



**EUROPEAN  
SPALLATION  
SOURCE**



# DMSC STAP

## Updates from DRAM

...

TORBEN NIELSEN

October 2022

# Agenda



1 DRAM

---

2 Data reduction updates

---

3 Modelling updates

---

4 Data analysis updates

---

---

---

---

---

---

---

# DRAM

## Data Reduction, Analysis and Modelling



Simon Heybrock



Neil Vaytet



Jan-Lukas Wynen



Piotr Rozyczko



Simon Ward



Andrew Sazonov



Peter Willendrup



Mads Bertelsen



Justin Bergmann

## Scope

The DRAM group is responsible for providing the data reduction, analysis and modelling software for all instruments at ESS.

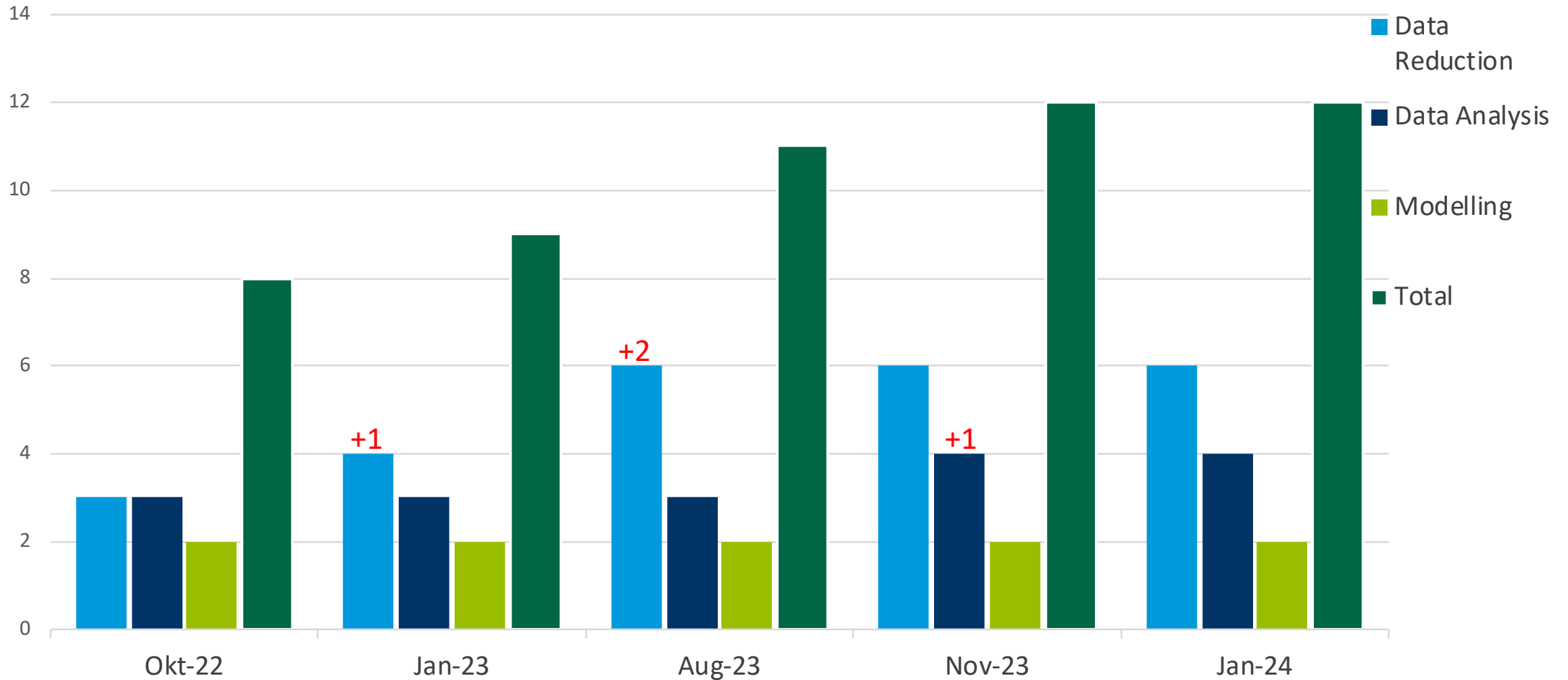
- ❑ 3 teams (9 persons)
  - ❑ Data Reduction (scipp)
  - ❑ Data Analysis (SasView, SpinW, EasyScience, external collaborations )
  - ❑ Modelling (McStas++, pan-learning.org, NMX)



