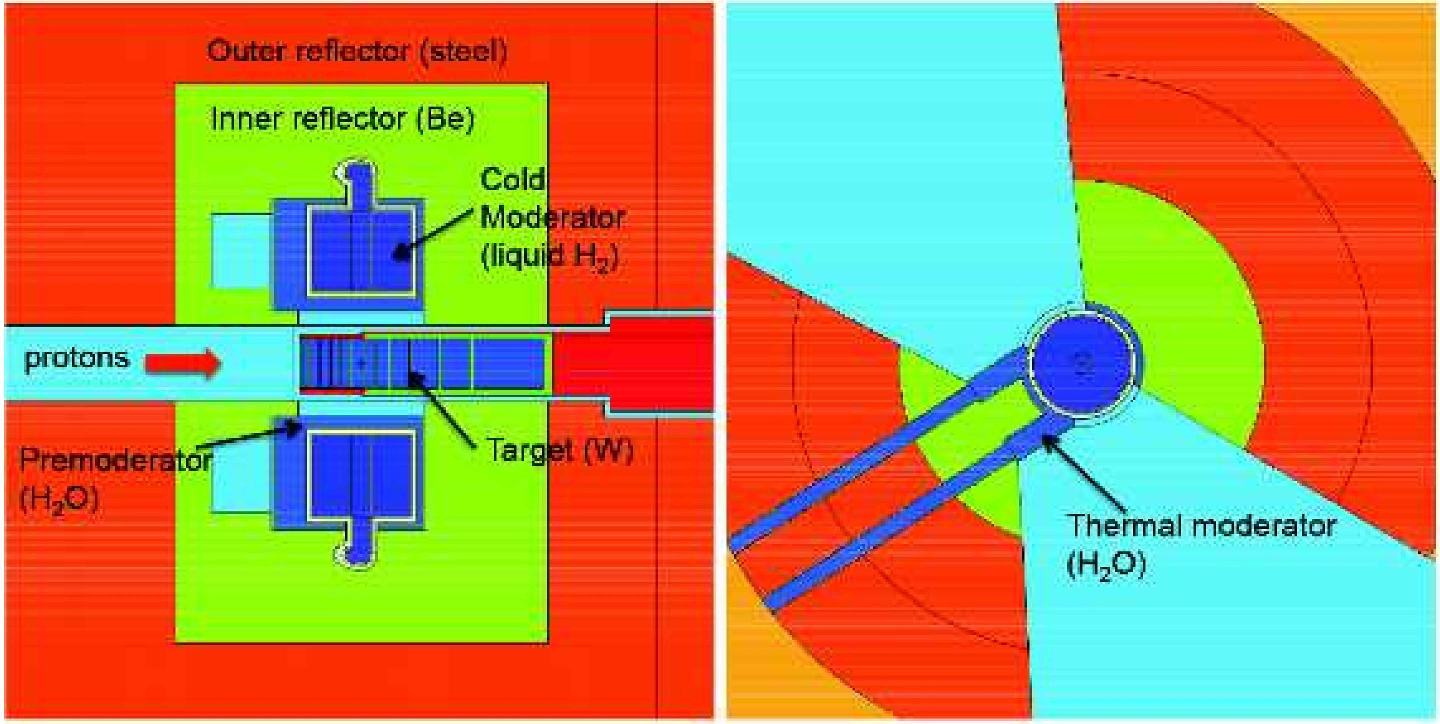
look at & run the instr.
Test_Scatter_log_Mon_nD_mvalue.instr

Example2: Reflectivity



Example3: Background along guide

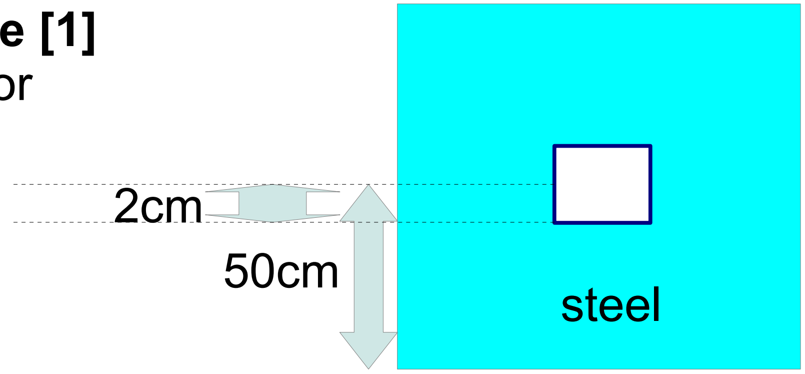
→**I. Neutrons generated with MCNPX**
II. Handed to McStas through SSW interface [1]
III. Unreflected neutrons returned to MCNPX for dose-rate calculation

