



Future developments in SasView

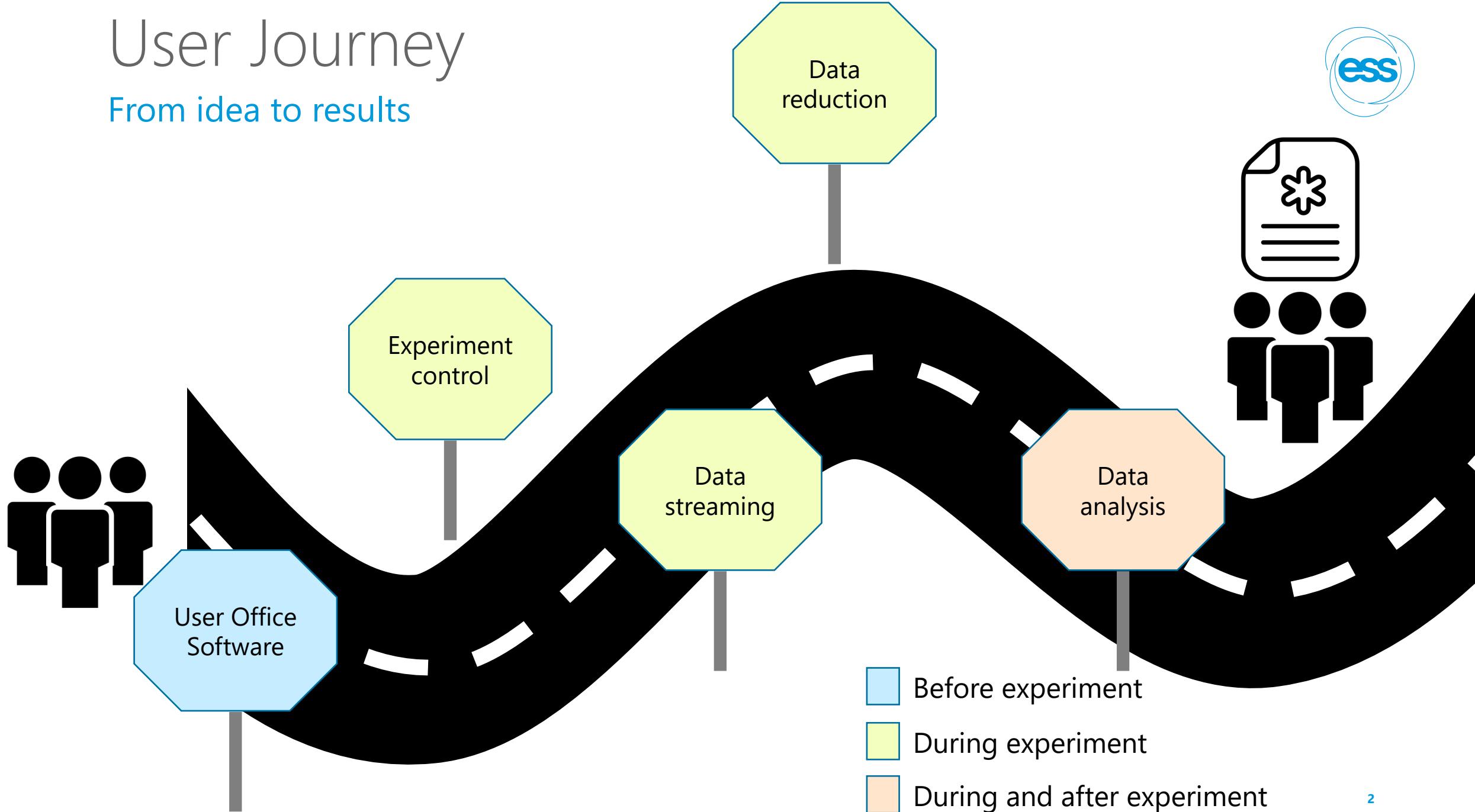
The challenge of meeting community aspirations

WOJCIECH POTRZEBOWSKI

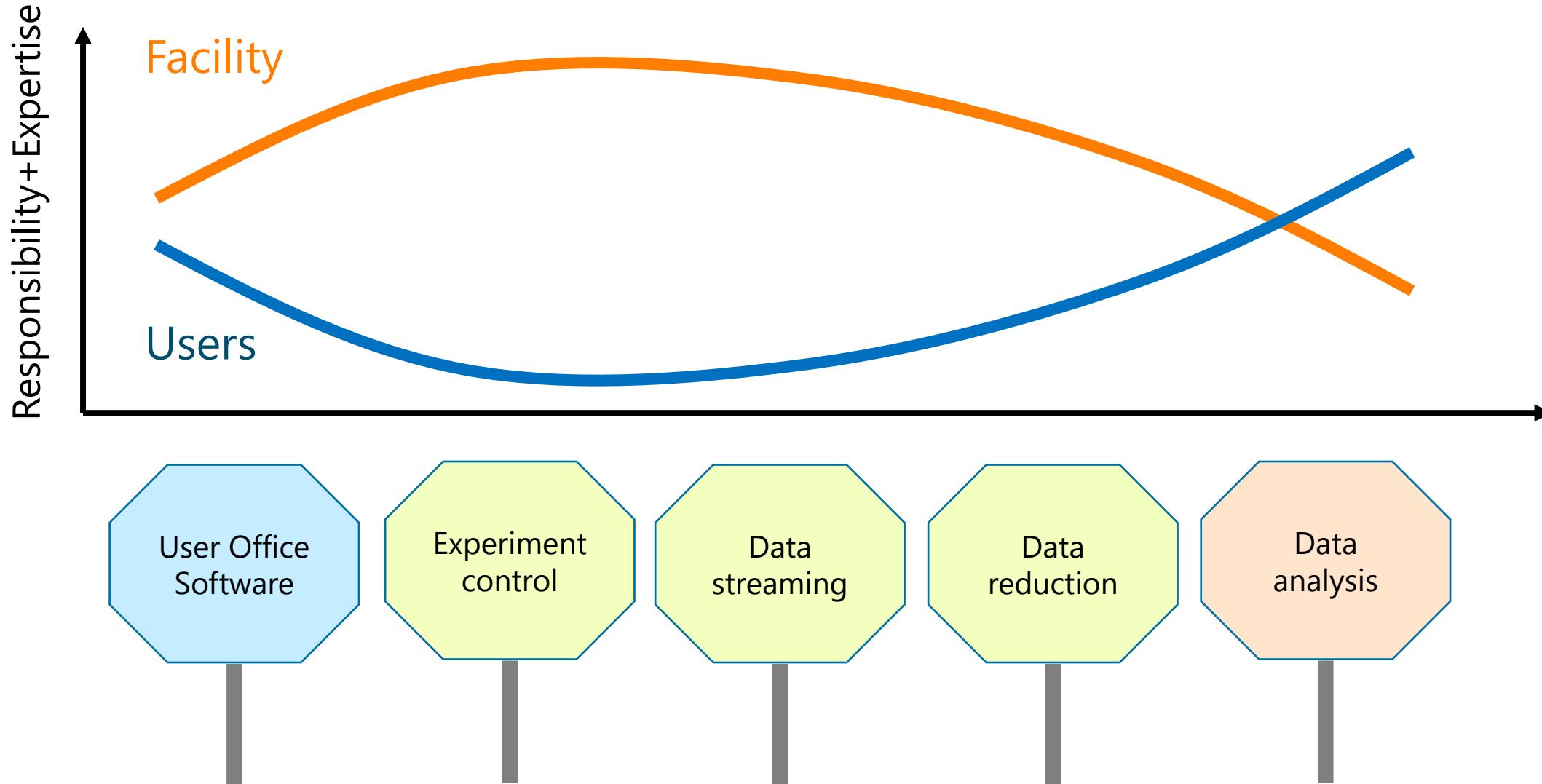
CanSAS 2023

User Journey

From idea to results



User Journey



Data analysis

Facts of life



Data analysis is part of the processing workflow

Reduced data on the disks is useless

Users expertise is critical for successful data analysis

Facilities should support data analysis software but shouldn't fully drive the process

