EUROPEAN
SPALLATION
SOURCE

Can we work together on data analysis?

Wojciech (Wojtek) Potrzebowski

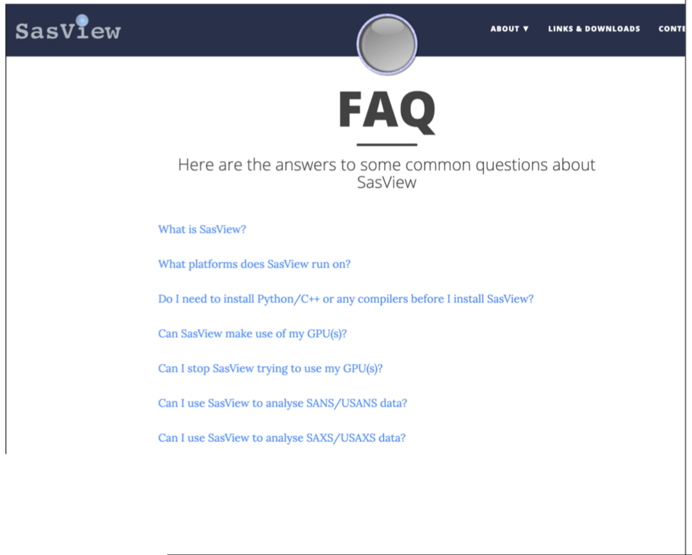
European Spallation Source ERIC

SasView as an example for collaborative data analysis software development

- SasView open, collaborative, community development platform for SAS data analysis
- SasView originates from NSF funded project DANSE (2006)
- Turned into Community project in 2013
- Supported by 9 facilities, 40 contributors (~15 active at any one time)
- Management team P. Butler (NIST), M. Doucet (ORNL), A. Jackson (ESS), S. King (ISIS)
- Biweekly calls
- Regular code camps
- External funding e.g. SINE2020

SasView resources, education and outreach

- Website
- Documentation
- Written Tutorials
- Video Tutorials (YouTube)
- Taught courses
  - Scattering schools
  - University courses
- E-learning
- Twitter
- Slack
- Mailing Lists
- Bootcamps & Regional Workshops
- (Marketplace)



FAQ

Here are the answers to some common questions about SasView

What is SasView?

What platforms does SasView run on?

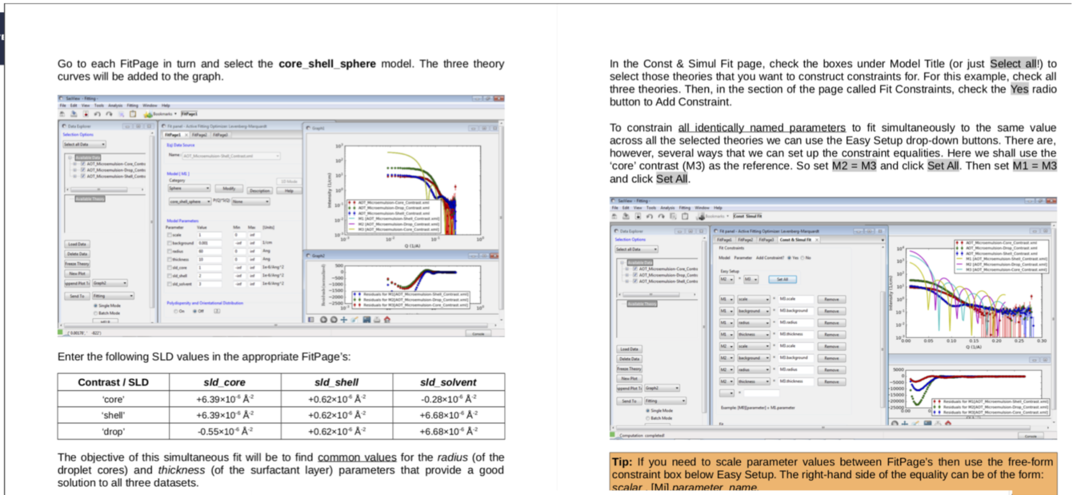
Do I need to install Python/C++ or any compilers before I install SasView?

