EUROPEAN
SPALLATION
SOURCE

The Data Reduction, Analysis, and Modelling Group (DRAM)

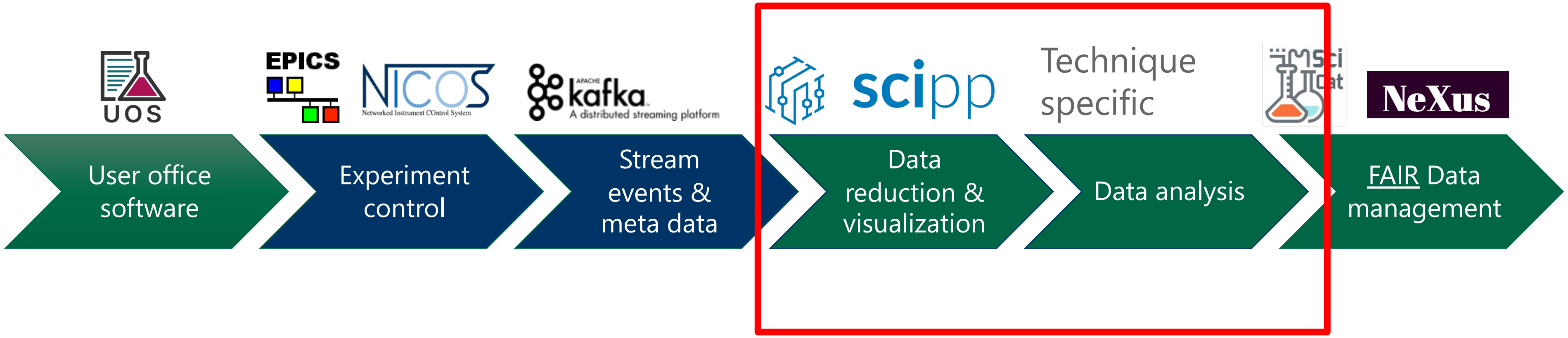
Science away day

TORBEN NIELSEN

Spring 2024

DMSC Scope for scientific computing

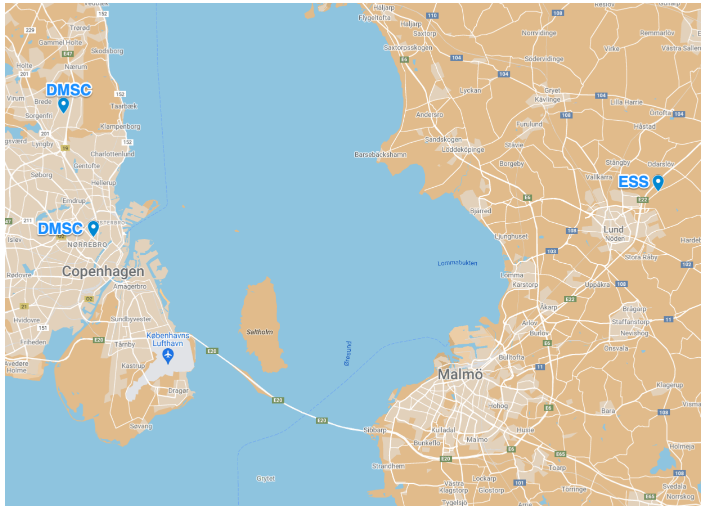
Support users with scientific computing at modern open science facility



UOS
EPICS
NICOS
Networked Instrument COntrol System
APACHE kafka
A distributed streaming platform
scipp
Technique specific
SciCat
NeXus

User office software → Experiment control → Stream events & meta data → Data reduction & visualization → Data analysis → FAIR Data management

DRAM: Data Reduction, Analysis and Modelling



Data Management & Software Centre (DMSC)

DRAM

Data Reduction, Analysis and Modelling - Staff

Scipp

Simon Heybrock

Neil Vaytet

Jan-Lukas Wynen

Sunyoung Yoo

Johannes Kasimir

Mridul Seth

SWAT

Piotr Rozyczko

Andrew Sazonov

Andreas Pedersen

Henrik Jacobsen

Christian Vedel

Modelling

Peter Willendrup

Mads Bertelsen

Thomas Kittlemann

➢3 teams (14+ persons)

1. Data Reduction (scipp)
2. Data Analysis (SasView, SpinW, EasyScience, external collaborations)
3. Modelling (McStas++, pan-learning.org, Detector Group)

Scope

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

Data reduction
-
scipp

scipp

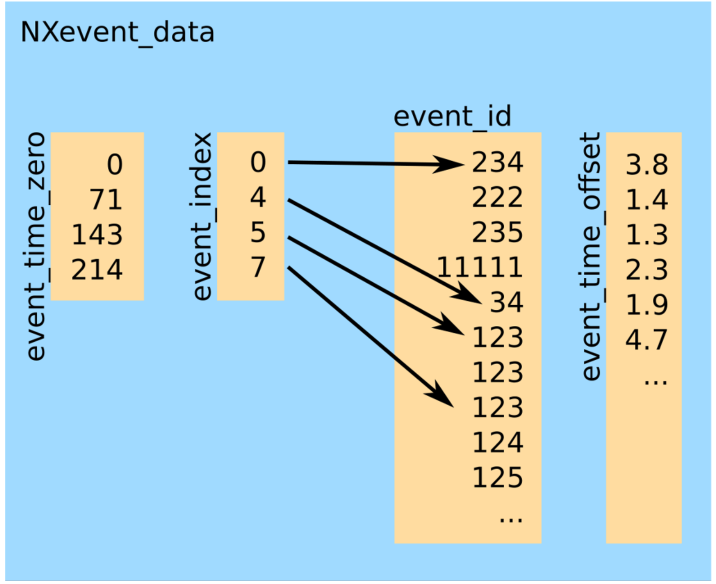
python™

jupyterhub

Data Reduction: convert detector data to physical data

(pixel position, detection time) → (λ, energy, θ, d-spacing, intensity, …)

Event data in NeXus file



NXevent_data

event_time_zero: 0, 71, 143, 214

event_index: 0, 4, 5, 7

event_id: 234, 222, 235, 11111, 34, 123, 123, 123, 124, 125, …

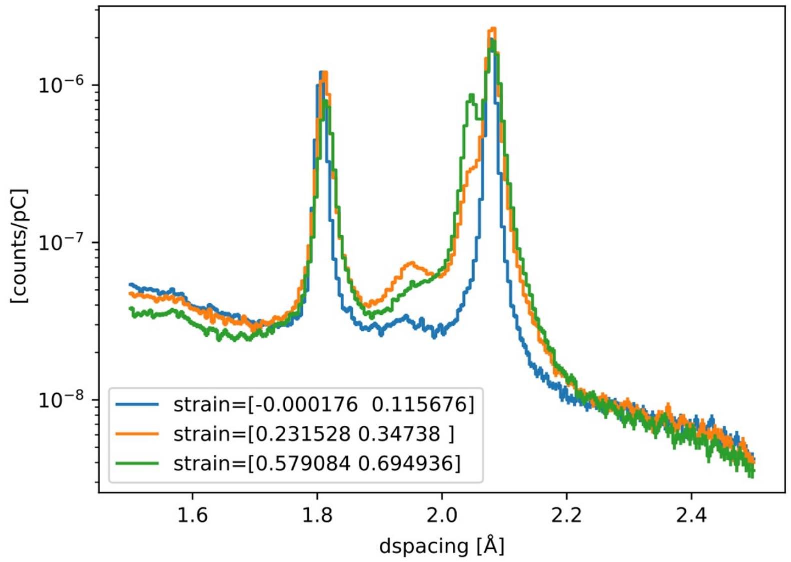
event_time_offset: 3.8, 1.4, 1.3, 2.3, 1.9, 4.7, …