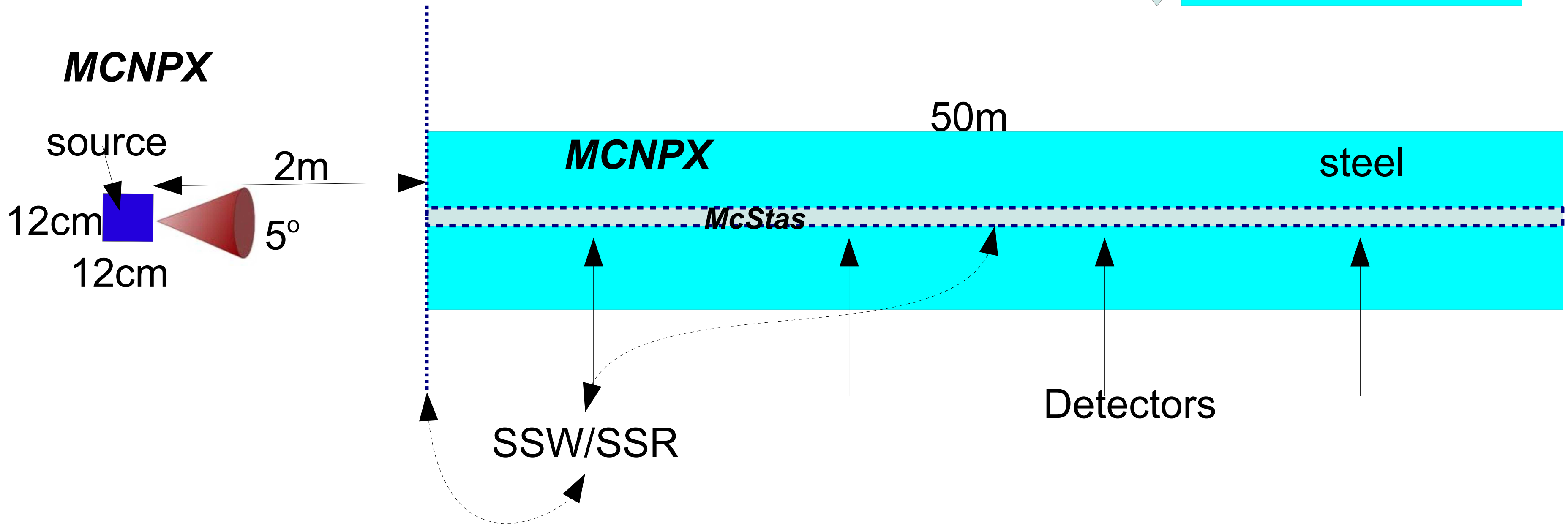


Example3: Background along guide

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

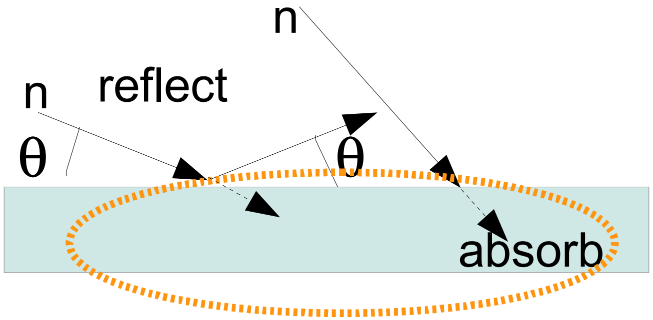




[1] E. Klinkby et al. 'Interfacing MCNPX and McStas for simulation of neutron transport.' Nucl. Instr & Meth A , 700: p106, 2013.

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**



<u>At each scattering:</u>

⇒

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

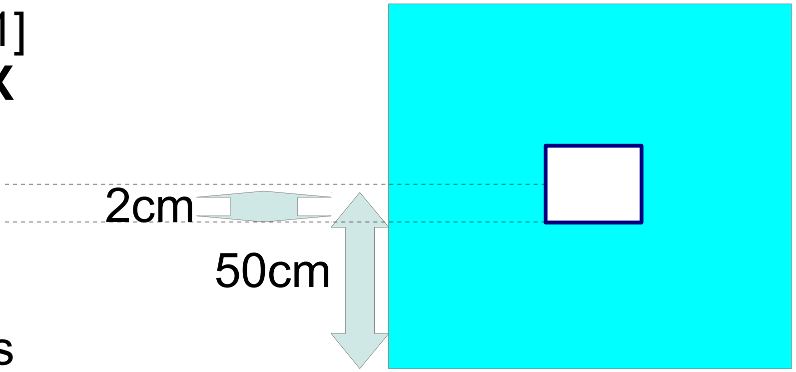
Transmitted state: $n_{trans} = (\vec{\mathbf{x}}, \vec{\mathbf{v}}_{in}, t, w_{trans})$