# DRAM - staff profile

P0/P6 – spring 2022

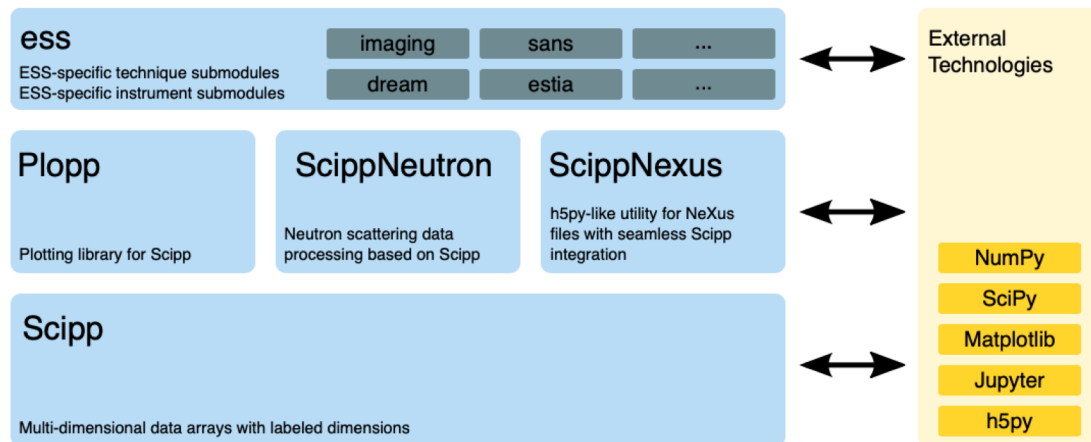
S





# scipp

scipp, scippneutron, scippnexus, plopp and ess



- ❑ Close collaboration between scipp and IDS team for defining data reduction workflows in scipp
- ❑ Joint meetings, pair-programming, and lively discussion on a dedicated Slack channel.

❑ scipp continues to be developed with input and requirements from instrument data scientists.

❑ Good progress

❑ Lots of features added & issues fixed

❑ From: <https://scipp.github.io>



# scipp

Documentation on <https://scipp.github.io>

The screenshot shows the scipp NeXus documentation website. The header includes the scipp NeXus logo, a search bar, and navigation links for 'Related projects', '0.3.0 (latest)', and 'Contents'. The main content area is titled 'Quick Start Guide' and 'Overview'. The overview text states: 'The NeXus Data Format is typically used to structure HDF5 files. An HDF5 file is a container for datasets and groups. Groups are folder-like and work like Python dictionaries. Datasets work like NumPy arrays. In addition, groups and datasets have a dictionary of attributes.' The left sidebar contains a 'GETTING STARTED' section with links for 'Installation' and 'Quick Start Guide'.

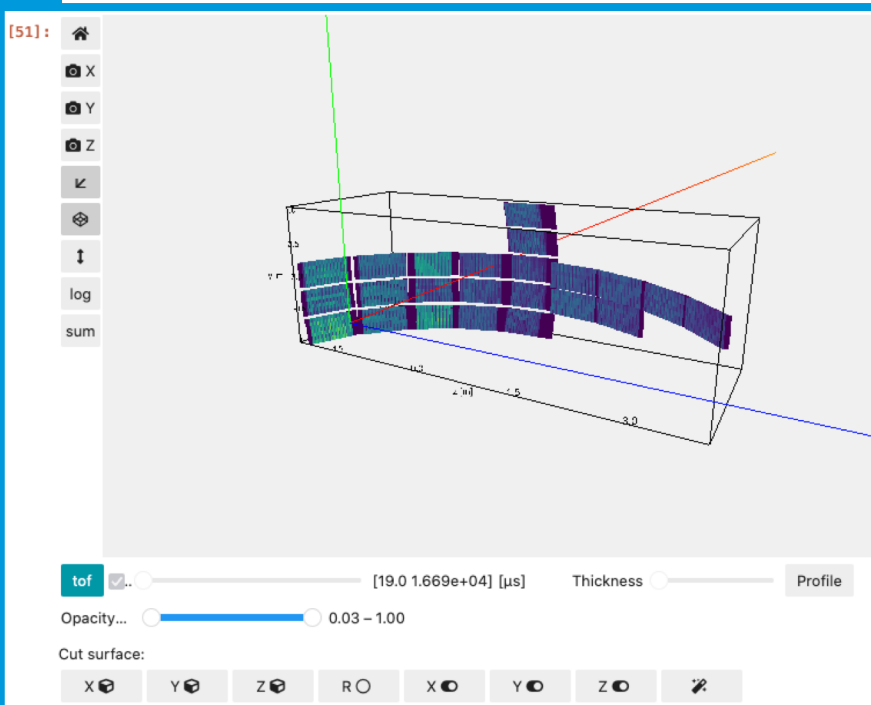
- ❑ scippnexus: An h5py-like utility for NeXus files with seamless Scipp integration