scipp

scipp.github.io

GitHub

Output with physical units

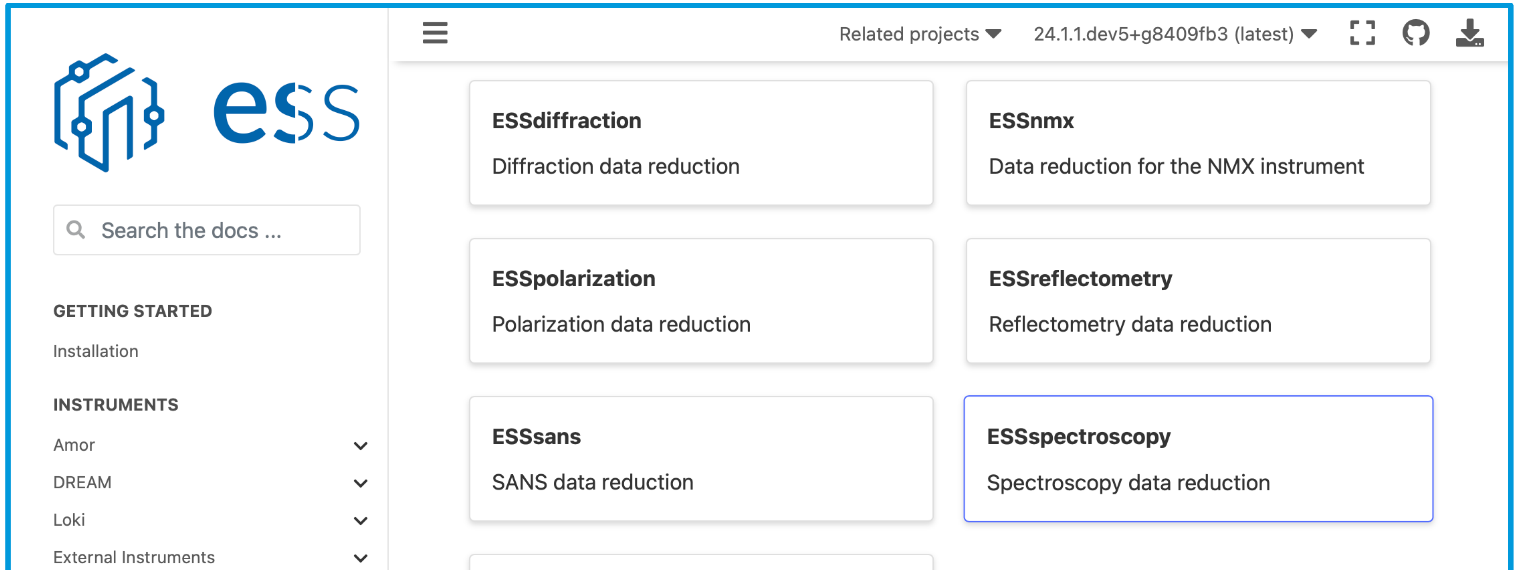


Data reduction workflows for ESS

On-line documentation // Getting started

❏ scipp.github.io/ess

GitHub



ess

Related projects ▾ 24.1.1.dev5+g8409fb3 (latest) ▾

Search the docs ...

GETTING STARTED

Installation

INSTRUMENTS

- Amor
- DREAM
- Loki
- External Instruments

TECHNIQUES

- Diffraction
- Reflectometry
- SANS
- Wavelength frame multiplication

ESSdiffraction
Diffraction data reduction

ESSnmx
Data reduction for the NMX instrument

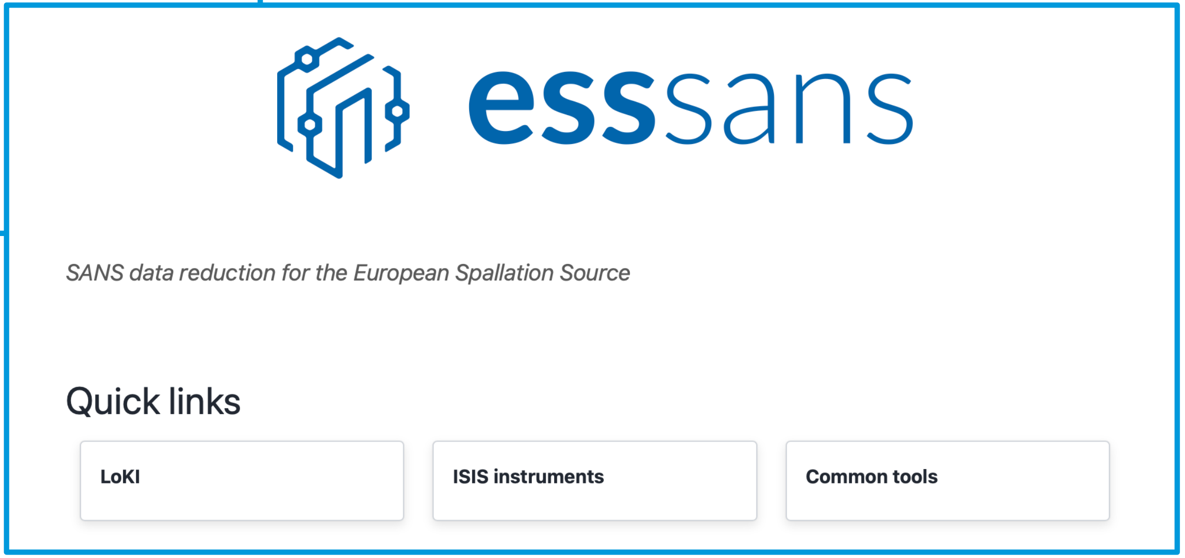
ESSpolarization
Polarization data reduction