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

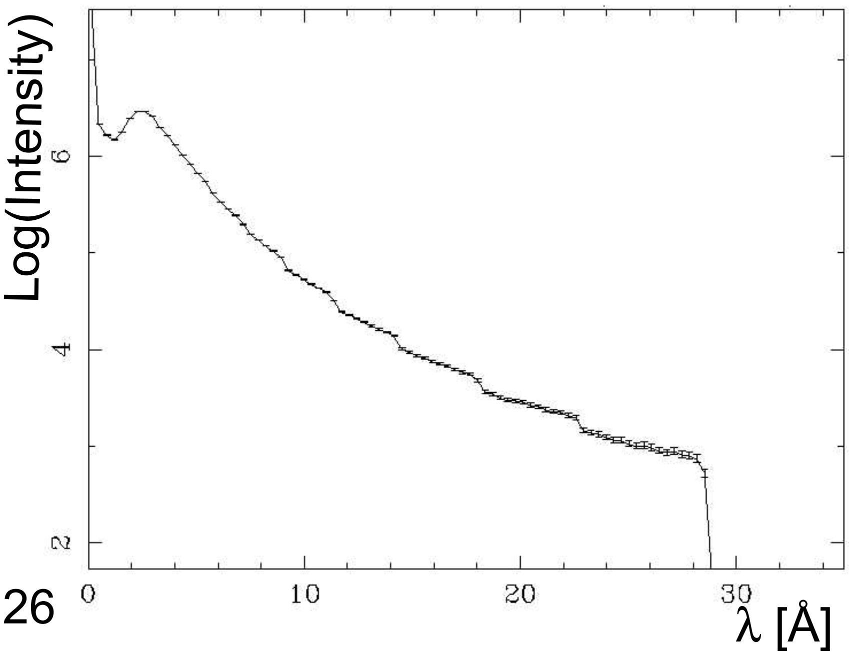
Example3: Background along guide

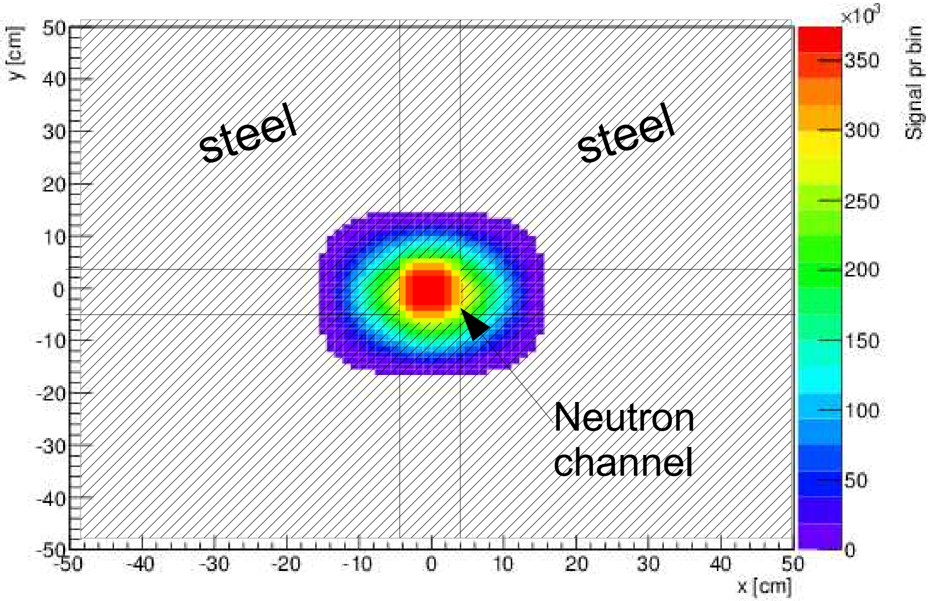
I. Neutrons generated with MCNPX
II. Handed to McStas through SSW interface [1]
→**III. Unreflected neutrons returned to MCNPX for dose-rate calculation**

Guide cross-section



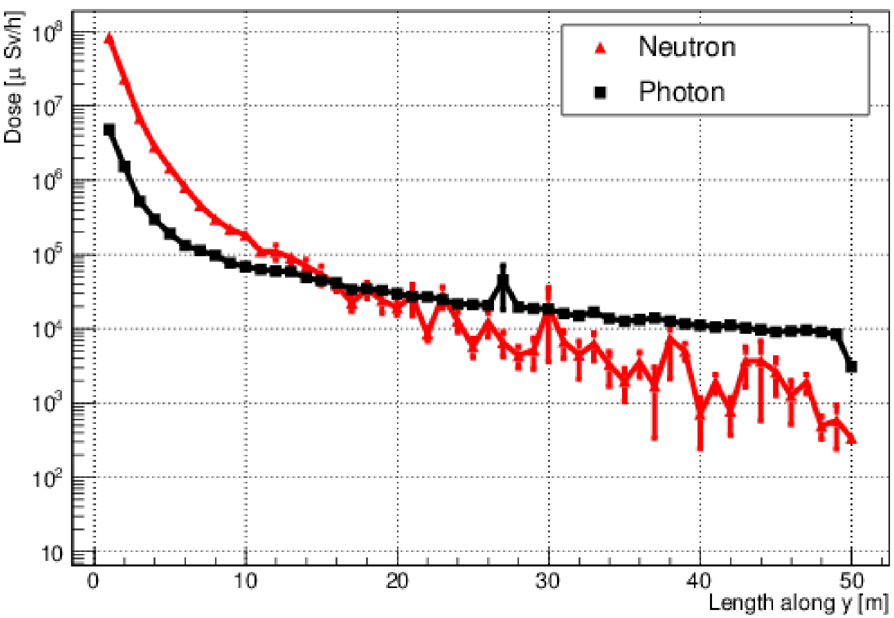
Guide end overilluminated by energetic neutrons