The screenshot shows the plopp documentation website. The header includes the plopp logo, a search bar, and navigation links for 'Related projects', '0.2.2 (latest)', and 'Contents'. The main content area is titled 'Installation' and 'Using pip'. The text states: 'You can install from pip using' followed by a code block: `pip install plopp[scipp]`. Below the code block, it says: 'This will install both plopp and scipp which is required to use plopp. If you already have scipp installed, you can leave the [scipp] part out:'. The left sidebar contains a 'USER GUIDE' section with links for 'Line plot', 'Image plot', and 'Slicer plot'.

- ❑ plopp: plotting library for scipp

# scippnexus

- POWGEN data
- PG3\_4844\_event.nxs



```
[39]: from scippnexus import NXdetector, NXdata
      f.entry.instrument[NXdetector]

[39]: {'bank102': <NXdetector "/entry/instrument/bank102">,
      'bank103': <NXdetector "/entry/instrument/bank103">,
      'bank104': <NXdetector "/entry/instrument/bank104">,
      'bank105': <NXdetector "/entry/instrument/bank105">,
      'bank106': <NXdetector "/entry/instrument/bank106">,
      'bank123': <NXdetector "/entry/instrument/bank123">,
      'bank124': <NXdetector "/entry/instrument/bank124">,
      'bank143': <NXdetector "/entry/instrument/bank143">,
      'bank144': <NXdetector "/entry/instrument/bank144">,
      'bank164': <NXdetector "/entry/instrument/bank164">,
      'bank184': <NXdetector "/entry/instrument/bank184">,
      'bank22': <NXdetector "/entry/instrument/bank22">,
      'bank23': <NXdetector "/entry/instrument/bank23">,
      'bank24': <NXdetector "/entry/instrument/bank24">,
      'bank42': <NXdetector "/entry/instrument/bank42">,
      'bank43': <NXdetector "/entry/instrument/bank43">,
      'bank44': <NXdetector "/entry/instrument/bank44">,
      'bank62': <NXdetector "/entry/instrument/bank62">,
      'bank63': <NXdetector "/entry/instrument/bank63">,
      'bank64': <NXdetector "/entry/instrument/bank64">,
      'bank82': <NXdetector "/entry/instrument/bank82">,
      'bank83': <NXdetector "/entry/instrument/bank83">,
      'bank84': <NXdetector "/entry/instrument/bank84">}
```

```
[40]: f.entry.instrument[NXdetector]['bank102'] [()]

[40]: scipp.DataArray (8.75 MB)

  Dimensions:      (x_pixel_offset: 154, y_pixel_offset: 7)
  Coordinates:
  azimuthal_angle  (x_pixel_offset, y_pixel_offset) float32 rad -0.3123652, -0.2958419, ..., -0.2258334E
  distance          (x_pixel_offset, y_pixel_offset) float32 m 3.2983806, 3.2987897, ..., 3.5141585, 3.5
  local_name        () string F2
  pixel_id          (x_pixel_offset, y_pixel_offset) int32 127500, 127501, ..., 128576, 128577
  polar_angle      (x_pixel_offset, y_pixel_offset) float32 rad 1.4804635, 1.4808731, ..., 1.266226, 1.26
  total_counts      () int32 456601
  x_pixel_offset    (x_pixel_offset) float32 m -0.3825, -0.3775, ..., 0.3775, 0.3825
  y_pixel_offset    (y_pixel_offset) float32 m -0.1629, -0.1086, ..., 0.1086, 0.1629

  Data:
  (x_pixel_offset, y_pixel_offset) DataArrayView binned data [len=0, len=0, ..., len=0, len=C
```

```
[41]: f.entry.instrument[NXdetector]['bank102']['event_time_offset'] [()]

[41]: scipp.Variable (1.74 MB)

(event: 456601) float32 μs 6027.5, 7976.1, ..., 10003.6, 3044.1
```

# plopp

new

ess  
ESS-specific technique submodules  
ESS-specific instrument submodules

imaging sans ...  
dream estia ...

