



EUROPEAN
SPALLATION
SOURCE



McStas and how-to generate event data for Mantid and scipp

TORBEN

Spring 2022 – McStas workshop @ DMSC

McStas and McStasScript

Histogram data / binned data / ascii data



←

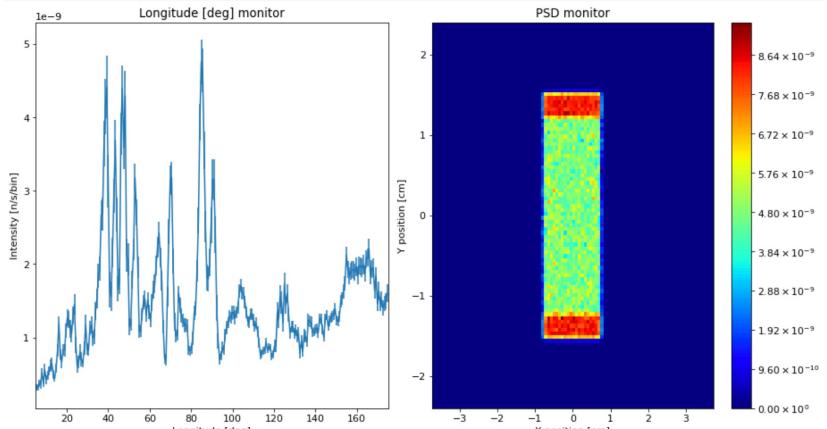
Contents

McStasScript documentation

Search the docs ...

```
read.  
plotter.make_sub_plot(data)
```

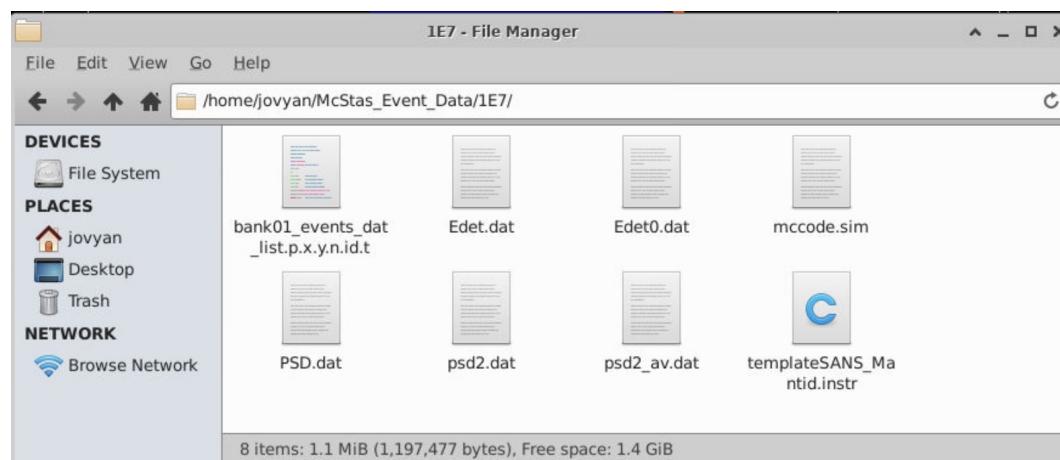
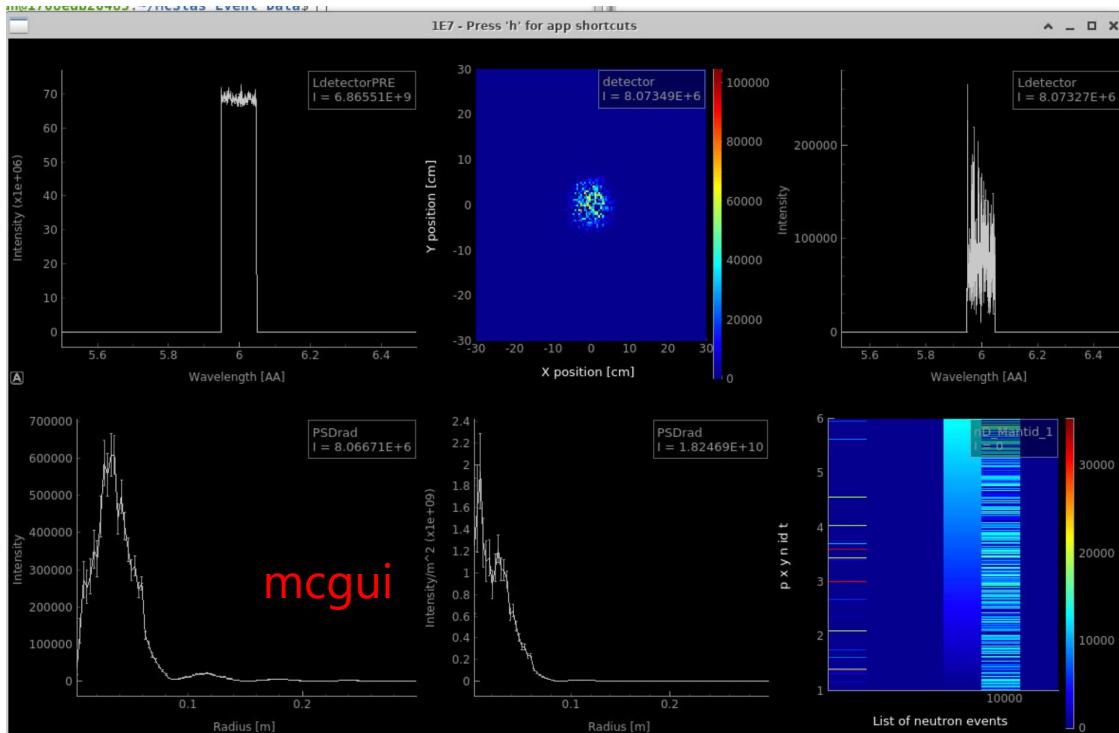
Plotting data with name banana
Plotting data with name PSD



