

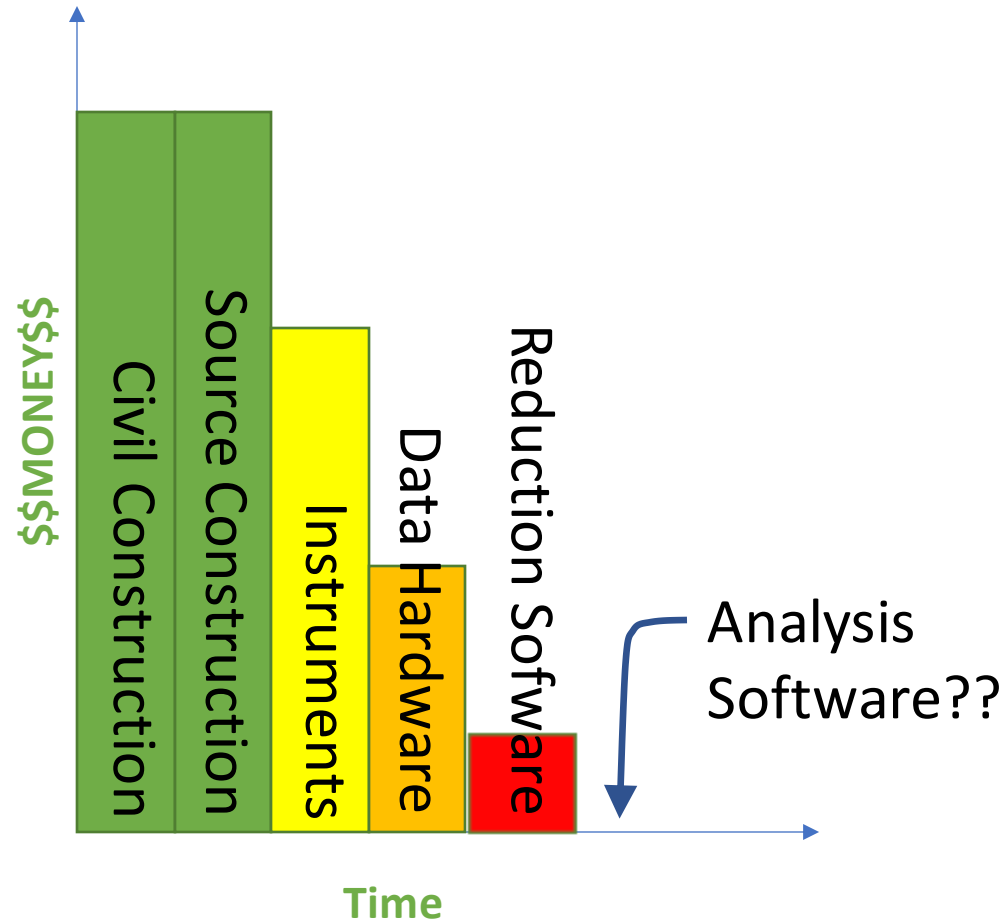
Addressing the Resource Problem Through Community

SasView

The resource Problem Part I

Analysis? Let's define that

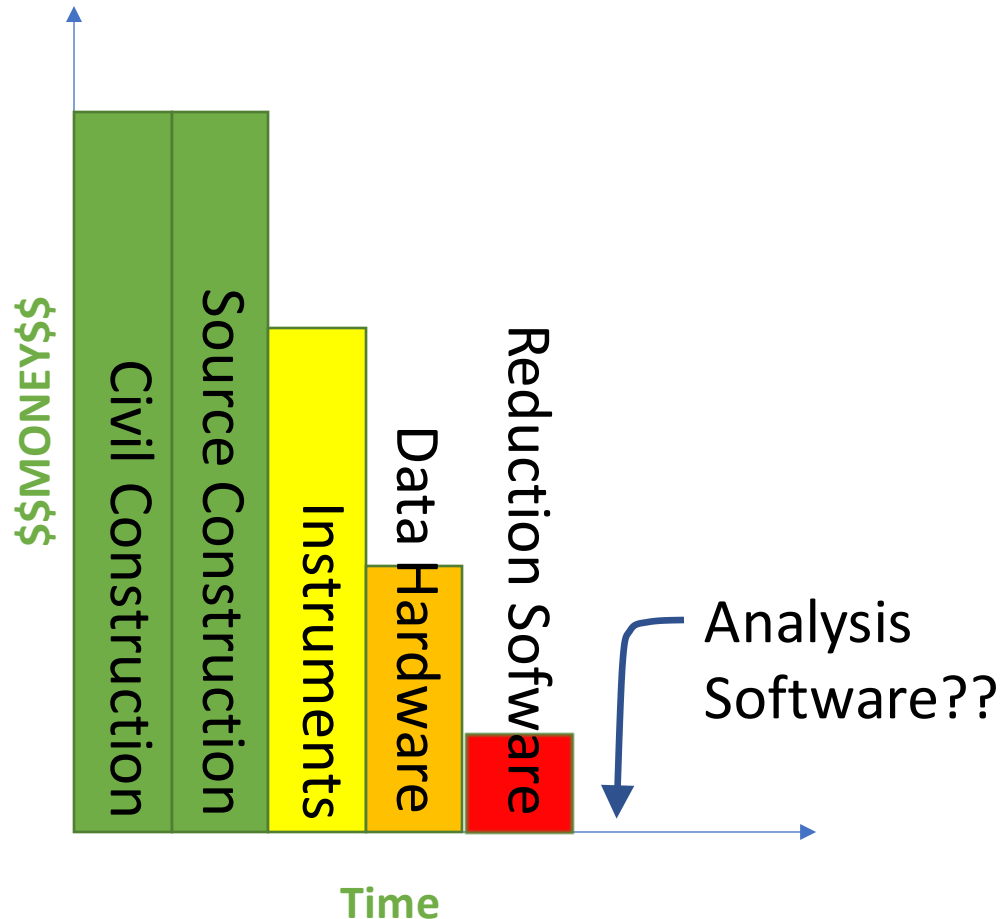
"Works on data from which all
instrumental artifacts have been
removed"
.... sortof



Instrument Independent Data

The resource Problem Part I

Analysis Software - Who's Job is it Anyway?



Scattering is an analysis tool and part of providing the tool should be the necessary software

