



EUROPEAN
SPALLATION
SOURCE



McStas - Mantid

Torben Nielsen

Anders Markvardsen

Peter Willendrup

MCSTAS WORKSHOP

May 2021



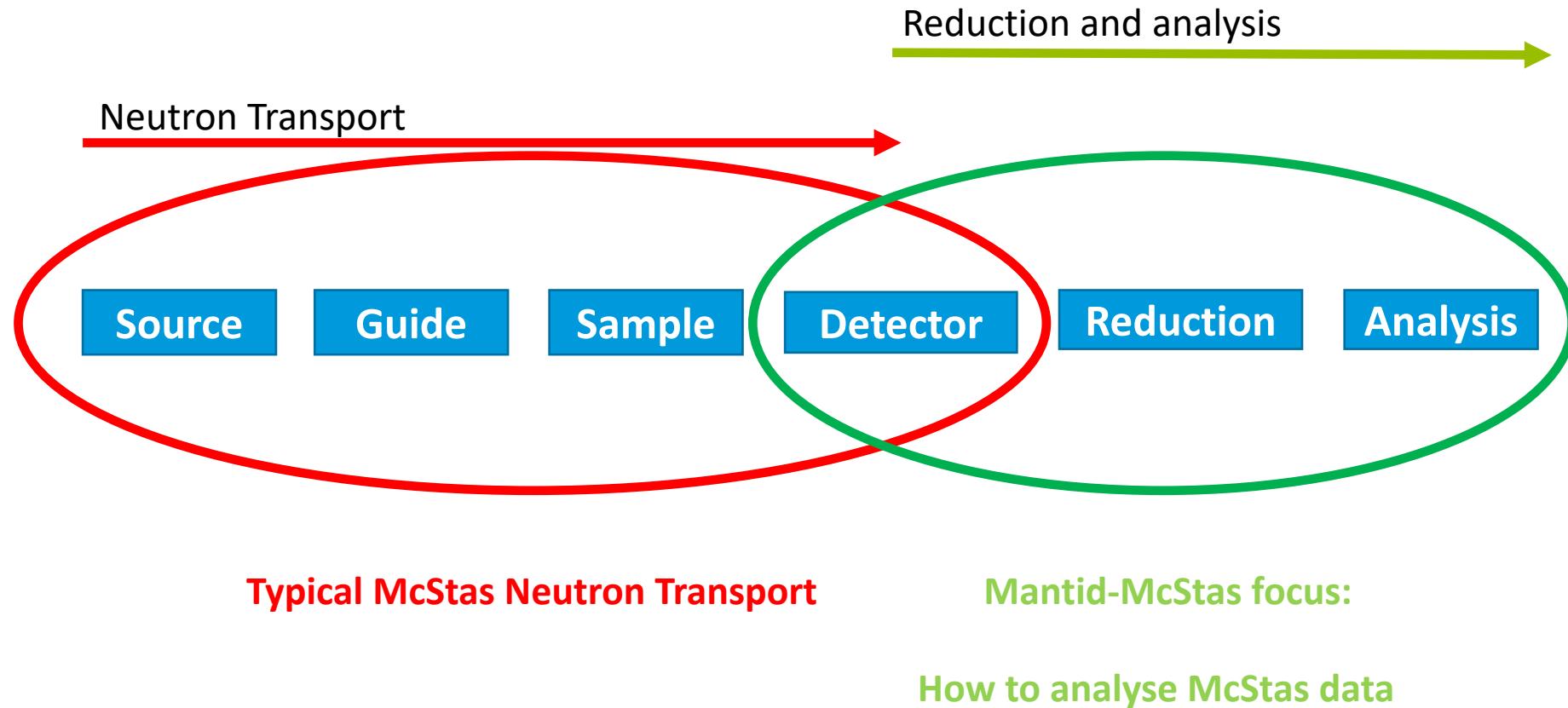
McStas event-data for data-reduction

Data-reduction = { Mantid, Scipp }

McStas-Mantid and the neutron pipeline



Focus: How to analyse McStas data



Comparison



Classic McStas

mcstas - transport



mcstas - plot



Post processing:
iFit / Matlab / own code

Mantid-McStas

mcstas - transport



mantidplot



Post processing:
Mantid algorithms

Motivation: new flexible dataflow



1. Combine the reduction framework and the neutron transport framework
2. Neutron transport, reduction and “analysis” in one go
3. View McStas data in Mantid (histogram data)
4. Import McStas event data to Mantid
5. Use already developed and tested algorithms in Mantid to process McStas event data (no need to reinvent methods)
6. Advanced use: Use McStas to quantify (perhaps remove) “spurries” signal on detector (can scatt. – multi. scatt)

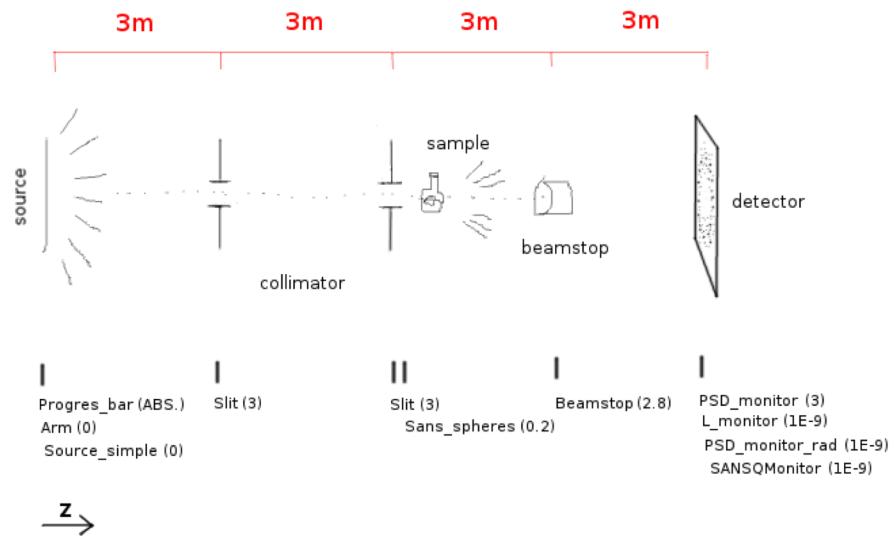
Illustration: simplified SANS instrument



