Writing Models in SasView

A Contributing to SasView workshop Tutorial

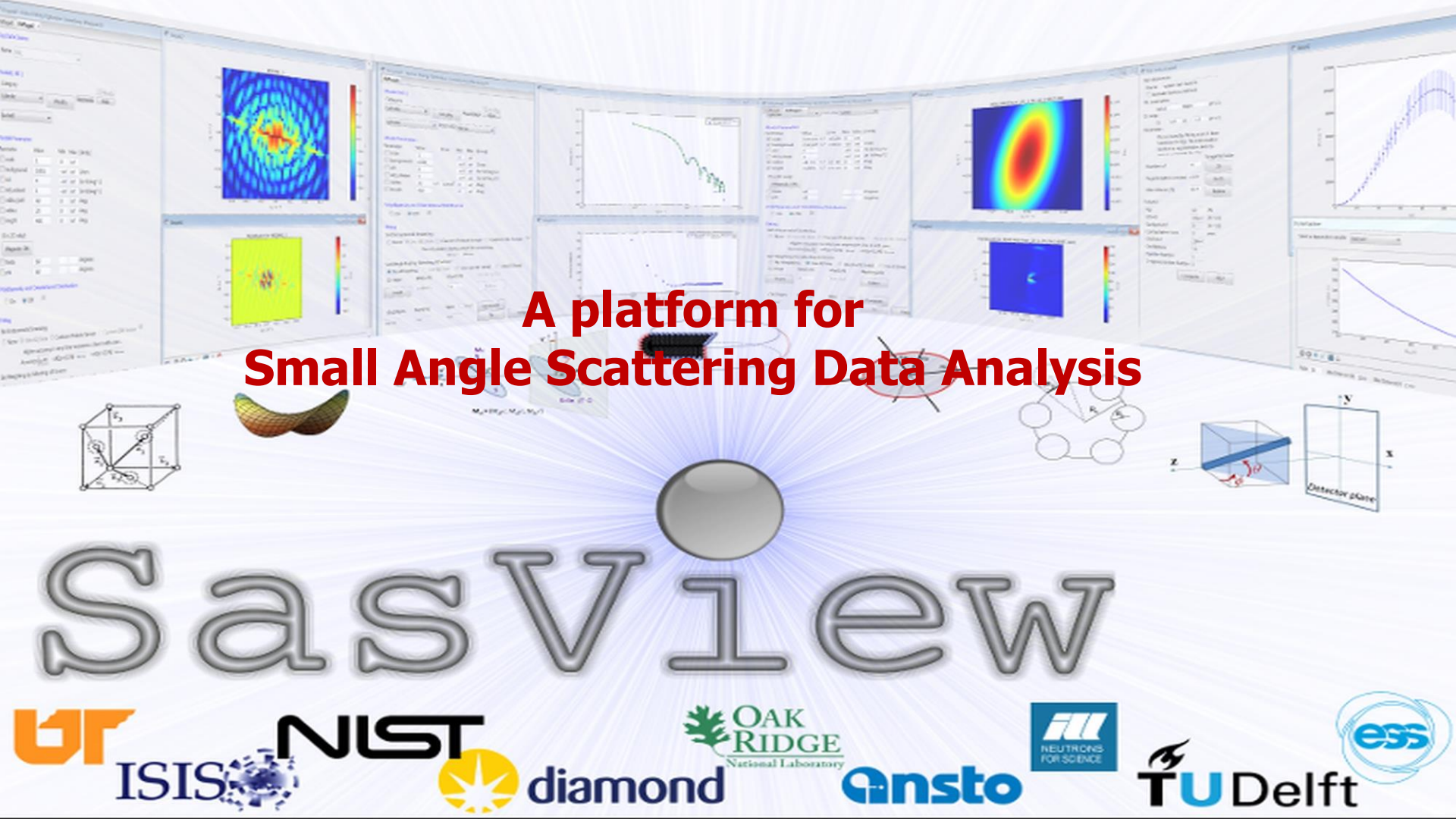
Paul Butler
Virtually Dec 9, 2021

Program

- Just in case … a bit about what SasView is today
  - What the program “is” and “is not”
  - What the program can do
  - SasView support resources available
- A bit of historical background and current status
- The SasView collaborative model
- Quizz: What is the shape of the future?
- Tutorial:
  - Very brief Introduction
  - a step by step walk through
  - from: general equation
  - To: reparameterization
  - … and everything in between?