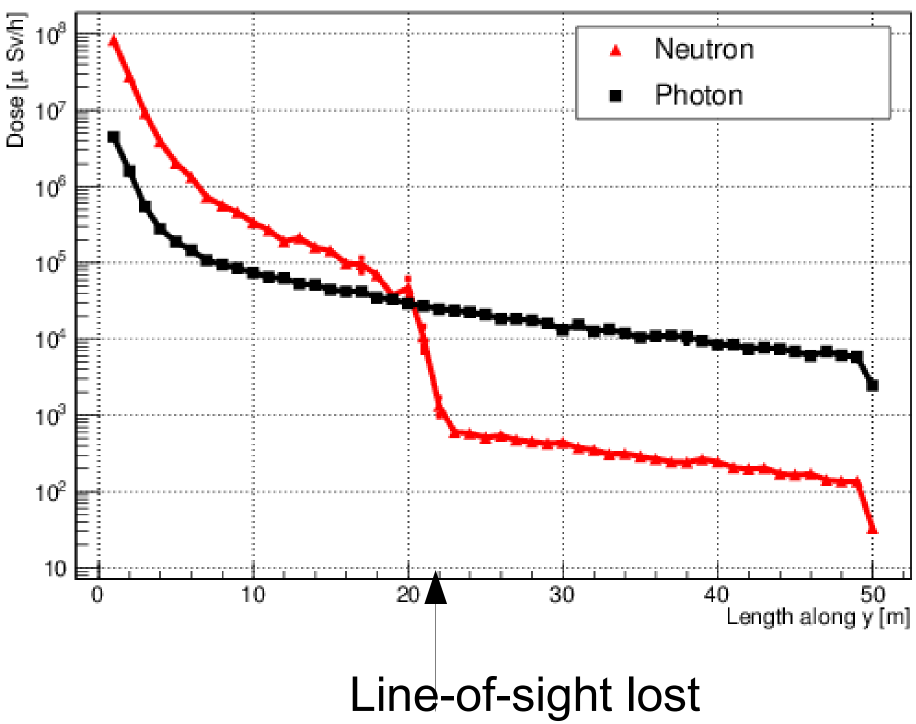


Example3: Background along guide

Straight guide



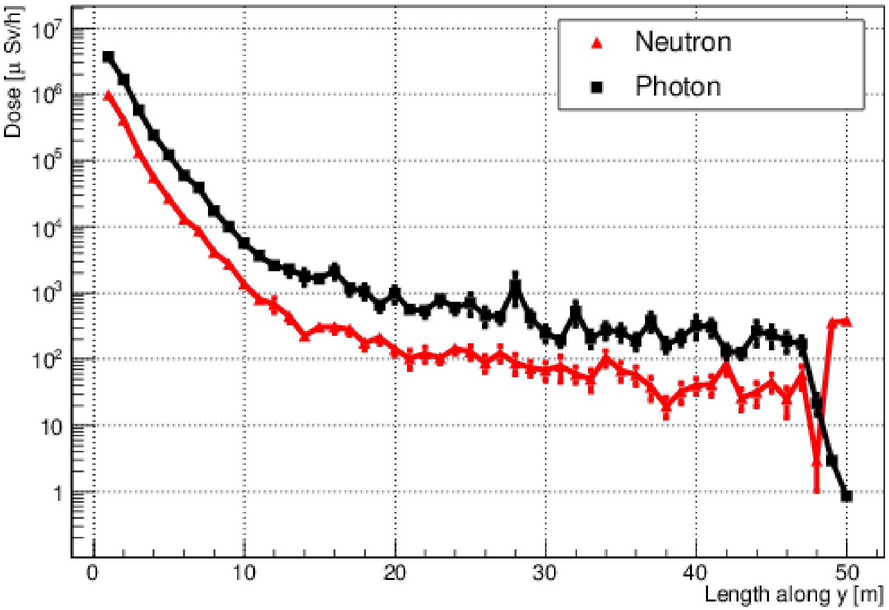
Curved guide ($r_{curvature}$=1500m)



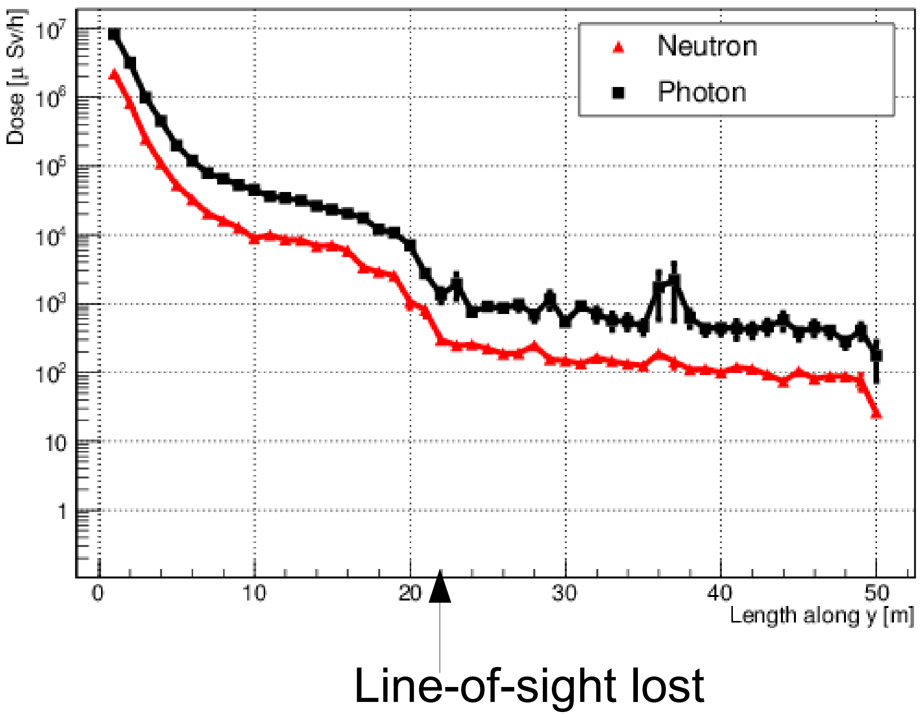
- Dose-rates, measured 5cm in the steel converted from flux according to official Swedish radiation protection procedures

Example3: Background along guide

Straight guide



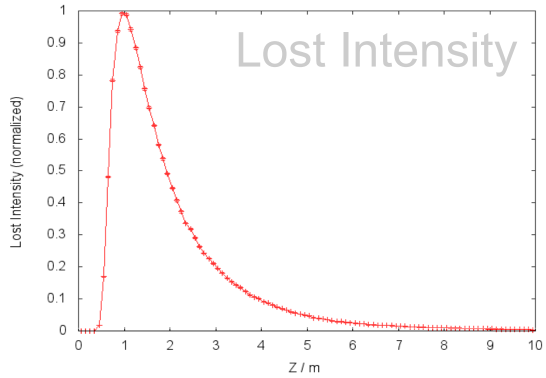
Curved guide ($r_{curvature}$=1500m)

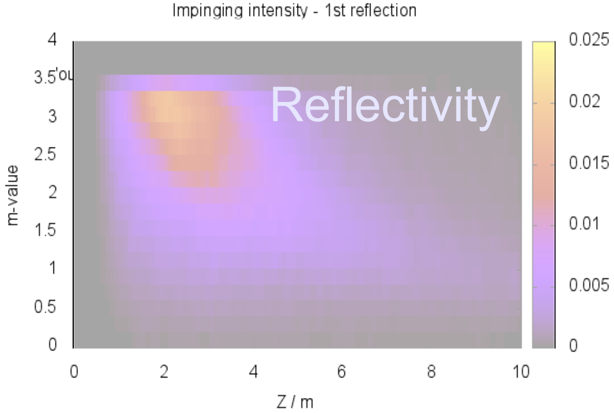


- Restricting to $\lambda \in \{0.5\ \text{Å} - 1.0\ \text{Å}\}$
- Photon dose-rate follows neutron dose-rate √