Customizing plots

As described in the data section of the documentation, the `McStasData` objects contain preferences on how they should be plotted. The same keywords can however be given to the plotters to override these preferences. When plotting a list of `McStasData` objects, a list can be given for each keyword.

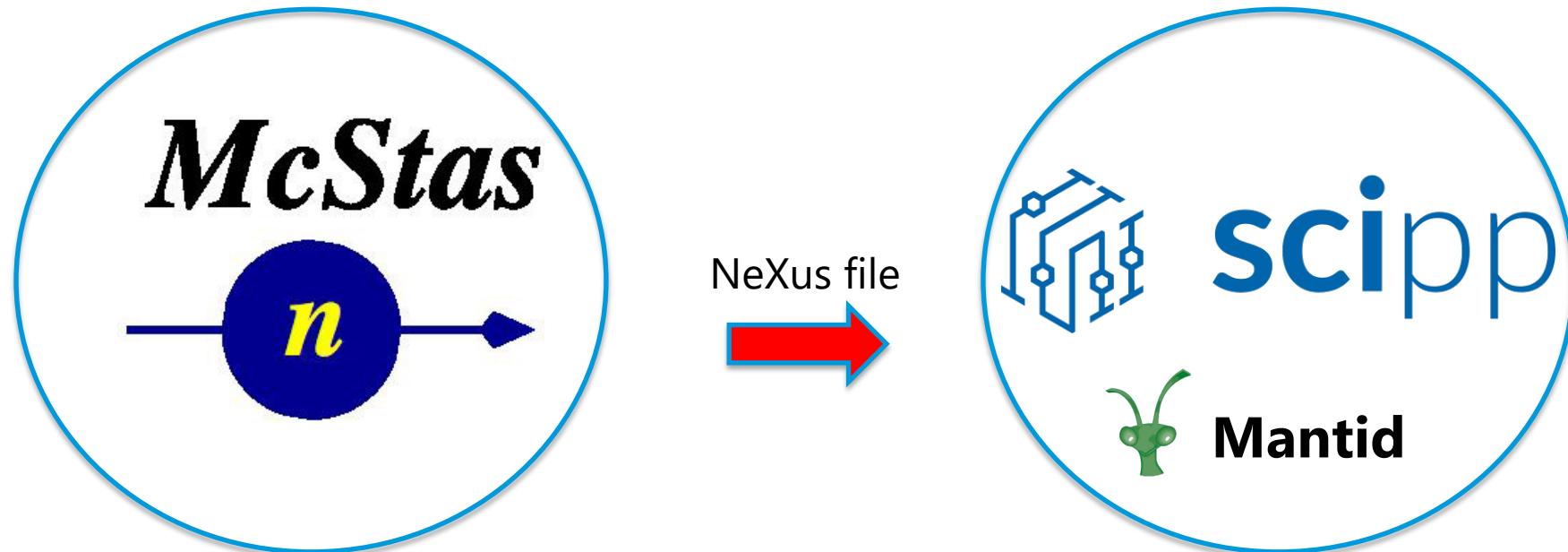
```
plotter.make_sub_plot(data, log=[False, True], fontsize=15)
```



McStas event data and The NeXus file



Process McStas event data as you would do with recorded data from a beamline



- <https://www.nexusformat.org>
- The McStas Nexus file must contain:
- Event data, i.e. each neutron has a pixel id and a time stamp
- An IDF McStas monitor_nD gives pixel id & time for each event
- *mcdisplay can auto-generate an IDF*

McStas Nexus file

Mantid/scipp compatible



HDFView 2.9

File Window Tools Help

Recent Files /home/ub/tmp/McStas_Event/McStas/Exp_Data/mccode.h5

Clear Text

Table View - events - /entry1/data/k01_events_dat_list_p_x_y_n_id_t/ - /home/ub/tmp/McStas_Event/McStas/Exp_Data/mccode.h5