→ the FACILITY'S JOB

Analysis is where the science is

→ the USER'S JOB

Or maybe

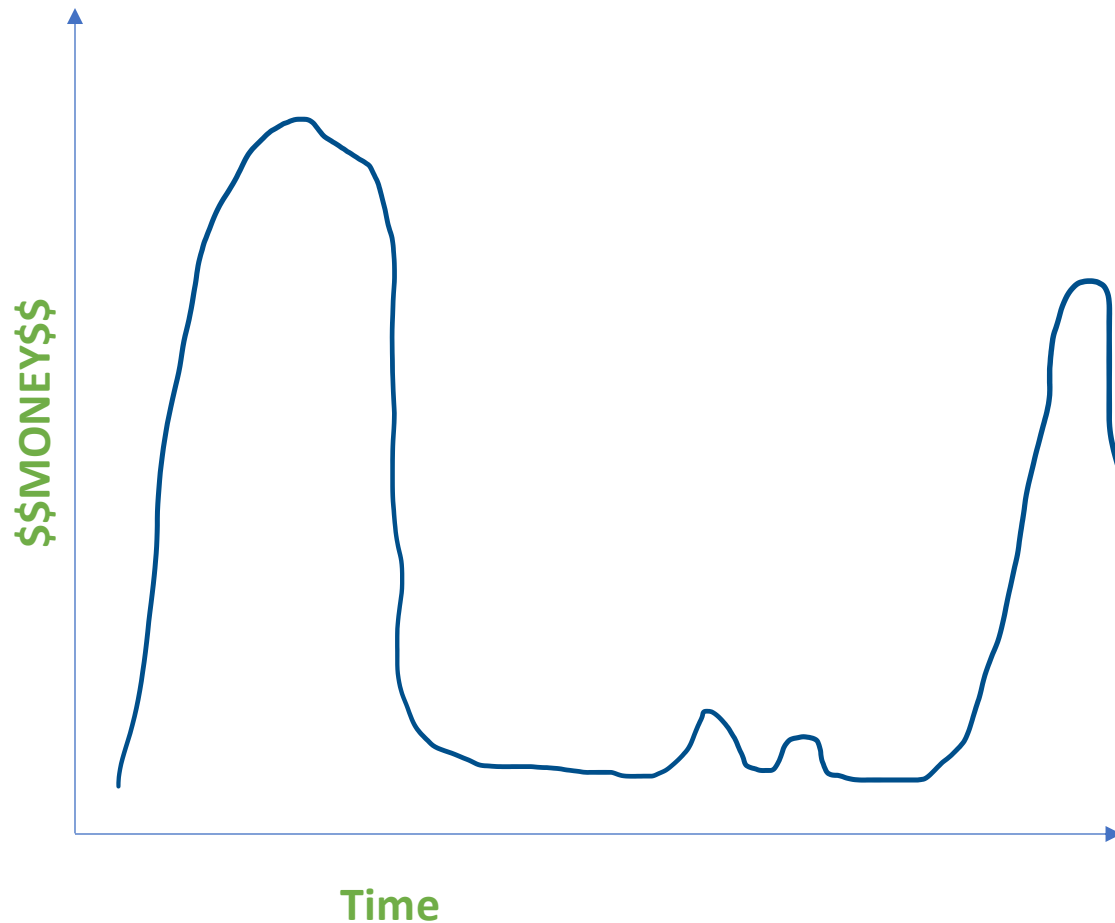
→ The Domain Science funding agencies?

The resource Problem Part II

The feast and famine roller coaster

The Valley(s) of death

Analysis Software - Who's Job is it Anyway?
- *The domain science funding agencies*



Facility directors discretion, NIH, NSF, DOE, etc.

Special funding (grants) do no fund long term maintenance and ongoing development. They fund “big new (transformative) ideas”

The resource Problem Part III

The unbounded problem

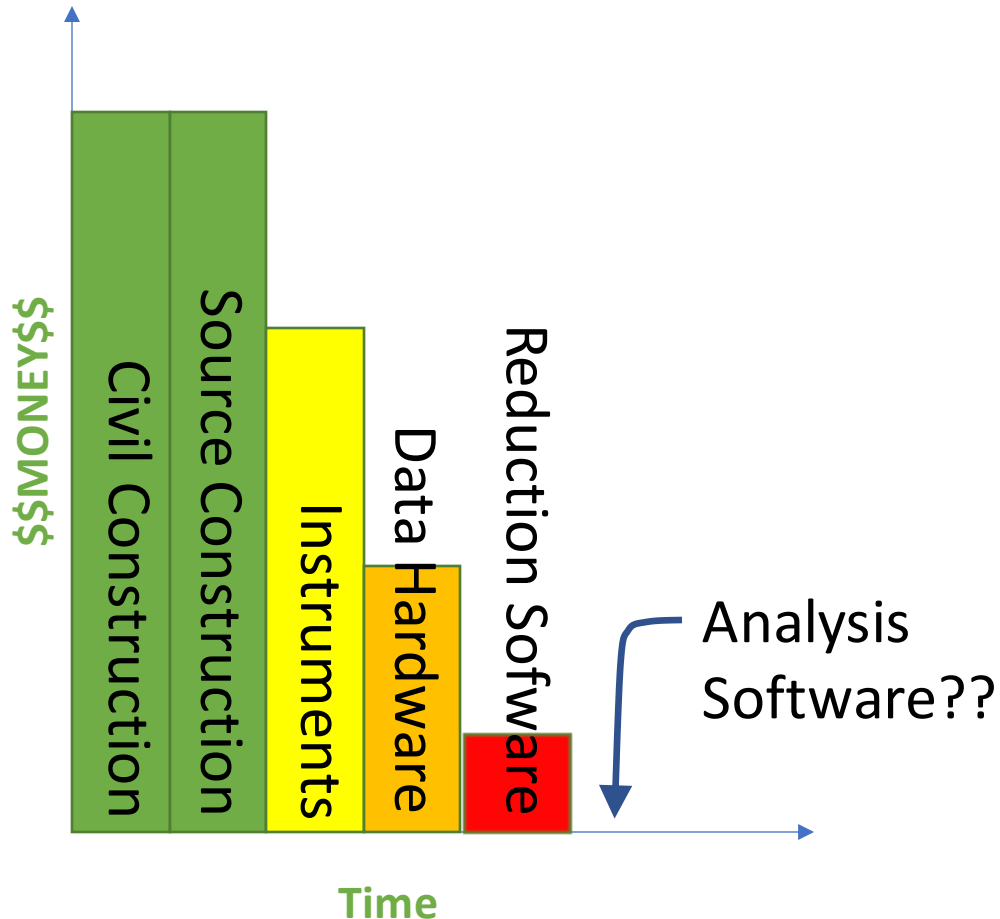
Fundamentally all these elements are relatively well defined problems...

EXCEPT ...

Analysis Software is really unbounded. The needs are nearly infinite and constantly evolving.

HOWEVER --- Analysis as defined here is also uniquely universal and ripe for collaborative pooling of finite resources

.... But beware the monoculture?



The Problem Statement

- A way to focus limited resources on top priorities (most useful to the science being done) in a world of infinite possibilities
- A way to harness funding for bold new ideas without losing the investment in the valleys of death
- A way to provide sustained maintenance and development in an uncertain funding environment (thriving through the famines)

FACTS OF LIFE:

- Resources are finite
- Needs are infinite

So, How Can we Solve the Problem?

Step One: Collaboration.
Let's all work together for the common good

The Reality: Collaboration is REALLY HARD!!

Lesson 1: Work with me NOT for me

Serendipity: Some history

2006



SansView was DANSE project output
~ 8.5% of funds were for SANS
+ BUMPS ... see later

2011

2012

NIST Supported initial transition from NSF funding

2013

Transition to Community project.

1st Code Camp at NIST April 2013

2014

2nd Code Camp at ISIS April 2014

2015

Move to GitHub

Rename to SasView

v3.0 released

v3.1 released

3rd Code Camp at ESS Feb 2015

2016

v4.0 released

4th Code Camp at TU Delft March 2016

5th Code Camp at ORNL Oct 2016

2017

v4.1 released

6th Code Camp at ILL/ESRF April 2017

7th Code Camp at DMSC October 2017

8th Code Camp at ESS Sept 2018

2018



v5.0b1 released

v4.2 released

v4.2.1 released


v5.0b2 released

1st SasView User Meeting at SAS2018

2019

9th Code Camp at ILL/ESRF March 2019

Some More Lessons Learned

- Nobody owns  fewer egos
- No proprietary platform (open source python – eventually some C)
- Must align with the needs and the priorities of each person/facility
- Community building is integral/essential to the effort
- "Small" money can be blessing
 - More freedom for experimentation
 - Provides better motivation for collaboration

Getting to collaboration was a huge amount of work but there was also a LOT of luck



The SasView Approach

An
“open, collaborative, community development”
platform for
Small Angle Scattering Data Analysis