For complete
See YouTube channel video by Steve King!!:
https://www.youtube.com/watch?v=Sbzf_wdlPnQ

Scripting Tutorial next Thursday, Dec 16
Led by Wojciech Potrzebowski


A platform for
Small Angle Scattering Data Analysis
SasView
UT
ISIS
NIST
diamond
OAK RIDGE
National Laboratory
ansto
NEUTRONS FOR SCIENCE
ess
TU Delft

SasView

Data Analysis eh?

…. So what exactly does that mean?

Only works on Reduced data

(All the instrumental artifacts are removed and only the science is left)

…. Sorta

Focus on analytical approaches for this package

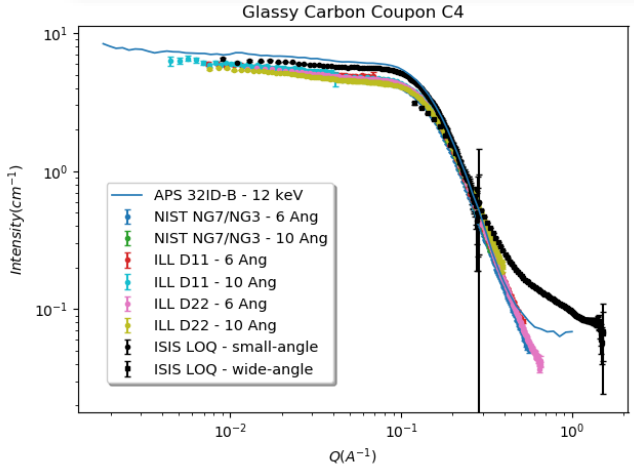
…. Sorta

Whatever anybody puts into it

…. Sorta

Experimental Data Flow

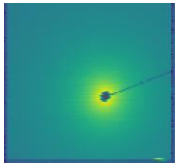
Reduced data should be reproducible anywhere!



http://www.cansas.org

Instrument-*dependent* data

Instrument-***in****dependent* data

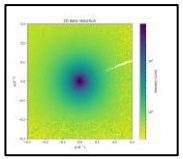


Data Acquisition System

'raw' data file (NXS)

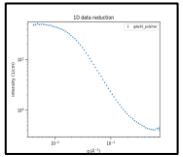
Data Reduction Programs (e.g. *Mantid*)

reduced data file (1D: HDF / XML / TXT; 2D: HDF / TXT)