Conclusions

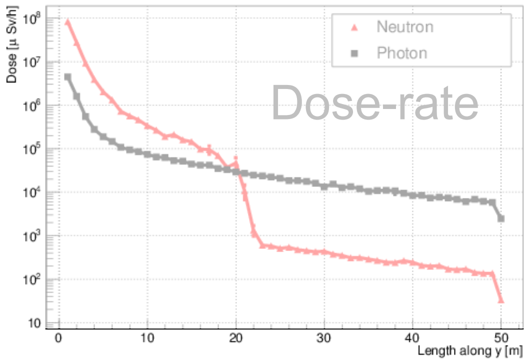
- Logging mechanism: ScatterLogger potentially useful for guide design:
  - Monitor ***intensity loss***
  - Optimizing use of high/low ***reflectivity*** mirrors
  - Calculate ***dose-rates*** along guide
- Works in instrument designers accustomed work-flow (***McStas***)





Prospects

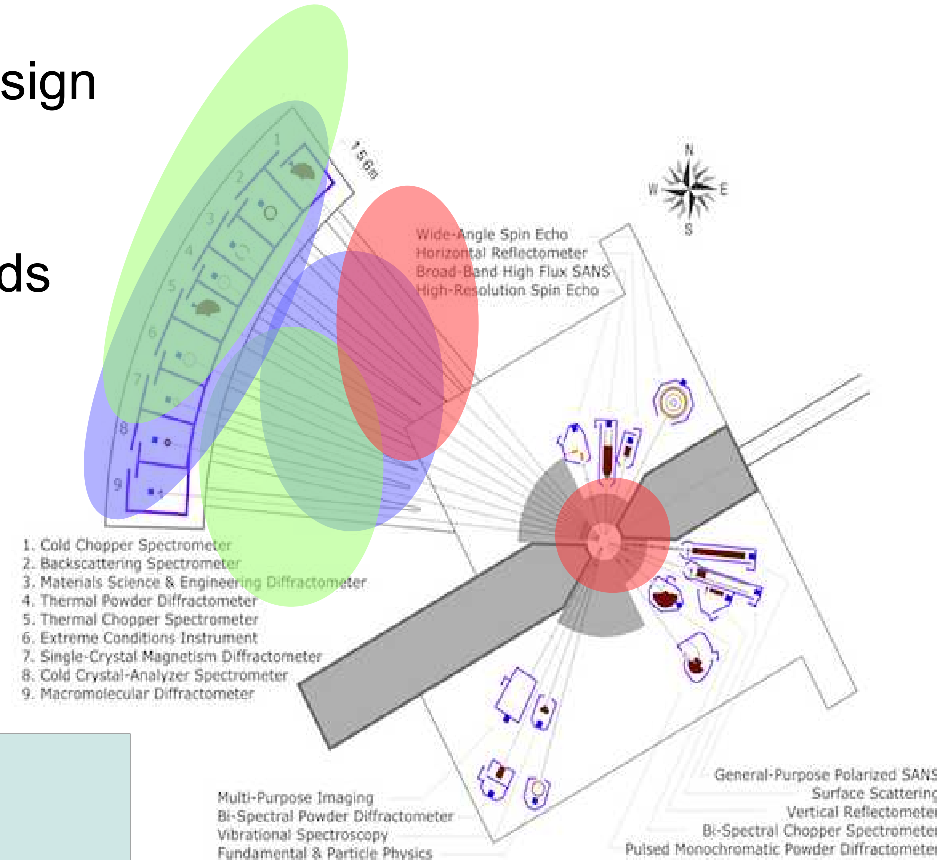
- Could be used to evaluate gamma and (energetic) neutron background at the sample position / instrument / detectors.
- Must be utilized on an instrument to instrument basis



Interfaces work. And they can be useful.
But are they sufficient and convenient to use?

Interfacing revisited

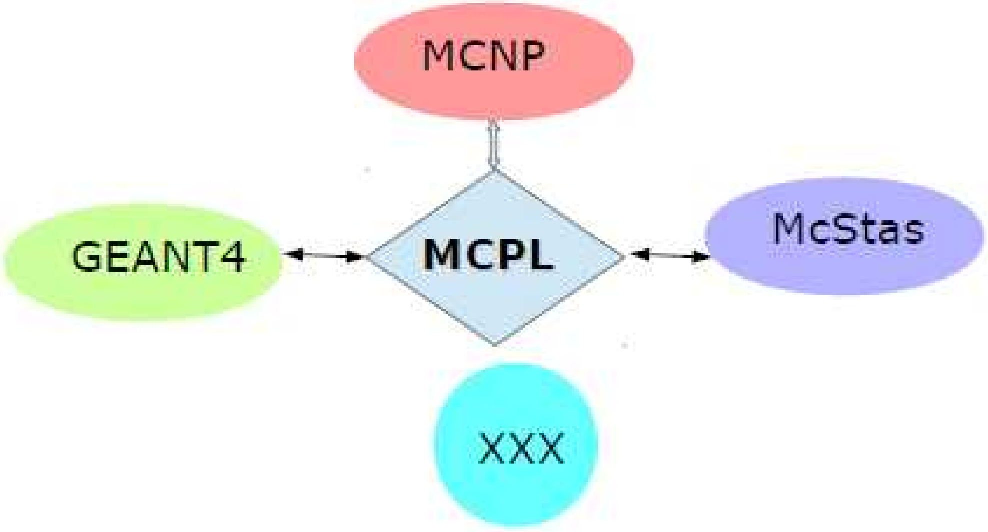
- *MCNP:* target, moderator, reflector design
- *McStas* (+*guide_bot*) for instrument design
- *GEANT4* for shielding and backgrounds
- Vitess & NADS & Particle swarms: shielding & optics
  - design documentation instruments
- *MCNP*: safety, dose-rates (future use of FLUKA or MARS)
- *GEANT4:* detector design