Open, Collaborative, Community Development

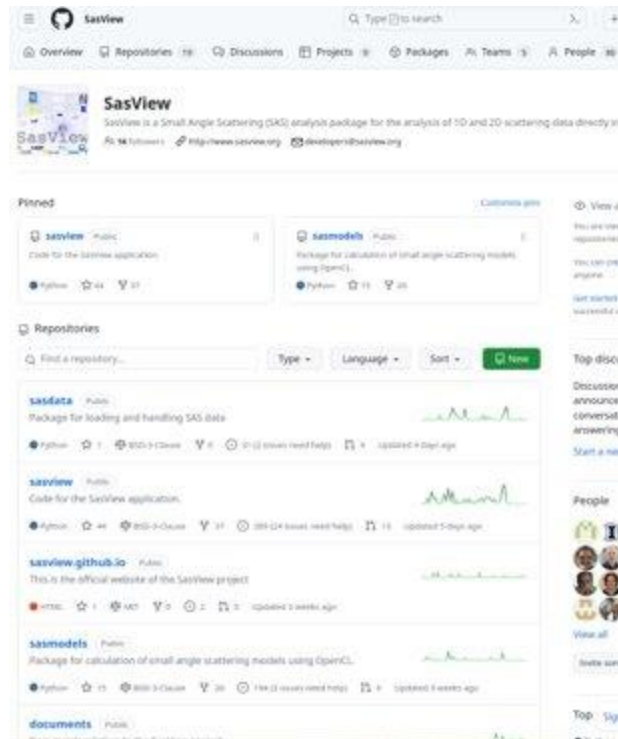
Code is **open source** and **publicly hosted** at Github

Released under **BSD 3-clause license**

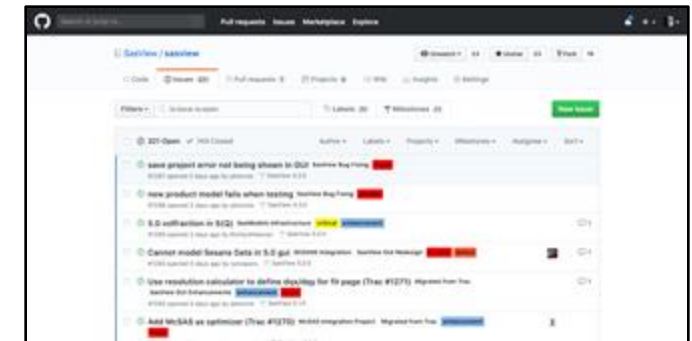
(Zenodo) DOI for each release

Rolling 5 Year Roadmap

Code Hosting, Issue
Tracking, Developer Wiki & CI on
Public Github repos



<https://www.sasview.org>



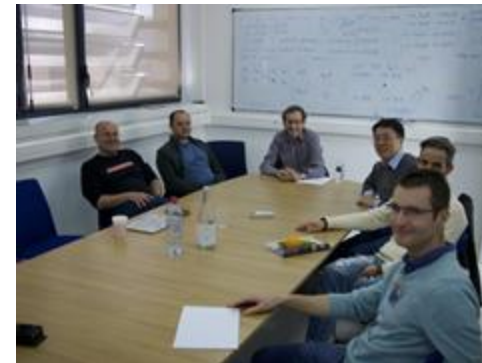
<https://github.com/SasView>



Open, Collaborative, Community Development



- Twice monthly zoom calls
- Regular 'camps' & 'hackathons'
- Developer's mailing list
- SasView slack
- Expertise sharing and helping
- Small leadership team to facilitate



**Collaboration also builds
Community**

<http://www.sasview.org>

<http://github.com/SasView>

Open, Collaborative, **Community Development**

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

- P. Butler, March 2019

No MOU ... all are invited and welcome

Two Basic "Rules"

He who pays the piper gets to choose the tune...

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

And ...

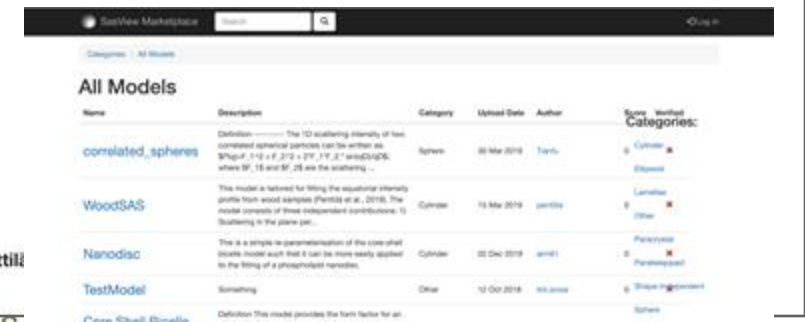
You cannot break existing experiences ...

- New dependencies vs long term maintenance (sustainability)
 - Code quality vs long term sustainability
- Changing/degrading the current user experience for the existing user base

<http://www.sasview.org>

<http://marketplace.sasview.org>

Model Marketplace for Users to share their models



Name	Description	Category	Upload Date	Author	Score	Method	Categories
correlated_spheres	Definition: The 1D scattering intensity of two correlated spherical particles can be written as $I(q) = 1 + 2V \cdot P(q) + V^2 \cdot S(q)$, where V is the volume of the particles, $P(q)$ is the form factor, and $S(q)$ is the structure factor.	Spheres	26 Mar 2019	Tarbi	0	Custom	Custom
WoodSAS	This model is optimized for fitting the equatorial intensity profiles from wood samples (Penttilä et al., 2016). The model consists of three independent contributions: 1) Scattering in the plane cell.	Cylinder	19 Mar 2019	penttila	0	Custom	Custom
Nanodisc	This is a simple re-parameterization of the core-shell sphere model such that it can be more easily applied to the fitting of a phospholipid membrane.	Cylinder	22 Dec 2018	ameli	0	Parameterized	Parameterized
TestModel	Something	Other	12 Oct 2018	Amelia	0	Shape Independent	Shape Independent
Core Shell Bicelle	Definition: This model provides the form factor for an ellipsoidal cylinder with a core-shell scattering length.	Other	12 Oct 2018	Amelia	0	Shape Independent	Shape Independent

Paavo A. Penttilä
@PaavoPenttila

The #WoodSAS model for analyzing small angle scattering data from wood is freely available at the @SasView Marketplace: marketplace.sasview.org/models/111/

Paavo A. Penttilä @PaavoPenttila

It's there, finally! The main outcome of my postdoc @ILLGrenoble: "Small-angle scattering model for efficient characterization of wood nanostructure and moisture behaviour" And it's all free! doi.org/10.1107/S16005...

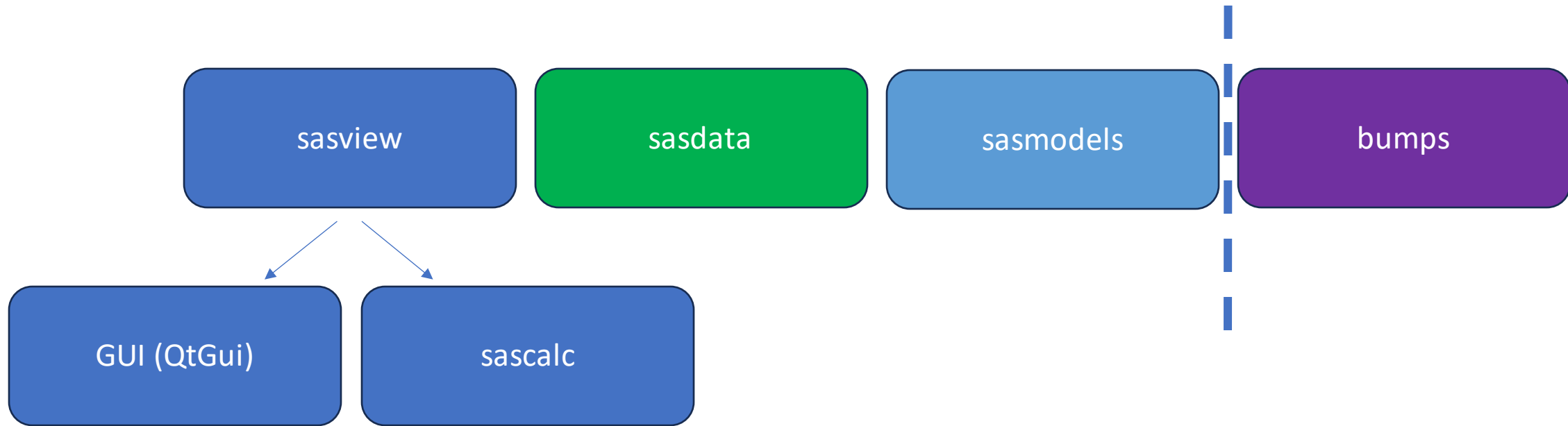
12:02 AM - 27 Mar 2019



SasView Today



Basic Structure of code base



- *Pip install sasmodels sasdata*
- Need to finish separating GUI from sascalc ... discussion on how
- Documenting entry points to GUI functionalities