Can SasView make use of my GPU(s)?

Can I stop SasView trying to use my GPU(s)?

Can I use SasView to analyse SANS/USANS data?

Can I use SasView to analyse SAXS/USAXS data?



Go to each FitPage in turn and select the **core_shell_sphere** model. The three theory curves will be added to the graph.

Enter the following SLD values in the appropriate FitPage's:

Contrast / SLD	*sld_core*	*sld_shell*	*sld_solvent*
'core'	+6.39×10^{-6} Å^{-2}	+0.62×10^{-6} Å^{-2}	-0.28×10^{-6} Å^{-2}
'shell'	+6.39×10^{-6} Å^{-2}	+0.62×10^{-6} Å^{-2}	+6.68×10^{-6} Å^{-2}
'drop'	-0.55×10^{-6} Å^{-2}	+0.62×10^{-6} Å^{-2}	+6.68×10^{-6} Å^{-2}

The objective of this simultaneous fit will be to find common values for the *radius* (of the droplet cores) and *thickness* (of the surfactant layer) parameters that provide a good solution to all three datasets.

(Depending on how meticulous the p
possible to find a common scale param
volume fraction of droplets in each sam

We now need to tell SasView that the

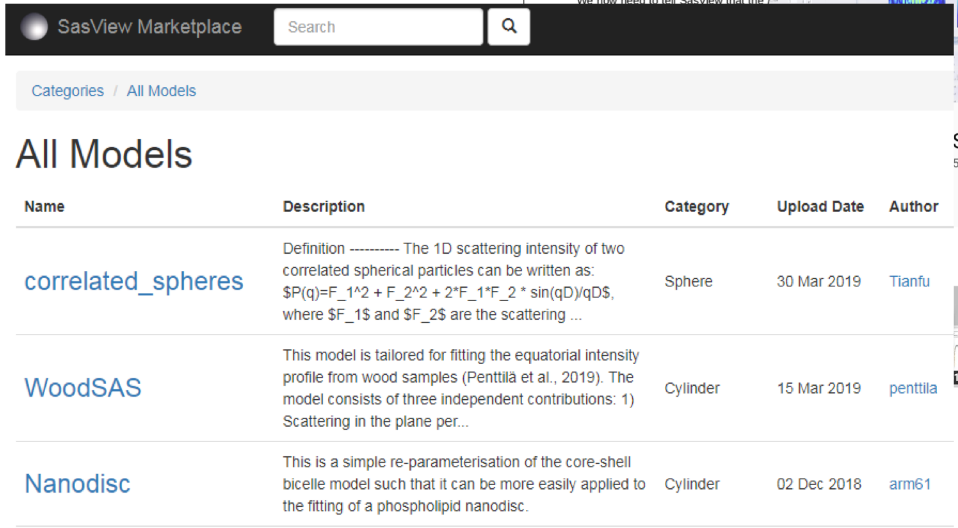
In the Const & Simul Fit page, check the boxes under Model Title (or just Select all!) to select those theories that you want to construct constraints for. For this example, check all three theories. Then, in the section of the page called Fit Constraints, check the Yes radio button to Add Constraint.

To constrain all identically named parameters to fit simultaneously to the same value across all the selected theories we can use the Easy Setup drop-down buttons. There are, however, several ways that we can set up the constraint equalities. Here we shall use the 'core' contrast (M3) as the reference. So set M2 = M3 and click Set All. Then set M1 = M3 and click Set All.

Tip: If you need to scale parameter values between FitPage's then use the free-form constraint box below Easy Setup. The right-hand side of the equality can be of the form: *scalar* . [Mi].*parameter_name*.