Table

	0	1	2	3	4	5
0	3.185987...	-0.01450...	0.007932...	0.0	18717.0	0.025524...
1	0.192848...	0.141796...	0.347311...	1.0	31227.0	0.046068...
2	2.98905...	-0.01409...	-5.33756...	2.0	18333.0	0.025703...
3	0.001396...	-0.00520...	-0.01711...	3.0	17759.0	0.025971...
4	3.701651...	0.251736...	-0.12745...	4.0	13776.0	0.008363...
5	0.032482...	-0.11949...	0.083006...	5.0	21385.0	0.018234...
6	2.859783...	0.004956...	0.005223...	6.0	18729.0	0.041370...
7	0.022012...	0.058801...	-0.08114...	7.0	15	
8	8.530934...	0.012643...	-0.03070...	8.0	17	
9	6.984646...	-0.00867...	0.012712...	9.0	18	
10	0.009140...	-0.07853...	-0.07291...	10.0	15	
11	7.150334...	-0.01082...	0.005151...	11.0	18	
12	0.001010...	0.457856...	0.087702...	12.0	21	
13	0.004965...	0.003146...	-0.04262...	13.0	16	
14	8.629457...	-0.03193...	-0.02392...	14.0	17	
15	7.072494...	0.015294...	-0.01158...	15.0	17	
16	3.898581...	-2.38144...	0.011657...	16.0	18	
17	3.479036...	-0.02370...	-0.00281...	17.0	18	
18	5.870513...	0.006430...	0.006773...	18.0	18	
19	0.010325...	-0.07164...	-0.01926...	19.0	17	
20	3.191063...	-0.00829...	-0.00252...	20.0	18	
21	3.746034...	-0.00570...	0.011053...	21.0	18	
22	0.003724...	0.038904...	0.029156...	22.0	19	
23	3.316878...	-0.00442...	6.097504...	23.0	18	
24	0.023456...	-0.03083...	0.084904...	24.0	21	
25	0.001505...	0.031248...	-0.02562...	25.0	17	

Text View - data - /entry1/instrument/instrument_xml/ - mccode.h5

Data selection: [0] ~ [0]

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- IDF generated using McStas McDisplay and the Mantid backend -->
<!-- For help on the notation used to specify an Instrument Definition File see
http://www.mantidproject.org/IDF -->
<instrument name="ISIS_SANS2d_Mantid.out" valid-from = "1900-01-31 23:59:59"
valid-to = "2100-01-31 23:59:59" last-modified="Tue Apr 4 14:17:50 2017">
<defaults>
    <length unit="meter"/>
    <angle unit="degree"/>
    <reference-frame>
        <!-- The z-axis is set parallel to and in the direction of the beam. The
y-axis points up and the coordinate system is right handed. -->
        <along-beam axis="z"/>
        <pointing-up axis="y"/>
        <handedness val="right"/>
    </reference-frame>
    <default-view axis-view="z"/>
</defaults>
```

data (22370, 2)
String, length = 7891, 1
Number of attributes = 0

Log Info Metadata

2022-07-04 PRESENTATION TITLE/FOOTER

How to use: Online documentation



Github McStas wiki pages
Archive - lanl.arXiv.org
Docs in Mantid/scipp



Search the docs ...

GETTING STARTED

Installation

USER GUIDE

Coordinate Transformations
GroupBy
From Mantid to Scipp
Instrument view

scippneutron.load(filename='', load_pulse_times=True, instrument_filename=None, error_connection=None, mantid_elg='Load', mantid_args=None, advanced_geometry=False)

Wrapper function to provide a load method for a Nexus file, hiding mantid specific code from the scipp interface. All other keyword arguments not specified in the parameters below are passed on to the mantid.Load function.

Example of use:

```
from scipp.neutron import load
d = sc.DataSet()
d["sample"] = load(filename="PG3_4844_event.nxs",
                   load_pulse_times=False,
                   mantid_args={'bankName': 'bank184',
                               'LoadMonitors': True})
```

LoadMcStas v1

Table of Contents

- Summary
- Properties
- Description
 - McStas compiling and running
 - McStas event data conventions
 - Tested versions
 - References

Loads a McStas Nexus file into an workspace.

Filename Browse

OutputWorkspace

LoadMcStas dialog.

Summary
Loads a McStas Nexus file into an workspace.

Properties

Name	Direction	Type	Default	Description
Filename	Input	string	Mandatory	The name of the Nexus file to load
OutputWorkspace	Output	Workspace	Mandatory	An output workspace.

Description

[1607.02498] McStas and Mantid Integration - Mozilla Firefox

McStas and Mantid ... x [1607.02498] McStas and ... x

lanl.arxiv.org/abs/1607.02498

Cornell University Library

lanl.arxiv.org > physics > arXiv:1607.02498

Physics > Instrumentation and Detectors

McStas and Mantid integration

Torben R Nielsen (1), Anders J Markvardsen (2), Peter K Willendrup (1,3) ((1) European Spallation Source ERIC, Denmark (2) ISIS Facility, Rutherford Appleton Laboratory, UK (3) Physics Department, Technical University of Denmark)

(Submitted on 8 Jul 2016 (v1), last revised 25 Aug 2016 (this version, v2))

McStas and Mantid are two well established software frameworks within the neutron scattering community. McStas has been primarily used for simulating the neutron transport of instruments, while Mantid has been primarily used for data reduction. We report here the status of our work done on the interoperability between the instrument simulation software McStas and the data reduction software Mantid. This provides a demonstration of how to successfully link together two software that otherwise have been developed independently, and in particular here show how this has been achieved for an instrument simulation software and a data reduction software. This paper will also provide examples of some of the expected future enhanced analysis that can be achieved from combining accurate instrument and sample simulations with software for correcting raw data. In the case of this work for raw data collected at large scale neutron facilities.

Comments: 17 pages, 12 figures, POSTPRINT with proofs of article submitted to Journal of Neutron Research

Subjects: Instrumentation and Detectors (physics.ins-det)

Cite as: arXiv:1607.02498 [physics.ins-det]

Bookmark (what is this?)

IDF xml data, TOF and pixel ID's



Mantid's IDF stores geometry information used in TOF analysis

This implies parsing information about:

1. where the neutron source is located,
2. where the sample is located,
3. where each individual detector pixel is located.

McStas event KEYWORDS



McStas instrument file name and the McStas defined name of the instrument must be the same:

E.g. **templateSANS_Mantid.instr** and "**DEFINE INSTRUMENT templateSANS_Mantid(....)**"

In the McStas instrument file the source must be named "**sourceMantid**"

E.g. "**COMPONENT sourceMantid = Source_simple(....)**"

In the McStas instrument file the sample must be named
"sampleMantid"

E.g. "**COMPONENT sampleMantid = Sans_spheres(....)**"

In the McStas instrument file the event monitors must be named
"nD_Mantid_#"

E.g. "**COMPONENT nD_Mantid_1 = Monitor_nD(....)**"

templateSANS.inst vs templateSANS_Mantid.inst

templateSANS.inst x templateSANS_Mantid.instr UNREGISTERED