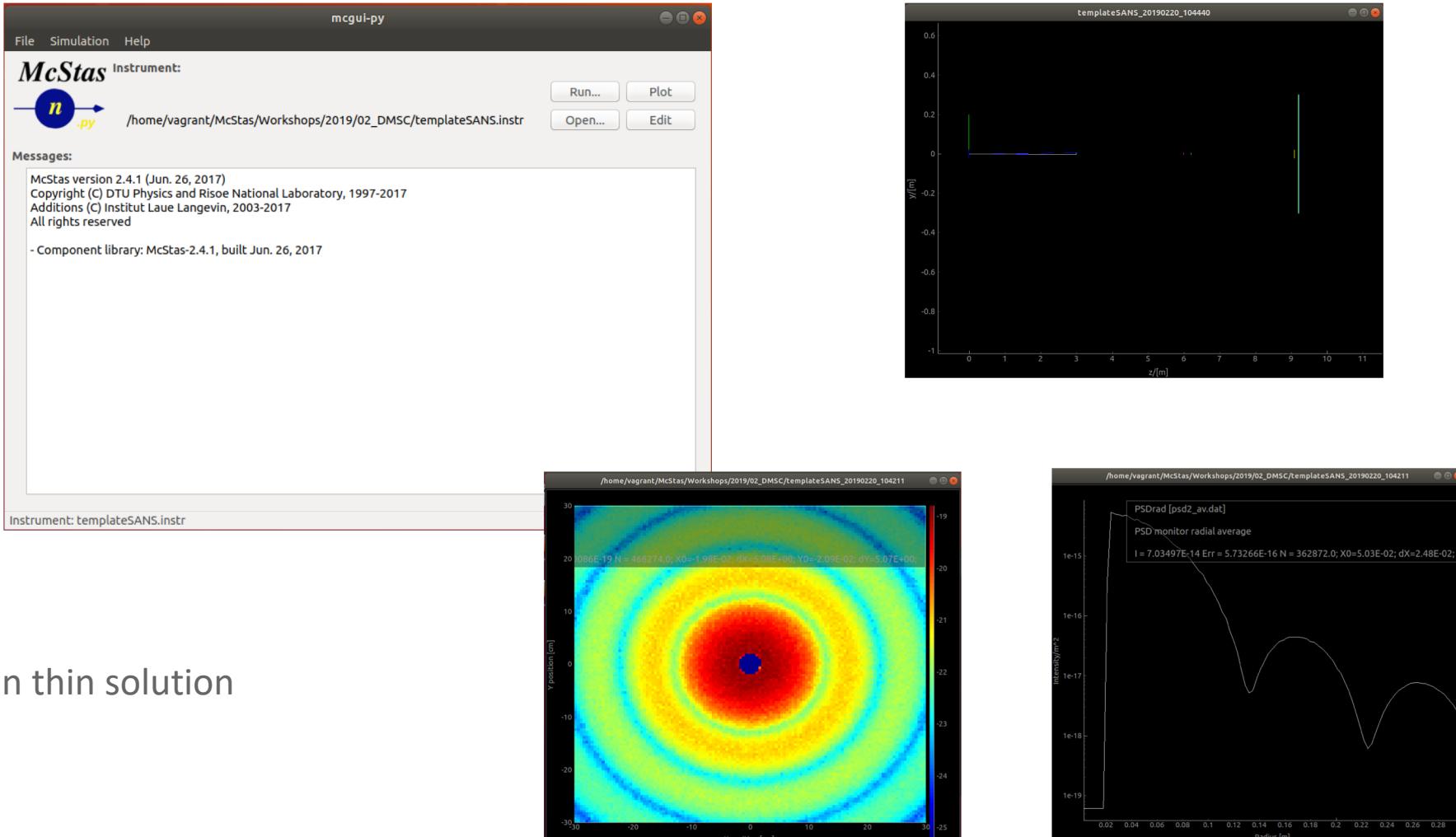
McStas wiki page on Github

The screenshot shows a GitHub repository page for 'McStasMcXtrace / McCode'. The page title is 'McStas tutorial: simplified SANS instrument'. It was last edited by Jakob Garde on Feb 8, 2016, with 29 revisions. The page content starts with a brief description of the tutorial: 'In this tutorial, you will write a simplified SANS instrument. When you have completed this tutorial, you will have learned the basics of mctas.' Below this, there is a bulleted list: '• Requirements: mctas 2.2a.'

- SANS
- Small angle neutron scattering
- Can give structural information of nano sized particles



McStas GUI: templateSANS.instr



Mantid / MantidWorkBench



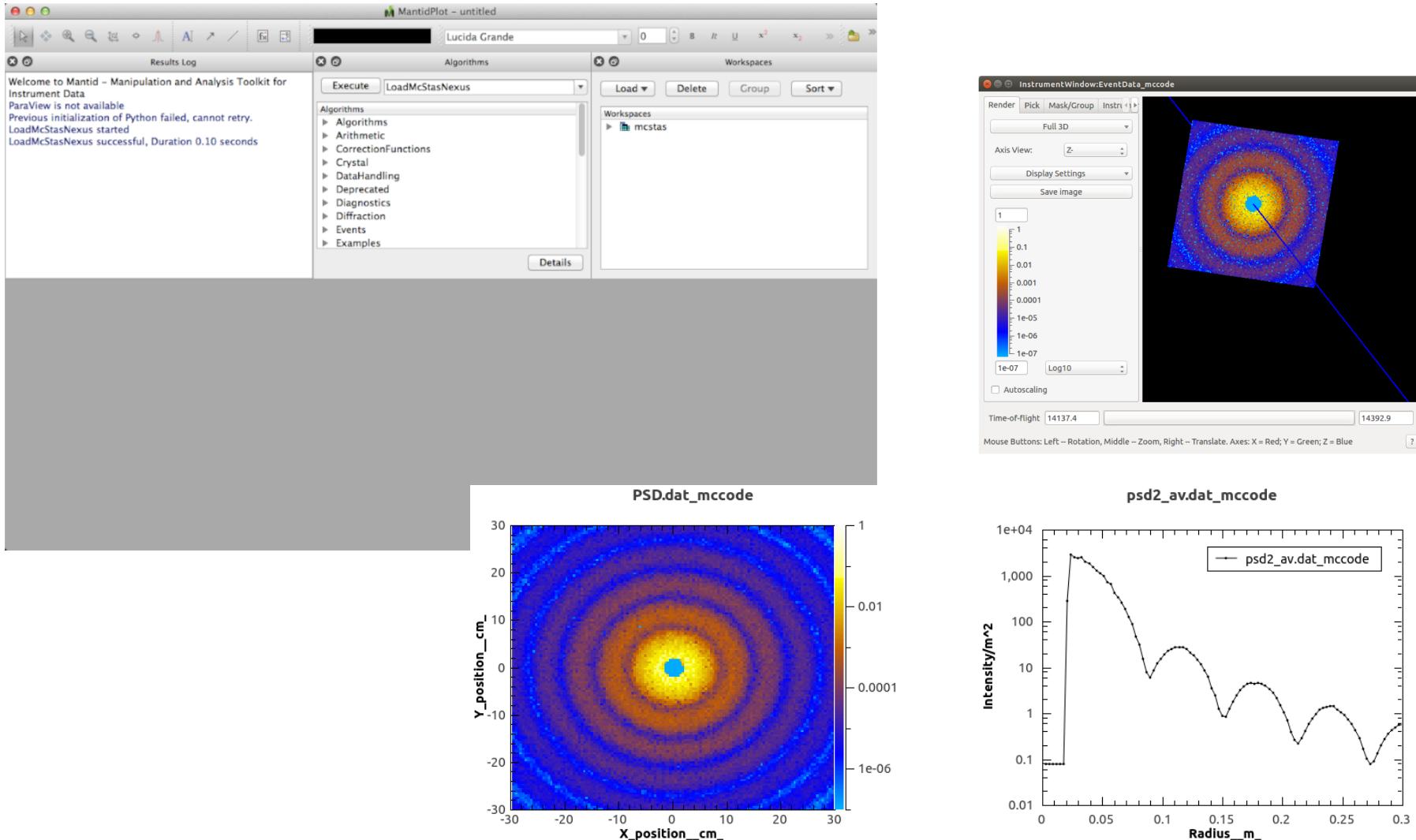
Desktop APP and python CLI: <https://www.mantidproject.org>