Plopp  
Plotting library for Scipp

ScippNeutron  
Neutron scattering data processing based on Scipp

ScippNexus  
h5py-like utility for NeXus files with seamless Scipp integration

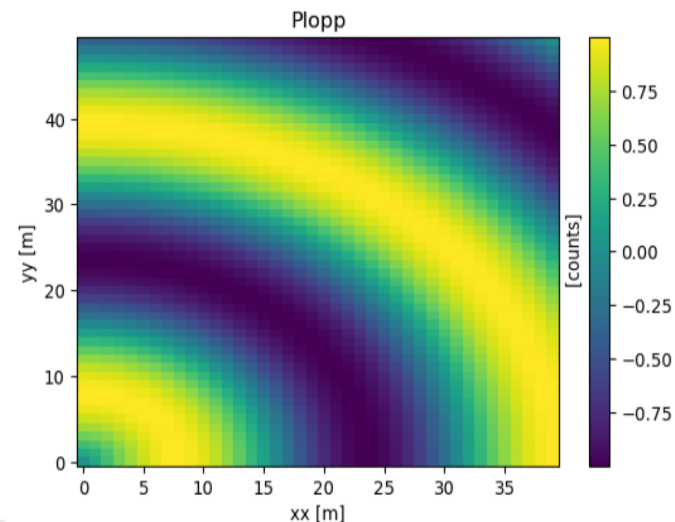
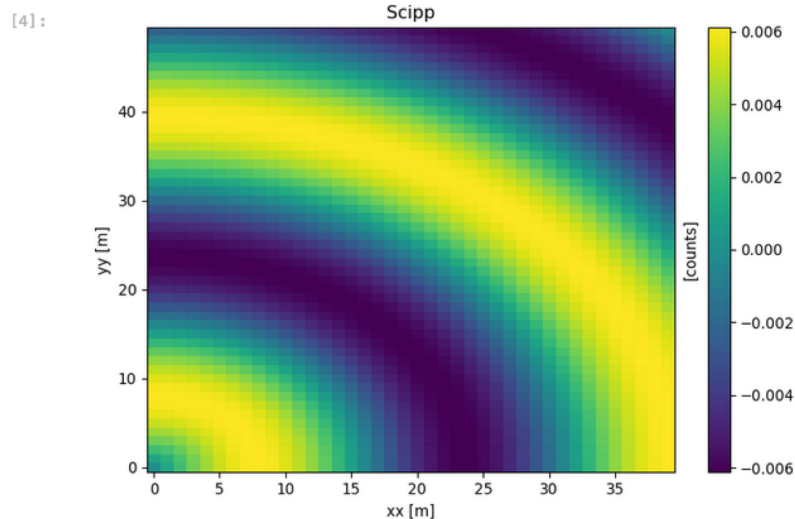
Scipp  
Multi-dimensional data arrays with labeled dimensions

## Issues with automatic resampling

```
[3]: print(da.min().value, da.max().value)
-0.9999902065507035 0.9999886166646338

[4]: s = sc.plot(da, title='Scipp')
p = pp.plot(da, title='Plopp')

ipw.HBox([s._to_widget(), p._to_widget()])
```



## Scipp vs Plopp

This notebook aims to illustrate the differences between the current (old) Scipp plotting, and the new Plopp package.

## Background

See [ADR0013](#) for more details.

The current plotting functionality automatically resamples 2-D (or higher) data as well as binned data of any dimension. There were three main drivers/motivations for this:

- We frequently deal with data with “long” dimensions, such as “pixel” or “spectrum” dimensions, which may exceed a million.
- We often deal with “ragged” data, i.e., data with 2-D coordinates that originate after coordinate transformations.
- We frequently plot binned data. Bins are often much larger than the target plot resolution, so histogramming must be done. This is combined with “long” dimensions and potentially high resolutions along other dimensions (exceeding 1000 or even 10000).

In the past we have therefore taken it as a given that resampling is a non-optional requirement. However, we have since then experienced a multitude of problems with this, ranging from simple bugs to complex and hard to maintain code. More importantly, the resulting plots are unpredictable and not faithful to the actual data.

One of the most compelling conclusions was:

**"Scientifically any plot generated with resampling is not usable, e.g., for publications."**

## New requirements

From [scipp#2475](#)

1. Faithful representation of data
2. Not tied to plotting/widget library
3. Updatable data
4. Both interactive and static figures
5. Customizable figures
6. Plots should still be visible in saved notebooks
7. API for adding custom tools with callback (or similar)

old

ess-notebooks

ess  
imaging sans ...  
dream estia ...

scippneutron

scipp

# scipp / IDS

## Data reduction workflow - - see [ess package](#)

Some of the highlights include the `ess` package being extended for powder diffraction support such that reduction workflows are now available directly for the users on the online documentation pages.