ESSreflectometry
Reflectometry data reduction

ESSsans
SANS data reduction

ESSspectroscopy
Spectroscopy data reduction

ESSreduce
Common tools for ESS data reduction



esssans

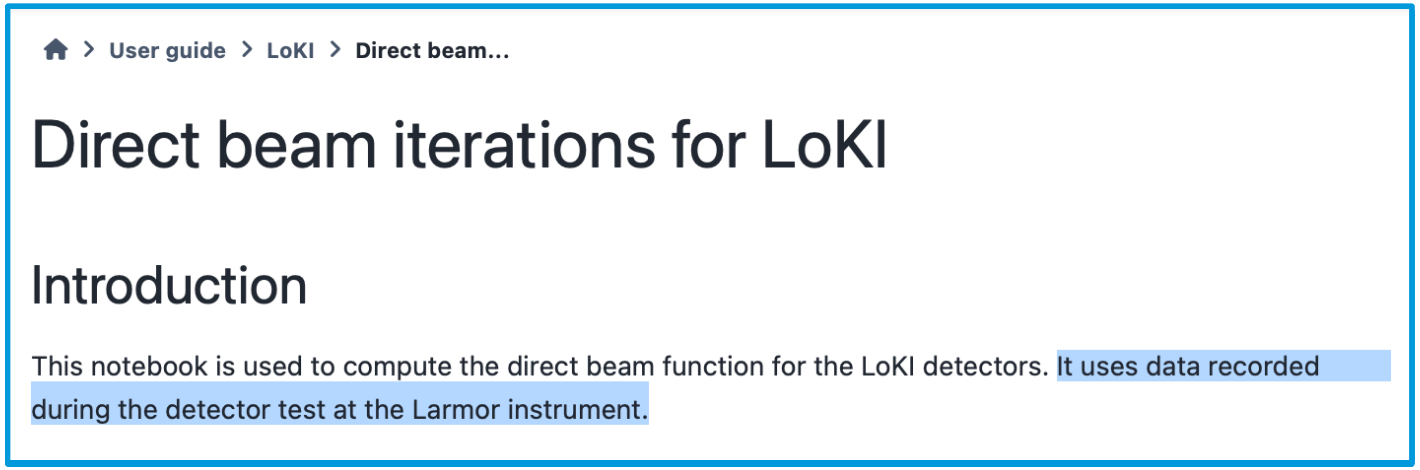
SANS data reduction for the European Spallation Source

Quick links

LoKI | **ISIS instruments** | **Common tools**

Data reduction workflows

Example – LOKI detector test data



User guide > LoKI > Direct beam...

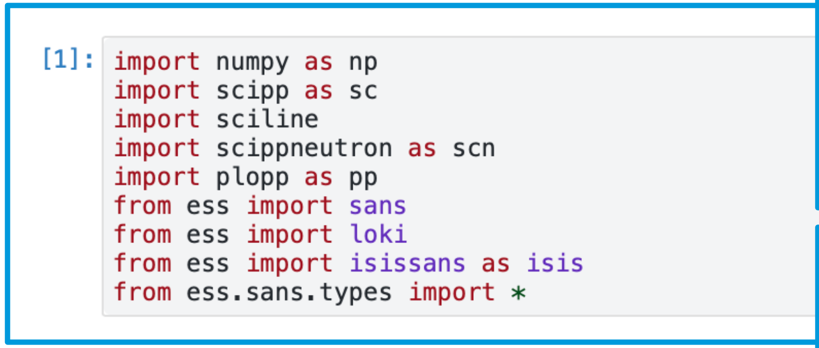
Direct beam iterations for LoKI

Introduction

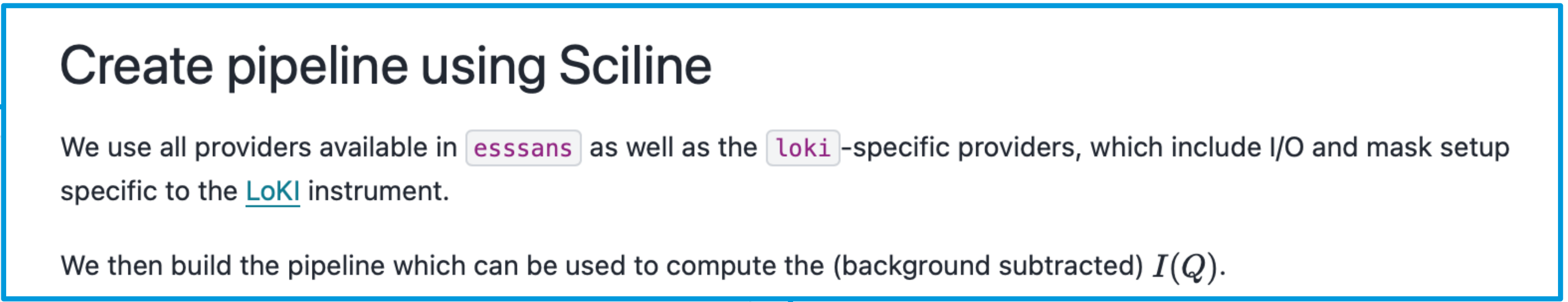
This notebook is used to compute the direct beam function for the LoKI detectors. It uses data recorded during the detector test at the Larmor instrument.

❑ scipp.github.io/esssans GitHub

Compute the (background subtracted) I(Q)



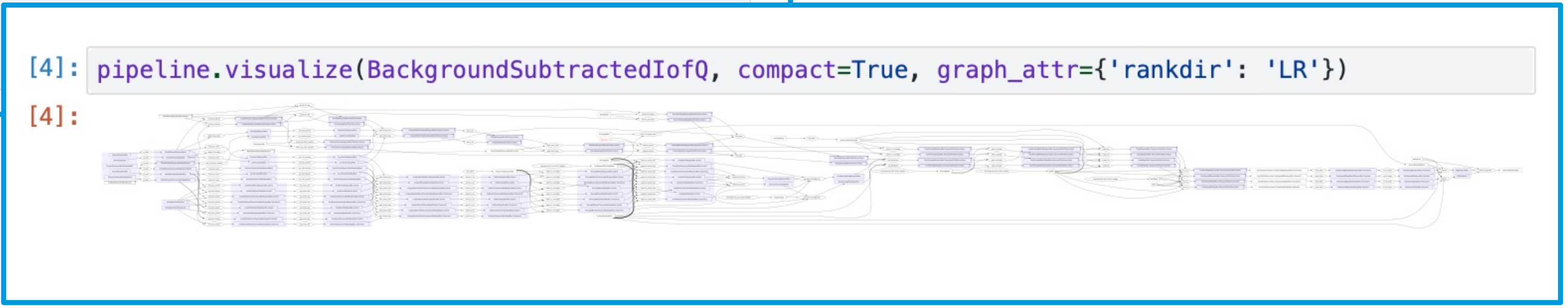
```
[1]: import numpy as np
     import scipp as sc
     import sciline
     import scippneutron as scn
     import plopp as pp
     from ess import sans
     from ess import loki
     from ess import isissans as isis
     from ess.sans.types import *
```



Create pipeline using Sciline

We use all providers available in `esssans` as well as the `loki`-specific providers, which include I/O and mask setup specific to the LoKI instrument.

We then build the pipeline which can be used to compute the (background subtracted) $I(Q)$.



```
[4]: pipeline.visualize(BackgroundSubtractedIofQ, compact=True, graph_attr={'rankdir': 'LR'})
[4]:
```

Data Analysis
-
SWAT

easydiffraction

easyreflectometry

python™

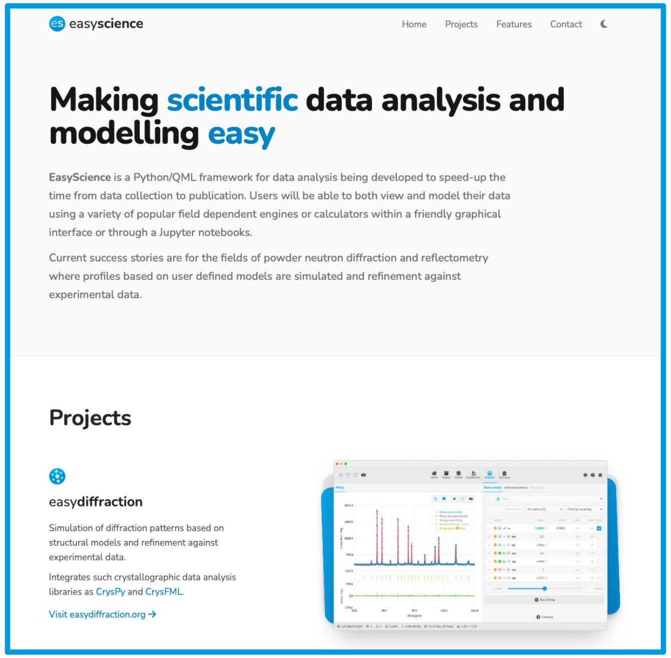
jupyterhub

Qt

Data Analysis

In-house projects

- ❑ EasyScince
- ❑ EasyDiffractionApp
  - ❑EasyDiffractionLib
- ❑ EasyReflectometryApp
  - ❑EasyReflectometryLib
- ❑ See https://easyscience.software