```

1 DEFINE INSTRUMENT templateSANS(lambda=6, dlambd=0.05, r=100, PHI=1e-3, Delta_Rho=0.6, sigma_abs=0.5)
2
3 TRACE
4
5 COMPONENT a1 = Progress_bar()
6 | AT (0,0,0) ABSOLUTE
7
8 COMPONENT arm = Arm()
9 | AT (0, 0, 0) ABSOLUTE
10
11 COMPONENT source = Source_simple(
12 | radius = 0.02, dist = 3, focus_xw = 0.01, focus_yh = 0.01,
13 | lambda0 = lambda, dlambd = dlambd, flux = 1e16)
14 | AT (0, 0, 0) RELATIVE arm
15
16 COMPONENT coll1 = Slit(
17 | radius = 0.005)
18 | AT (0, 0, 3) RELATIVE arm
19
20 COMPONENT coll2 = Slit(
21 | radius = 0.005)
22 | AT (0, 0, 6) RELATIVE arm
23
24
25
26
27
28
29
30 SPLIT COMPONENT sample = Sans_spheres(
31 | R=r, Phi=PHI, Delta_rho=Delta_Rho, sigma_abs=sigma_abs,
32 | xwidth=0.01, yheight=0.01, zdepth=0.005, focus_xw=0.6, focus_yh=0.
33 | 6, target_index=2)
34 | AT (0,0,0.2) RELATIVE coll2
35
36 COMPONENT STOP = Beamstop(
37 | radius = 0.02)
38 | AT (0, 0, 2.9) RELATIVE sample

```

Line 1, Column 1 Spaces: 2 Plain Text

templateSANS.instr x templateSANS_Mantid.instr UNREGISTERED

```

1 DEFINE INSTRUMENT templateSANS_Mantid(lambda=6, dlambd=0.05, r=150, PHI=1e-3, Delta_Rho=0.6, sigma_abs=0.0)
2
3 TRACE
4
5 COMPONENT a1 = Progress_bar()
6 | AT (0,0,0) ABSOLUTE
7
8 COMPONENT arm = Arm()
9 | AT (0, 0, 0) ABSOLUTE
10
11 COMPONENT sourceMantid = Source_simple(
12 | radius = 0.02, dist = 3, focus_xw = 0.01, focus_yh = 0.01,
13 | lambda0 = lambda, dlambd = dlambd, flux = 1e16)
14 | AT (0, 0, 0) RELATIVE arm
15
16 COMPONENT coll1 = Slit(
17 | radius = 0.005)
18 | AT (0, 0, 3) RELATIVE arm
19
20 COMPONENT coll2 = Slit(
21 | radius = 0.005)
22 | AT (0, 0, 6) RELATIVE arm
23
24 COMPONENT LdetectorPRE = L_monitor(
25 | nL = 1000, filename = "Edet0.dat", xmin = -0.3,
26 | xmax = 0.3, ymin = -0.3, ymax = 0.3, Lmin = 5.5,
27 | Lmax = 6.5)
28 | AT (0,0,0.05) RELATIVE coll2
29
30 SPLIT COMPONENT sampleMantid = Sans_spheres(
31 | R=r, Phi=PHI, Delta_rho=Delta_Rho, sigma_abs=sigma_abs,
32 | xwidth=0.01, yheight=0.01, zdepth=0.005)
33 | AT (0,0,0.2) RELATIVE coll2
34 EXTEND %{
35 | | | if (!SCATTERED) ABSORB;
36 %}

```

Line 1, Column 1 Spaces: 2 Plain Text

templateSANS.inst vs templateSANS_Mantid.inst

Two code snippets are shown side-by-side, illustrating the differences between templateSANS.inst and templateSANS_Mantid.inst.

Left Window (templateSANS.inst):