- Data reduction for POWGEN
- This notebook shows a basic reduction workflow for powder diffraction for the SNS POWGEN instrument. It serves mainly to develop and present routines for powder diffraction and will eventually be removed in favor of a workflow for DREAM at ESS.
- See [ess docs on io pages](#)

The screenshot shows the ESS documentation page for the 'Load data' section. The left sidebar contains a navigation menu with categories: GETTING STARTED (Installation), INSTRUMENTS (Amor, Loki, External Instruments, POWGEN, Reference), TECHNIQUES (Diffraction, Reflectometry, SANS, Wavelength frame multiplication), and UTILITIES (Modules). The main content area is titled 'Load data' and includes a search bar, a note about the `load` function, and a code cell showing the execution of `powgen.load`. The output displays the dimensions and coordinates of the loaded data array.

```
[3]: sample = powgen.load(powgen.data.sample_file())
Downloading file 'PG3_4844_event.nxs' from 'https://public.esss.dk/groups/scipp/ess/'
[4]: sample
[4]: scipp.DataArray (291.86 MB)
  Dimensions: (spectrum: 24794, tof: 1)
  Coordinates:
  detector_info () Dataset <scipp.Dataset> Dimensions: Sizes[deteci
  position (spectrum) vector3 m [1.17451004 -1.01106149 -2.03796699], [
  sample_position () vector3 m [0.0 0.0]
  source_position () vector3 m [0.0. -60.]
  spectrum (spectrum) int32 1, 2, ..., 24793, 24794
  tof (tof [bin-edge]) float64 μs 19.0, 1.669e+04
  Data:
  (spectrum, tof) DataArrayView binned data [len=0, len=0, ..., len=0, len=C
  Attributes: (56)
```

The screenshot shows the ESS documentation page for a notebook workflow. The left sidebar is identical to the previous screenshot. The main content area shows a code cell with the following code:

```
[19]: all_spectra = diffraction.normalize_by_vanadium(
      sample_dspacing.bins.concat('spectrum'),
      vanadium=vana_dspacing.bins.concat('spectrum'),
      edges=dspacing_edges,
)
[20]: sc.histogram(all_spectra, bins=dspacing_edges).plot()
[20]:
```

The output of the code cell is a plot of the histogram. The x-axis is labeled 'dspacing [Å]' and ranges from 0.0 to 2.0. The y-axis is labeled '[dimensionless]' and ranges from 0 to 8. The plot shows a series of peaks, with the most prominent peak at approximately 1.4 Å.



# scipp

## A bunch of releases

**scipp: #10:** v0.16.4 - v0.16.3 - v0.16.4 - v0.16.2 - v0.16.1 - v0.16.0 - v0.15.1 - v0.15.0 - v0.14.1 - v0.14.0 - v0.13.1 (April 01 2022)

**scippneutron: #9:** v0.9.4 - v0.9.3 - v0.9.2 - v0.9.1 - v0.9.0 - v0.9.4 - v0.8.0 - v0.7.0 - v0.6.0 - v0.5.3 (30 March 2022)

**ess: #7:** v0.9.0 - v0.8.0 - v0.9.0 - v0.7.1 - v0.7.0 - v0.6.0 - v0.5.0 - v0.4.0 (Feb 28 2022)

**scippnexus: #21:** v0.3.2 - v0.3.1 - v0.3.2 - v0.3.0 - v0.2.1 - v0.2.0 - v0.1.8 - v0.1.7 - v0.1.6 - v0.1.5 - v0.1.4 - v0.1.3 - v0.1.2 - v0.1.1 - v0.1.0 - v0.0.9 - v0.0.8 - v0.0.7 - v0.0.6 - v0.0.5 - v0.0.4 (31 March 2022)