The image shows a desktop environment with three main windows:

- MantidPlot - untitled**: A graphical user interface for data visualization. It features a toolbar at the top, a font selection dropdown (Helvetica), and a results log area. The log displays initialization and execution times. Below the log is a large green banner with the Mantid logo and "version 5.0.0". The main workspace contains a "Personal Setup" dialog box with fields for Default Facility (TEST_LIVE), Default Instrument (ISIS_Kafka_Event), and Set data directories. A "Report usage data" checkbox is checked. A "Privacy Policy" link is also present. A "Cancel" and "Set" button are at the bottom right. A mantis illustration is visible in the background.
- Worksheets**: A Jupyter QtConsole window showing Python 3.8.3 and IPython 7.14.0 environments. The console output includes details about the Python version and license information. The code cell "In [1]:" is empty.
- Mantid Workbench**: A central workspace for managing workspaces and running algorithms. It includes tabs for "Editor" and "Messages". The "Editor" tab shows a Python script with imports for mantid, numpy, and matplotlib.pyplot, followed by imports for mantid.simpleapi, matplotplib.pyplot, and numpy. The "Messages" tab displays a welcome message for Mantid 6.0.0, citation information, and a link to the release note. The "Algorithms" panel on the left lists various categories of algorithms: Arithmetic, CorrectionFunctions, Crystal, DataHandling, Diagnostics, Diffraction, Events, Examples, ILL, ISIS, Inelastic, MDALgorithms, Muon, Optimization, PythonAlgorithms, Reflectometry, SANS, and SINQ. A "Details" button is located at the bottom of this panel.

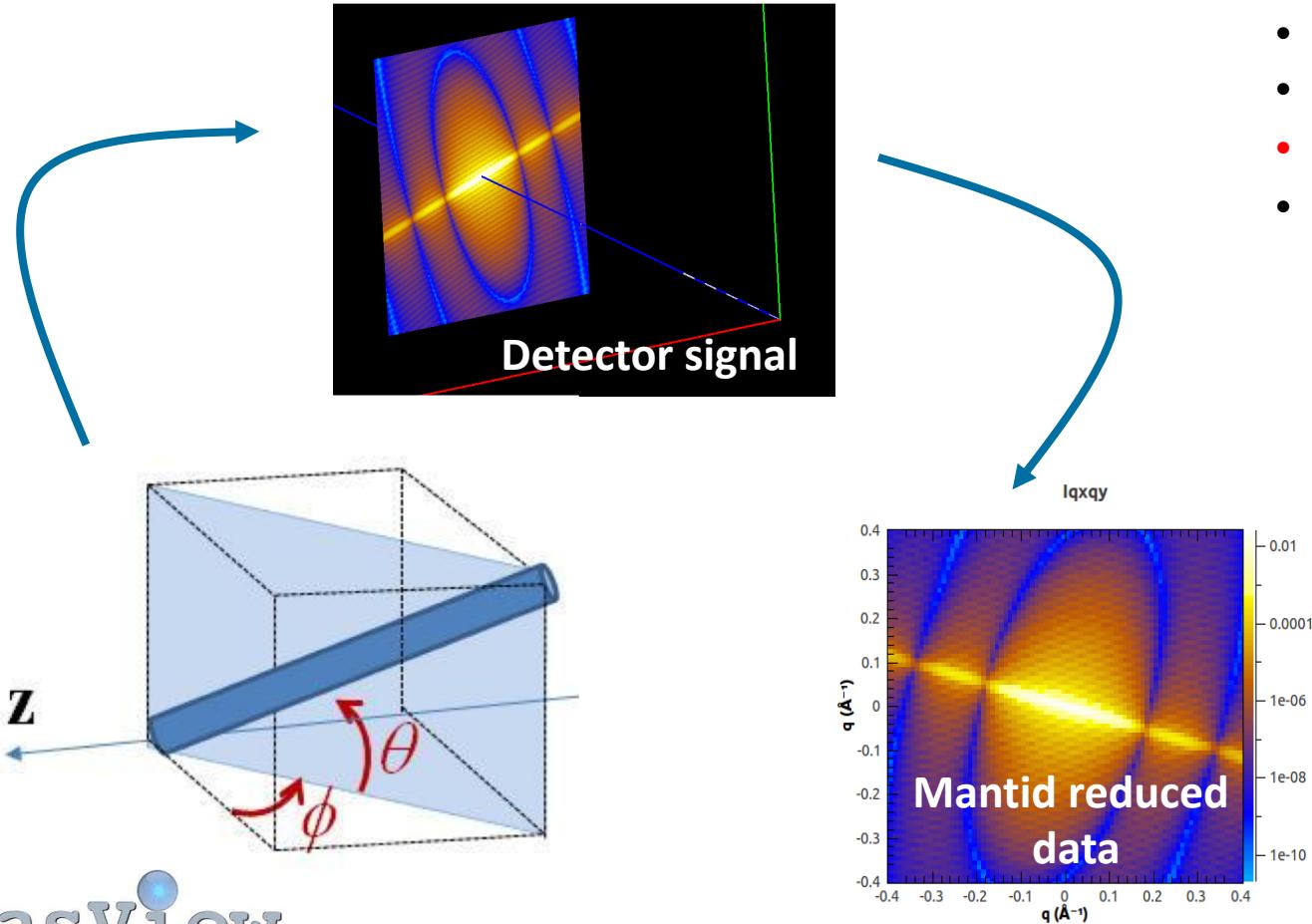
Mantid GUI: templateSANS.instr

Using Mantid 5.0:



Example: Usage of Mantid algorithms

Let's try to use a 2D scattering kernel. And use Mantid for data reduction



- SasView_model.comp
- 2D scattering kernel from SasView
- **2D reduction in Mantid: Qxy**
- Can be send back to SasView for fitting