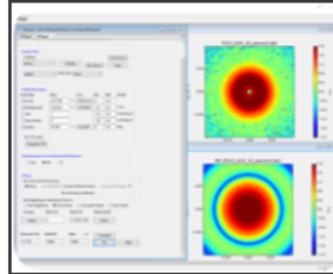
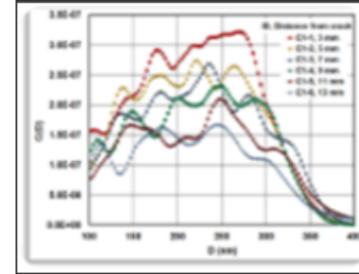
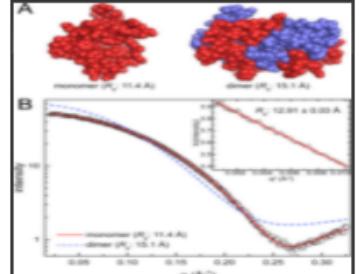
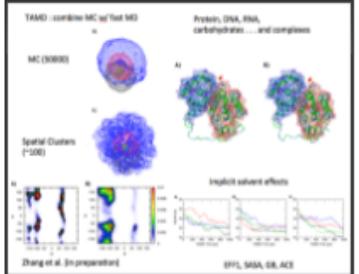
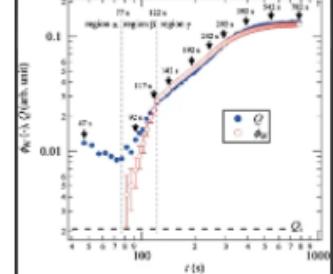
->We need open and collaborative software tools in order to succeed!

Data analysis

Few different options for SAS data



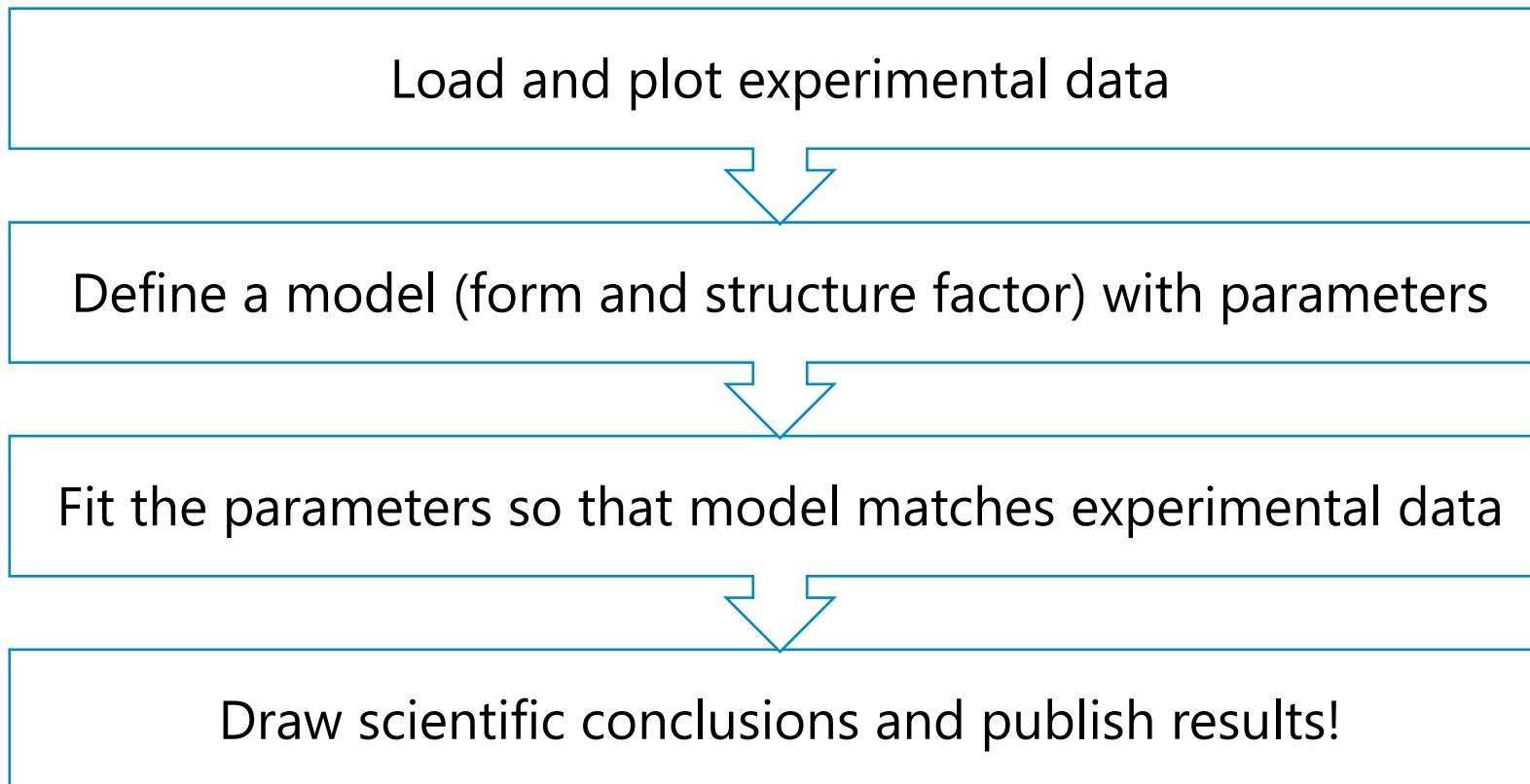
Not in SasView scope

Model-Fitting Methods	Real-Space Methods	Ab-Initio Methods	MC/TMD Methods	Other Methods
 <p>Example of 2D model-fitting using the SasView application 10.1179/1743284714Y.00000000577</p>	 <p>Cavity size distributions in a steel weldment as derived from SANS 10.1179/1743284714Y.00000000577</p>	 <p>Ab-initio modelling of polcalcin constrained by SAXS 10.1002/pro.3376</p>	 <p>MC & TAMD modelling of proteins constrained by SANS 10.1016/j.jmgm.2017.02.010</p>	 <p>Time evolution of the invariant during crystallisation of P4MP1 10.1038/pj.2012.204</p>

<https://www.isis.stfc.ac.uk/Pages/SANSdataanalysisOverview.aspx>

Typical SasView workflow

For reduced data:

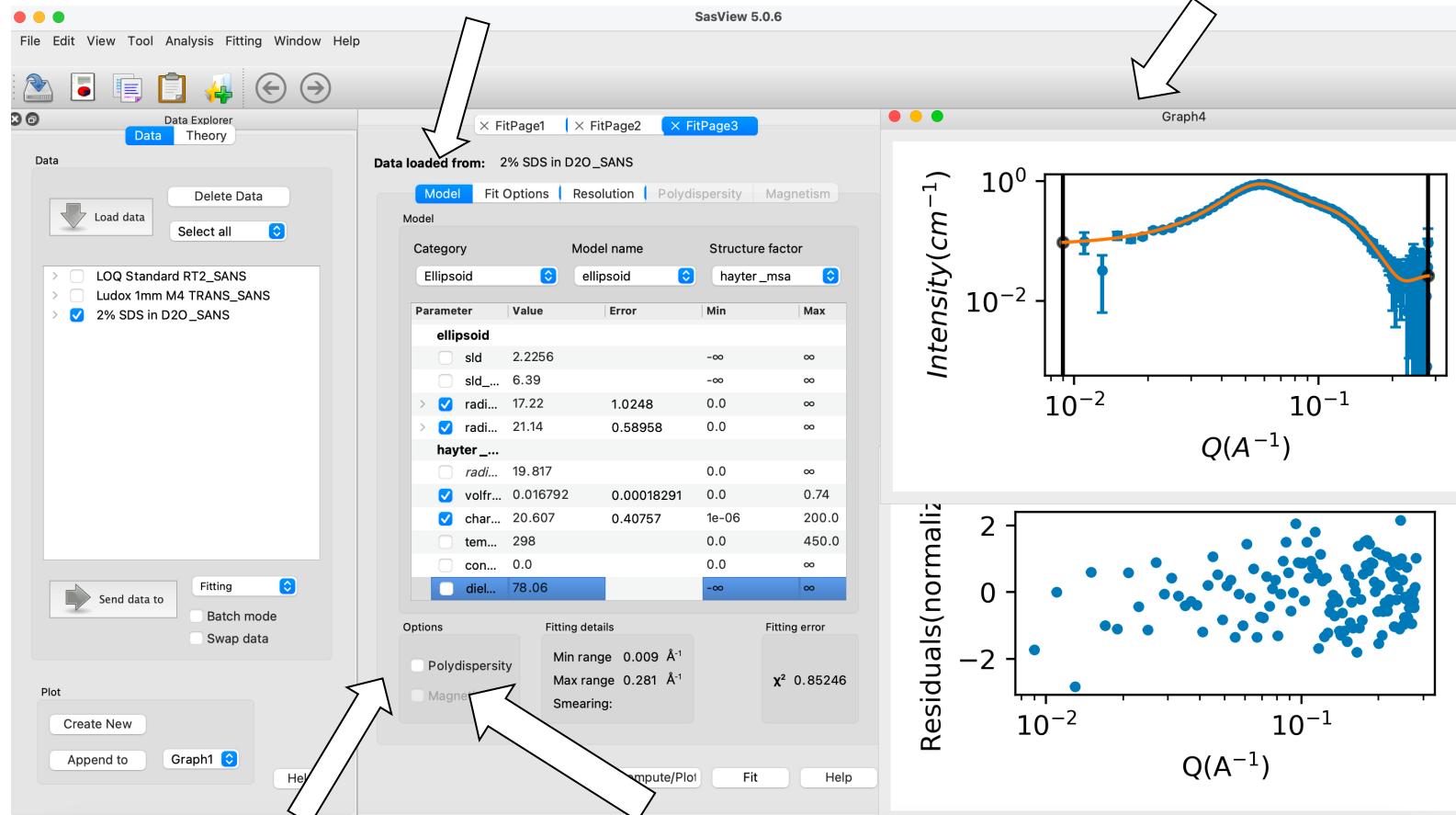


Fitting in SasView

70+ models to explain data



Wide choice of built-in models (> 70)
 $F(Q)$, $S(Q)$ & $F(Q)*S(Q)$



Single, batch and simultaneous fitting

Plugin models

SasView provides tools and infrastructure for custom/plugin models



- Dedicated editor
- Syntax and performance testing
- Directly available in SasView ecosystem
- Community developed models can be deposited to marketplace:
<https://marketplace.sasview.org/>

The screenshot shows the SasView Marketplace interface. At the top, there's a search bar and a log-in button. Below it, a grid of model cards:

- OrientedMagneticChains**: Sphere, 21 May 2021, krycket, 0, Ellipsoid, Lameillae, Other, Paracrystal, Parallelepiped, Shape-Independent, Sphere, Structure Factor.
- Magnetic vortex in a disc**: Cylinder, 10 Mar 2021, dehoni, 0, Ellipsoid, Lameillae, Other, Paracrystal, Parallelepiped, Shape-Independent, Sphere, Structure Factor.
- Field-dependent magnetic SANS of misaligned magnetic moments in bulk ferromagnets**: Sphere, 17 Feb 2021, dehoni, 0, Ellipsoid, Lameillae, Other, Paracrystal, Parallelepiped, Shape-Independent, Sphere, Structure Factor.
- SANS of bulk ferromagnets**: Sphere, 17 Feb 2021, dehoni, 0, Ellipsoid, Lameillae, Other, Paracrystal, Parallelepiped, Shape-Independent, Sphere, Structure Factor.
- core_shell_ellipsoid_tied and core_shell_ellipsoid_repar**: Ellipsoid, 16 Feb 2021, richard, 0, Ellipsoid, Lameillae, Other, Paracrystal, Parallelepiped, Shape-Independent, Sphere, Structure Factor.

The Model Editor interface has two tabs: "Plugin Definition" and "Model editor".

Plugin Definition Tab:

- Plugin name:** my_broad_peak
- Description:** My special broad peak model
- Fit parameters:**
 - Non-polydisperse:** porod_scale (1.0e-05), porod_exp (3), lorentz_length (50), lorentz_scale (10), lorentz_exp (2.0)
 - Polydisperse:** (empty)
- Function(x):**

```
z = abs(q - peak_pos) * lorentz_length
lq = (porod_scale / q ** porod_exp + lorentz_scale / (1 + z ** lorentz_exp))
return lq
```

Model Editor Tab:

- Model:**

```
"""
Definition
-----
Calculates my_broad_peak.

My special broad peak model

References
-----
Authorship and Verification
-----
"""

**Author:** --- **Date:** 2018YY-09m-20d
**Last Modified by:** --- **Date:** 2018YY-09m-20d
**Last Reviewed by:** --- **Date:** 2018YY-09m-20d

from math import *
from numpy import inf

name = "my_broad_peak"
title = "User model for my_broad_peak"
description = "My special broad peak model"

parameters = [
    # {"name": "units", "units": "default", "lower": None, "upper": None, "type": "float", "description": "# porod_scale", "value": 1e-05, "min": -inf, "max": inf, "step": 1e-05}, {"name": "porod_exp", "units": "None", "lower": -inf, "upper": inf, "type": "float", "description": "# porod_exp", "value": 3.0, "min": -inf, "max": inf, "step": 0.1}, {"name": "lorentz_length", "units": "None", "lower": -inf, "upper": inf, "type": "float", "description": "# lorentz_length", "value": 50.0, "min": -inf, "max": inf, "step": 1.0}, {"name": "lorentz_scale", "units": "None", "lower": -inf, "upper": inf, "type": "float", "description": "# lorentz_scale", "value": 10.0, "min": -inf, "max": inf, "step": 0.1}, {"name": "lorentz_exp", "units": "None", "lower": -inf, "upper": inf, "type": "float", "description": "# lorentz_exp", "value": 2.0, "min": -inf, "max": inf, "step": 0.1}, {"name": "peak_pos", "units": "None", "lower": -inf, "upper": inf, "type": "float", "description": "# peak_pos", "value": 0.1, "min": -inf, "max": inf, "step": 0.01}, {"name": "q", "units": "None", "lower": -inf, "upper": inf, "type": "float", "description": "# q", "value": 0.01, "min": -inf, "max": inf, "step": 0.001}],
```