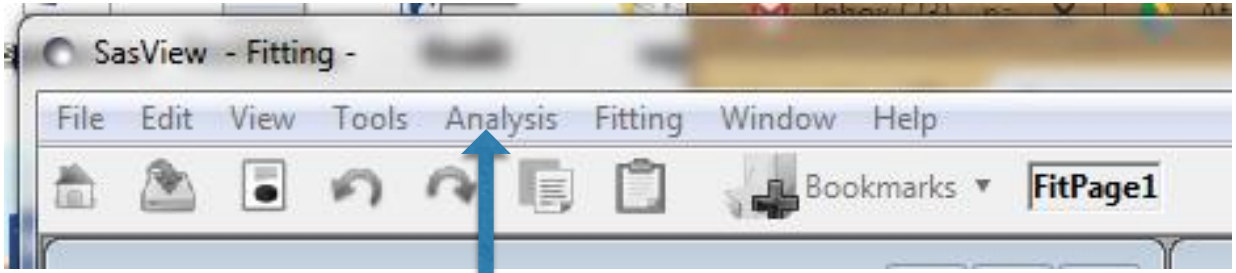


Data Analysis Programs

Paper / Thesis



Perspectives on the data



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

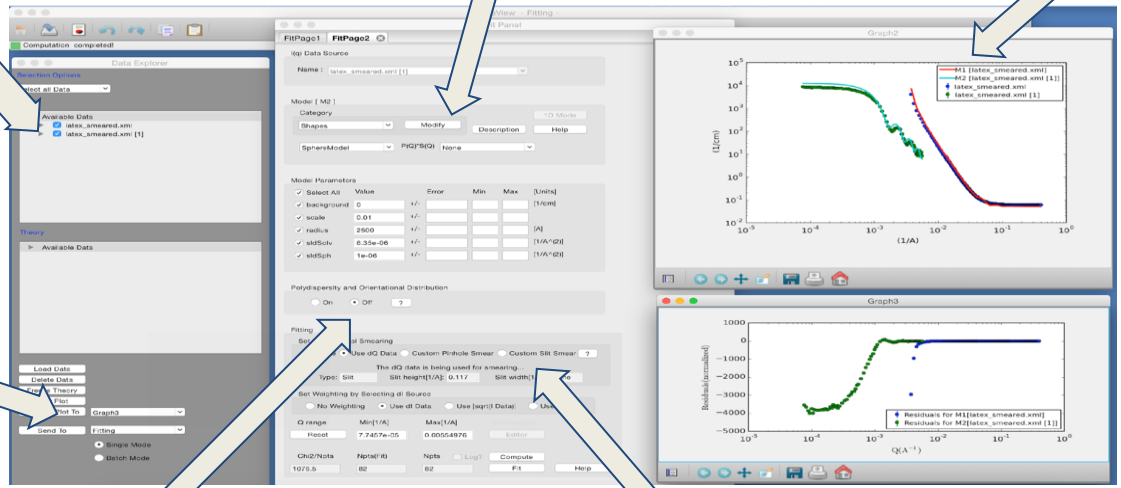
- Fitting
- Invariant
- Pr Inversion
- Correlation Function