DATA MANAGEMENT

Recently moved to sasdata

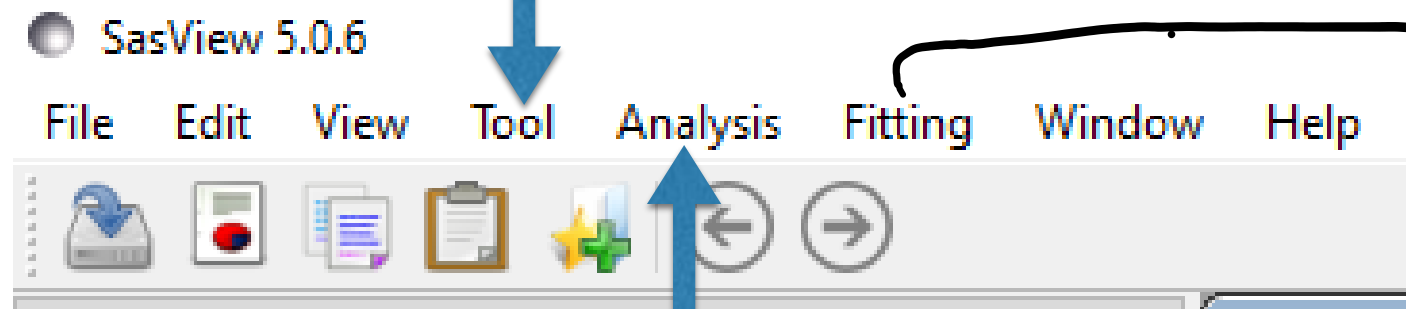
- Most all 1D data types supported on input
- Standard data supported (cansas1D and NXcanSAS)
- Some image and odd data supported through converters
- All data output in NXcanSAS format
- Calculated curves associated with data (PQ, SQ, desmeared model, etc)
- Project saving capabilities (currently a bit fragile)