def lq(x, porod_scale, porod_exp, lorentz_length, lorentz_scale, lorentz_exp, peak_pos, q):
 """Absolute scattering"""
 z = abs(q - peak_pos) * lorentz_length
 lq = (porod_scale / q ** porod_exp + lorentz_scale / (1 + z ** lorentz_exp))
 return lq

uncomment the following if lq works for vector x
lq.vectorized = True

#def lqxy(x, y, porod_scale, porod_exp, lorentz_length, lorentz_scale, lorentz_exp,
peak_pos, q):
 """Absolute scattering of oriented particles."""
 # ...
 # return oriented_form(x, y, args)

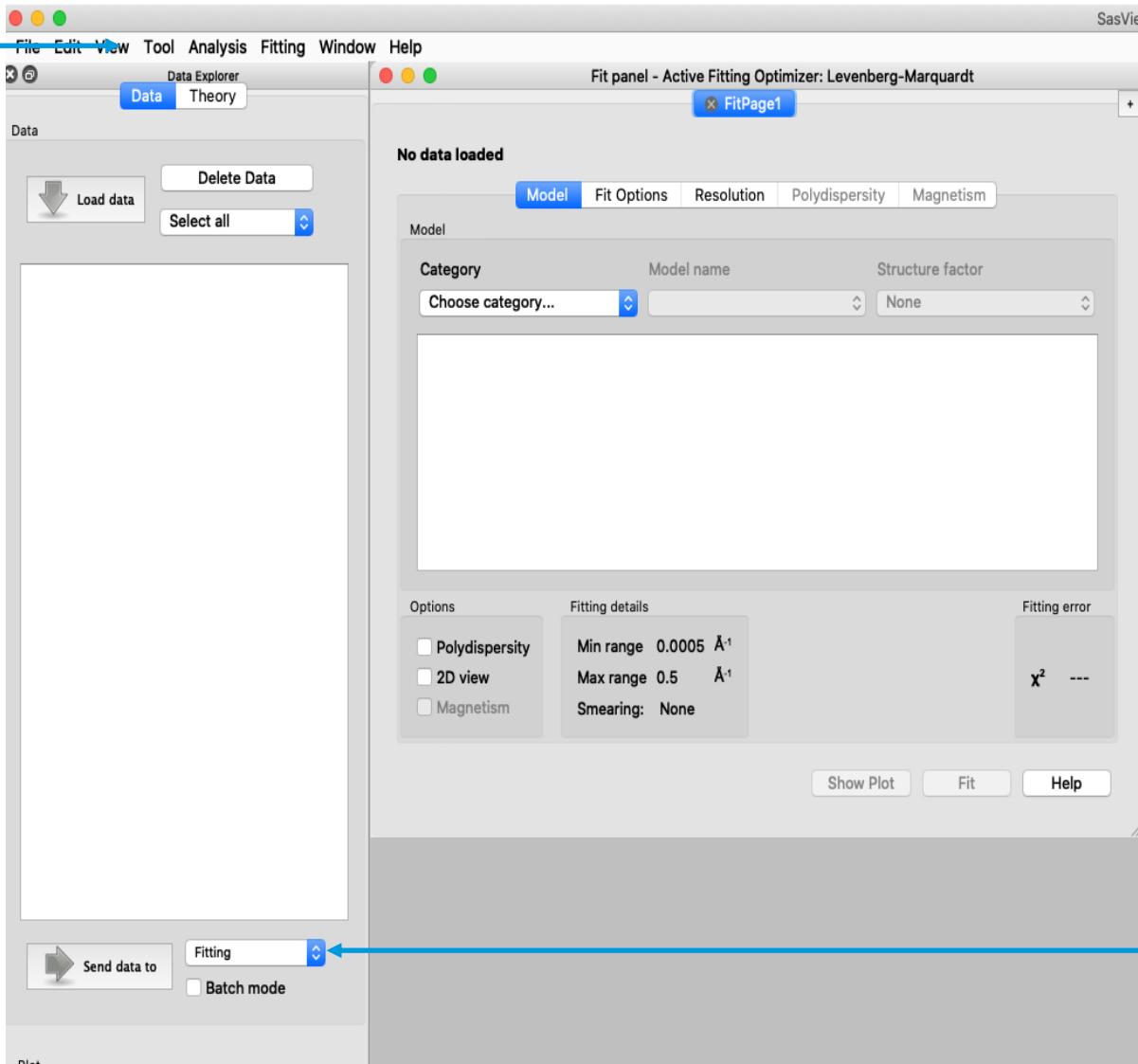
uncomment the following if lqxy works for vector x, y
- Help, Apply, Cancel**

Other SasView functionality

Various utility tools and calculators

Tools

- Data Operation
- SLD calculator
- Density/Volume calculator
- Slit Size Calculator
- Kiessig Thickness Calculator
- Q Resolution Estimator
- Generic Scattering calculator
- Orientation Viewer
- Python Shell/Editor
- Image Viewer
- File Converter



Analysis

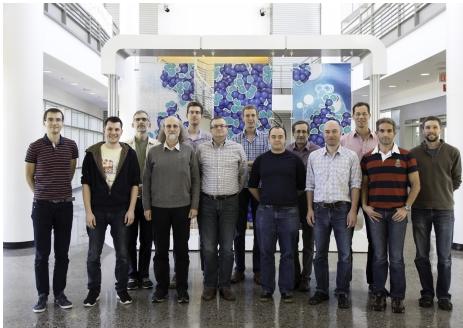
- Fitting
- Invariant
- Pr Inversion
- Correlation Function

The SasView approach

An open, collaborative, community development



- Community driven project
- Supported by 9 facilities, 40 contributors (~15 active at any one time)
- Small leadership team: Paul Butler (NIST), Steve King (ISIS), Wojciech Potrzebowski (ESS)
- Biweekly calls
- Regular SasView camps and hackathons
- Hosted on github:
<https://github.com/SasView/sasview>



Pictures from Code Camps: V, VI, VIII, X

www.sasview.org

SasView collaboration

Basic rules

Those who bring the resources (time, effort, funds) choose what to work on

but cannot break existing experiences ...