Mantid

Qxy v1

Table of Contents

- Summary
 - See Also
- Properties
- Description
- Source

Summary

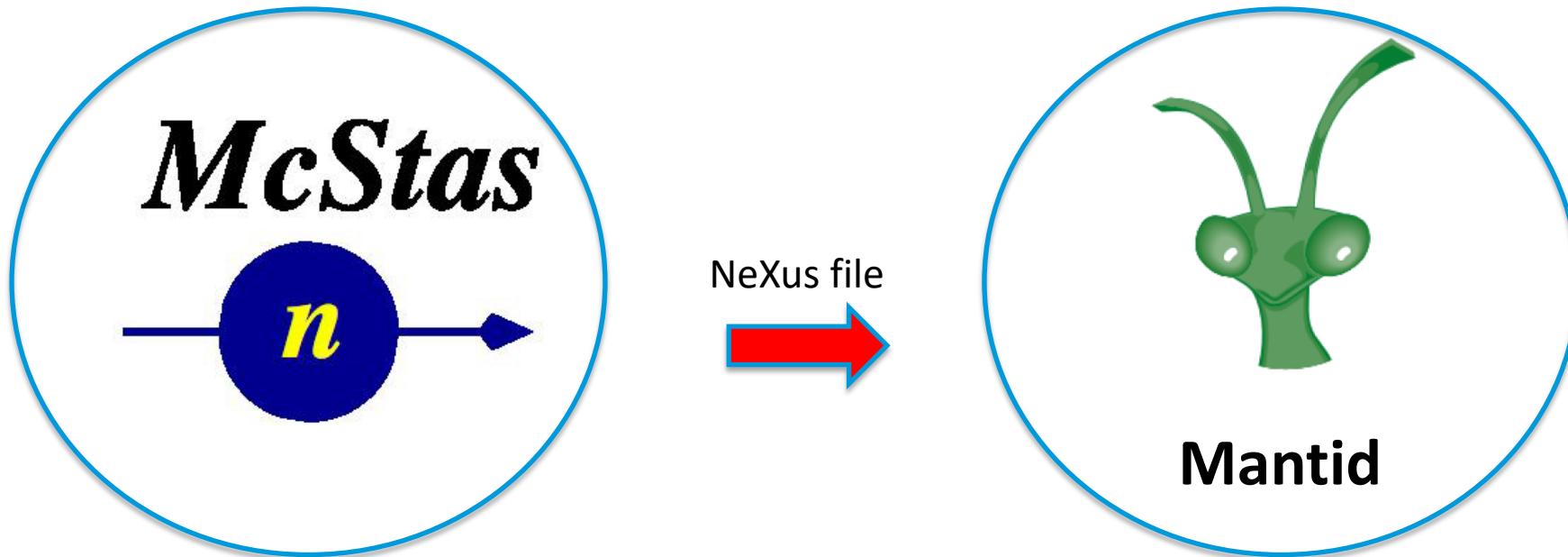
Performs the final part of a SANS (LOQ/SANS2D) two dimensional (in Q) data reduction.

See Also

Properties

| Name | Direction | Type | Default | Description |
|---------------------|-----------|-----------------|-----------|---|
| InputWorkspace | Input | MatrixWorkspace | Mandatory | The corrected data in units of wavelength. |
| OutputWorkspace | Output | MatrixWorkspace | Mandatory | The name to use for the corrected workspace. |
| MaxQxy | Input | number | Mandatory | The upper limit of the Qx-Qy grid (goes from -MaxQxy to +MaxQxy). |
| DeltaQ | Input | number | Mandatory | The dimension of a Qx-Qy cell. |
| IQxQyLogBinning | Input | boolean | False | I(qx,qy) log binning when binning is not specified. |
| PixelAdj | Input | MatrixWorkspace | | The scaling to apply to each spectrum e.g. for detector efficiency, must have just one bin per spectrum and the same number of spectra as DetBankWorkspace. |
| WavelengthAdj | Input | MatrixWorkspace | | The scaling to apply to each bin to account for monitor counts, transmission fraction, etc. Must be one spectrum with the same binning as the InputWorkspace, the same units (counts) and the same [[ConvertToDistribution distribution status]]. |
| AccountForGravity | Input | boolean | False | Whether to correct for the effects of gravity. |
| SolidAngleWeighting | Input | boolean | True | If true, pixels will be weighted by their solid angle. |

The NeXus file



- 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*

<https://www.nexusformat.org>

<https://manual.nexusformat.org>

NeXus

NeXus is developed as an international standard by scientists and programmers representing major scientific facilities in Europe, Asia, Australia, and North America in order to facilitate greater cooperation in the analysis and visualization of neutron, x-ray, and muon data.

Home
[GitHub Organisation](#)
 © 2021 NIAC

About the NeXus Data Format

NeXus is a common data format for neutron, x-ray, and muon science. It is being developed as an international standard by scientists and programmers representing major scientific facilities in order to facilitate greater cooperation in the analysis and visualization of neutron, x-ray, and muon data.

Documentation:

- Most recent publication to cite:
J. Appl. Cryst. (2015), **48**, 301-305 doi:[10.1107/S1600576714027575](https://doi.org/10.1107/S1600576714027575)
- [User Manual](#):
 - Introduction to the concepts behind the NeXus data format
 - Design: The hierarchical design of NeXus files
 - NeXus Class Definitions: description of each NXDL specification
 - base classes: components that might be used in any NeXus data file
 - application definitions: layout specifications for a specific purpose
 - contributed definitions: propositions from the community
 - Utilities: Software applications that browse, plot, and analyze NeXus data
 - FAQ: Commonly asked questions about NeXus
- Facilities using NeXus

Discussion and Development:

- Most recent meetings: [NIAC2020](#) and [Code Camp 2021-1](#)
- Teleconferences
- Current Active Projects
- NeXus International Advisory Committee
- Bug / Issue Reporting
- Mailing Lists

2.1.2. Python Examples using h5py — nexus v2020.10 documentation

2.1.2. Python Examples using h5py ¶

One way to gain a quick familiarity with NeXus is to start working with some data. For at least the first few examples in this section, we have a simple two-column set of 1-D data, collected as part of a series of alignment scans by the APS USAXS instrument during the time it was stationed at beam line 32ID. We will show how to write this data using the [Python](#) language and the [h5py](#) package [1] using [h5py](#) calls directly rather than using the NeXus API. The actual data to be written was extracted (elsewhere) from a [spec](#) [2] data file and read as a text block from a file by the Python source code. Our examples will start with the simplest case and add only mild complexity with each new case since these examples are meant for those who are unfamiliar with NeXus.

[1] [h5py: https://www.h5py.org/](https://www.h5py.org/)
[2] [SPEC: http://certif.com/spec.html](http://certif.com/spec.html)

The data shown plotted in the next figure will be written to the NeXus HDF5 file using only two NeXus base classes, `NXentry` and `NXdata`, in the first example and then minor variations on this structure in the next two examples. The data model is identical to the one in the [Introduction](#) chapter except that the names will be different, as shown below:



data structure, (from introduction)

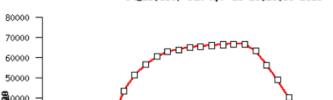
our h5py example

```

1 /entry:NXentry
2   /mr_scan:NXdata
3     /mr : float64[31]
4       /100 : int32[31]

```

84_25.dat, Sun Apr 25 10:20:56 2010



McStas Nexus file



HDFView 2.9

File Window Tools Help

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

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

Event data

| | 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 | 2.989055... | -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 | 18720.0 | 0.041270... |
| 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... | 0.005191... | 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