1D Analysis

Data management
Common data formats supported, including NXCansas & *cansas1D*

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

Simultaneous fitting

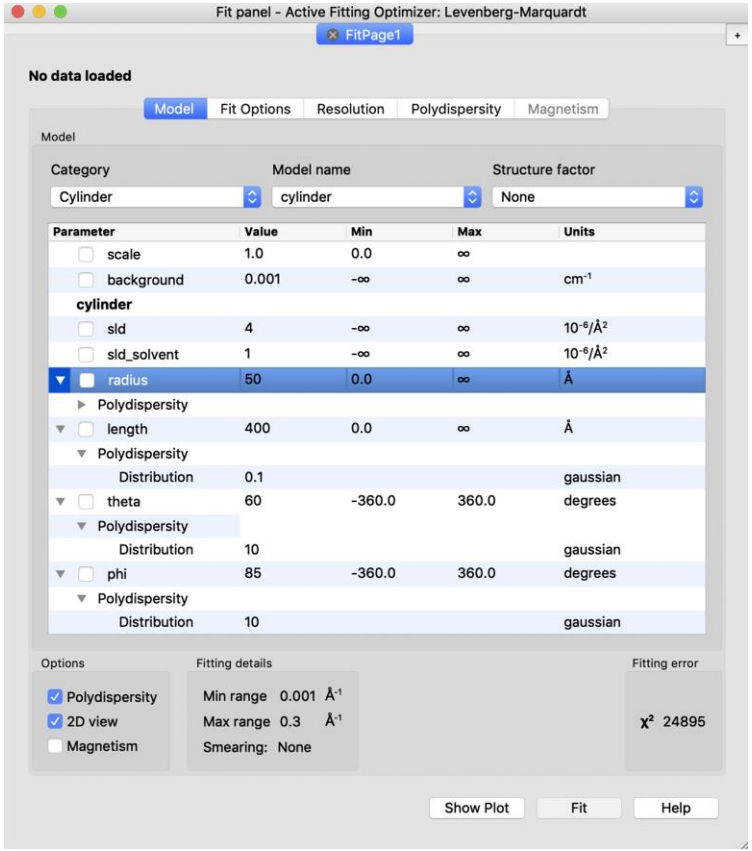


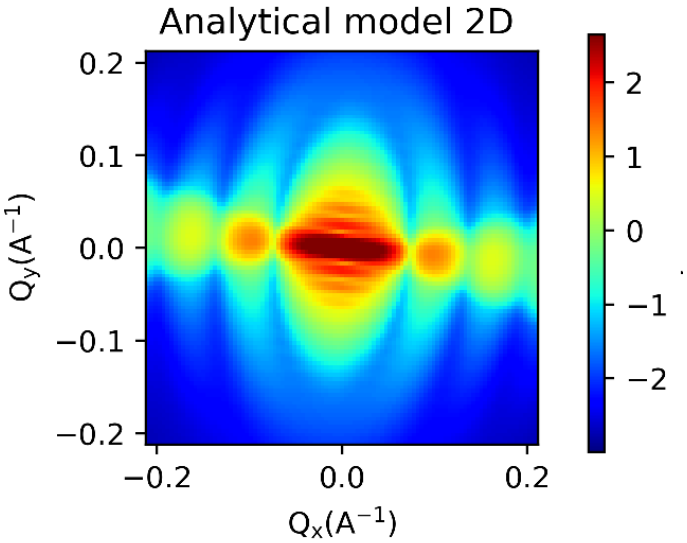
Analysis Tool Choice & Plotting

Generic parameter polydispersity
Choice of distribution and distribution parameters

Resolution smearing (pinhole and slit)
Automatically from data or provide parameters

2D Analysis



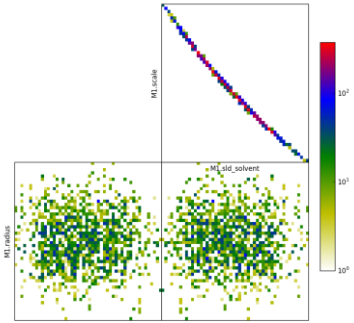


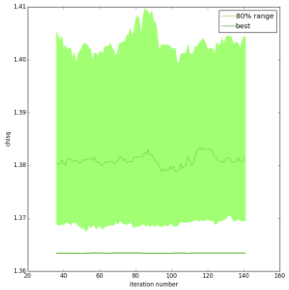
Orientational polydispersity = “jitter”

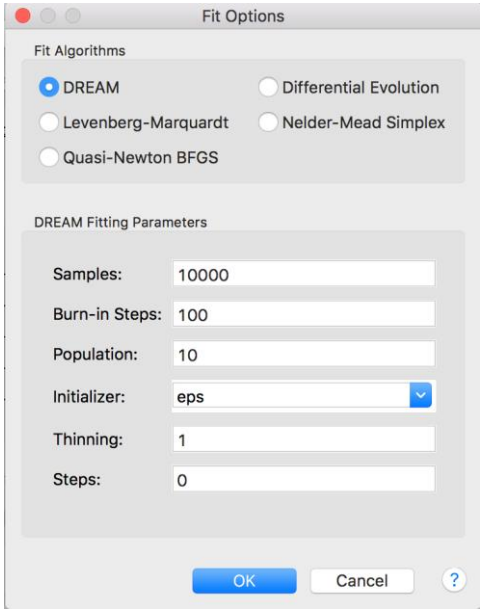
Decouples the frame for the object’s orientation with respect to the beam and the “jitter” around the axis of the object.

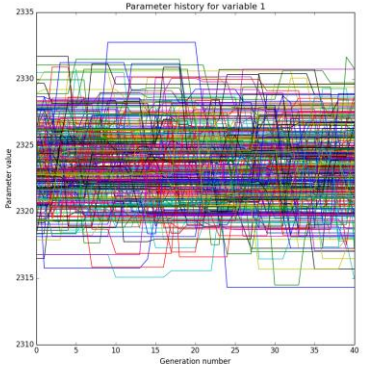
Turning on GPU Option highly recommended for fitting