```

COMPONENT detector = PSD_monitor(          UNREGISTERED
COMPONENT nD_Mantid_1 = Monitor_nD(      .
1 COMPONENT detector = PSD_monitor(        .
2   nx = 128, ny = 128, filename = "PSD.dat", xmin = -0.3,
3   xmax = 0.3, ymin = -0.3, ymax = 0.3)
4   AT (0, 0, 3) RELATIVE sample
5

```

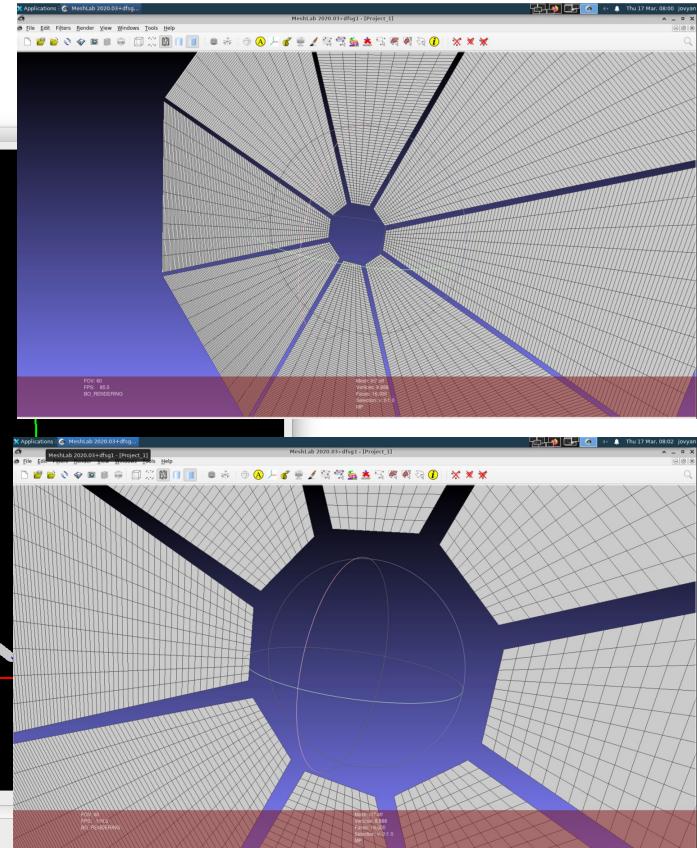
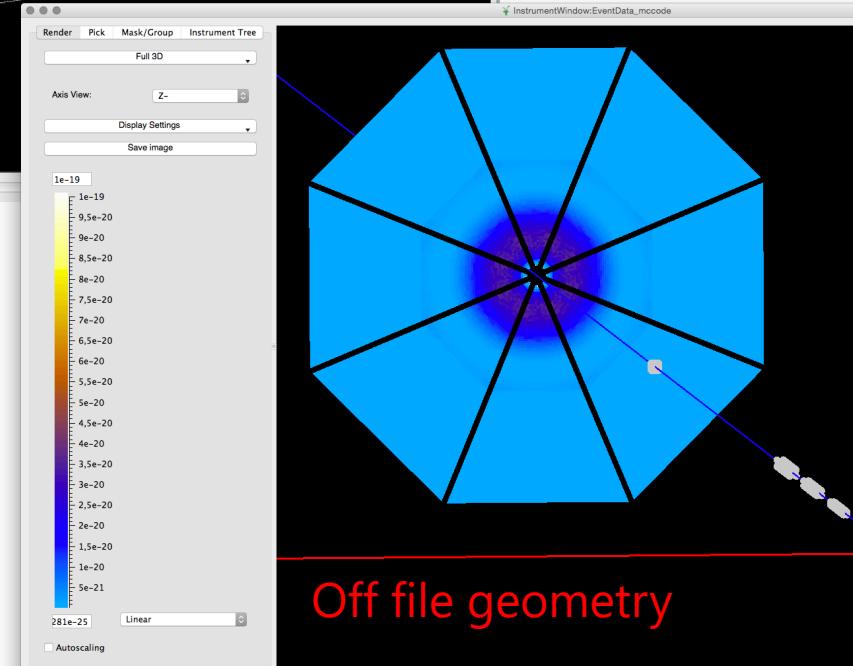
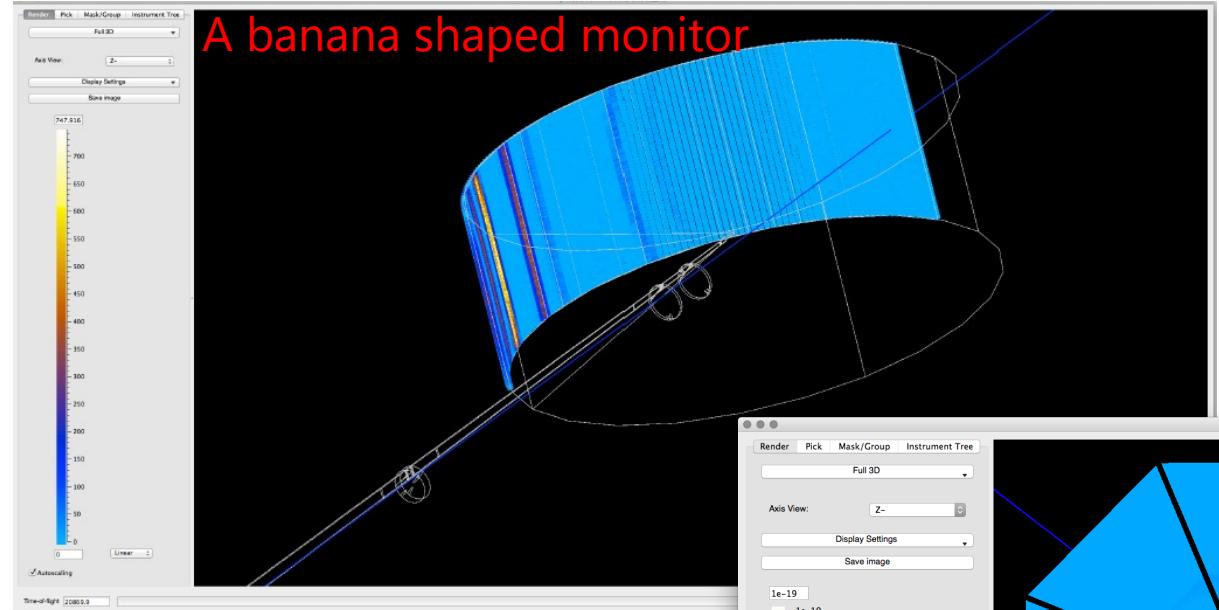
Right Window (templateSANS_Mantid.inst):