IDF xml data

```
<?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

McStas Nexus files (aka. mccode.h5)



Event data (templateSANS_Mantid.instr)

HDFView 3.1.0

Recent Files

Clear Text

Editor

Redu... run... Reduce_L...

bench

mccode.h5

entry1

data

Edet0_dat

Edet_dat

PSD_dat

k01_events_dat_list_p_x_y_n_id_t

acquisition_mode

distance

events

psd2_av_dat

psd2_dat

duration

end_time

instrument

program_name

simulation

start_time

title

events at /entry1/data/k01_events_dat_list_p_x_y_n_id_t/ [mccode.h5 in /Users/torbennielsen/ECD...

Event data

0-based

| | p | x | y | n | id | tof |
|----|--------------|--------------|--------------|------|---------|--------------|
| | 0 | 1 | 2 | 3 | 4 | 5 |
| 0 | 0.0913800... | -0.225551... | 0.1925273... | 0.0 | 13455.0 | 0.0142448... |
| 1 | 22.019941... | 0.0420603... | 0.1131936... | 1.0 | 11336.0 | 0.0142298... |
| 2 | 0.3301759... | 0.0807967... | 0.2567819... | 2.0 | 15185.0 | 0.0143417... |
| 3 | 0.3621693... | -0.239800... | -0.209276... | 3.0 | 2444.0 | 0.0142652... |
| 4 | 0.5076892... | -0.272713... | 0.1527182... | 4.0 | 12293.0 | 0.0142334... |
| 5 | 1.3581475... | 0.0836564... | 0.2396197... | 5.0 | 14801.0 | 0.0142137... |
| 6 | 0.3057814... | 0.0887109... | -0.203683... | 6.0 | 2642.0 | 0.0142206... |
| 7 | 0.8617839... | 0.2167541... | -0.005328... | 7.0 | 8046.0 | 0.0141667... |
| 8 | 16.073920... | -0.013003... | -0.136530... | 8.0 | 4413.0 | 0.0142430... |
| 9 | 4.3700886... | -0.062985... | 0.1794850... | 9.0 | 13106.0 | 0.0142948... |
| 10 | 0.0127532... | -0.156116... | -0.237719... | 10.0 | 1694.0 | 0.0142245... |
| 11 | 0.3700904... | -0.050861... | -0.199255... | 11.0 | 2741.0 | 0.0143691... |
| 12 | 3.7775476... | -0.115059... | 0.1344954... | 12.0 | 11815.0 | 0.0142526... |
| 13 | 0.5987287... | 0.1539585... | -0.223370... | 13.0 | 2144.0 | 0.0141577... |
| 14 | 1.0285914... | -0.178457... | -0.194771... | 14.0 | 2841.0 | 0.0143350... |
| 15 | 1.2972624... | -0.140932... | -0.203484... | 15.0 | 2593.0 | 0.0143330... |
| 16 | 0.0030315... | 0.2563316... | 0.2651014... | 16.0 | 15478.0 | 0.0143101... |

ncount at /entry1/data/PSD_dat/ [mccode.h5 in /Users/torbennielsen/ECD...

events at /entry1/data/k01_events_dat_list_p_x_y_n_id_t/ [mccode.h5 in /Users/torbennielsen/ECD...

IDF xml data, TOF and pixel ID's



Mantid's IDF store geometry information use in TOF analysis

This implies parsing information about:

- where the neutron source is located,
- where the sample is located,
- where each individual detector pixel is located.