Making scientific data analysis and modelling easy

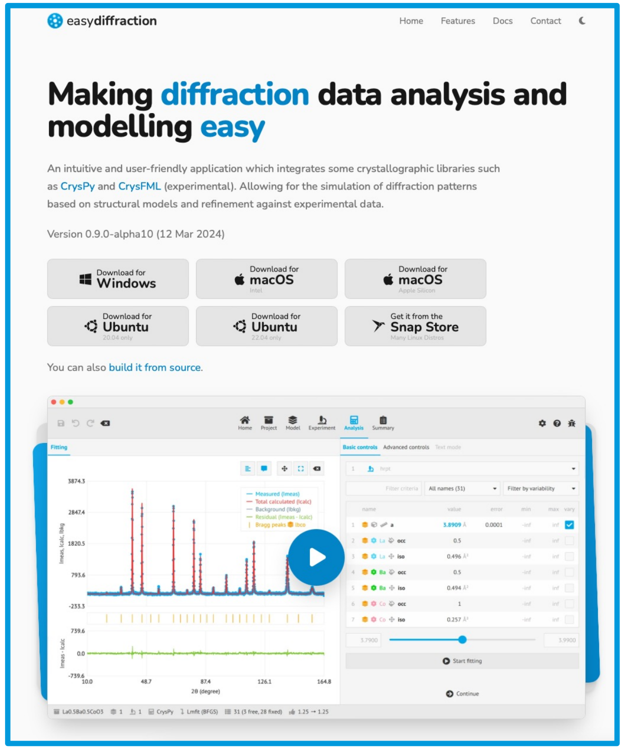
EasyScience is a Python/QML framework for data analysis being developed to speed-up the time from data collection to publication. Users will be able to both view and model their data using a variety of popular field dependent engines or calculators within a friendly graphical interface or through a Jupyter notebooks.

Current success stories are for the fields of powder neutron diffraction and reflectometry where profiles based on user defined models are simulated and refinement against experimental data.

Projects

Simulation of diffraction patterns based on structural models and refinement against experimental data.

Integrates such crystallographic data analysis libraries as CrysPy and CrysFML.

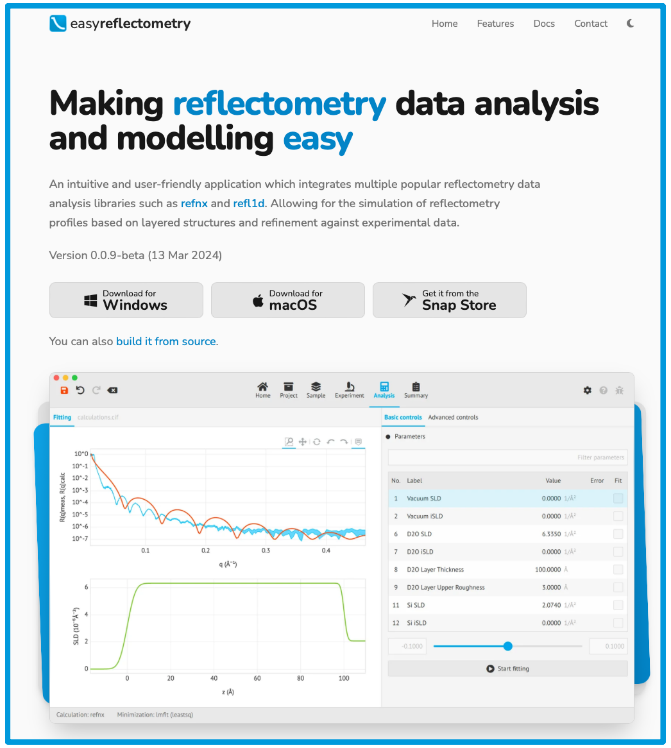


Making diffraction data analysis and modelling easy

An intuitive and user-friendly application which integrates some crystallographic libraries such as CrysPy and CrysFML (experimental). Allowing for the simulation of diffraction patterns based on structural models and refinement against experimental data.

Version 0.9.0-alpha10 (12 Mar 2024)

You can also build it from source.



Making reflectometry data analysis and modelling easy

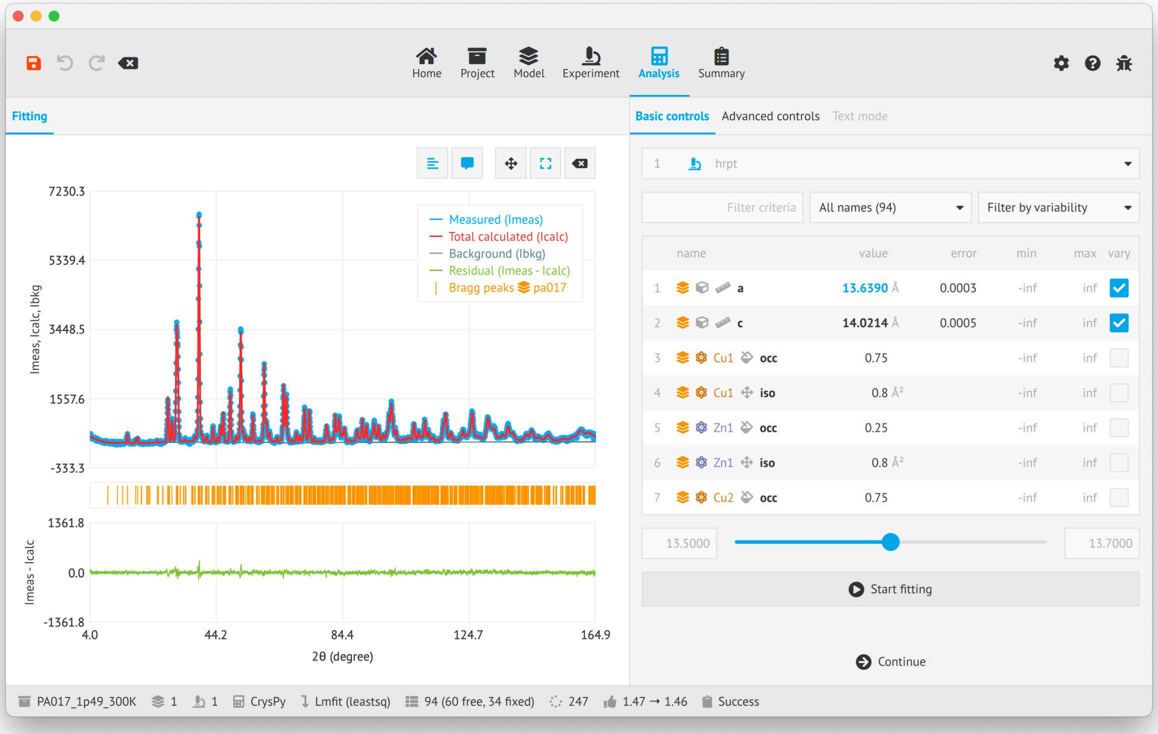
An intuitive and user-friendly application which integrates multiple popular reflectometry data analysis libraries such as refnx and refl1d. Allowing for the simulation of reflectometry profiles based on layered structures and refinement against experimental data.