Student projects

Summer 2022 and 2023



Plotting improvements, model calculations with hypergeometric series - **Nouhalia Agouzal** and **Dorian Lozano**, supervisor: Miguel Gonzales (ILL)

Simultaneous SAXS and SANS fitting- **Iestyn Cadwallader-Jones**, supervisor: Miguel Gonzales (ILL)

Extending scattering calculator for Interacting bio-molecules – **Alex Zheng**, supervisor Yun Liu (NIST)

Local documentation generator – **Brayden Miller**, supervisor Paul Butler (NIST)

Web-based API, **Xale Shan**, supervisor Jeff Kryzwon (NIST)

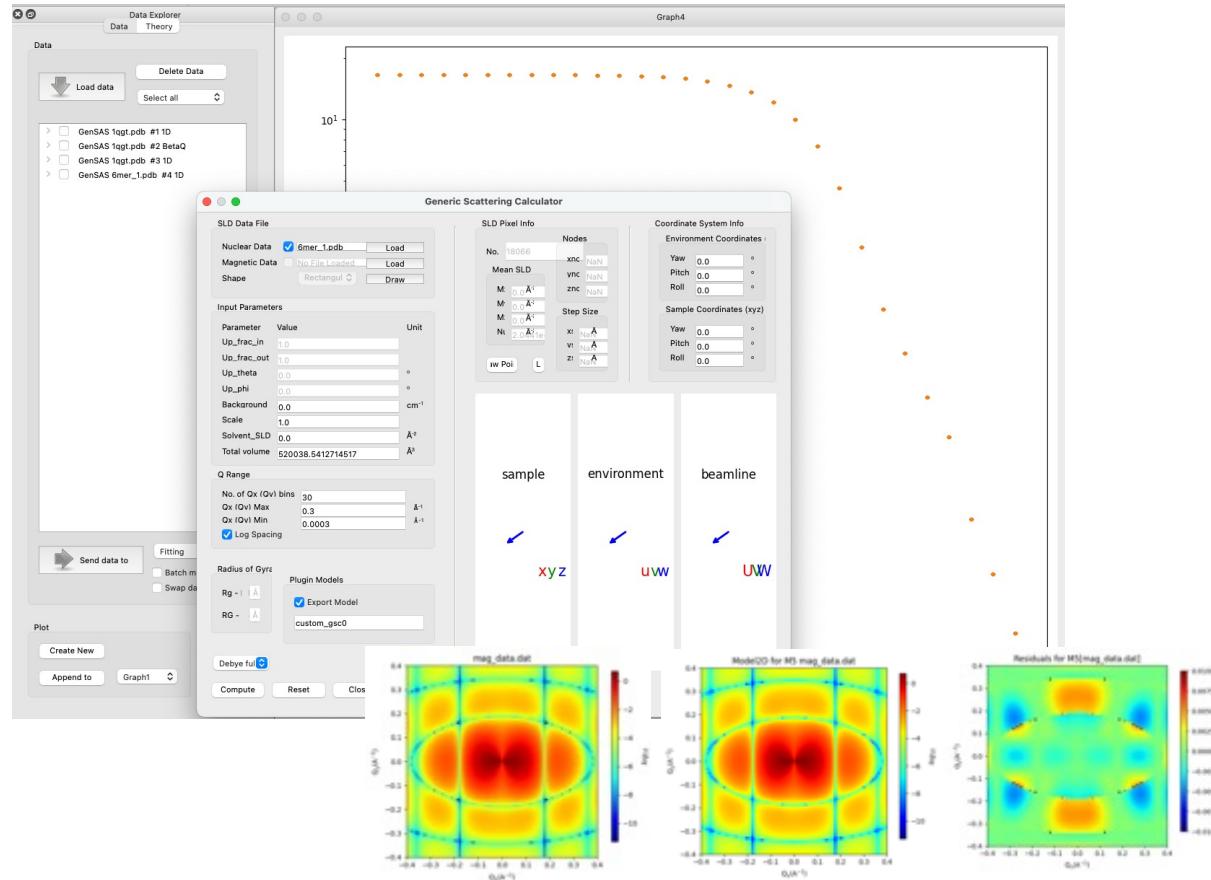
Wedge slicer - **Ellis Hewis**, supervisor Dirk Honecker (ISIS)

Extending scattering calculator for magnetic systems – **Ruben Lopes**, supervisor Dirk Honecker (ISIS)

Generic Scattering calculator improvements



Available in SasView 5.0.6



Magnetic/polarized SANS

- Generic Scattering Calculator supports magnetic/polarized SANS
- Coordinate data in VTK formats supported

Interacting bio-molecules

- Generate scattering profile from PDB file
- Save as a custom/plugin model
- Use for fitting with structure factors
- (PDB reader and engine refactored)
- (No hydration layer)

<https://github.com/SasView/sasview/releases/tag/v5.0.6>

<https://github.com/SasView/sasview/releases/tag/v6.0.0-alpha>

Magnetic SANS workshops



2-day in-person workshop in 2022

1-day follow-up hybrid workshop in 2023

Participants from universities and facilities

Gather requirements for Magnetic SANS
data analysis

A number of GitHub issues created

Ongoing work on slicers and porting
MuMag functionality (**Michael Adams**,
Univ. of Luxembourg)

Trying to arrange student projects



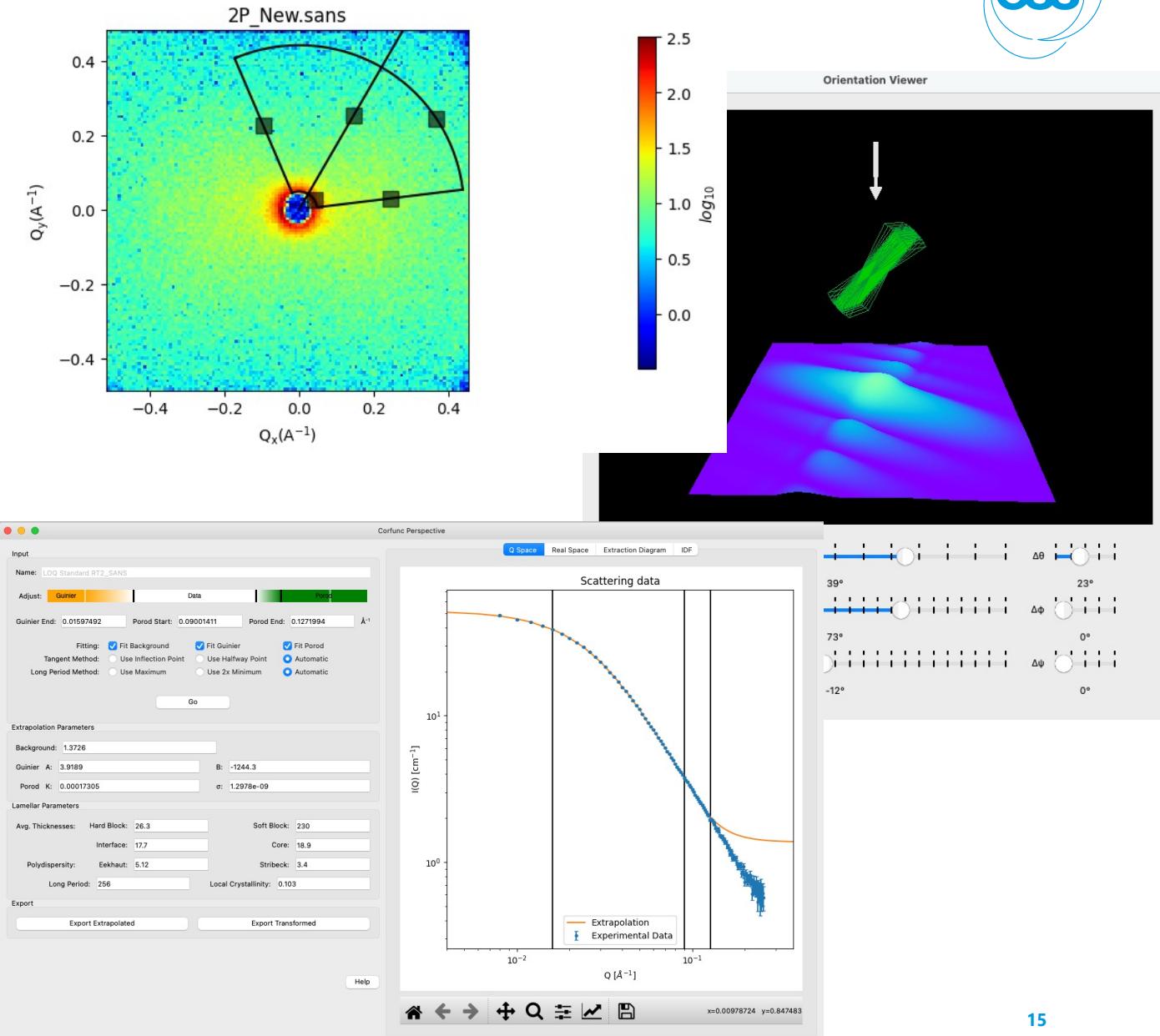
<https://www.tandfonline.com/doi/full/10.1080/10448632.2022.2126691>