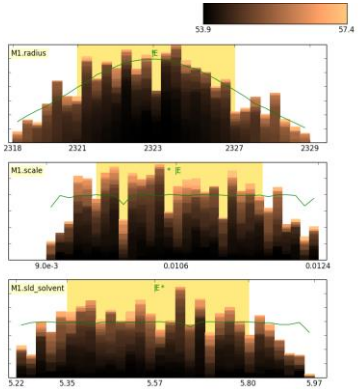
Fitting Algorithms











Uses bumps package from P. Kienzle

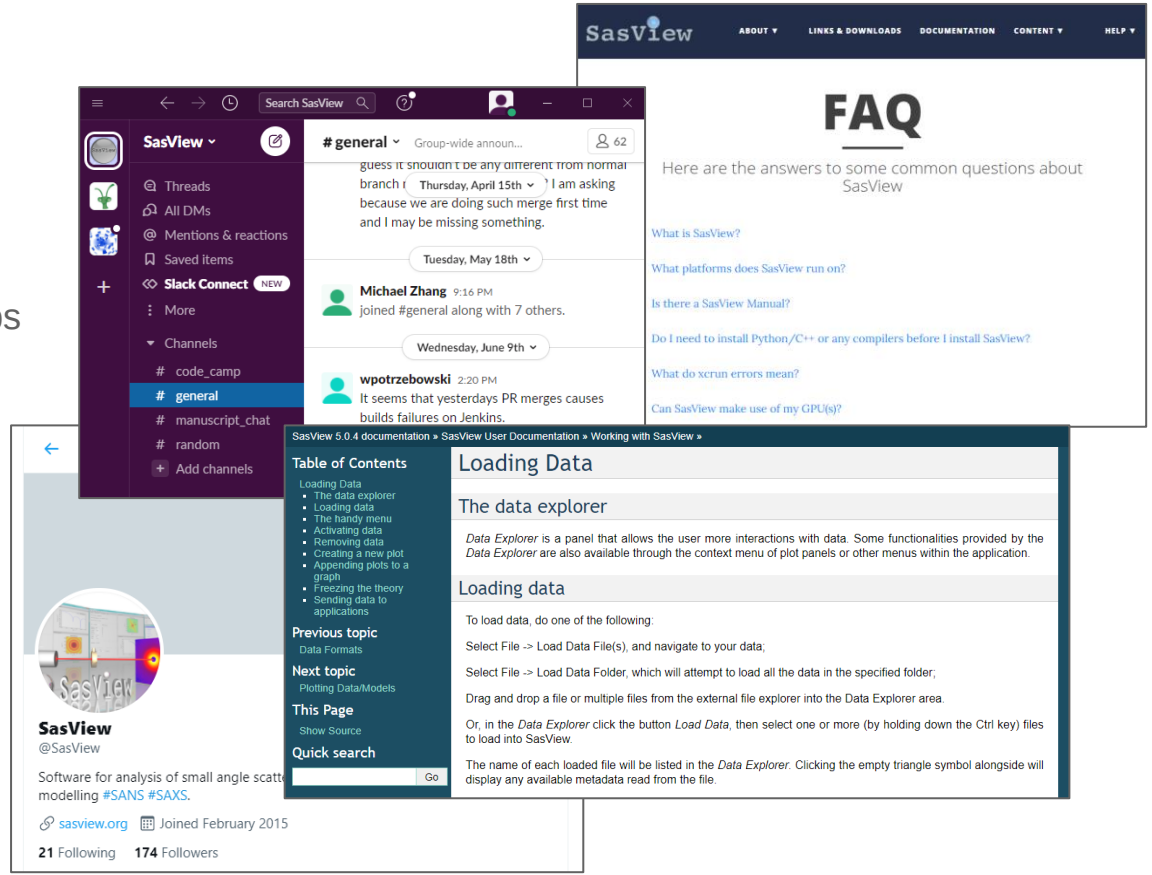
provides user choice of fitting engine

Conventional and Bayseian

Support Resources

- **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)