⇒ Interfacing **is** important.
⇒ *MCNP* ⇔ *McStas* is insufficient
⇒ Present implementation proved to be restricted to expert use only
⇒ A **common file format** would facilitate 'cradle to grave' simulations, without intermediate loss of information
(e.g. through fitting)

MCPL: MCNP-Geant4-McStas-XXX coupling

- Rather than using converters to read the MCNP SSW format directly from within McStas, an intermediate event format is defined, containing the full event information
- *position, momentum, weight, time, particle ID, custom flags*



- The data format is binary for performance reasons, test-functionality allow the user to view the file content

MCPL: MCNP-Geant4-McStas-XXX coupling

```
klinkby@laptop:~/projects/dg_dgcode$ ess_mcpl_tool ./packages/Validation/UnitTests/MCPLTests/data/reffile_skip123.mcpl
Opened MCPL file reffile_skip123.mcpl:

  Basic info
    Format             : MCPL-2
    No. of particles   : 123
    Header storage     : 59 bytes
    Data storage       : 8364 bytes

  Custom meta data
    Source             : "MyMCApp"
    Number of comments : 0
    Number of blobs    : 0

  Particle data format
    User flags         : no
    Polarisation info  : no
    Fixed part. type   : no
    FP precision       : double
    Endianness         : little
    Storage            : 68 bytes/particle

index     pdgcode   ekin[MeV]       x[cm]       y[cm]       z[cm]          ux          uy          uz    time[ms]      weight
    0        2112       1.234           0           0           0           0           1           0           0           1
    1        2112           0           0           0        0.01        0.01           0    -0.99995           0           1
    2        2112       1.234           0           0        0.02        0.02           0      0.9998           0           1
    3        2112           0           0           0        0.03        0.03    -0.99955           0           0           1
    4        2112       1.234           0           0        0.04        0.04           0      0.9992           0           1
    5        2112           0           0           0        0.05        0.05           0    -0.99875           0           1
    6        2112       1.234           0           0        0.06        0.06      0.9982           0           0           1
    7        2112           0           0           0        0.07        0.07           0    -0.99755           0           1
    8        2112       1.234           0           0        0.08        0.08           0     0.99679           0           1
    9        2112           0           0           0        0.09        0.09    -0.99594           0           0           1
```

- Developed within the software framework of the ESS Detector Group – Thomas Kittelmann is the main developer
- Core software (written in c) is stable but some examples + documentation are missing. Expected "release" by autumn 2016. Until then, contact me and/or Thomas for instructions
- First use-case: Geant4 – MCNPX comparisons: arXiv:1509.03036

Hands on session. My hands only, but still....

MCPL: status & prospects

- MCPL format is defined and stable
- Core software is maintained within the ESS detector group software framework: *dgcode*
  https://ess-ics.atlassian.net/wiki/pages/viewpage.action?spaceKey=DG&title=CodingFramework

- **Converters to/from:**
  - GEANT4 exist
    https://ess-ics.atlassian.net/wiki/display/DG/Geant4SimulationFramework
  - McStas has been written, ~1day of validation remains
  - MCNPX has been written, ~1-2days of validation remains (mcnp5,6 in particular)

- ⇒ Final touch prior to distribution. ~2weeks
- For now: use within the *dg* framework

- If you're interested in a standalone distribution / info, please contact Thomas Kittelmann or me.

Backup slides

$\Rightarrow$

Reflectivity curve

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

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
```

Specialized pseudo neutron state function

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

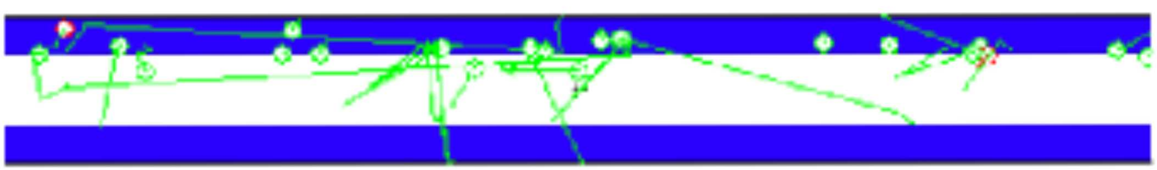
```
/*position comes from "new" state*/
ns_tilde[0]=S1->_x;ns_tilde[1]=S1->_y;ns_tilde[2]=S1->_z;

/*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:

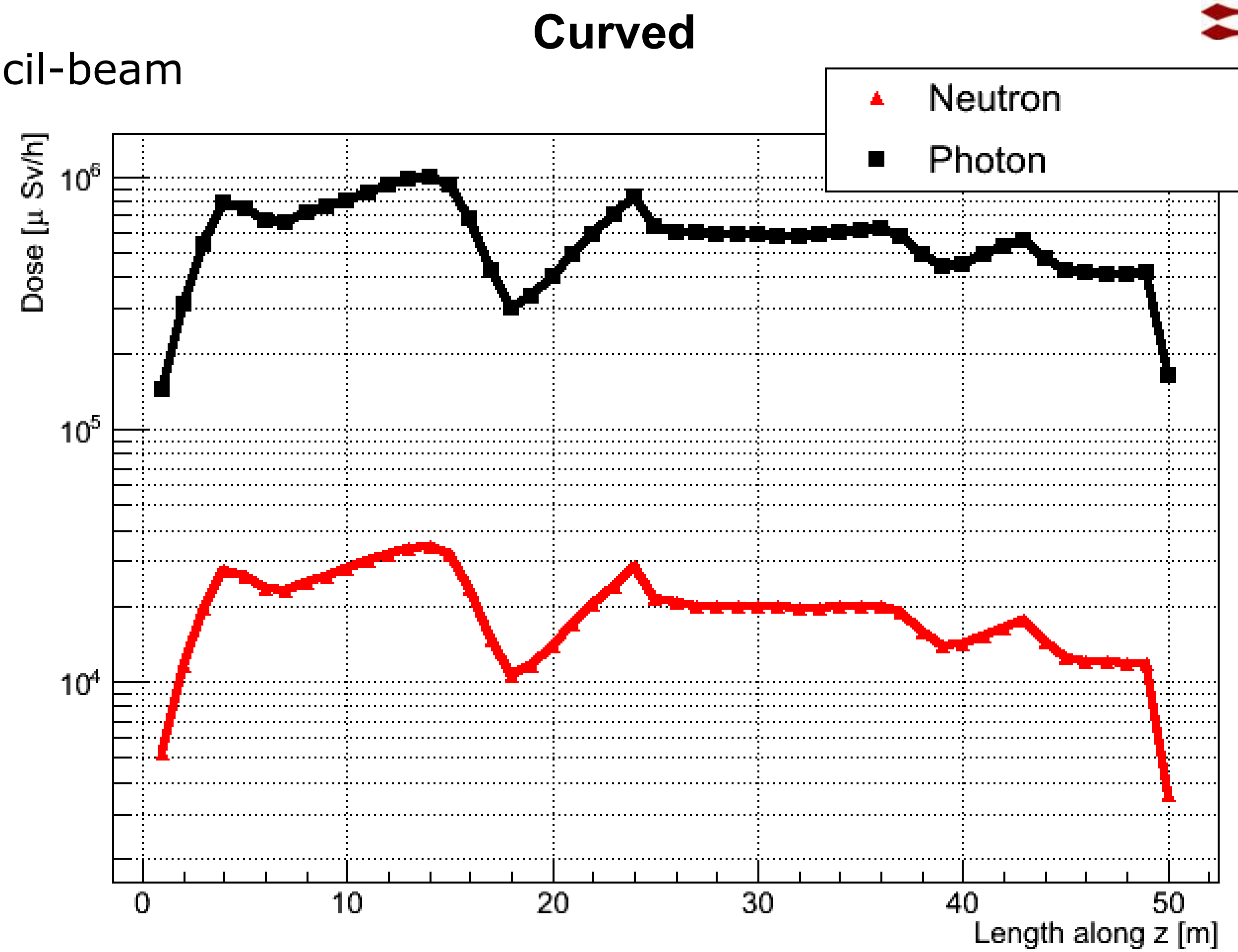


Neutron and gamma trajectories (a few example events from MCNPX → Vised)