```

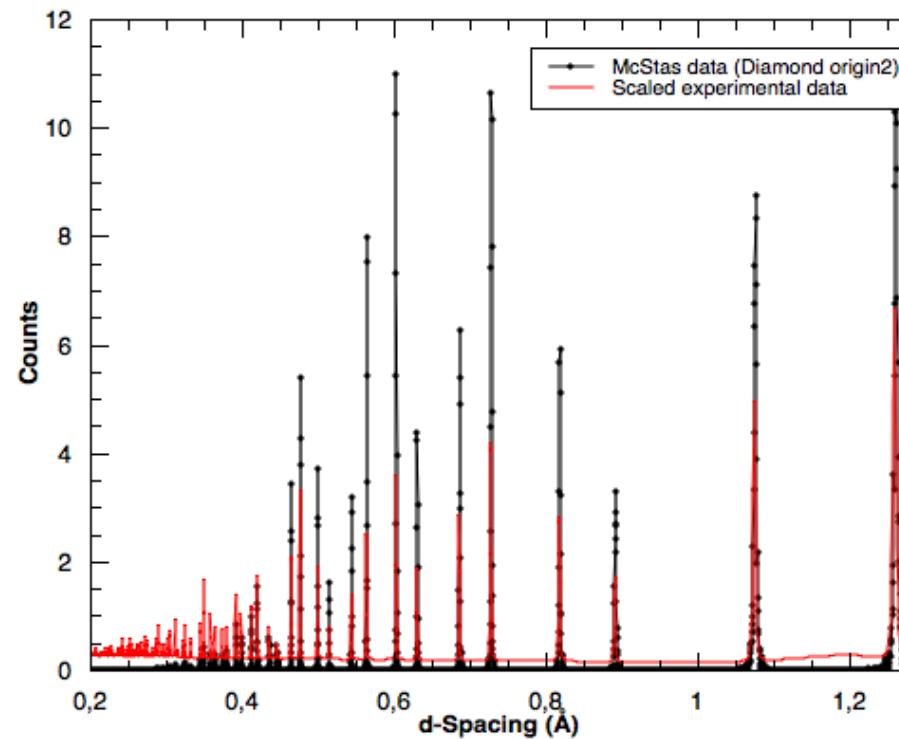
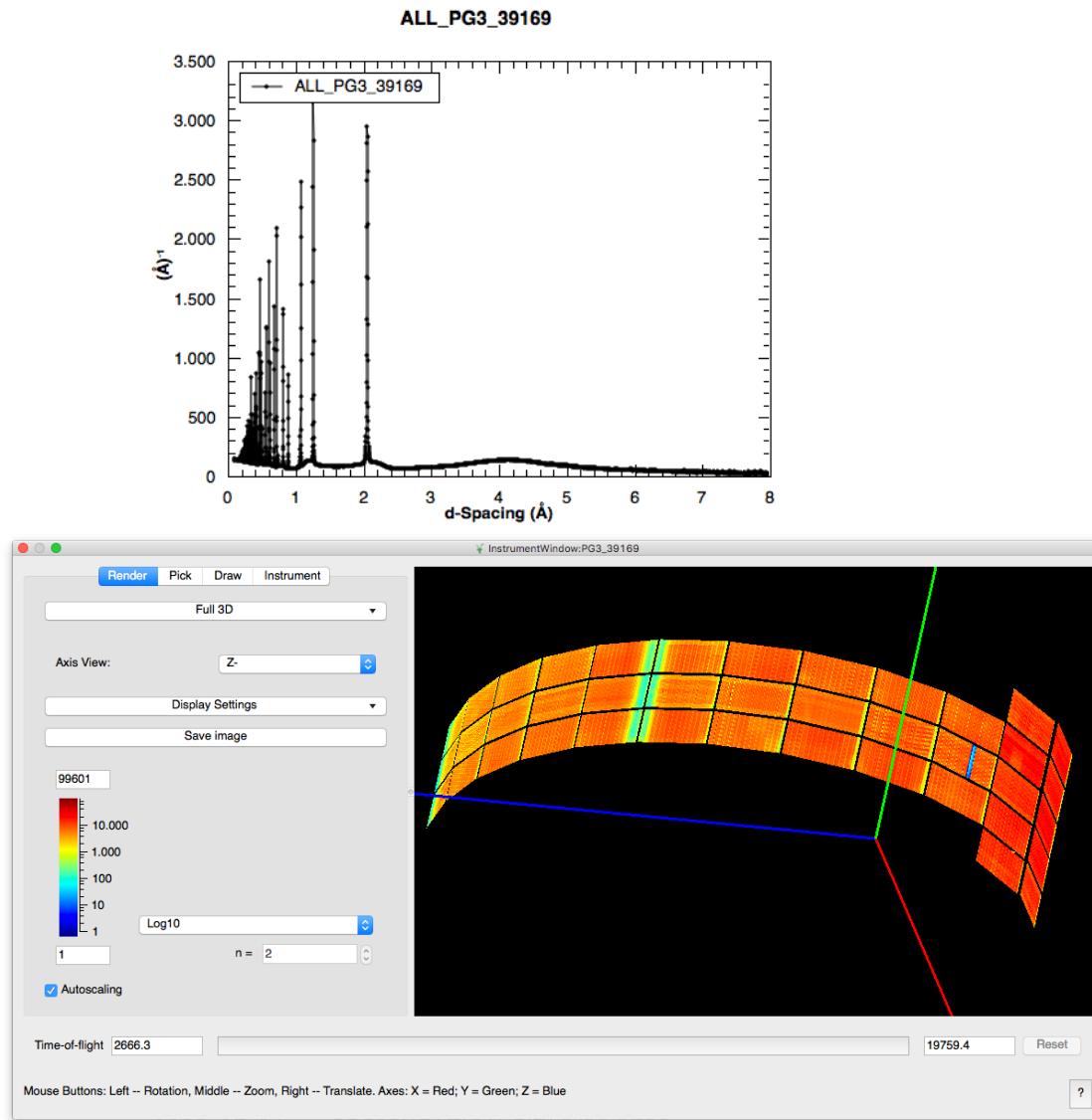
COMPONENT nD_Mantid_1 = Monitor_nD(          UNREGISTERED
COMPONENT detector = PSD_monitor(      .
1 COMPONENT nD_Mantid_1 = Monitor_nD(    .
2   options ="mantid square x limits=[-0.3 0.3] bins=128
3   y limits=[-0.3 0.3] bins=128, neutron pixel min=0 t,
4   list all neutrons",
5   xmin = -0.3,
6   xmax = 0.3,
7   ymin = -0.3,
8   ymax = 0.3,
9   restore_neutron = 1,
10  filename = "bank01_events.dat")
10  AT (0, 0, 3.2) RELATIVE sampleMantid

```

Examples of detectors: IN5 (ILL) and LoKI (ESS)



McStas and experimental data: POWGEN



By CelineD @ ESS + SNS collaboration

How to use: How to run simulation

Github wiki pages



McStas GUI

Generating McStas event data for Mantid can be achieved from the McStas GUI `mogui`. Below we show how to setup the simulation on Windows 7, OSX 10.12, and Ubuntu 16.04. For `McStas` we use version 2.4.1. For `Mantid` use version 3.4 or later.

Windows users need to download the latest version of `mccode-r.c` in order to generate IDF data in the `mccode.h5` file which Mantid can read. Copy the downloaded file to the directory where your McStas instrument file is located.

1. Open the McStas configuration file. In `mogui` go to: File -> Configuration
2. Change the setting as shown in figures below:
 - o In the section `mcrun` select `mcrun --format=Nexus`
 - o In the section `mcplot` select `mcplot-pyqtgraph`
 - o In the section `mcdisplay` select `mcdisplay-pl --format=Mantid` (Windows) or `mcdisplay.pl --format=Mantid` (OSX or Ubuntu)
 - o In the section `C flags` select this line depending on your OS
 - Windows: `-g -O2 -lm -DUSE_NEXUS -lNexus-0 -I "C:/Program Files (x86)/NeXus Data Format/include/nexus" -L "C:/Program Files (x86)/NeXus Data Format/lib/nexus"`
 - OSX: `-g -O2 -lm -DUSE_NEXUS -lNexus -I/usr/local/include/nexus`
 - Ubuntu: `-g -O2 -lm -DUSE_NEXUS -lNexus`

McStas CLI

Generating McStas event data for Mantid can also be controlled from the terminal, using this command line interface (CLI). Below we show how to setup the simulation for OSX 10.12 and Ubuntu 14.04. The procedure for Windows is similar. For `McStas` we use version 2.4.1.

Mac OS X and Ubuntu

1. Compile McStas instrument file