Version 0.0.9-beta (13 Mar 2024)

You can also build it from source.

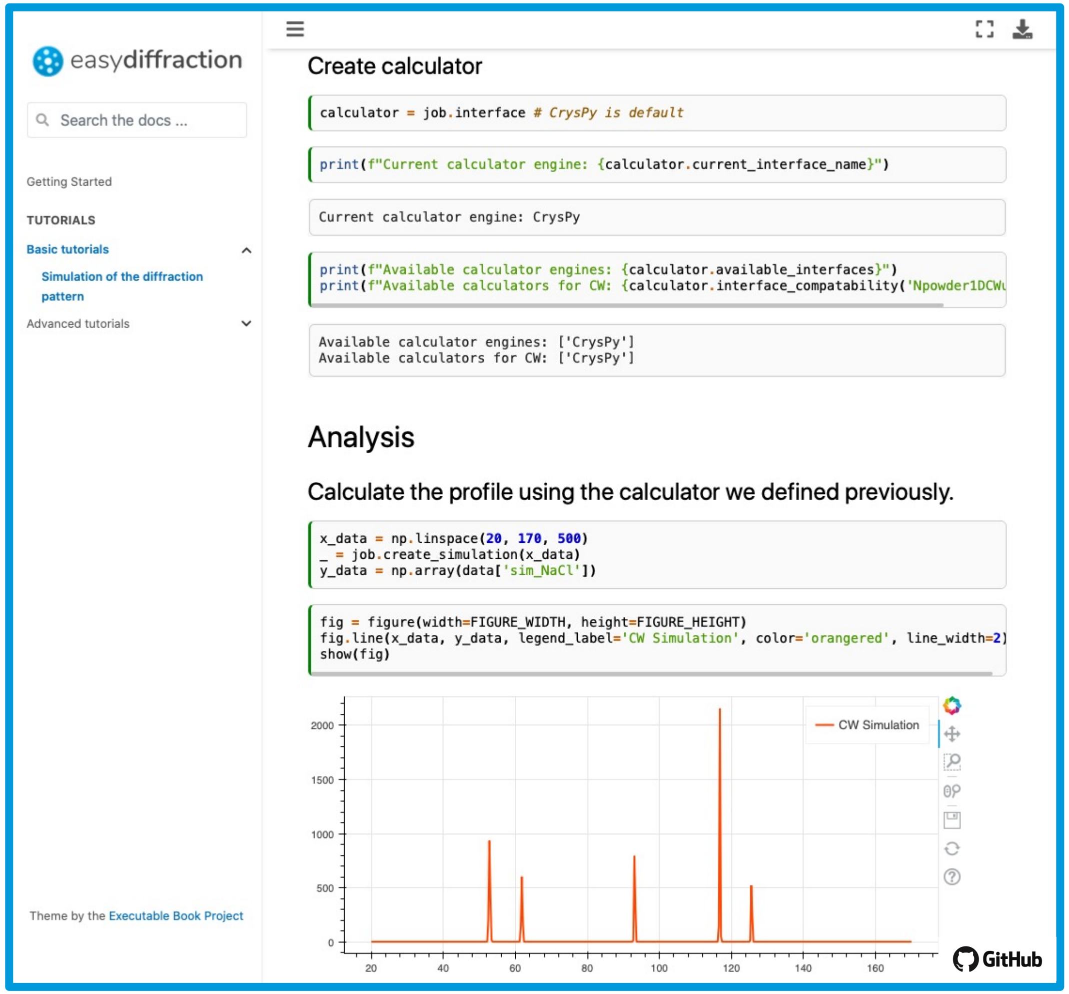
easyDiffraction

App & Lib for Jupyter notebook



	name	value	error	min	max	vary
1	a	13.6390 Å	0.0003	-inf	inf	☑
2	c	14.0214 Å	0.0005	-inf	inf	☑
3	Cu1 occ	0.75		-inf	inf	□
4	Cu1 iso	0.8 Å²		-inf	inf	□
5	Zn1 occ	0.25		-inf	inf	□
6	Zn1 iso	0.8 Å²		-inf	inf	□
7	Cu2 occ	0.75		-inf	inf	□

PA017_1p49_300K 1 1 CrysPy Lmfit (leastsq) 94 (60 free, 34 fixed) 247 1.47 → 1.46 Success



Create calculator

```
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_compatability('Npowder1DCWu
```

```
Available calculator engines: ['CrysPy']
Available calculators for CW: ['CrysPy']
```

Analysis

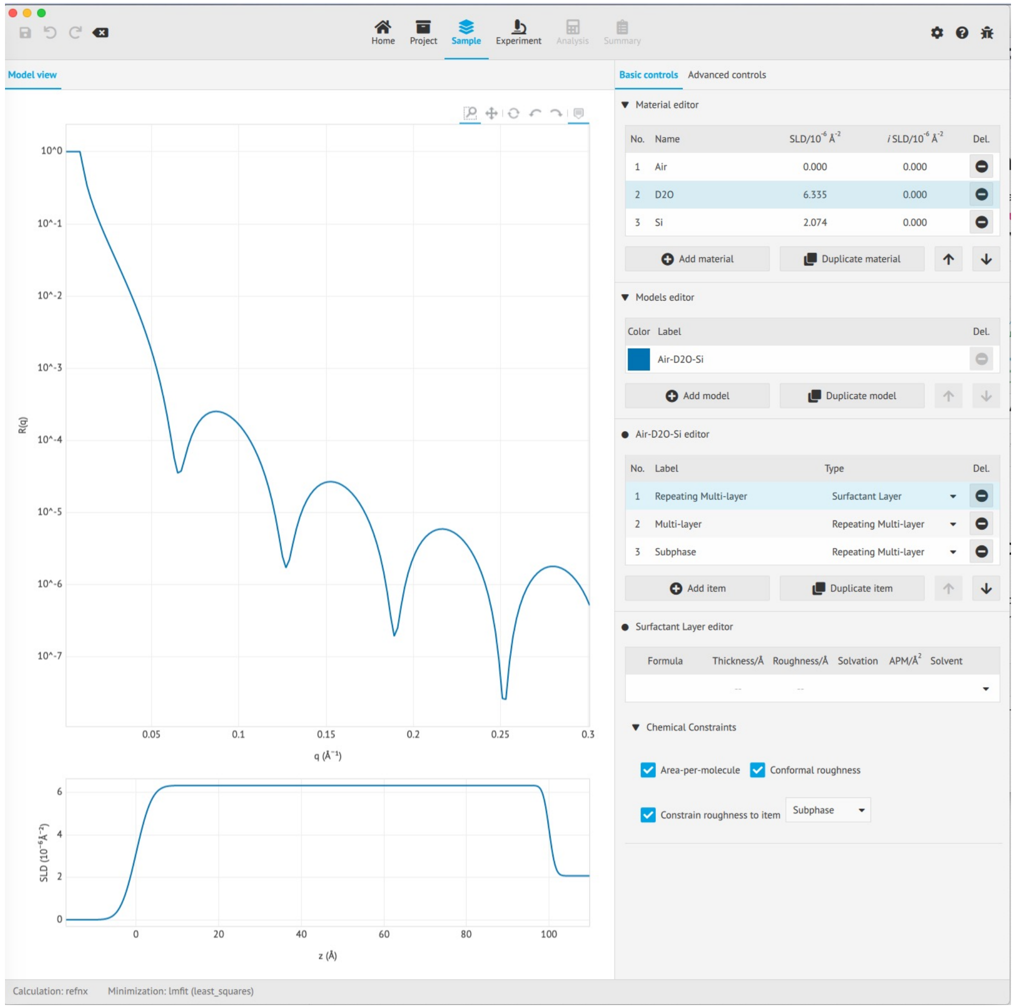
Calculate the profile using the calculator we defined previously.

```
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)
```