SasView Toolkit for SAS data Analysis

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

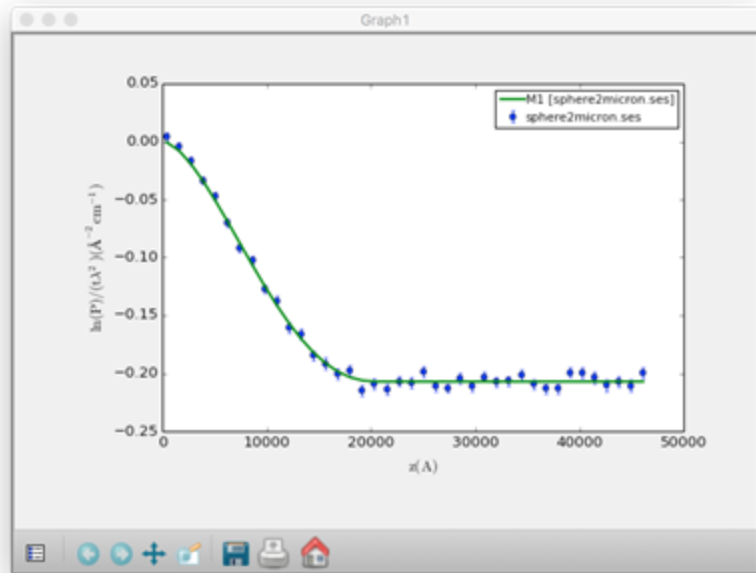
Perspectives on the data



Analysis

- Fitting
- Invariant
- Pr Inversion
- Correlation Function

Fitting in SasView: 1D



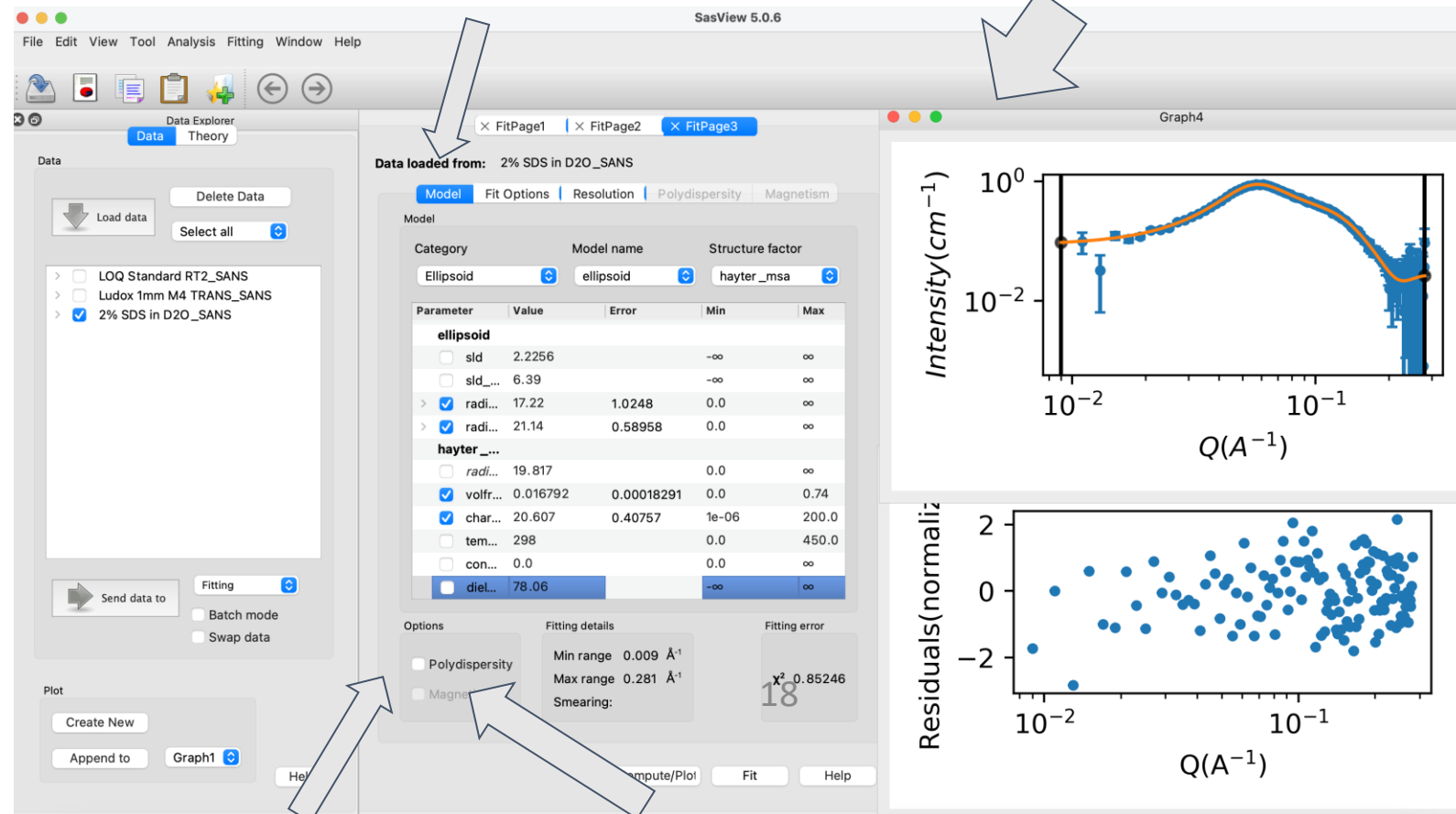
Also SESANS data

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

Analysis Tool Choice
&
Plotting

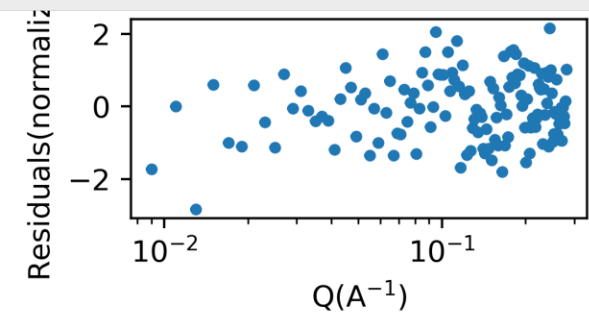
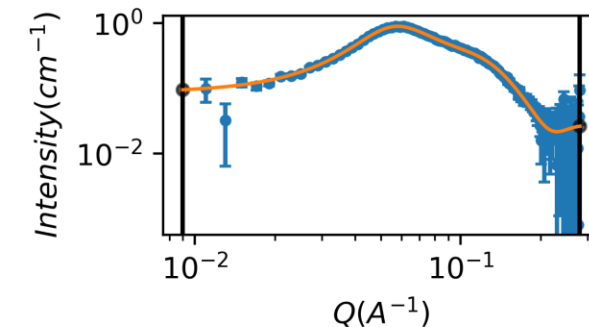
Wide choice of built-in models (> 70)
 $P(Q)$, $S(Q)$ & $P(Q)*S(Q)$

Single, batch and simultaneous
1D and 2D fitting

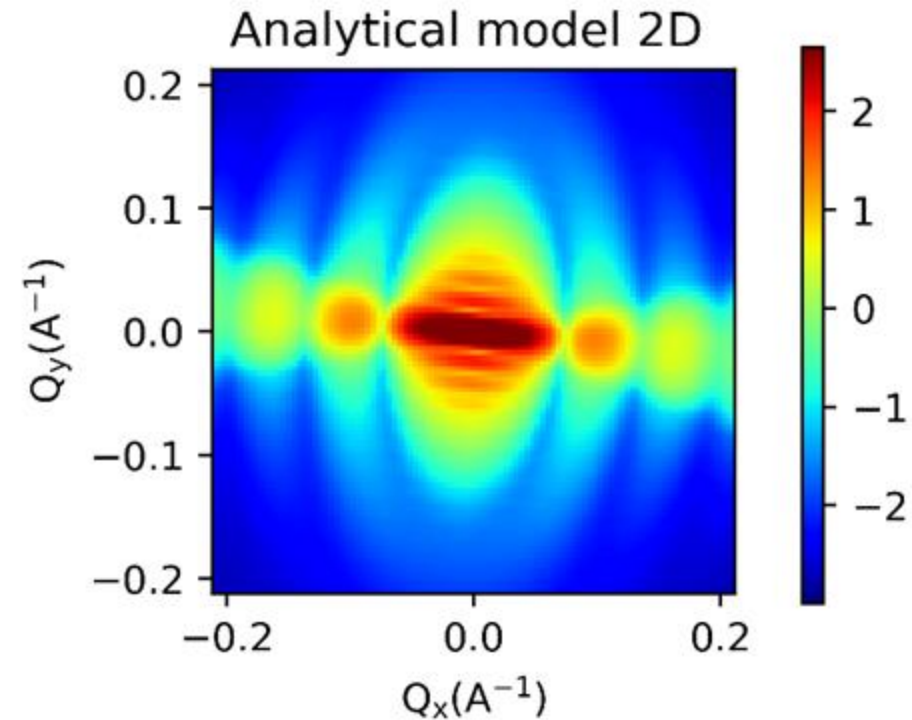
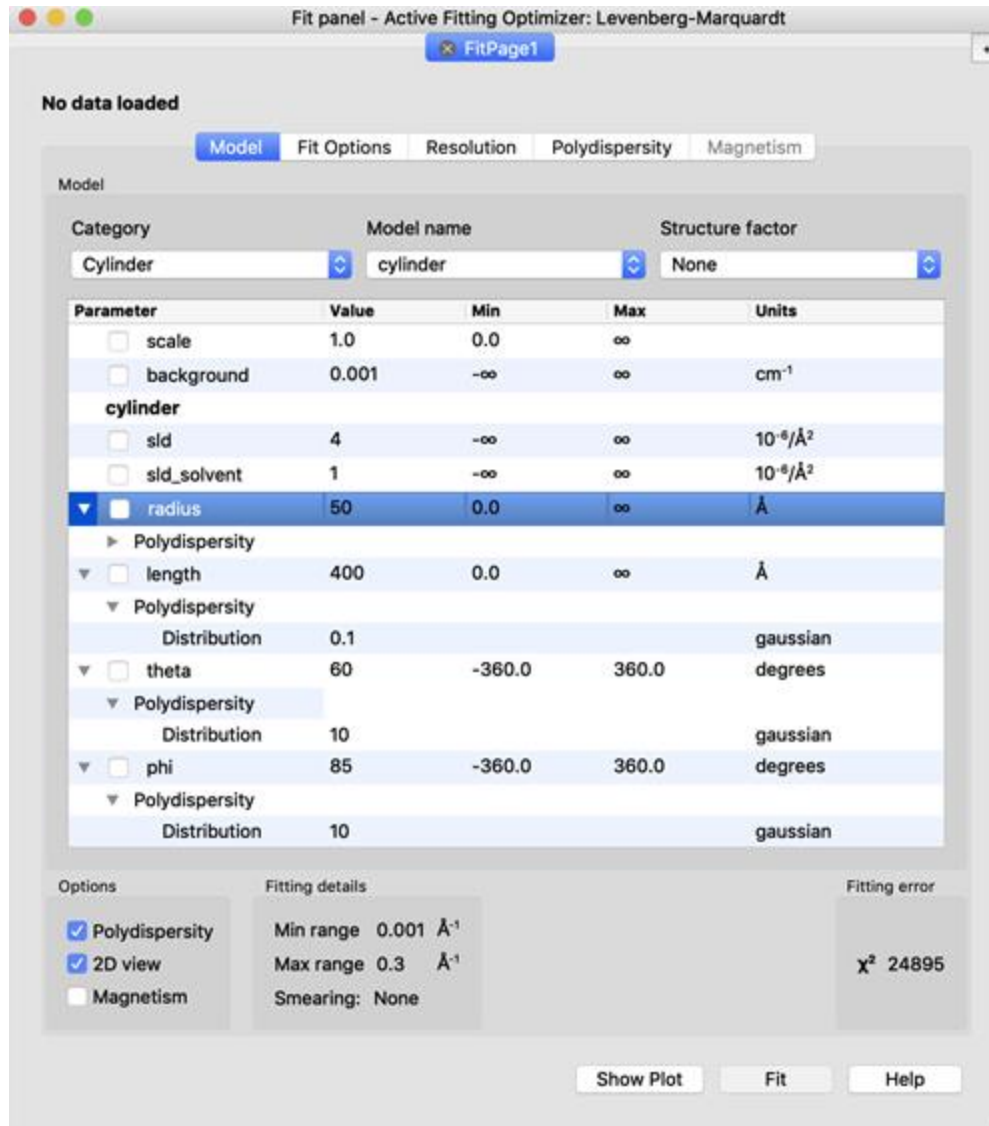


Polydispersity (choice of distribution and
distribution parameters)

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



Fitting in SasView: 2D

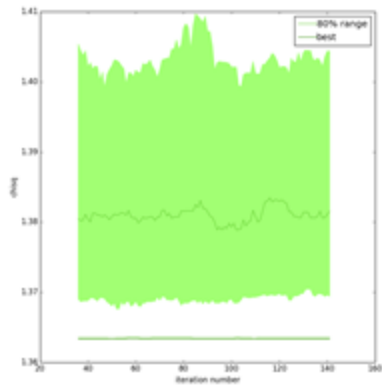
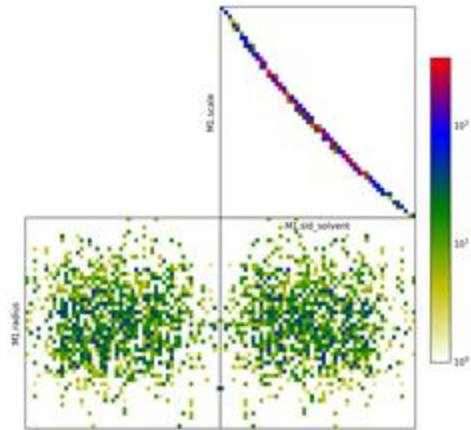