**plopp: #8: :** v0.3.0 - v0.2.2 - v0.2.1 v0.2.0 – v0.1 – v0.03 – v0.0.2 – v0.0.1 (25 Aug 2022)

**mpltoolbox: #7:** v0.1 – v0.0.8 – v0.0.7 – v0.0.6 – v0.0.5 – v0.0.4 – v0.0.3 (11 Jul 2022)

# 62 - releases

# Modelling



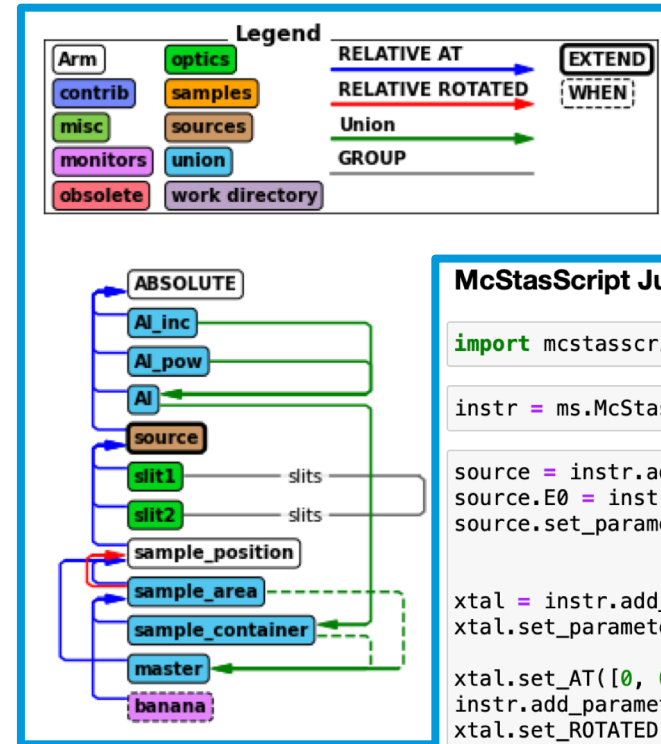
## McStas Python API with tools, diagrams and widgets

### New developments

- Diagrams of instrument objects
- Improved error checking
- Compatibility with McStas 3.X
- Interactive quiz system in jupyter notebooks
- Dynamic splits of instruments using MCPL

### In collaboration with PaNOSC

- Instrument database with API



### McStasScript Jupyter notebook

```
import mcstas as ms

instr = ms.McStas_instr("demo")

source = instr.add_component("source", "Source_simple")
source.E0 = instr.add_parameter("energy", comment="[meV]")
source.set_parameters(xwidth=0.12, yheight=0.03, dE="0.1*energy",
                    dist=2.0, focus_xw=0.01, focus_yh=0.01)

xtal = instr.add_component("crystal", "Single_crystal")
xtal.set_parameters(radius=0.005, reflections='YBaCu0.lau',
                  ax=3.8186, by=3.886, cz=11.6777, mosaic=30)
xtal.set_AT([0, 0, 2.0], RELATIVE=source)
instr.add_parameter("sample_rotation", value=0, comment="[deg]")
xtal.set_ROTATED([0, "sample_rotation", 0], RELATIVE=source)

instr.set_parameters(energy=5.0, sample_rotation=37.0)
instr.show_parameters()

energy          = 5.0    // [meV]
sample_rotation = 37.0  // [deg]

data = instr.backengine()
```



# Modelling: workshops and conferences



## Events attended since april

Participated in following events

- DMSC McStas workshop
- McStas training days (May-July 2022)
  
- ICNS Argentina  
Neutron Instrumentation and Innovation award for Mads Bertelsen
- PaNOSC Summer school
- ESS-ILL User meeting
- PaNOSC WP 5 sprint (online)



# McStasScript



## Modelling overview

**mcstasscript: #10:** 0.0.51, 0.0.50, 0.0.49, 0.0.48, 0.0.46, 0.0.44, 0.0.43, 0.0.42, 0.0.41, 0.0.40  
(March 22 2022)

**guide\_bot: #10:** 0.0.27 - 0.0.26 - 0.0.25 - 0.0.24 - 0.0.23 - 0.0.22 - 0.0.21 - 0.0.20 - 0.0.19 - 0.0.18 -  
0.0.16 - 0.0.15 (Feb 16 2022)

# Data Analysis

## Updates since last STAP



### For easyScience core:

- extension for multiple datasets/fit functions fitting
- IO rework of easyCore for speedup in XML encoding, and hence a more responsive GUI.
- Started work on asymmetric peaks fitting in easyDiffraction

### For easyDiffraction:

- easyDiffraction v0.8.4 was released
- polarized support in the GUI
- changing how experimental data are handled
- starting working on file-less interface to CrysFML
- Improved CrysPy interface

### For easyReflectometry:

- easyReflectometry v0.0.5 was released
- surfactant Layer in GUI was added
- general bug fixes and minor improvements

# easyDiffraction 0.8.4

## Polarization support for CW using CrysPy

easyDiffraction	CW	TOF
un-polarized	✓ CrysFML ✓ CrysPy ✓ GSAS-II	✓ CrysPy
polarized	✓ CrysPy	

Fe3O4

Short description: neutrons, powder, constant wavelength, polarised, 672@LLB

Phases: Fe3O4

Experiments: PoINPDST

Modified: 14.07.2022 16:47

Basic controls

- Get started
- Create a new project
- Continue without a project
- Open an existing project
- Save project as...