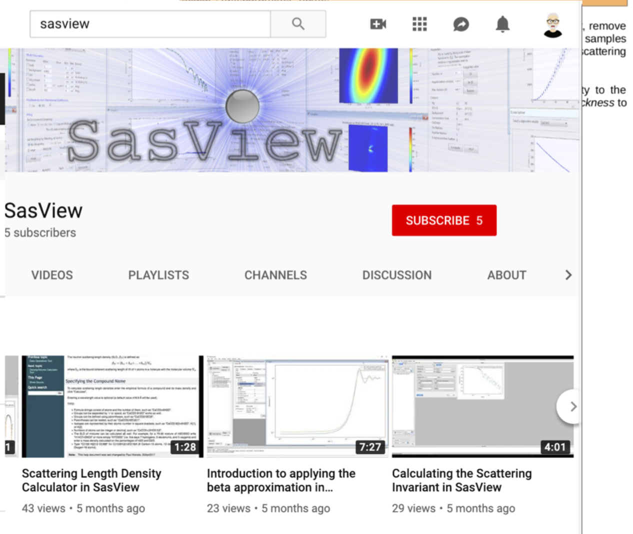
, remove
samples
scattering

ty to the
ckness to



All Models

Name	Description	Category	Upload Date	Author
correlated_spheres	Definition ---------- The 1D scattering intensity of two correlated spherical particles can be written as: $P(q)=F_1^2 + F_2^2 + 2*F_1*F_2 * sin(qD)/qD$, where F_1 and F_2 are the scattering ...	Sphere	30 Mar 2019	Tianfu
WoodSAS	This model is tailored for fitting the equatorial intensity profile from wood samples (Penttilä et al., 2019). The model consists of three independent contributions: 1) Scattering in the plane per...	Cylinder	15 Mar 2019	penttila
Nanodisc	This is a simple re-parameterisation of the core-shell bicelle model such that it can be more easily applied to the fitting of a phospholipid nanodisc.	Cylinder	02 Dec 2018	arm61



SasView
5 subscribers

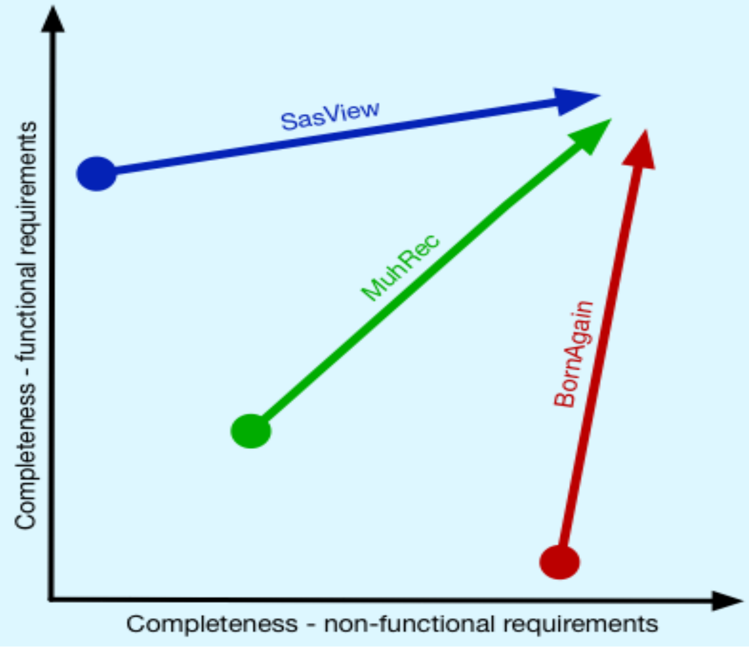
Scattering Length Density Calculator in SasView
43 views • 5 months ago