easyReflectometry

App & Lib for Jupyter notebook



▼ Material editor

No.	Name	SLD/10^{-6} Å^{-2}	*i* SLD/10^{-6} Å^{-2}	Del.
1	Air	0.000	0.000	⊖
2	D2O	6.335	0.000	⊖
3	Si	2.074	0.000	⊖

▼ Models editor

Color	Label	Del.
🟦	Air-D2O-Si	⊖

● Air-D2O-Si editor

No.	Label	Type	Del.
1	Repeating Multi-layer	Surfactant Layer	⊖
2	Multi-layer	Repeating Multi-layer	⊖
3	Subphase	Repeating Multi-layer	⊖

● Surfactant Layer editor

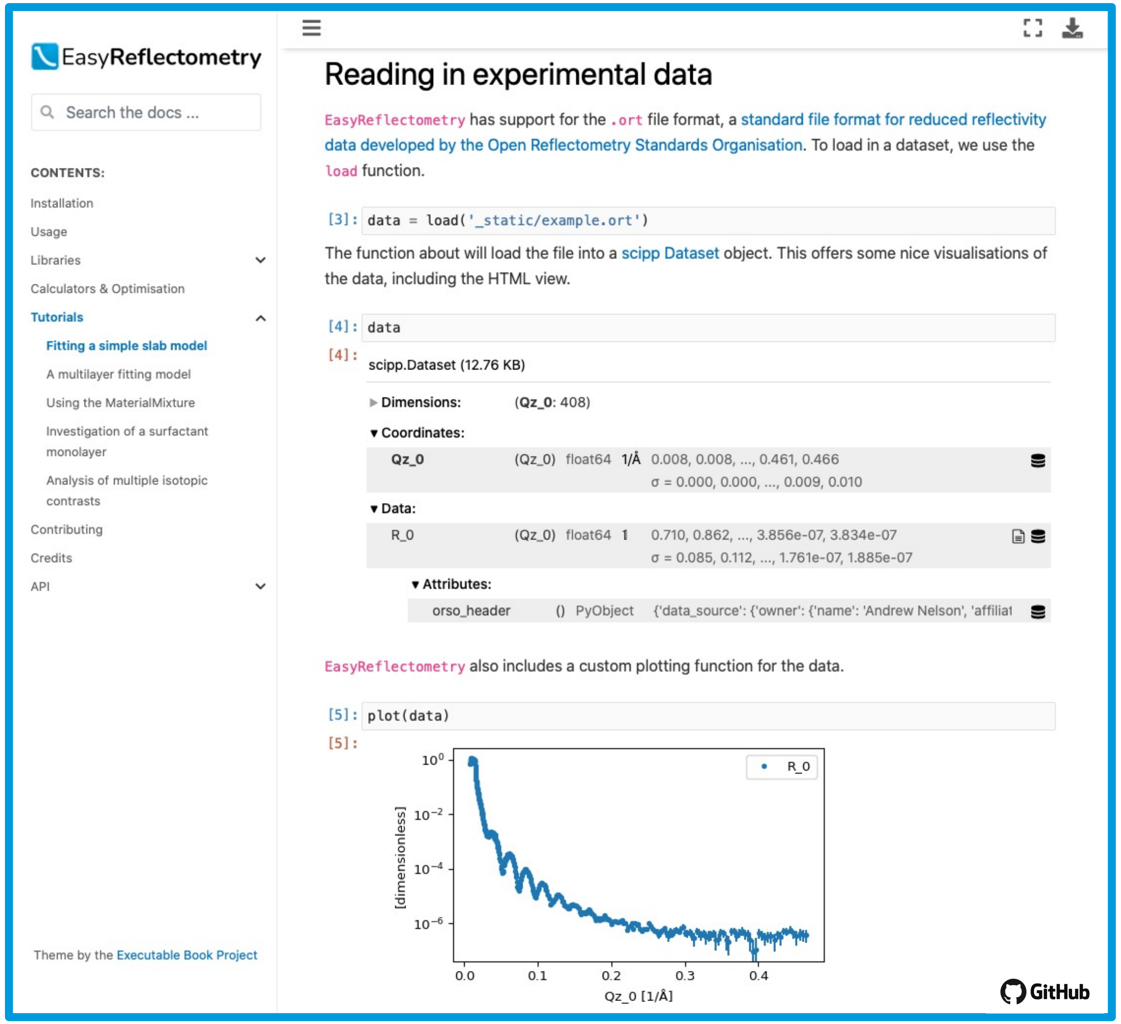
Formula	Thickness/Å	Roughness/Å	Solvation	APM/Å^2	Solvent
	--	--			

▼ Chemical Constraints

☑ Area-per-molecule ☑ Conformal roughness

☑ Constrain roughness to item: Subphase

Calculation: refnx Minimization: lmfit (least_squares)



CONTENTS:

- Installation
- Usage
- Libraries
- Calculators & Optimisation
- Tutorials
  - Fitting a simple slab model
  - A multilayer fitting model
  - Using the MaterialMixture
  - Investigation of a surfactant monolayer
  - Analysis of multiple isotopic contrasts
- Contributing
- Credits
- API

Theme by the Executable Book Project

Reading in experimental data

`EasyReflectometry` has support for the `.ort` file format, a standard file format for reduced reflectivity data developed by the Open Reflectometry Standards Organisation. To load in a dataset, we use the `load` function.

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

The function about will load the file into a scipp Dataset object. This offers some nice visualisations of the data, including the HTML view.

```
[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
```

`EasyReflectometry` also includes a custom plotting function for the data.