Structure view

Basic controls

- Structural phases
- No Samples Added/Loaded
- Add new phase from CIF
- Symmetry and cell parameters
- Atoms, atomic coordinates and occupancies
- Atomic displacement parameters
- Magnetic susceptibility parameters

Fitting calculations.cif

Basic controls

- Parameters
- Calculation
  - Engine: CrysPy
- Minimization
  - Minimizer: lmfit
  - Method: least\_squares
- Fitting constraints
- Fitting components

Y-axis: I\_meas, I\_calc, I\_fit

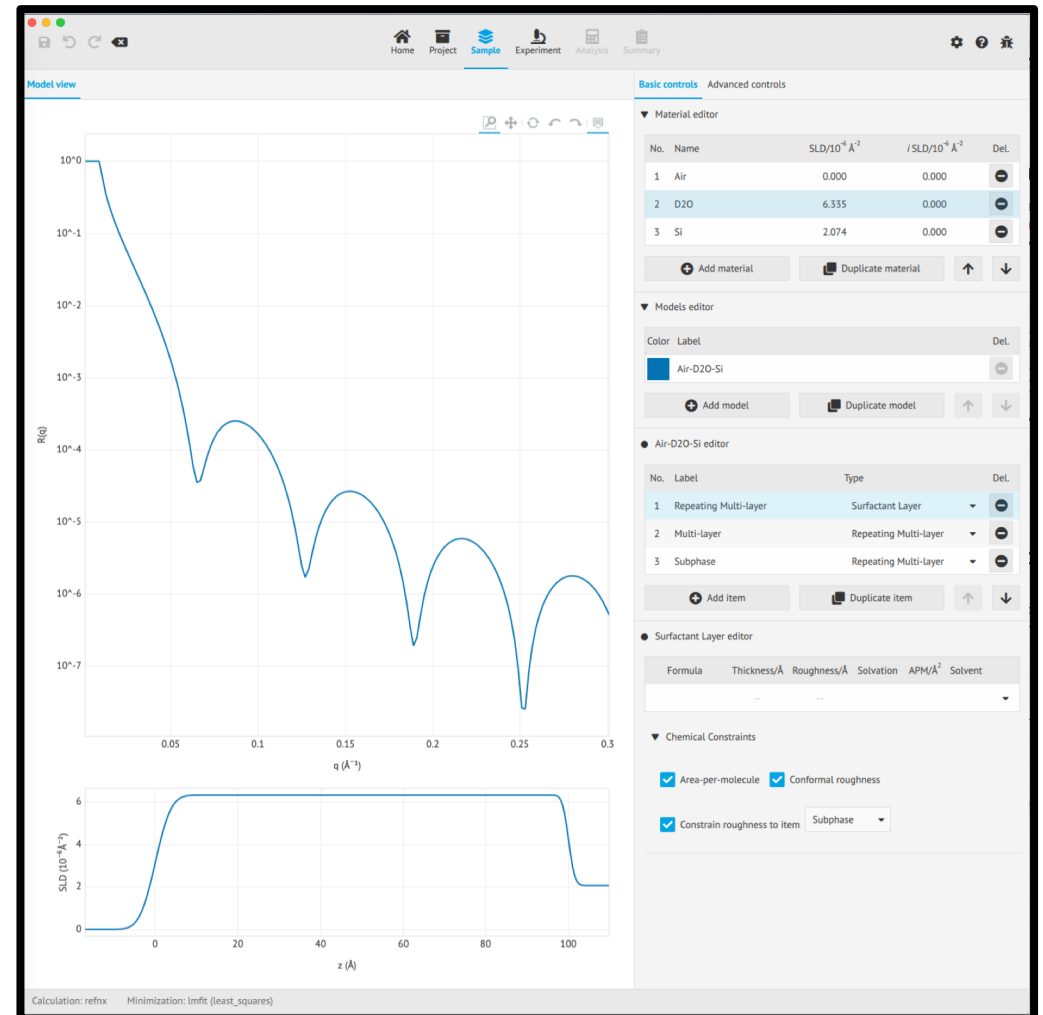
X-axis: 2θ (deg)

# easyReflectometry

v0.0.5



- ☐ surfactant Layer in GUI was added
- ☐ general bug fixes and minor improvements
- ☐ Python notebooks