- In other words
- Establishing k_i and k_f

How to use: Online documentation



Github McStas wiki pages
Archive - lanl.arXiv.org
Build-in Mantid

The screenshot shows a GitHub repository page for 'McStasMcXtrace / McCode'. The main content is titled 'McStas and Mantid' and discusses the integration between the two software packages. It includes a 'Table of Contents' section with links to various documentation pages like 'Introduction', 'Background and Motivation', and 'The Nexus data format'. On the right side, there is a sidebar with a 'Pages 60' section containing a list of other documentation pages such as 'Access to the code tree', 'Building McStas McXtrace', 'Builds and platforms overview', 'Developer literature', 'Generate Vtess modules via mcstas2vtess', 'How McStas McXtrace works overview', 'How to update the MCPL shipped with McStas McXtrace', 'How to use Eclipse with PyDev', 'mcdisplay pyqtgraph', 'mcdisplay variants table overview', 'mcdisplay webgl', 'mcgui', and 'MCPL related issue on Ubuntu'.

The screenshot shows a 'LoadMcStas v1' dialog box and its corresponding documentation page on lanl.arXiv.org. The dialog box has fields for 'Filename' (with a 'Browse' button), 'OutputWorkspace' (with a 'Browse' button), and buttons for '?', 'Run', and 'Cancel'. Below the dialog, the 'Summary' section states: 'Loads a McStas NeXus file into an workspace.' The 'Properties' table defines the parameters:

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

The documentation page also includes sections for 'Table of Contents' (with links to 'Summary', 'Properties', 'Description', 'McStas compiling and running', 'McStas event data conventions', 'Tested versions', and 'References'), 'Summary' (repeating the dialog's function), and 'Properties' (repeating the table definition). At the bottom, it provides download options ('PDF only'), current browse context ('physics.Ins-det'), and references & citations.

McStas event KEYWORDS



1. 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(....)**”

2. In the McStas instrument file the source must be named “**sourceMantid**”

E.g. “**COMPONENT sourceMantid = Source_simple(....)**”

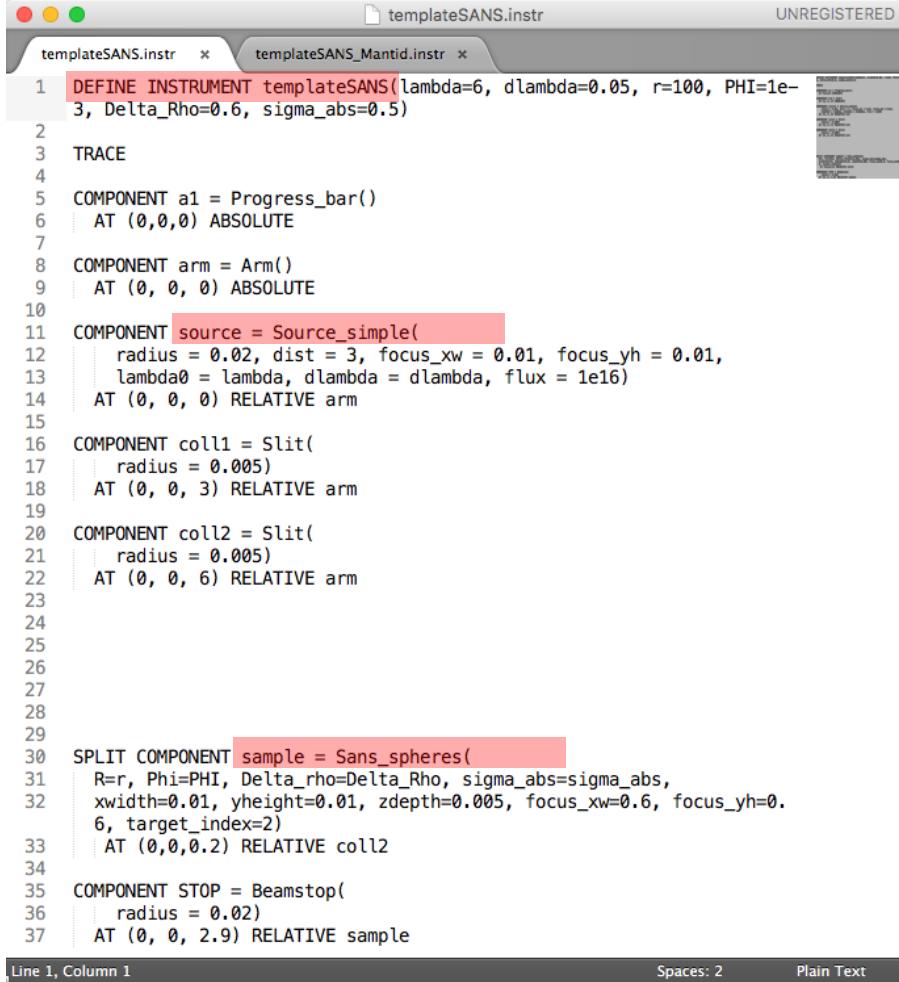
3. In the McStas instrument file the sample must be named “**sampleMantid**”

E.g. “**COMPONENT sampleMantid = Sans_spheres(....)**”

4. In the McStas instrument file the event monitors must be named
“**nD_Mantid_#**”

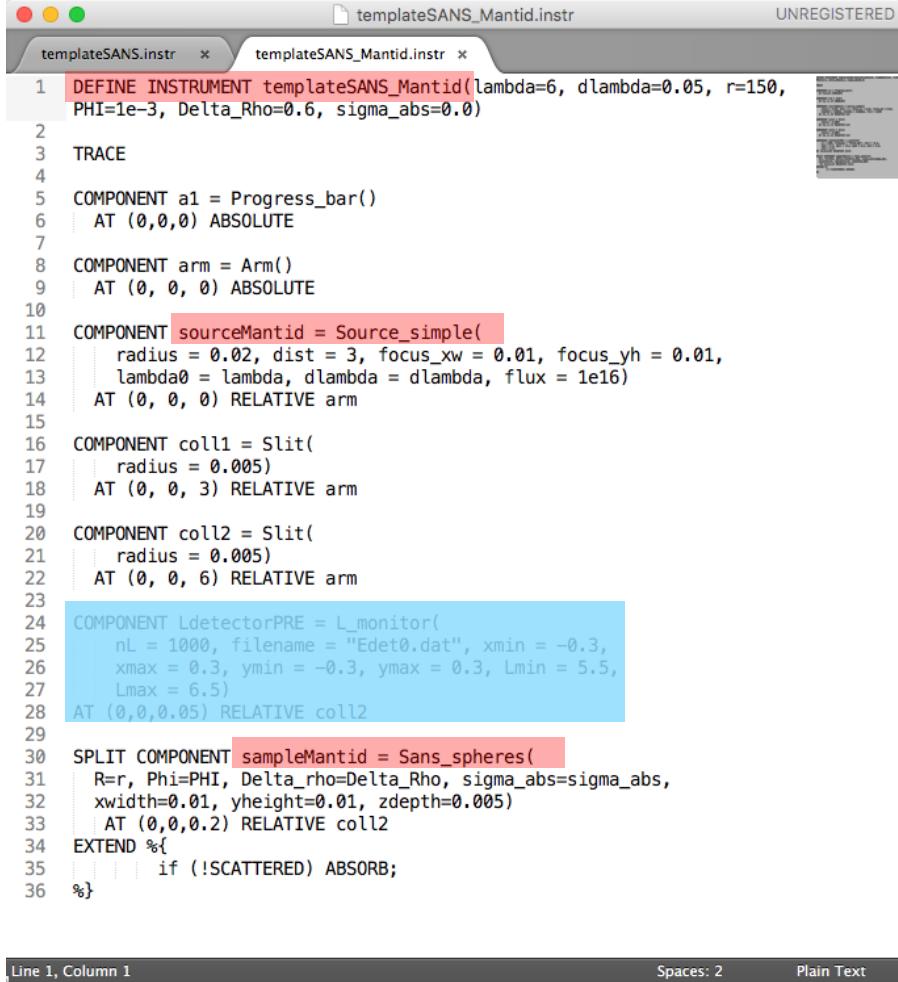
E.g. “**COMPONENT nD_Mantid_1 = Monitor_nD(....)**”

templateSANS.instr vs templateSANS_Mantid.instr