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

- ***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
-

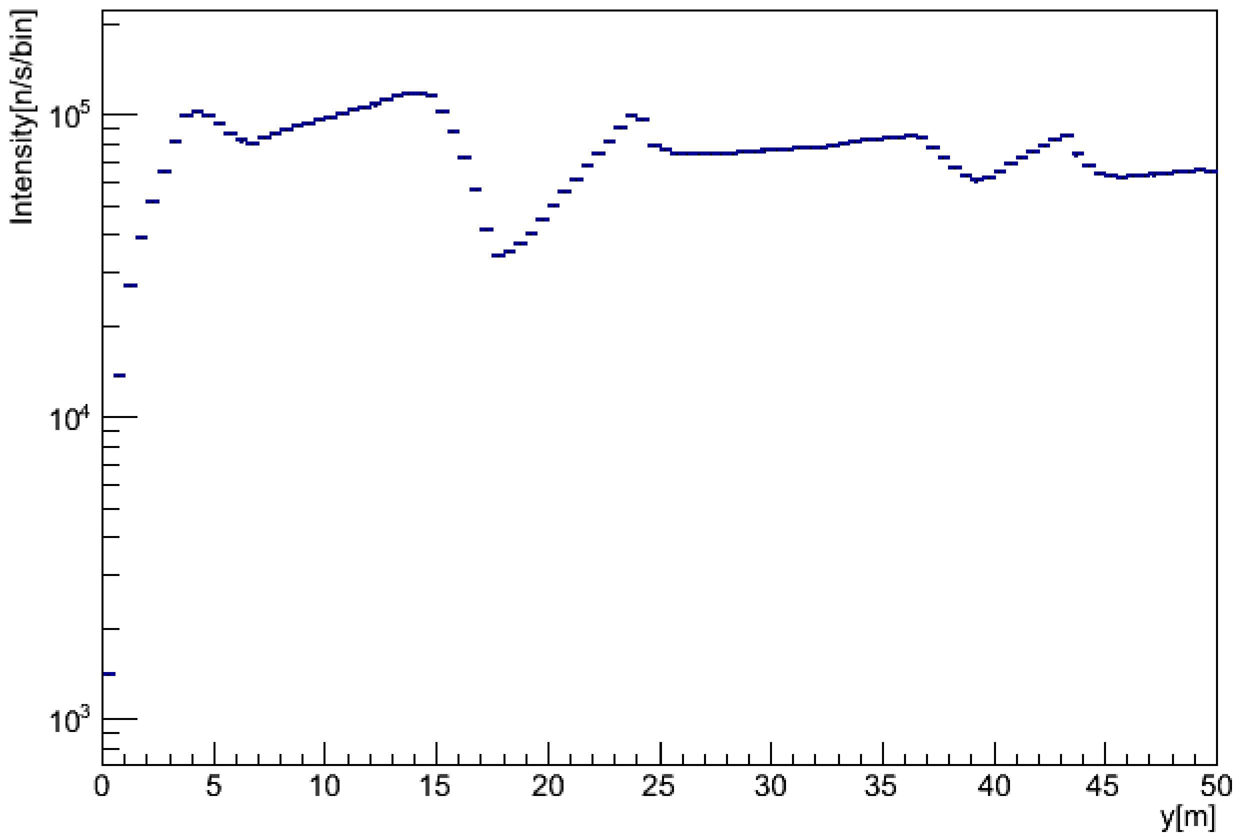
Dose rates: cold-pencil-beam



i.e. perfect match between curve structure

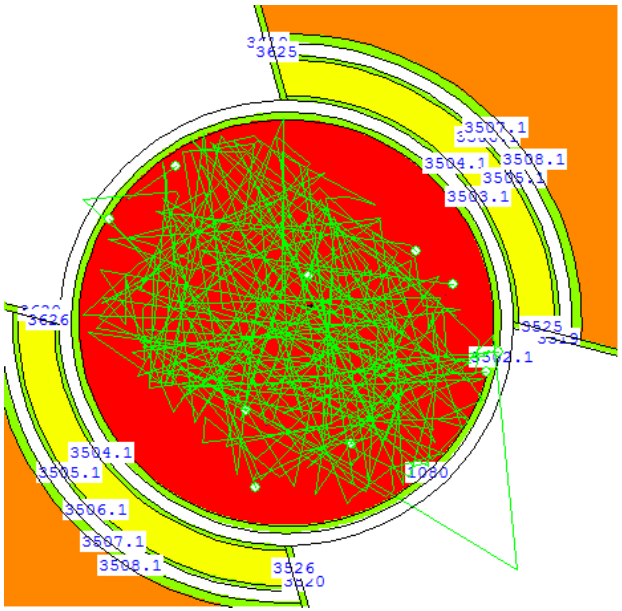
Q: how thus this match McStas-only treatment showing intensity loss?

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

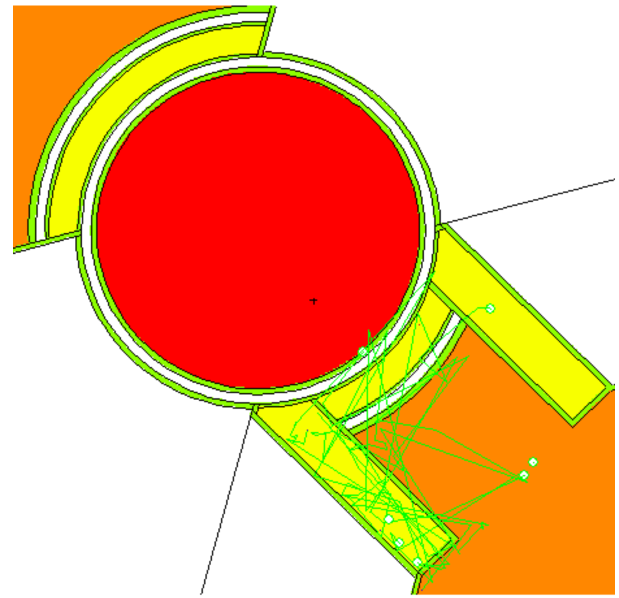


Tally contributions

- Contributions to cold spectrum



- Contributions to thermal spectrum



MCPL - commands

```
emacs -nw input_deck
mcnpx z      ip i=input_deck
mcnpx i=input_deck n=test_ssw_file.

ess_mcnp_ssw2mcpl -h
ess_mcnp_ssw2mcpl -s ~/projects/PID/ssw_to_mcpl_to_ssw/test_ssw_file.w
ess_mcnp_ssw2mcpl -s ~/projects/PID/ssw_to_mcpl_to_ssw/test_ssw_file.w -c
~/projects/PID/ssw_to_mcpl_to_ssw/test_spheres
ess_mcpl_tool output.mcpl.gz


ess_mcpl_tool output.mcpl.gz  -bmcnp_config_file | tee input_deck
diff input_deck  ~/projects/PID/ssw_to_mcpl_to_ssw/test_spheres
ess_mcplextra_browse output.mcpl.gz


----------------

ess_g4mcpltests_mcplwrite
ess_mcnp_mcpl2ssw recordfwd.mcpl.gz  ~/projects/PID/ssw_to_mcpl_to_ssw/reference_ssw_file.w g4_out.w  3
cp ../read_deck .
mcnpx i=read_deck rssa=g4_out.w
ess_mcnp_ssw2mcpl wssa
ess_mcplextra_browse output.mcpl.gz
```

Input mcnpx deck

```
Test spheres v1
3   0 -3                    IMP:n,p=1  $ SPACE
4   0 -4 #3                   IMP:n,p=1  $ SPACE
5   0 -5 #4 #3                  IMP:n,p=1  $ SPACE
100 0  5                    IMP:n,p =0 $OUTSIDE WORLD

c
3 SO 100
4 SO 1000
5 SO 1500

C MATERIALS
C
C ////////////////////////
C
C
MODE n p # k / | e u v w z *
NPS 1E6
c sdef x=-5 y=0 z=0 vec 1 0 0 dir=1  par=n erg=1$ → 99242 035 000 004
sdef  erg=1$ → 99242 035 000 004
c
PRDMP  2J  -1
phys:n 300
phys:h 300
phys:p 300 3j 1
PHYS:# 1000 3j 1
ssw 3
```

Hands on

look at & run the instr.
Test_Scatter_log_Mon_nD_mvalue.instr