# Documentation

<https://docs.easydiffraction.org/lib/>

<https://docs.easyreflectometry.org>



The screenshot shows the 'easydiffraction' documentation page. The main content is titled 'Create calculator' and includes the following code blocks:

```
calculator = job.interface # CrysPy is default
print(f"Current calculator engine: {calculator.current_interface_name}")

Current calculator engine: CrysPy

print(f"Available calculator engines: {calculator.available_interfaces}")
print(f"Available calculators for CW: {calculator.interface_compatibility('Npowder1DCW')}
```

Below the code is the text 'Analysis' and 'Calculate the profile using the calculator we defined previously.' followed by another code block:

```
x_data = np.linspace(20, 170, 500)
_ = job.create_simulation(x_data)
y_data = np.array(data['sim_NaCl'])

fig = figure(width=FIGURE_WIDTH, height=FIGURE_HEIGHT)
fig.line(x_data, y_data, legend_label='CW Simulation', color='orangered', line_width=2)
show(fig)
```

The bottom part of the screenshot shows a plot of the simulation profile. The x-axis is labeled from 20 to 160, and the y-axis is labeled from 0 to 2000. The plot shows a series of peaks, with the most prominent one at approximately x=115. The legend indicates 'CW Simulation'.

The screenshot shows the 'EasyReflectometry' documentation page. The main content is titled 'Reading in experimental data' and includes the following code blocks:

```
[3]: data = load('_static/example.ort')
```

The text explains that the function will load the file into a `scipp Dataset` object. Below this is another code block:

```
[4]: data
[4]: scipp.Dataset (12.76 KB)
    Dimensions: (Qz_0: 408)
    Coordinates:
      Qz_0 (Qz_0) float64 1/Å 0.008, 0.008, ..., 0.461, 0.466
           σ = 0.000, 0.000, ..., 0.009, 0.010
    Data:
      R_0 (Qz_0) float64 1 0.710, 0.862, ..., 3.856e-07, 3.834e-07
           σ = 0.085, 0.112, ..., 1.761e-07, 1.885e-07
    Attributes:
      orso_header () PyObject ('data_source': {'owner': {'name': 'Andrew Nelson', 'affiliat
```

Below the code is the text 'EasyReflectometry also includes a custom plotting function for the data.' followed by another code block:

```
[5]: p1ot(data)
[5]:
```

The bottom part of the screenshot shows a plot of the data. The x-axis is labeled 'Qz\_0 [1/Å]' and ranges from 0.0 to 0.4. The y-axis is labeled '[dimensionless]' and ranges from 10<sup>0</sup> to 10<sup>-6</sup>. The plot shows a series of peaks, with the most prominent one at approximately x=0.05. The legend indicates 'R\_0'.

# Collaborations – getting people onboard



**EasyReflectometry**

Search the docs ...

**CONTENTS:**

- Installation
- Usage
- Libraries
- Calculators & Optimisation
- Tutorials
- Contributing**
- Credits
- API

## Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

## Types of Contributions

### Report Bugs

Report bugs at <https://github.com/easyScience/EasyReflectometryLib/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

**easydiffraction** Home Features Contact Docs

## Get in touch

**General questions**

For general questions or comments, please contact us at [support@easydiffraction.org](mailto:support@easydiffraction.org), or fill out the form.

**Issues and new features**

For bug reports and feature requests, please use [GitHub Issue Tracker](#) instead (free registration required).

Name

Email

Subject

Message

Send Message

This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.

**SasView** ABOUT LINKS & DOWNLOADS DOCUMENTATION CONTENT HELP

## Why get involved

No matter whether you are a student, numerous studies have shown that there are many benefits to joining collaborative software projects. See [here](#) for more information on why! But in a nutshell, you will not only be helping to develop a piece of software that you (and perhaps colleagues around you) rely on, but you will be developing your own skills set too.

And if you are just starting out using small-angle scattering, contributing to SasView would be a great way to meet people with a wealth of experience! We offer discounted consultancy rates to contributors. (In case you are wondering, that is a joke!)

## Ways to contribute

There are many ways that you can contribute to SasView, and not all of them require coding!

We need contributors that can...

- Respond to queries posted to [help@sasview.org](mailto:help@sasview.org);
- Teach others how to get the best from SasView;

**scipp**

Search the docs ...

Related projects 0.16.4 (latest)

## Contributing to scipp

### Overview

Contributions, bug reports, and ideas are always welcome. The following section outlines the scope of scipp. If in doubt whether a feature falls within the scope of scipp please [ask on github](#) before implementing functionality, to reduce the risk of rejected pull requests. Asking and discussing first is generally always a good idea, since our road map is not very mature at this point.

**GETTING STARTED**

- What is Scipp?
- Installation
- Quick start
- Frequently Asked Questions



# Finish presentation