```
templateSANS.instr * 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 * 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



COMPONENT detector = PSD_monitor(UNREGISTERED

COMPONENT detector = PSD_monitor(● 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
```

Line 5, Column 1 Tab Size: 4 Plain Text

COMPONENT detector = PSD_monitor(UNREGISTERED

COMPONENT detector = PSD_monitor(● COMPONENT nD_Mantid_1 = Monitor_nD(●

```
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")
11 AT (0, 0, 3.2) RELATIVE sampleMantid
12
```

Line 10, Column 3 Tab Size: 4 Plain Text

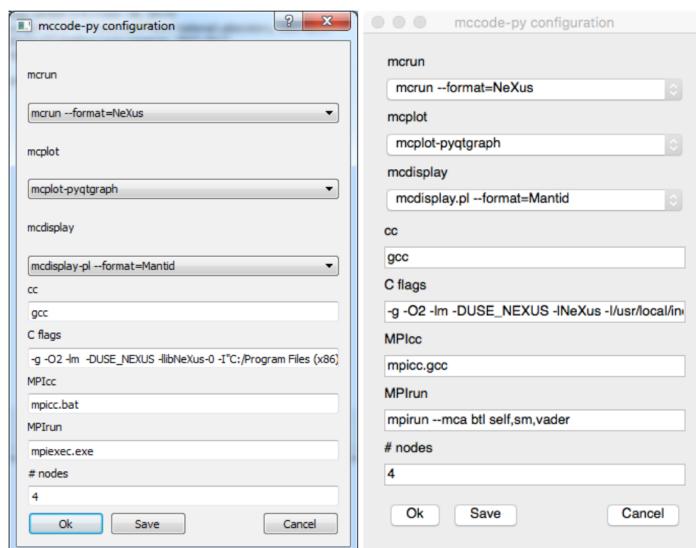
How to run the simulation



McStas GUI

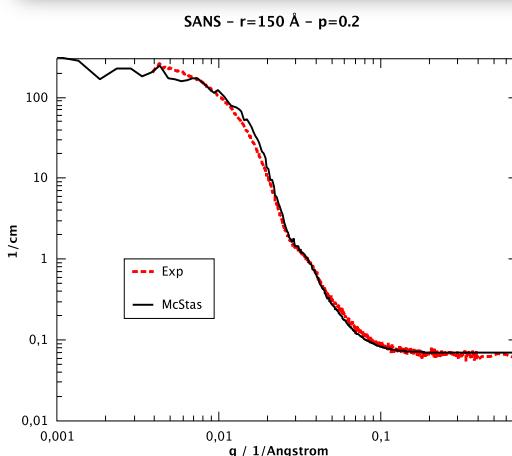
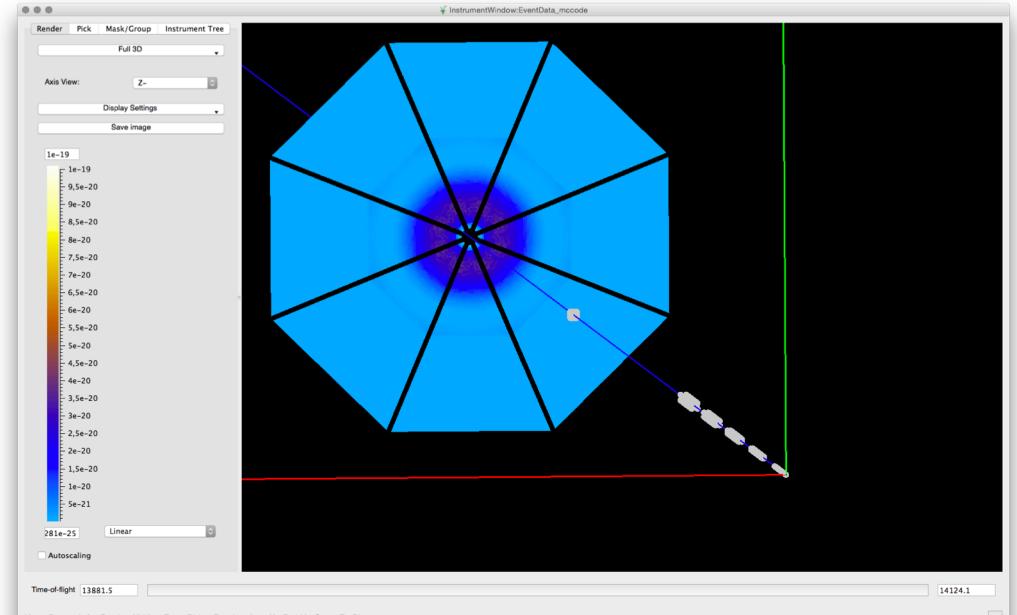
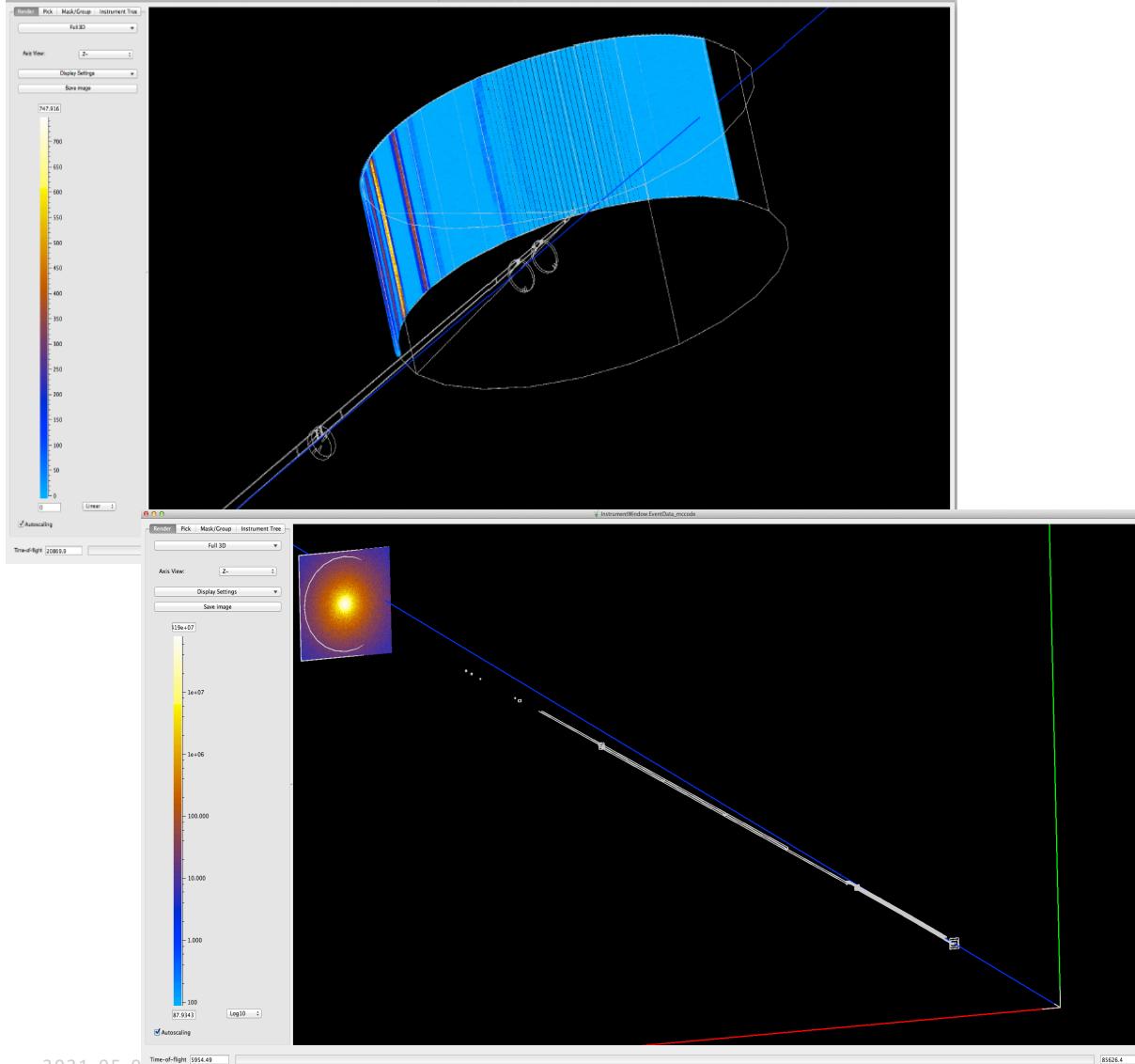
Generating McStas event data for Mantid can be achieved from the McStas GUI `mogui`. Below we show how to setup the simulation on Windows 7, OSX 10.12, and Ubuntu 16.04. For `McStas` we use version 2.4.1. For `Mantid` use version 3.4 or later.