Orientational polydispersity = “jitter”

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

Choice of Fitting Algorithms



Fit Options

Fit Algorithms

- ☒ DREAM
- ☐ Levenberg-Marquardt
- ☐ Quasi-Newton BFGS
- ☐ Differential Evolution
- ☐ Nelder-Mead Simplex

DREAM Fitting Parameters

Samples: 10000

Burn-in Steps: 100

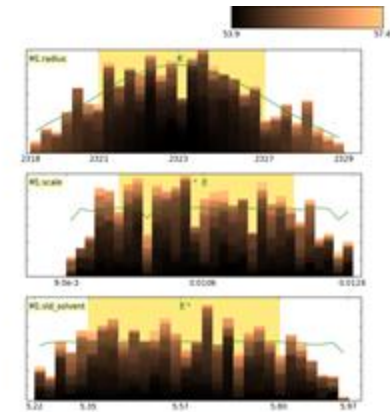
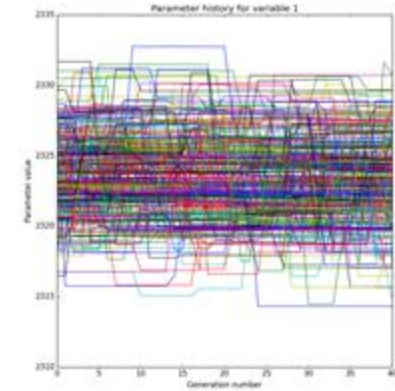
Population: 10

Initializer: eps

Thinning: 1

Steps: 0

OK Cancel ?



Uses bumps package from P. Kienzle
(also has DANSE origins)

Invariant Analysis

Invariant

I(q) Data Source

For more information, click on Details button.

Name: latex_smeared.xml [1]

Total Q Range (1/A): Min: 7.7457e-05 Max: 0.00554976

Outputs

Volume Fraction 2.26e-13 +/- 1.67e-18

Specific Surface +/- [1/A]

Invariant Total [Q*] 0.000445 +/- 7.4e-06 [1/(cm*A^3)]

Details? Compute HELP

Customized Inputs

Background: 0.0 [1/cm] Scale: 1.0

Contrast: 1.0 [1/ Porod Constant: [1/(cm*A^4)] (optional)

Extrapolation

Extrapolation Min: 1e-05 Max: 10

Maximum Q Range [1/A]:

Low Q

☒ Enable Extrapolate Low Q

Npts 10

☒ Guinier

☐ Power Law

☐ Fix ☐ Fit

Power 4.0

High Q

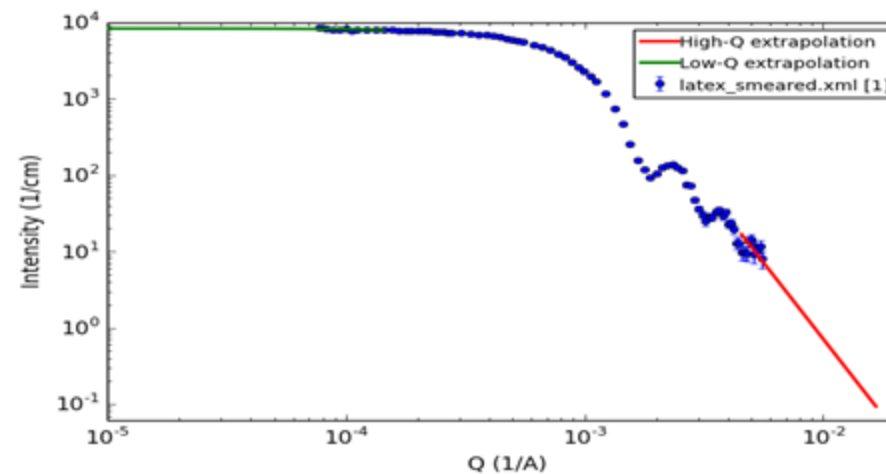
☒ Enable Extrapolate high-Q

Npts 10

Power Law

☒ Fix ☐ Fit

Power 4



P(r) Inversion

P(r) control panel

I(q) data source

Name:

☒ Estimate background level

Slit parameters

Height Width [\AA^{-1}]

Q range

Q min Q max [\AA^{-1}]

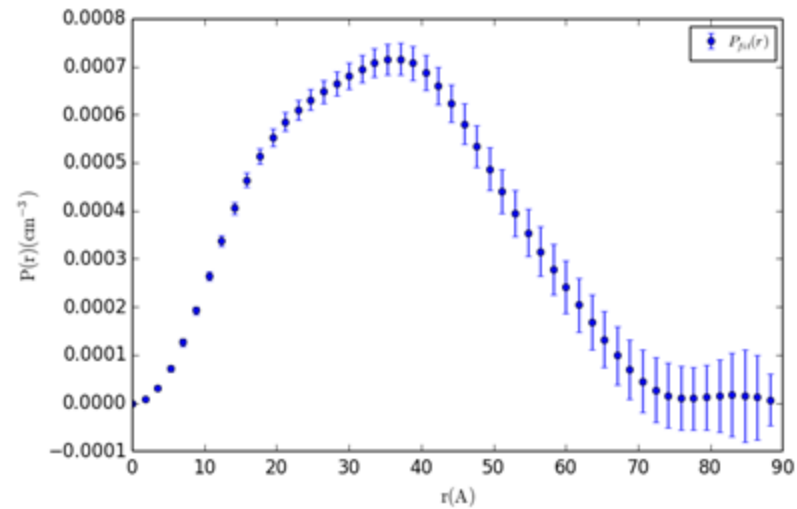
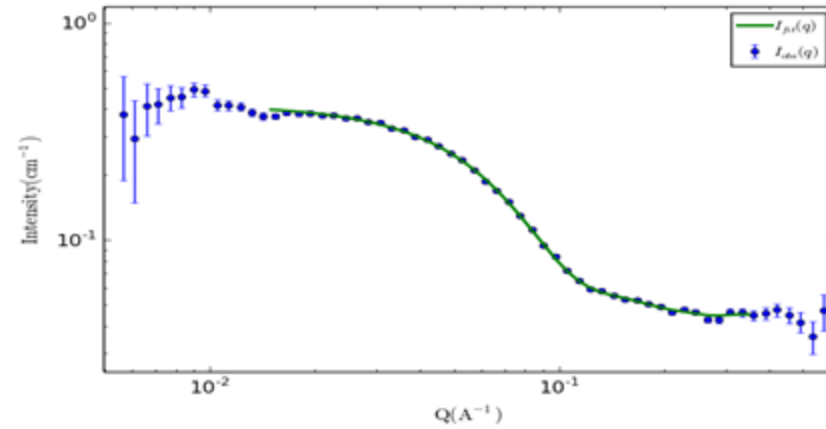
Parameters

P(r) is found by fitting a set of base functions to I(Q). The minimization involves a regularization term to ensure a smooth P(r). The regularization constant gives the size of that term. The suggested value is the value above which the output P(r) will have only one peak.