<https://www.linxs.se/events/2023/1follow-up-magnetic-sans-data-and-software-h4crj>

New features in SasView

6.0.0-alpha release

- Orientation viewer available
- Corfunc perspective refactored
- Simultaneous fitting allows for a weighting scheme
- Preferences panel with display and plotting options
- Improved label handling on plots
- Residuals plots refactored
- PDB reader refactored
- Wedge slicer added
- Sasdata package separated
- Custom Model writing tutorial

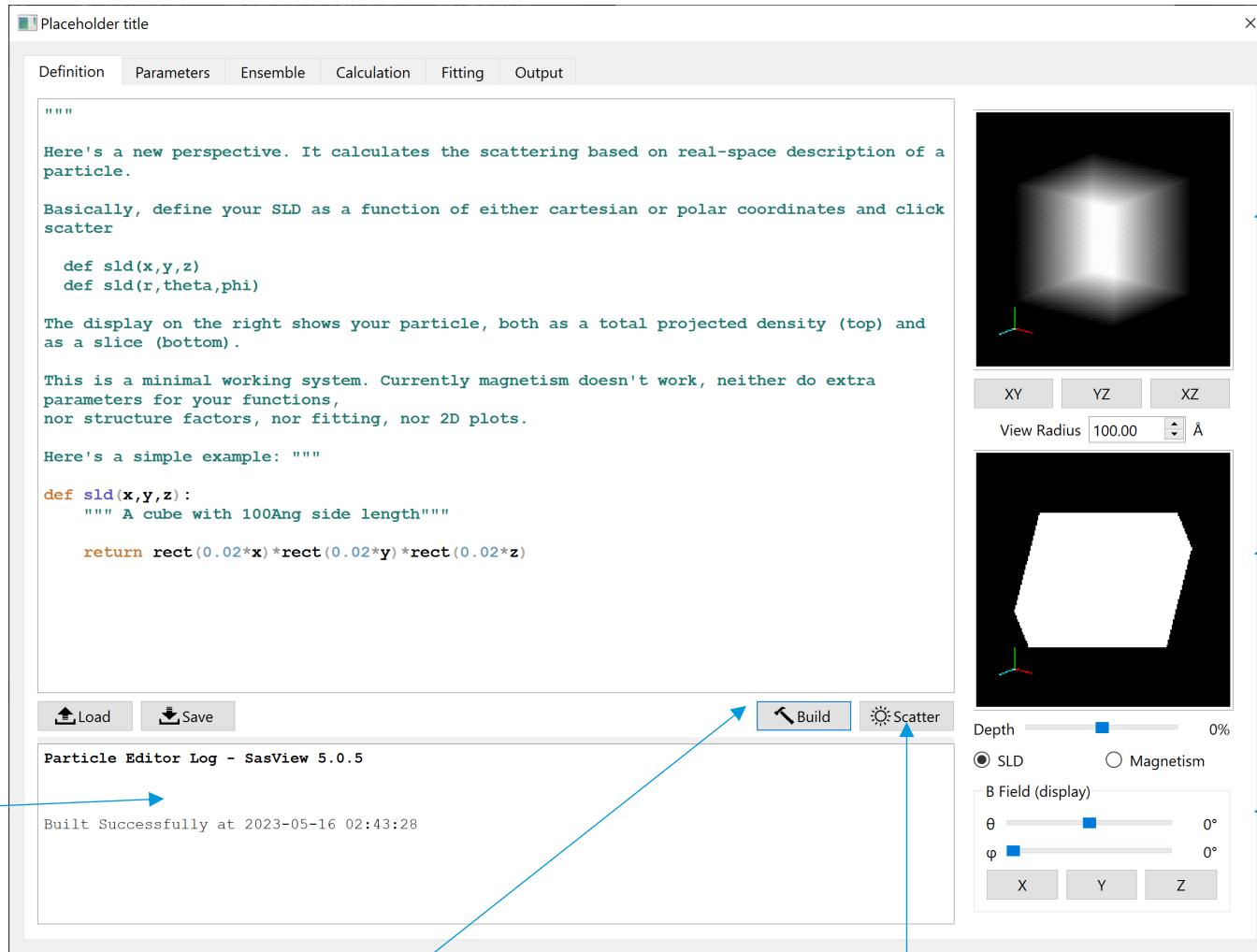


Particle Editor

Features beyond SasView-6.0.0



Define functions
sld and (optionally)
magnetism



Feedback on
code and
calculations

"Recompile" and
update display

Compute
scattering

sld / magnetism
"x-ray" projection

sld / magnetism
cross section

Magnetic field
controls for display

Beyond Graphical User Interface (GUI)