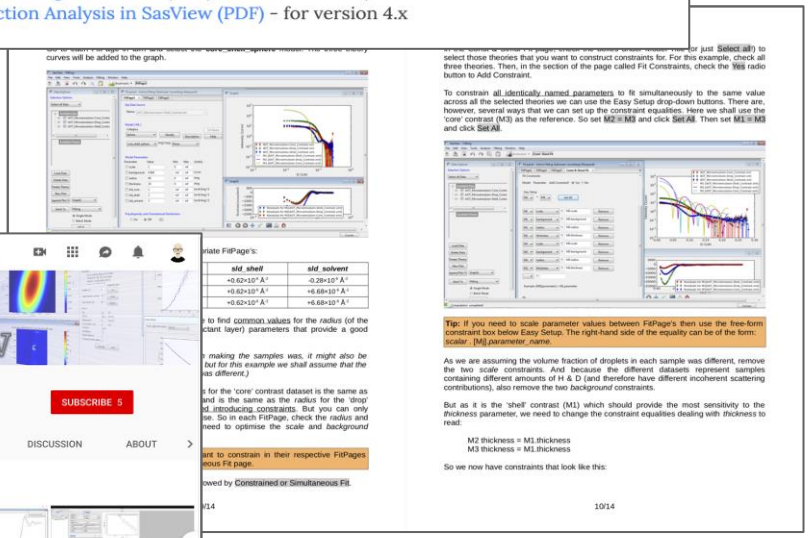
- **help@sasview.org**
- **users@sasview.org**