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

```
[5]:
```

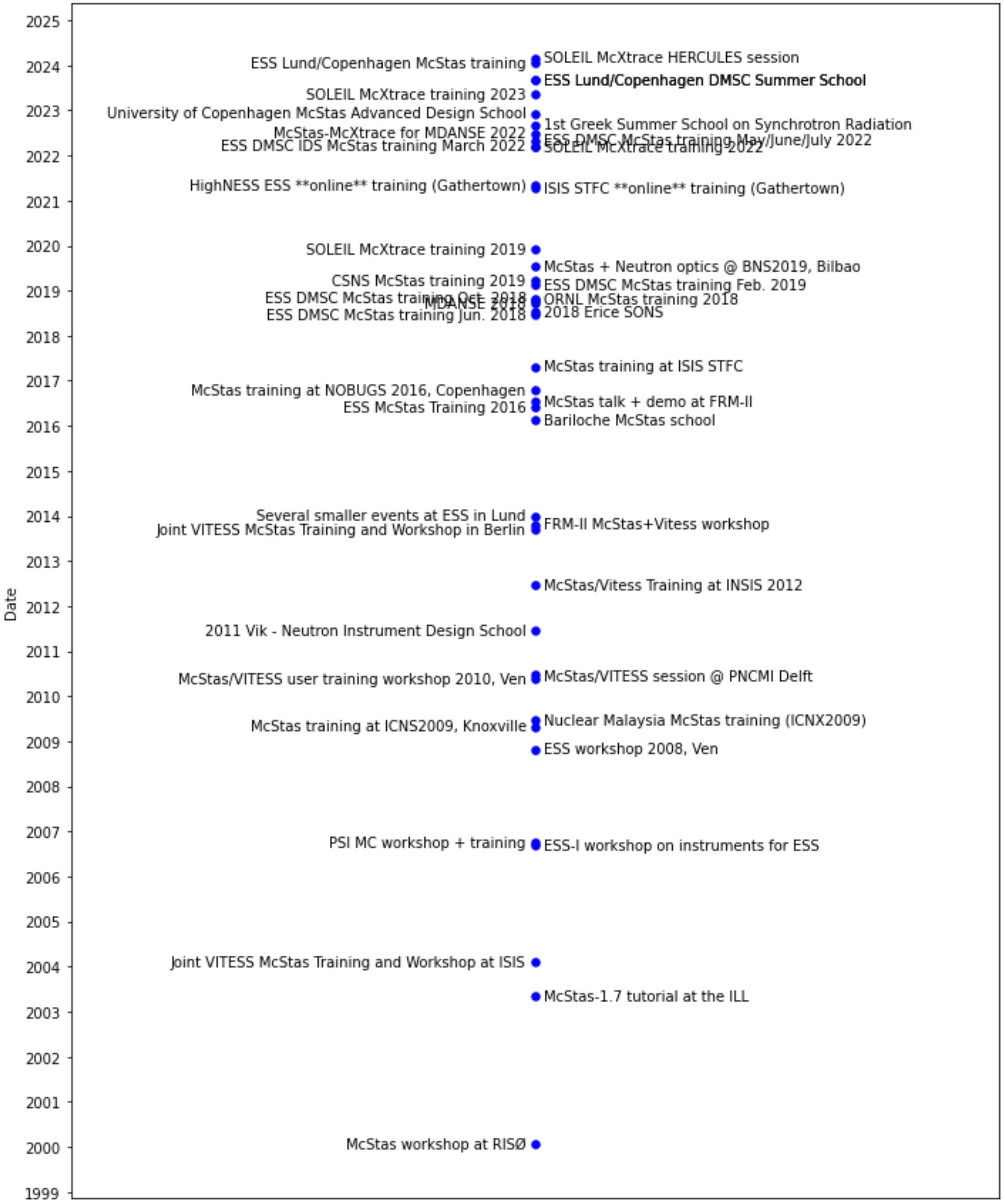
Modelling
-
McStas

McStas

Instrument simulations for ESS

- ❑ Maintain and develop McStas (for ESS and others)
- ❑ Help and train users

- Competence in community
  - – Over 40 schools
    - – more than 20 outside Nordic countries
  - – University courses
  - – e-learning https://e-learning.pan-training.eu
  - – Super users at facilities

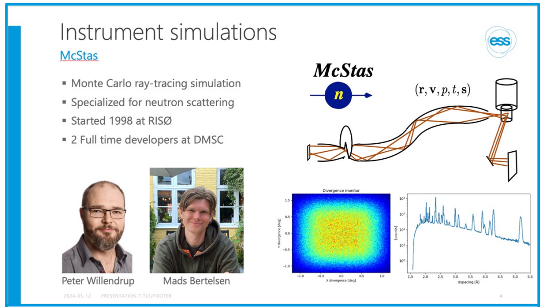


McStas

Instrument simulations for ESS

- ❑ See presentation from Mads B.
    - ❑ Introduction to simulation efforts

- ❑ McStas & McStasScript
- ❑ Union & GPU



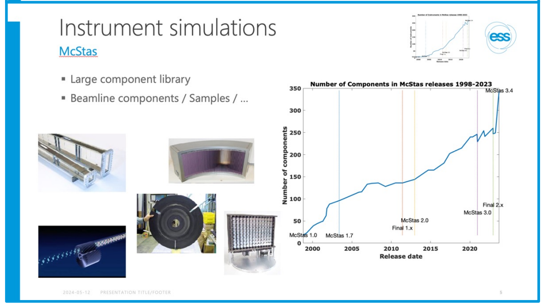
Instrument simulations

McStas

- Monte Carlo ray-tracing simulation
- Specialized for neutron scattering
- Started 1998 at RISØ
- 2 Full time developers at DMSC