		Suggested value
Number of terms	<input type="text" value="15"/>	<input type="text" value="15"/>
Regularization constant	<input type="text" value="4e+09"/>	<input type="text" value="4e+09"/>
Max distance [Å]	<input type="text" value="90"/>	<input type="button" value="Explore"/>

Outputs

Rg	<input type="text" value="27"/>	[Å]
I(Q=0)	<input type="text" value="0.37"/>	[\AA^{-1}]
Background	<input type="text" value="0.046"/>	[\AA^{-1}]
Computation time	<input type="text" value="0.0011"/>	secs
Chi2/dof	<input type="text" value="46"/>	
Oscillations	<input type="text" value="1.3"/>	
Positive fraction	<input type="text" value="1"/>	
1-sigma positive fraction	<input type="text" value="0.99"/>	



Correlation Function Analysis

Recently refactored

CCP13 (Fiber Diffraction) legacy code (Fortran)
(ISIS summer student)

Correlation Function

I(Q) Data Source
Name: ISIS_98929.TXT

Input Parameters
Corfunc will use all values in the lower range for Guinier back extrapolation, and all values in the upper range for Porod forward extrapolation.

Q
Lower: 0.0 - 0.01425
Upper: 0.175 - 0.285

Extrapolation Parameters

Guinier:
A: 3.73237
B: -19249.4

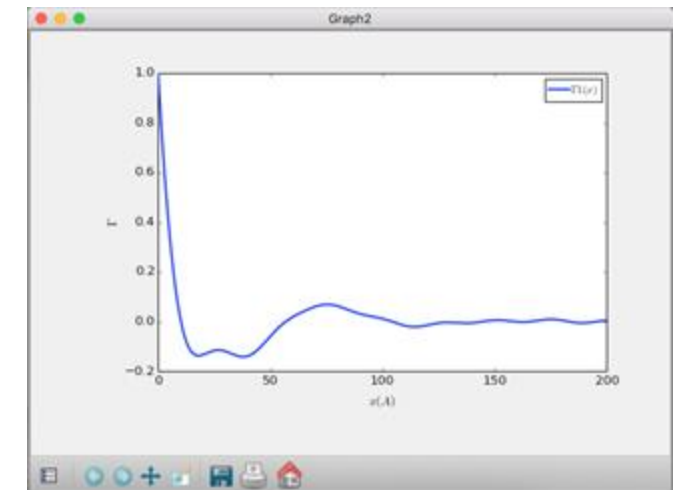
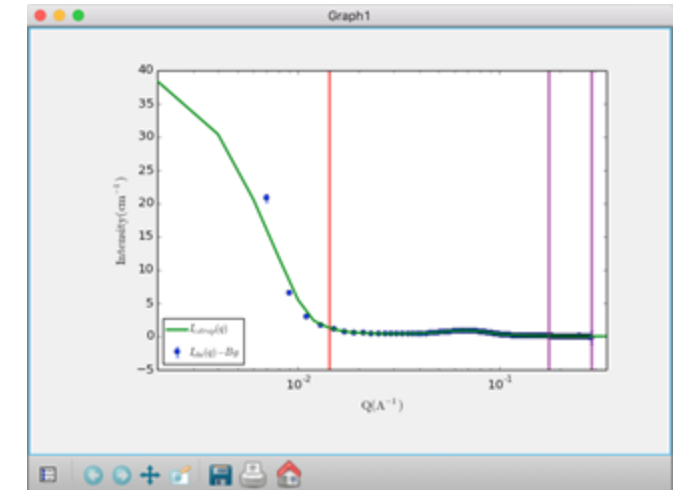
Porod:
K: 9.47650E-05
 σ : 9.76646E-13
Bg: 0.283226761551
Calculate Bg

Transform Type
☒ Fourier ☐ Hilbert

Output Parameters

Polydispersity:	0.182174
Average Hard Block Thickness (Å):	7.47393
Long Period (Å):	26.8946
Average Interface Thickness (Å):	0.0
Average Core Thickness:	1.65335
Local Crystallinity:	0.277897

Controls
Extrapolate
Transform
Compute Parameters
HELP



Plugging 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/>

SasView Marketplace	Search	Q	Log In				
OrientedMagneticChains	<p>with the option of adding a magnetic SLD to each layer. The chain scattering is the incoherent sum of a user-defined combination of sin...</p>	Sphere	21 May 2021	krycat	0	✖	Ellipsoid Lamellae Other Paracrystal Parallelepiped Shape-Independent Sphere Structure Factor All Models
Magnetic vortex in a disc	<p>This model describes the approximated scattering of a magnetic vortex in a flat ferromagnetic cylinders made of isotropic material [Metlov2016]. The circular cylinder with radius \$R\$ and length \$L\$...</p>	Cylinder	10 Mar 2021	dehoni	0	✖	
Field-dependent magnetic SANS of misaligned magnetic moments in bulk ferromagnets	<p>For bulk ferromagnets, this model allows to analyze the field-dependent purely magnetic SANS. The misalignment scattering is obtained by subtracting the reference scattering at a high (saturating) ...</p>	Sphere	17 Feb 2021	dehoni	0	✖	
SANS of bulk ferromagnets	<p>This model is a micromagnetic approach to analyse the SANS that arises from nanoscale variations in the magnitude and orientation of the magnetization in bulk ferromagnets in the approach to magnet...</p>	Sphere	17 Feb 2021	dehoni	0	✖	
core_shell_ellipsoid_tied and core_shell_ellipsoid_repar	<p>Two methods, both requiring sasview v5, to produce a core_shell_ellipsoid with solvent in the shell. Parameters include the dry_shell / core volume ratio, the local fraction of solvent in the she...</p>	Ellipsoid	16 Feb 2021	richardh	0	✖	

Model Editor - my_broad_peak*

Plugin Definition Model editor

Plugin name

my_broad_peak ☐ Overwrite existing plugin model of this name

Description

My special broad peak model

Fit parameters

Non-polydisperse

	Parameters	Initial value
	porod_scale	1.0e-05
2	porod_exp	3
3	lorentz_length	50
4	lorentz_scale	10
5	lorentz_exp	2.0

Polydisperse

Parameters	Initial value

Function(x)

```

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

```

Help Apply Cancel

The screenshot shows a software interface with a 'Plugin Definition' window. The window has two tabs: 'Plugin Definition' (active) and 'Model editor'. The 'Plugin Definition' tab contains the following content:

Model

Definition

~~~~~

Calculates my\_broad\_peak.

My special broad peak model

**References**

~~~~~

Authorship and Verification

~~~~~

```

* **Author:** --- **Date:** 2018YYY-09m-20d
* **Last Modified by:** --- **Date:** 2018YYY-09m-20d
* **Last Reviewed by:** --- **Date:** 2018YYY-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", default, [lower, upper], "type", "description"],
[porod_scale, "", 1e-05, [-inf, inf], "", ],
[porod_exp, "", 3.0, [-inf, inf], "", ],
[lorentz_length, "", 50.0, [-inf, inf], "", ],
[lorentz_scale, "", 10.0, [-inf, inf], "", ],
[lorentz_exp, "", 2.0, [-inf, inf], "", ],
[peak_pos, "", 0.1, [-inf, inf], "", ],
[q, "", 0.01, [-inf, inf], "", ],
]

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

```

At the bottom of the window, there are three buttons: 'Help', 'Apply', and 'Cancel'.

# Beyond the GUI

## *Running SasView from scripts*

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

```
import pylab
from bumps.names import *
from sasmodels.core import load_model
from sasmodels.bumps_model import Model, Experiment
from sasmodels.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)

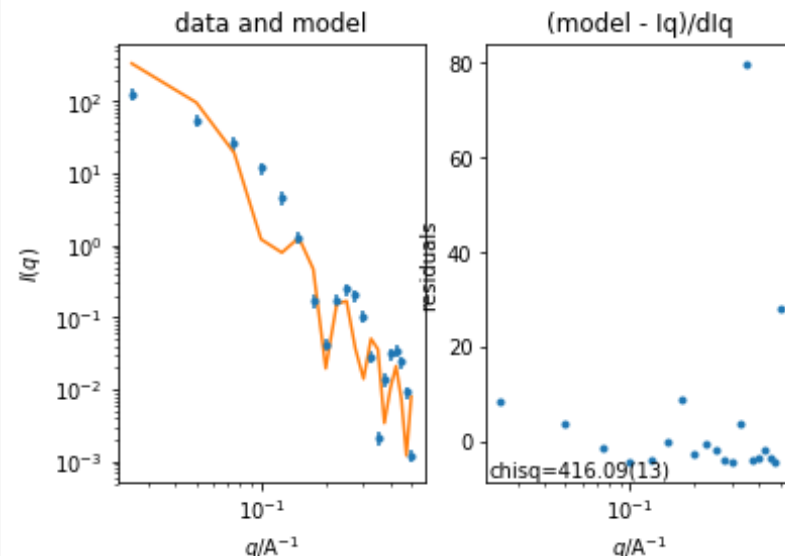
# SET THE FITTING PARAMETERS
model.radius.range(1, 50)
model.length.range(1, 500)