Running SasView from script

- Useful for batch jobs and reproducibility
- Scripts can be run on computer cluster

```
import pylab
from bumps.names import *
from sasmmodels.core import load_model
from sasmmodels.bumps_model import Model, Experiment
from sasmmodels.data import load_data

from bumps.fitters import fit
from bumps.formatnum import format_uncertainty

test_data = load_data('cyl_400_20.txt')
kernel = load_model('cylinder')

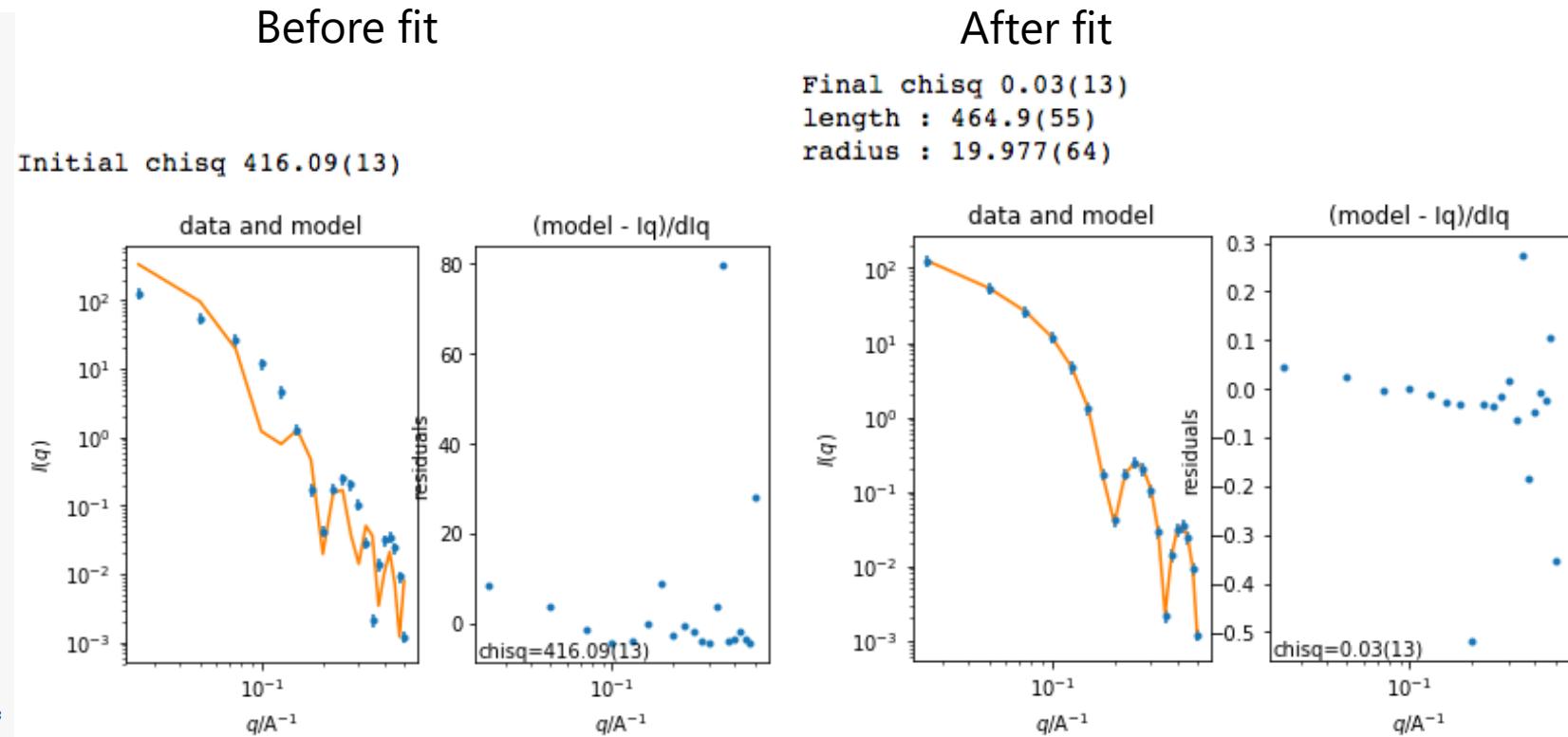
test_data.dy = 0.2*test_data.y

pars = dict(radius=35,
            length=350,
            background=0.0,
            scale=1.0,
            sld=4.0,
            sld_solvent=1.0)
model = Model(kernel, **pars)

# SET THE FITTING PARAMETERS
model.radius.range(1, 50)
model.length.range(1, 500)

M = Experiment(data=test_data, model=model)
problem = FitProblem(M)
print("Initial chisq", problem.chisq_str())
problem.plot()
pylab.show()

result = fit(problem, method='amoeba')
print("Final chisq", problem.chisq_str())
for k, v, dv in zip(problem.labels(), result.x, result.dx):
    print(k, ":", format_uncertainty(v, dv))
problem.plot()
pylab.show()
```

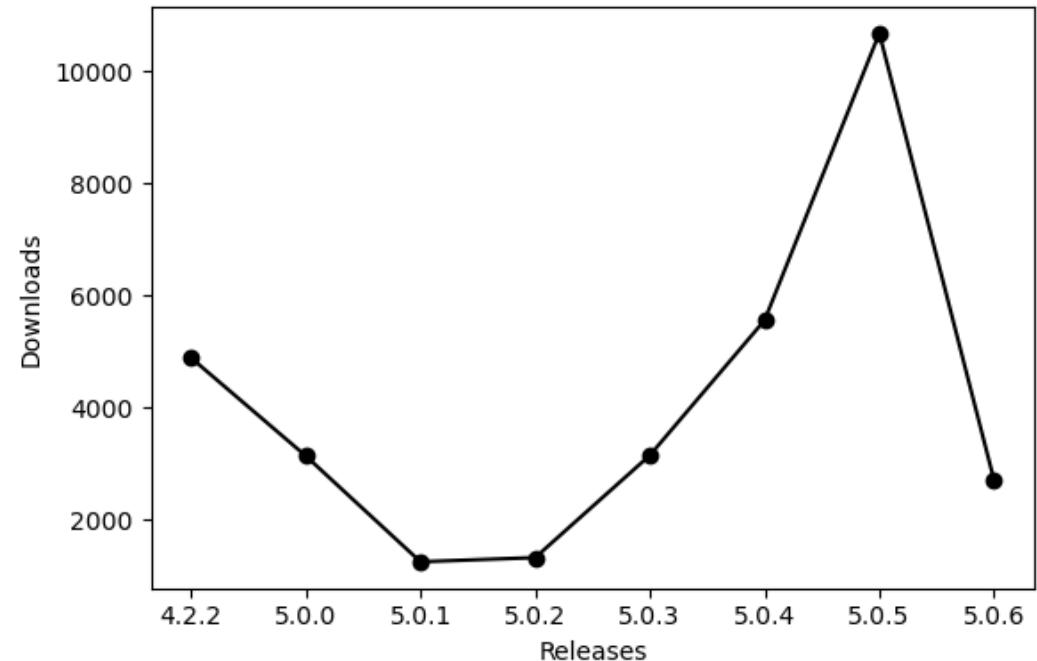


Pros and Cons of joining SasView



PROS

- You will join a friendly and international community
- You will have the opportunity to learn and discuss a technique and scientific computing,
- Your work will have an impact on a great number of users
- Develop your own skill set and improve your CV



Pros and Cons of joining SasView



PROS

- You will join a friendly and international community
- You will have the opportunity to learn and discuss a technique and scientific computing,
- Your work will have an impact on a great number of users
- Develop your own skill set and improve your CV

CONS

- The entry barrier for contribution is rather high
- Turnover (from an idea to release) is rather slow
- Need for accommodating requirements of a wide community
- Half-done projects stay usually in sandbox

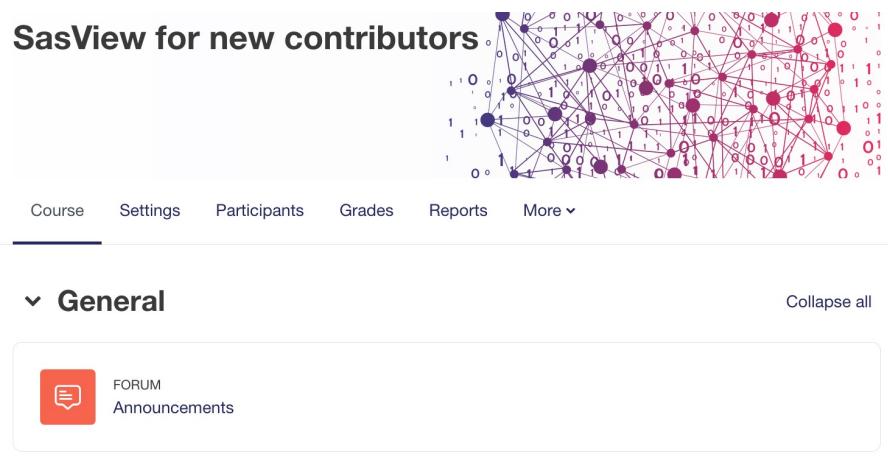
Reducing entry barrier

We are trying to do better



- Faster release cycles
- E-learning course
- Contributor Camp

SasView for new contributors



The screenshot shows the SasView for new contributors course interface. At the top, there's a navigation bar with links for Course, Settings, Participants, Grades, Reports, and More. Below the navigation, there are two expandable sections: 'General' and 'Get involved!'. The 'General' section contains a 'FORUM Announcements' box. The 'Get involved!' section contains text about the benefits of contributing and a note about discounted consultancy rates.