```
$ mcstas templateSANS_Mantid.instr --trace
```
2. Generate the IDF for Mantid

```
$ mcdisplay.pl templateSANS_Mantid.instr --format=Mantid -n0
```
3. Generate the executable for Mac OS X

```
$ gcc -o templateSANS_Mantid.out templateSANS_Mantid.c -lm -DUSE_NEXUS -lNexus -I/usr/local/include/nexus/
```
- And for Ubuntu
4. Run the executable and use the default parameters

```
$ ./templateSANS_Mantid.out --format=Nexus -d 1E8
```

13

How to use CLI: Generate the IDF



- How-to generate McStas event data for Mantid/scipp

Step 1. Generate c file:

```
mcstas templateSANS_Mantid.instr --trace
```

Step 2. Generate IDF:

```
mcdisplay-mantid.pl templateSANS_Mantid.instr -n0
```

A new xml IDF file is then generated on disk

How to use CLI: Make NeXus file for Mantid/scipp



- How-to generate McStas event data for Mantid/scipp

Step 1. Compile c code:

```
gcc -o templateSANS_Mantid.out templateSANS_Mantid.c -DUSE_NEXUS -INeXus -Im -I/usr/include/nexus
```

Step 2. Run simulation:

```
templateSANS_Mantid.out --format=Nexus
```

Now IDF is embedded in the NeXus file to be read by Mantid/scipp



Demo



Let's try to do a test



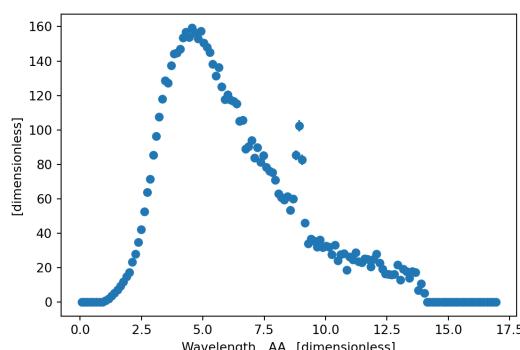
Exercise



Compile a Mantid/scipp instrument and generate event data

- * choose your instrument
- * compile and run (use 2.7.1)
- * load into scipp
- * view instrument
- * plot monitors (in scipp)

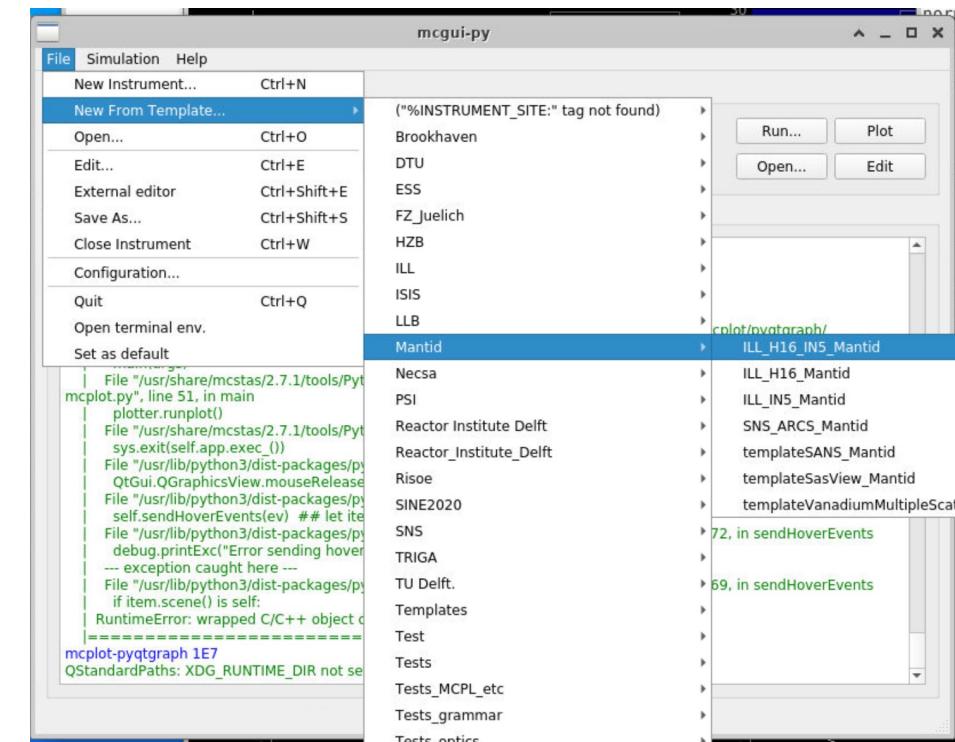
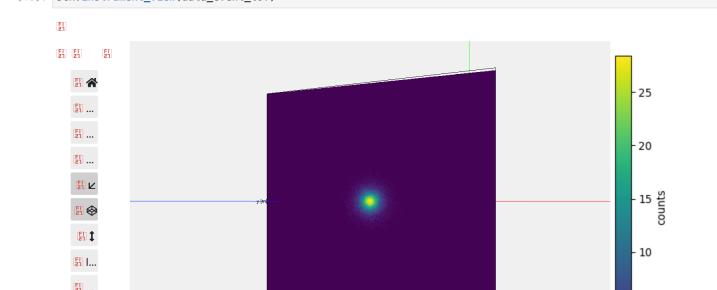
```
[10]: data_mcstas['lmonitor_post.dat'].plot()
```



```
[35]: file_name_mcstas = '1E7/mccode.h5'  
[36]: data_mcstas = scn.load(file_name_mcstas)  
Load identical. Load started.  
[38]: data_mcstas['EventData']  
scipp.DataArray (25.27 MB)
```

```
Dimensions: (spectrum: 40000, tof: 1)  
Coordinates:  
position (spectrum) vector3 m [-0.05627572 -0.4975 23.55533914], [-0.05127716 -0.4  
sample_position (spectrum) vector3 m [ 0.347518 0.19.6445 ]  
source_position (spectrum) vector3 m [-0.125071 0. -0.0488352]  
spectrum (spectrum) int32 1, 2, ..., 39999, 40000  
tof (tof [bin-edge]) float64 us 5953.265, 8.399e+04  
Data:  
(spectrum, tof) DataArrayView binned data [len=2, len=4, ..., len=0, len=2]  
Attributes:  
instrument_name (spectrum) string McStas  
sample (spectrum) PyObject <mantid.api._api.Sample object at 0x7fde005964a0>
```

```
[39]: data_event_tof = data_mcstas['EventData']  
[40]: scn.instrument_view(data_event_tof)
```





Finish presentation