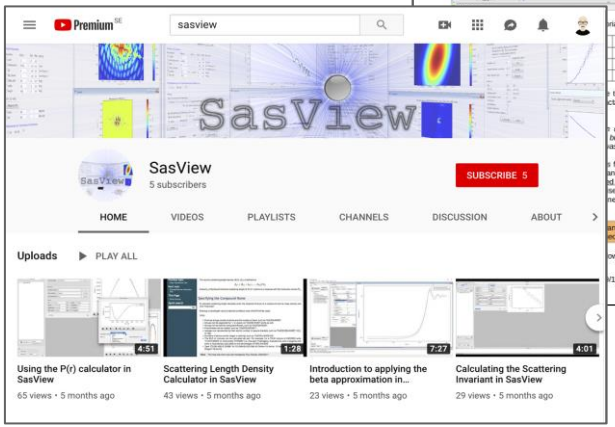


Support Resources

- 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





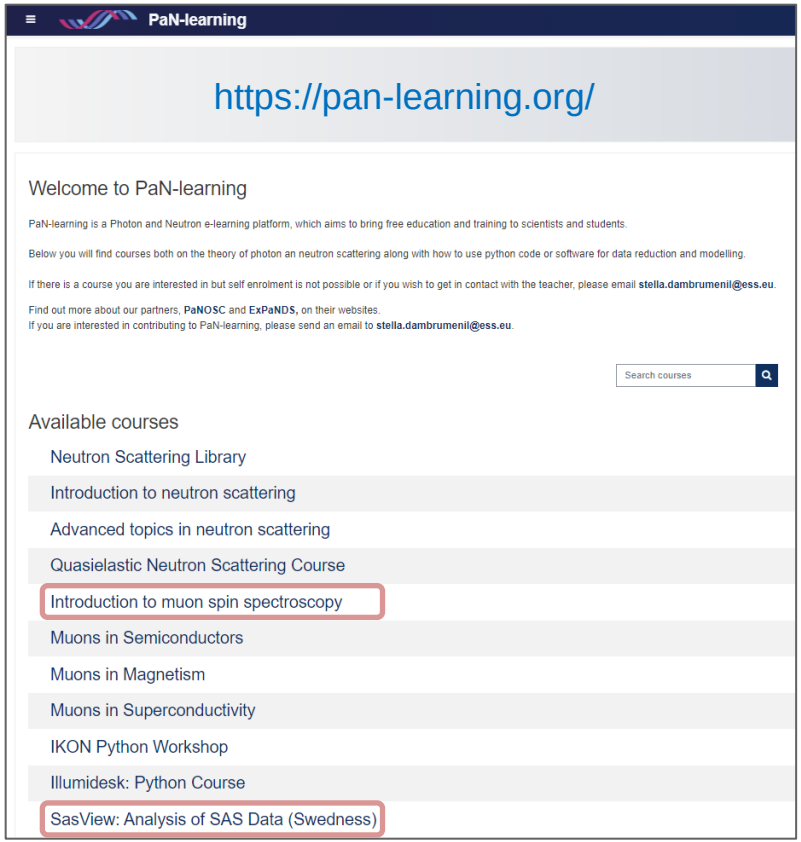
Support Resources

- 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

All the work of ISIS Sandwich Student Michael Oakley

moodle

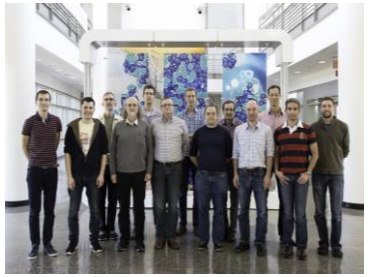


History and Status of SasView

- 2006; originates in NSF *DANSE* project
- 2013; transitions into a community project
- 2016; Sine2020 project funded
- 2021; Essentially a “volunteer army”

- ~40 contributors from 9 organisations (~15 active at any one time)

- small management team:
  Paul Butler (NIST), Wojciech Potrzebowski (ESS), Steve King (ISIS), [Mathieu Doucet (ORNL), Andrew Jackson (ESS)]



Scientific Software Development and The cyberinfrastructure revolution

- **Never enough resources to achieve the vision we have**
- **No resources for long term maintenance and support.**

Problem:

- To reap the benefit of investment in software developments requires foundational long term support.
- If entity that supports the development also must support the "maintenance" forever, the entity will soon cease to be able to fund new projects.

CONCLUSION: This paradigm is broken!!!

FACTS OF LIFE:

- **Resources are finite**
- **Needs are infinite**

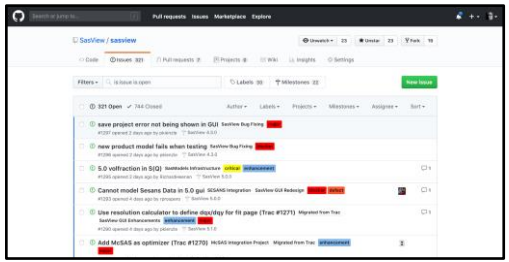
Open, Collaborative, Community **Development**

Code is open source and publicly hosted at Github
Released under BSD 3-clause license