Get involved!

If you have signed up to this course you most likely already know why you are doing this. It may be still be worth reiterating basic context and rules. No matter whether you are a senior academic, an industrial researcher, or a graduate student, numerous studies have shown there are many benefits to joining collaborative software projects. See here for a great overview of 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!)

<https://e-learning.pan-training.eu/course/view.php?id=52>

Releases / v6.0.0-alpha

Release 6.0.0-alpha Pre-release

wpotrzebowski released this 3 weeks ago · 42 commits to release_6.0.0 since this release · v6.0.0-alpha · d4e3a80

New features

- Orientation viewer
- Corfunc refactored

CodeCampXII

Paul Butler edited this page on Sep 13 · 14 revisions

Contributor Camp XII Planning edit

After being derailed by the Covid19 lockdowns and ensuing craziness, this SasView Contributor Camp aims to reclaim the pre-pandemic community building momentum. The camp will bring together seasoned SasView contributors less seasoned contributors and new contributors from a variety of backgrounds to further the development needs of the SasView community. Activities (work below) will include writing/reviewing/editing documentation, testing and reporting bugs, reviewing/testing code or documentation developed at the camp (Pull Request reviews below), fixing bugs, adding enhancements, writing tutorials, creating video tutorials and training course development.

All are welcome, especially students and postdocs. For those brand new to the camp, we may arrange for some zoom training sessions ahead of time as needed to help get people up to speed ahead of time, depending on how they would like to participate. There is no charge to participate, however as a community project, all participants are responsible for their own travel and lodging.

Dates edit

The twelfth SasView Contributor Camp (formerly known as code camp) will be held in Newark DE, US, starting Jan 16 on the University of Delaware campus.

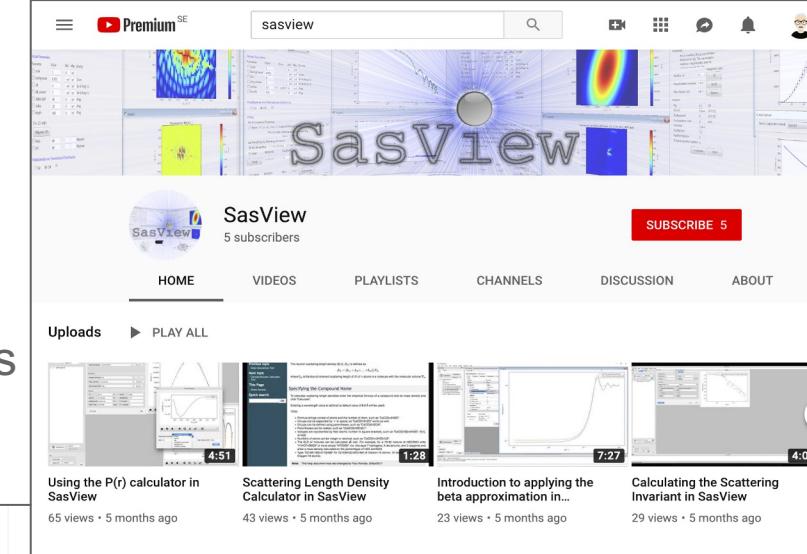
Save the date!
Jan 16th – 22nd 20204
University of Delaware, US

Other resources for users and developers



Documentation and communication channels

- Website
- Documentation
 - in-program & online
- Written Tutorials
- Video Tutorials (YouTube)
- Taught Courses
 - scattering schools/workshops
 - university courses
- Bootcamps & regional workshops
- e-Learning
- Slack
- Twitter
- (Marketplace)
- help@sasview.org
- users@sasview.org



The screenshot shows the SasView YouTube channel page. It features a banner image of the SasView software interface with various plots and data. Below the banner, there's a channel summary with the name 'SasView' and 5 subscribers, a 'SUBSCRIBE' button, and links to 'HOME', 'VIDEOS', 'PLAYLISTS', 'CHANNELS', 'DISCUSSION', and 'ABOUT'. Under the 'VIDEOS' tab, four video thumbnails are displayed with their titles and view counts: 'Using the P(r) calculator in SasView' (65 views), 'Scattering Length Density Calculator in SasView' (43 views), 'Introduction to applying the beta approximation in...' (23 views), and 'Calculating the Scattering Invariant in SasView' (29 views). At the bottom of the channel page, there's a sidebar with a circular profile picture, a title 'SasView @SasView', a description 'Software for analysis of small angle scattering modelling #SANS #SAXS.', and social media links for 'Go' and 'Quick search'.

FAQ

Here are the answers to some common questions about SasView

- What is SasView?
- What platforms does SasView run on?
- Is there a SasView Manual?
- Do I need to install Python/C++ or any compilers before I install SasView?
- What do xcrun errors mean?
- Can SasView make use of my GPU(s)?

[th SasView »](#)

Conclusions



SasView is an open, collaboratively developed software tool

There is a lot of development going on

Contributions are always welcome

We are all learning and need to understand boundary conditions

There are many ways to contribute

How to contribute

You don't have to be coding ninja to help out!

Respond to queries posted to help@sasview.org or github

Teach others how to get the best from SasView

Write and improve documentation

Write and record tutorials

Test SasView (over and over!) and write bug reports

Provide new plugin models

Deploy and improve automated testing

Review the code contributions of others

Develop code in Python3/C++ and PyQt/Matplotlib





Thank you!
On behalf of SasView community

SasView version 5.0.6

Doucet, Mathieu¹ [ID](#); Cho, Jae Hie²; Alina, Gervaise²; Attala, Ziggy³; Bakker, Jurrian⁴;
Beaucage, Peter⁵ [ID](#); Bouwman, Wim⁶; Bourne, Robert³; Butler, Paul¹⁵ [ID](#);
Cadwallader-Jones, Iestyn⁷; Campbell, Kieran⁸; Cooper-Benun, Torin³; Durniak, Celine⁹;
Forster, Laura¹⁰; Gilbert, Peter⁵ [ID](#); Gonzalez, Miguel⁷ [ID](#); Heenan, Richard³ [ID](#);
Jackson, Andrew⁹ [ID](#); King, Stephen³ [ID](#); Kienzle, Paul⁵; Krzywon, Jeff⁵ [ID](#); Maranville, Brian⁵ [ID](#);
Martinez, Nicolas⁷; Murphy, Ryan⁵ [ID](#); Nielsen, Torben⁹; O'Driscoll, Lewis³;
Potrzebowski, Wojciech⁹ [ID](#); Prescott, Stuart¹¹ [ID](#); Ferraz Leal, Ricardo¹; Rozyczko, Piotr⁹ [ID](#);
Snow, Tim¹⁰ [ID](#); Washington, Adam³; Wilkins, Lucas³; Wolf, Caitlyn⁵ [ID](#)

<https://zenodo.org/records/7581379>