Introduction to applying the beta approximation in...
23 views • 5 months ago

Calculating the Scattering Invariant in SasView
29 views • 5 months ago

Lessons learned from SasView

- SasView has engaged user community that always asks for more – which is good!
- But not everything can be done with limited resources
- Therefore right model for engaging community is critical
- The balance between functional and non-functional requirements (e.g. code quality) is critical
- Non-functional requirements are not scientifically exciting but are important. One either needs a lot of resources or better start from the right place

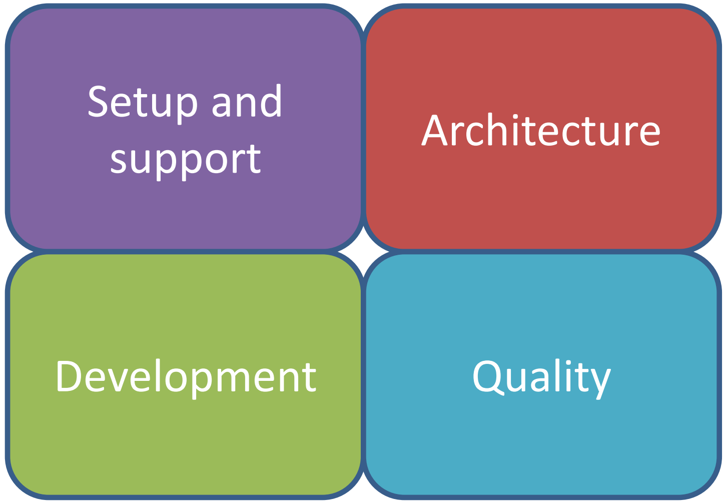

SasView
MuhRec
BornAgain
Completeness - functional requirements
Completeness - non-functional requirements

Discussion topic 1

- Do we need SasView like platform for reflectometry software?
- How do we engage user community?
- What can we share between software packages e.g. model library?
- How can we start? Is there a proper funding model?

Standards and Guidelines in SINE2020

- SINE2020 is EU founded project including work package on data treatment (WP 10)
- Guidelines were derived initially from a questionnaire to the neutron community
- Based on five different neutron/muon data analysis software tools: BornAgain, Mantid, McStas, Muhrec, SasView
- Addresses standards and guidelines in four areas:



Setup and support

Architecture

Development

Quality

- Planned to continue/update standard and guidelines and have annual meetings

Software setup, communication and support

- As a project grows the advice is to apply a governance model.
- Use an Open Source license, such as GPL.
- Provide user support, as a minimum through email support.
- Provide version controlled user documentation.
- Have regular releases, and aim for a minimum of once a year.
- With every new release, as a minimum, email out release notes to user mailing-list.
- Support multiple operation systems.

Software architecture

- Design software to be modular and to support plug-ins
- Where software needs to support scripting, the preference is for Python scripting
- Where the software uses a GUI, the advice is to use a recent version of Qt
- Design software to be able to accommodate different data formats.
- Support NeXus/HDF5 data formats.

Software development

- Use the programming languages C++ and/or Python.
- Factor into your project plan that you will be doing maintenance work on your code (~10-30%).
- Using a development methodology is advised, such as an agile methodology.
- Use Git for version control.
- Use a build tool, such as CMake.
- Use tool that makes user installation easy
- Use an issue/ticketing system, such as Github.
- Provide version controlled developer documentation.

Software quality

- Use a tool for continuous integration and automated testing, such as Jenkins.
- For software with a large user base, it is advised to have a user testing period prior to releases.
- Do code reviews, using a tool such as GitHub.
- Use static and/or dynamic analysis tools at least occasionally.
- It is advised to follow a coding standard.

- More at https://epubs.stfc.ac.uk/work/43005596 (Credits Anders Markvardsen)

Discussion topic 2

- Would it be useful to have similar guidelines for reflectometry software?
- What is missing or unnecessary?
- Should we also standardize on input for user requirements i.e. as a part of communication and support? If yes, how can achieve it e.g. github issues?
- Should we align with Standard and Guidelines annual meeting e.g. invite a representative to SXNS meeting in Lund (June 2020)?