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 -I "C:/Program Files (x86)/Nexus Data Format/include/nexus" -L "C:/Program Files (x86)/Nexus Data Format/lib/nexus"`
 - OS X: `-g -O2 -lm -DUSE_NEXUS -lNeXus -I/usr/local/include/nexus`
 - Ubuntu: `-g -O2 -lm -DUSE_NEXUS -lNeXus`

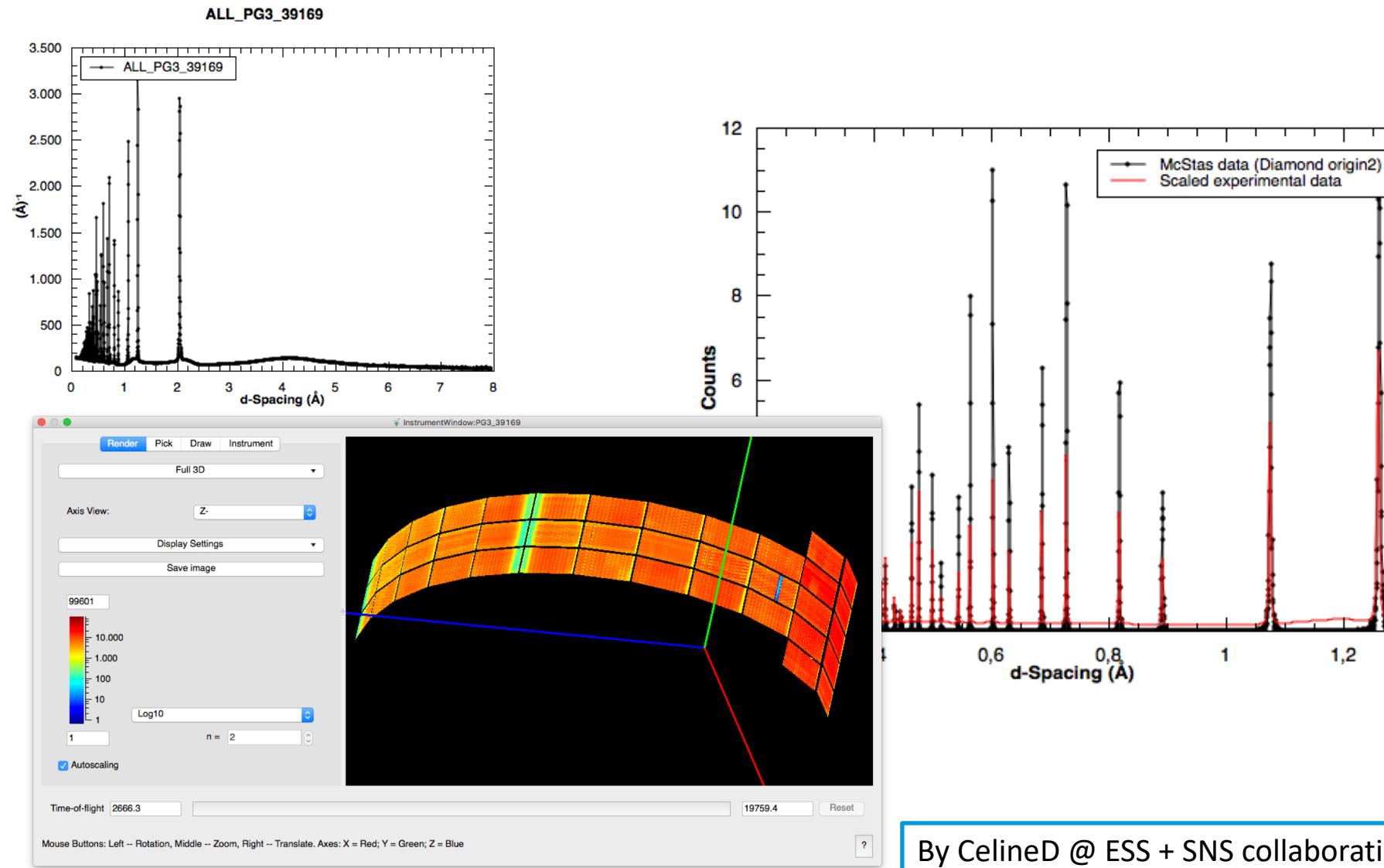


Wiki page on GitHub

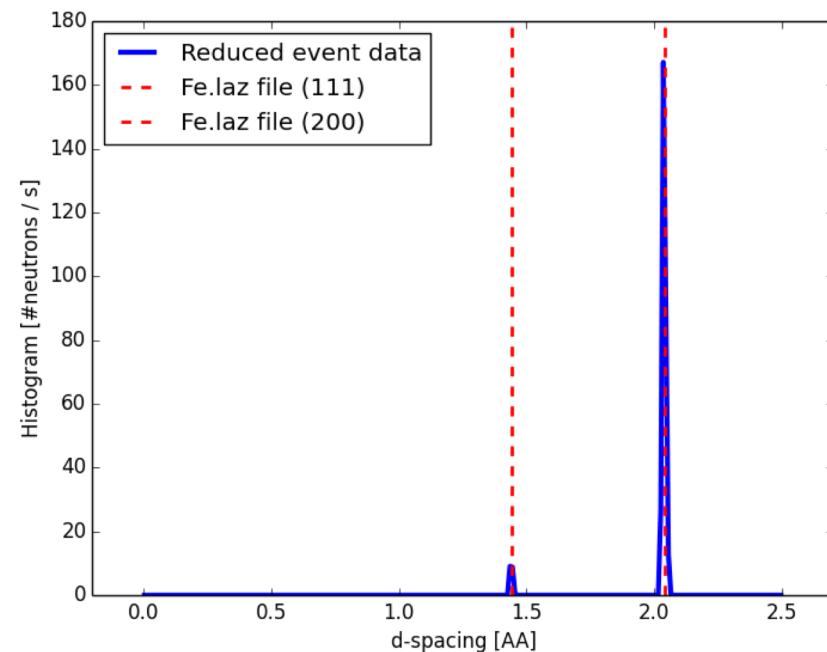
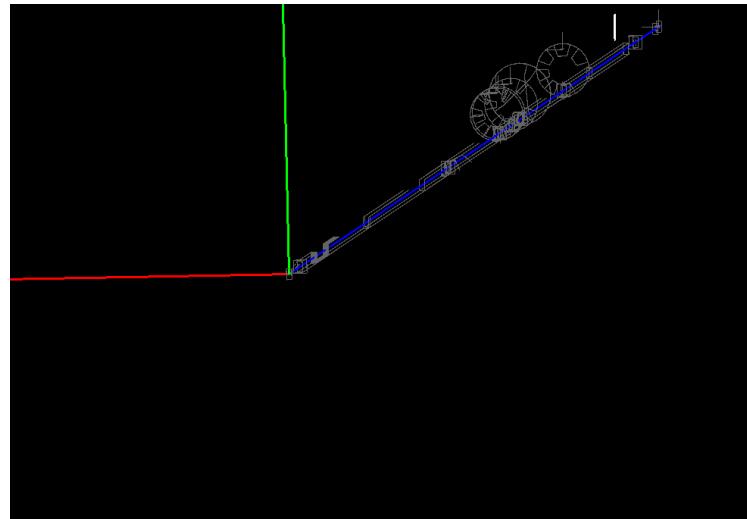
Examples of detectors: IN5 (ILL), LoKI (ESS), SANS2D (ISIS)



McStas and experimental data: POWGEN



Example – V20 – Powder diffraction Fe



Summary



1. Easier to post-process (event) data than making a completely new McStas simulation / component
 - E.g.:
 - Rebin as needed
 - Sum selective over detectors
 - Use already developed methods in Mantid
2. Mantid process McStas data as if they were equal to experimental twin data

Presentation headline

Sub-headline to strengthen the headline above



Presentation headline

Sub-headline to strengthen the headline above