Peter Willendrup

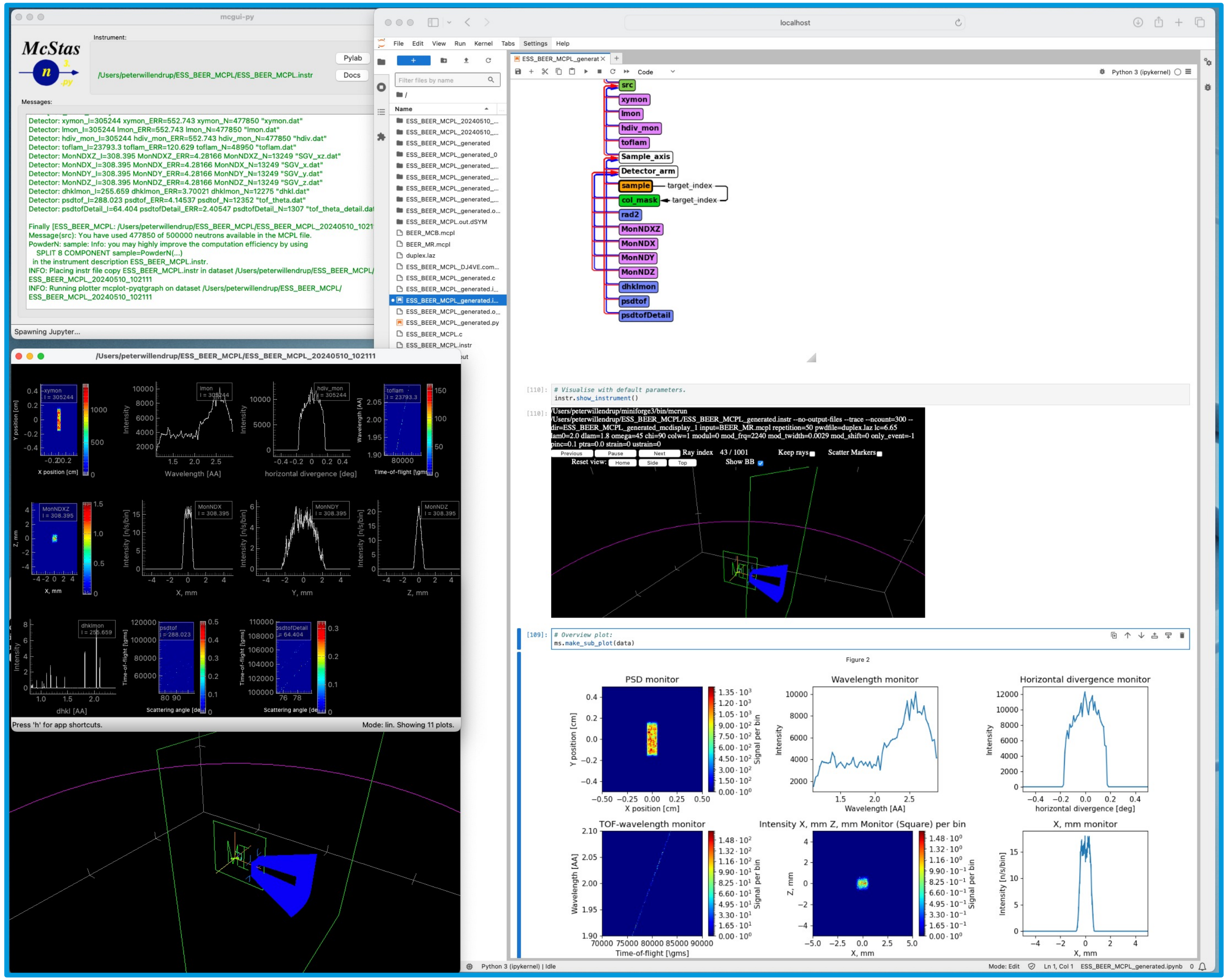
Mads Bertelsen



Instrument simulations

McStas

- Large component library
- Beamline components / Samples / …



Instrument:

/Users/peterwillendrup/ESS_BEER_MCPL/ESS_BEER_MCPL.instr

Messages:

```
Detector: xymon_I=305244 xymon_ERR=552.743 xymon_N=477850 "xymon.dat"
Detector: lmon_I=305244 lmon_ERR=552.743 lmon_N=477850 "lmon.dat"
Detector: hdiv_mon_I=305244 hdiv_mon_ERR=552.743 hdiv_mon_N=477850 "hdiv.dat"
Detector: toflam_I=23793.3 toflam_ERR=120.629 toflam_N=48950 "toflam.dat"
Detector: MonNDXZ_I=308.395 MonNDXZ_ERR=4.28166 MonNDXZ_N=13249 "SGV_xz.dat"
Detector: MonNDX_I=308.395 MonNDX_ERR=4.28166 MonNDX_N=13249 "SGV_x.dat"
Detector: MonNDY_I=308.395 MonNDY_ERR=4.28166 MonNDY_N=13249 "SGV_y.dat"
Detector: MonNDZ_I=308.395 MonNDZ_ERR=4.28166 MonNDZ_N=13249 "SGV_z.dat"
Detector: dhklmon_I=255.659 dhklmon_ERR=3.70021 dhklmon_N=12275 "dhkl.dat"
Detector: psdtof_I=288.023 psdtof_ERR=4.14537 psdtof_N=12352 "tof_theta.dat"
Detector: psdtofDetail_I=64.404 psdtofDetail_ERR=2.40547 psdtofDetail_N=1307 "tof_theta_detail.dat

Finally [ESS_BEER_MCPL: /Users/peterwillendrup/ESS_BEER_MCPL/ESS_BEER_MCPL_20240510_1021
Message(src): You have used 477850 of 500000 neutrons available in the MCPL file.
PowderN: sample: Info: you may highly improve the computation efficiency by using
   SPLIT 8 COMPONENT sample=PowderN(...)
 in the instrument description ESS_BEER_MCPL.instr.
INFO: Placing instr file copy ESS_BEER_MCPL.instr in dataset /Users/peterwillendrup/ESS_BEER_MCPL/
ESS_BEER_MCPL_20240510_102111
INFO: Running plotter mcplot-pyqtgraph on dataset /Users/peterwillendrup/ESS_BEER_MCPL/
ESS_BEER_MCPL_20240510_102111
```

Spawning Jupyter...

/Users/peterwillendrup/ESS_BEER_MCPL/ESS_BEER_MCPL_20240510_102111

Press 'h' for app shortcuts. Mode: lin. Showing 11 plots.

```
[110]: # Visualise with default parameters.
       instr.show_instrument()
```

```
[110]: /Users/peterwillendrup/miniforge3/bin/mcrun
/Users/peterwillendrup/ESS_BEER_MCPL/ESS_BEER_MCPL_generated.instr --no-output-files --trace --ncount=300 --
dir=ESS_BEER_MCPL_generated_mcdisplay_1 input=BEER_MR.mcpl repetition=50 pwdfile=duplex.laz lc=6.65
lam0=2.0 dlam=1.8 omega=45 chi=90 colw=1 modul=0 mod_frq=2240 mod_twidth=0.0029 mod_shift=0 only_event=-1
pinc=0.1 ptra=0.0 strain=0 ustrain=0
```

Ray index 43 / 1001

```
[109]: # Overview plot:
       ms.make_sub_plot(data)
```

Python 3 (ipykernel) | Idle

McStas McGui & McStasScript – same instr file

Looking ahead

Going forwards

Next steps, contacts, and ways of collaboration

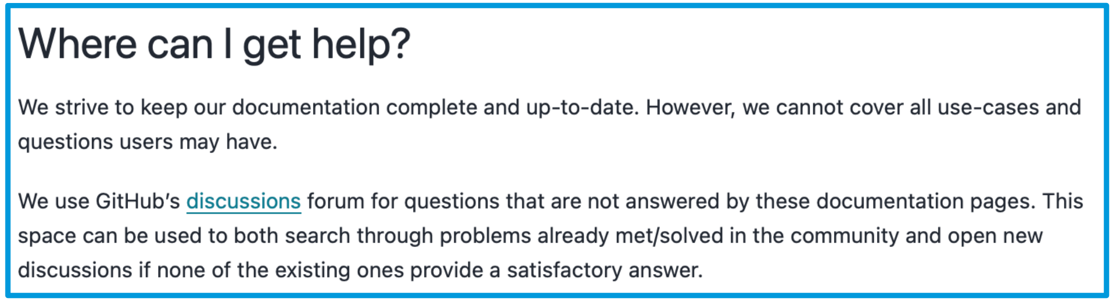
Suggestions

- ❑ Try out the reduction software scipp
- ❑ Advantage to be familiar with scipp and python

Where to find more information?

- ❑ Ask the DRAM teams
- ❑ On-line Documentation
- ❑ Python training (IKON)
- ❑ DMSC summer school

scipp



Where can I get help?

We strive to keep our documentation complete and up-to-date. However, we cannot cover all use-cases and questions users may have.

We use GitHub's discussions forum for questions that are not answered by these documentation pages. This space can be used to both search through problems already met/solved in the community and open new discussions if none of the existing ones provide a satisfactory answer.

easyreflectometry

Get in touch

?

General questions

For general questions or comments, please contact us at support@easyreflectometry.org, or fill out the form.

GitHub

Questions?