DOI for **each release**

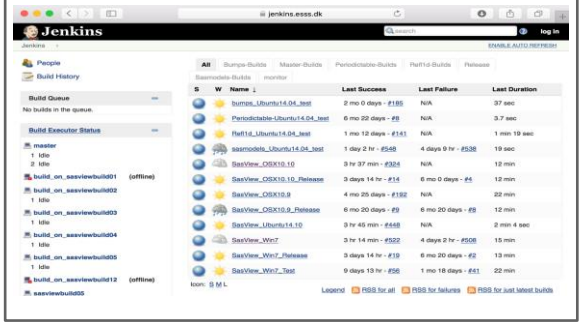
Rolling **5 Year Roadmap**

Code Hosting, **Issue Tracking** & Developer Wiki on Github



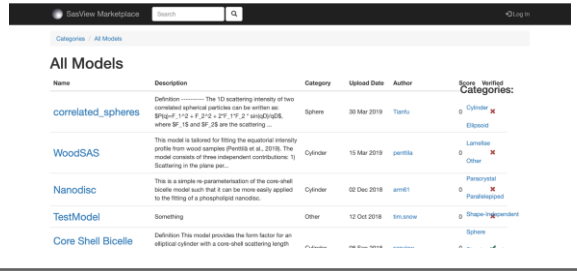
https://github.com/SasView

Automated Builds



http://build.sasview.org/

Model Marketplace for Users to share their models



http://marketplace.sasview.org

python™ Qt

https://www.sasview.org

matplotlib

Open, Collaborative, Community Development

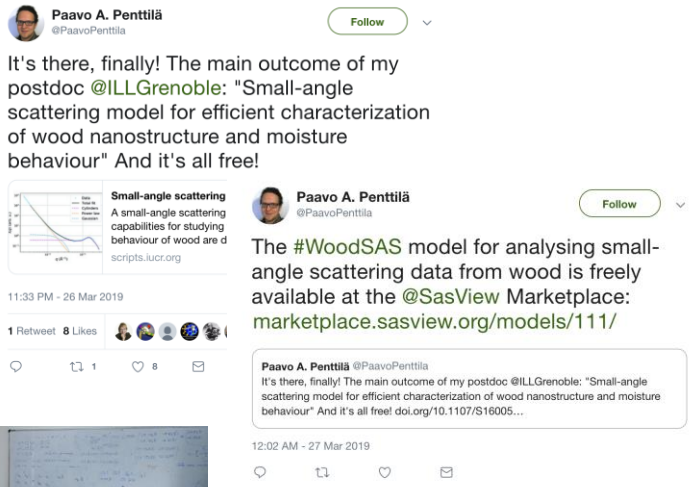
We work together towards common goals formulated through community input, with two guiding principles ...

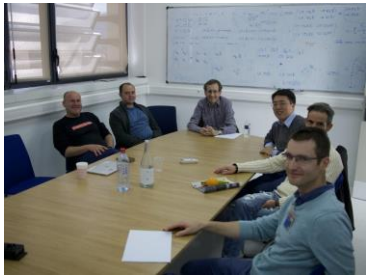
He who pays the piper …
Those who bring the resources (time and effort, or funds to buy time and effort) choose what to work on.

You break it, you bought it ...
You are not allowed to break what is already there for others. If you break it, you fix it.

Open, **Collaborative**, Community **Development**

Twice monthly zoom calls
Regular ‘camps’ & ‘hackathons’





Ask not what the community is going to do for you,
ask what you can do for the community

\- P. Butler, March 2019

http://www.sasview.org

http://github.com/SasView

Quiz time

What will the Future SasView be able to do?

What to expect

We will be bold (or foolish)
and use SasView 5.0.5 beta.

No prior experience is required. However, a knowledge of SAS would be helpful along with a knowledge of enough maths to write the necessary functions

These few slides are intended to provide a quick orientation on a couple of fundamental aspects and on the outline of the demo tutorial. They are intended to be followed immediately by a live, hands-on, demo/tour using SasView to write models.

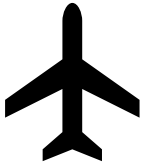
SOME FUNDAMENTALS

POSTULATE: creating a new model in SasView is EASY

- There is ZERO difference between built in model files and custom/plugin model files.
  - But there are some differences in how the GUI handles them…
- You do NOT need anything but SasView to create a model
  - Though it may help for complicated things ….
- You must know:
  - The equation you want to use (I(Q) = ….. what?)
  - What are the adjustable parameters in that model
  - …. And then the complexities of polydispersity, orientation, integer parameters, “multiplicity” etc.

MORE FUNDAMENTALS

POSTULATE: creating a new model in SasView is EASY



There are three types of model files that can be used (ALL include *mymodel.py*). We will only cover two of them.

NOTE: SasView provides “magic” and tools specific to SAS data, and the models and GUI make assumptions about the data being SAS data. *HOWEVER*, fundamentally, SasView can fit/model any data with any equation that can be given analytically (DLS? Reflectivity?)

NOTE3: SasView 4.2 is no longer being worked on and will soon be obsolete.

THE PLAN

POSTULATE: creating a new model in SasView is EASY

- Write a simple python model (MODEL TYPE I) with no polydispersity
- Rewrite using special functions (and even other packages, e.g.scipy)
- Add polydispersity and the form_volume function
- Enable use with structure factor (effective radius)
- Rewrite in C (MODEL TYPE II and III)
- Using (and creating) C library functions
- Using cylinder model to look at orientation.
- Use of various other flags
- MAYBE .. multiplicity and Reparameterization…. A powerful new tool
- How to make the model available immediately to the community