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()
```

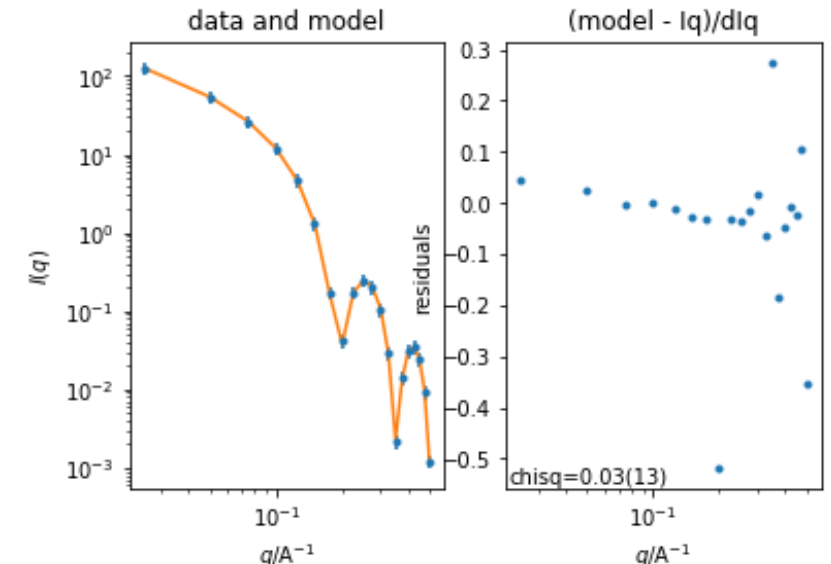
Before fit

Initial chisq 416.09(13)



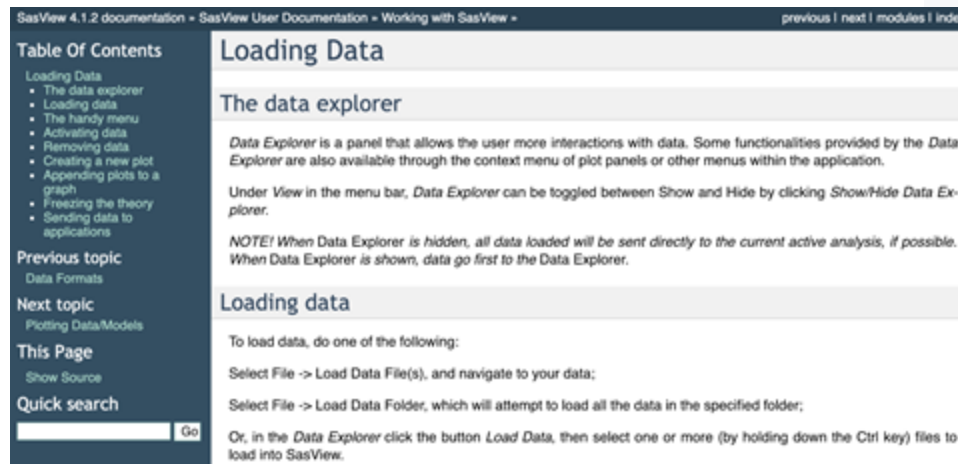
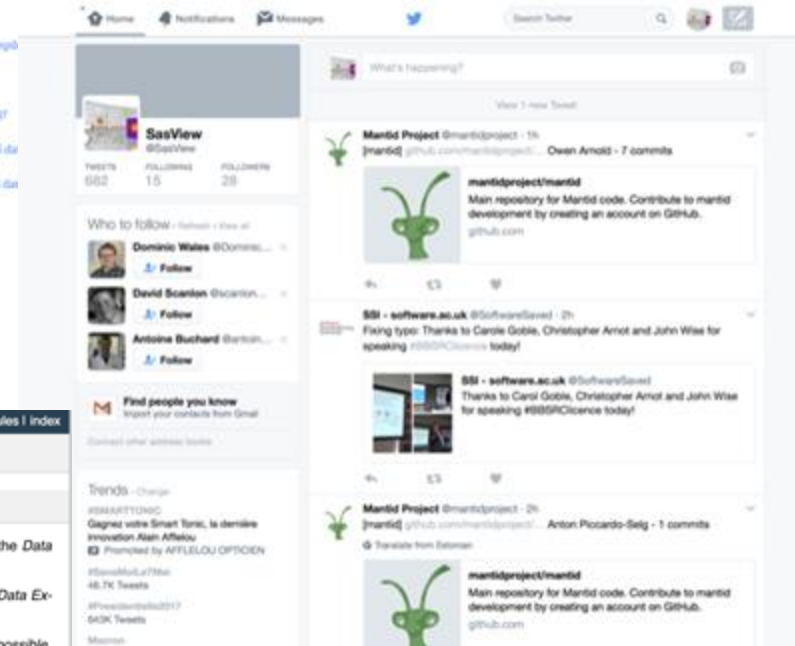
After fit

Final chisq 0.03(13)  
length : 464.9(55)  
radius : 19.977(64)



# Resources, Education & Outreach

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



# Resources, Education & Outreach


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

**SasView** ABOUT LINKS

**Tutorials:**

- [Old SasView tutorial \(PDF\)](#) - still useful
- [Getting started with SasView \(PDF\)](#)
- [Basic 1D Fitting in SasView \(PDF\)](#) - for versions 3.x/4.x
- [Simultaneous 1D Fitting in SasView \(PDF\)](#) - for versions 3.x/4.x
- [Correlation Function Analysis in SasView \(PDF\)](#) - for version 4.x

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 | slit core                               | slit shell                              | slit solvent                            |
|----------------|-----------------------------------------|-----------------------------------------|-----------------------------------------|
| 'core'         | $+6.39 \times 10^{-4} \text{ \AA}^{-2}$ | $+6.62 \times 10^{-4} \text{ \AA}^{-2}$ | $-0.29 \times 10^{-4} \text{ \AA}^{-2}$ |
| 'shell'        | $+6.39 \times 10^{-4} \text{ \AA}^{-2}$ | $+6.62 \times 10^{-4} \text{ \AA}^{-2}$ | $+6.66 \times 10^{-4} \text{ \AA}^{-2}$ |
| 'slit'         | $-0.55 \times 10^{-4} \text{ \AA}^{-2}$ | $+6.62 \times 10^{-4} \text{ \AA}^{-2}$ | $+6.66 \times 10^{-4} \text{ \AA}^{-2}$ |

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 possible to find a common scale parameter volume fraction of droplets in each sample.

We now need to tell SasView that the radius for the 'shell' contrast data is contrast dataset, and so on. This is constrain parameters that are free to thickness parameters. However, we parameters so check those too.

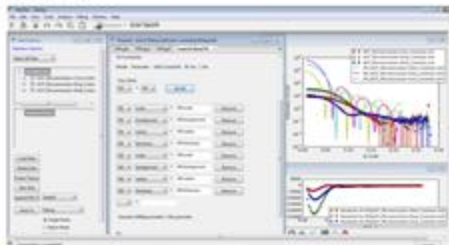
**Tip:** Always check the parameters in before opening the Contrasted or Set

Now go to the Menu Bar and click **Fit**

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 **Y00** 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 **Select**. Then set **M3=M3** and click **Select**.

**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 \* M0 parameter name**



**SasView** 5 subscribers

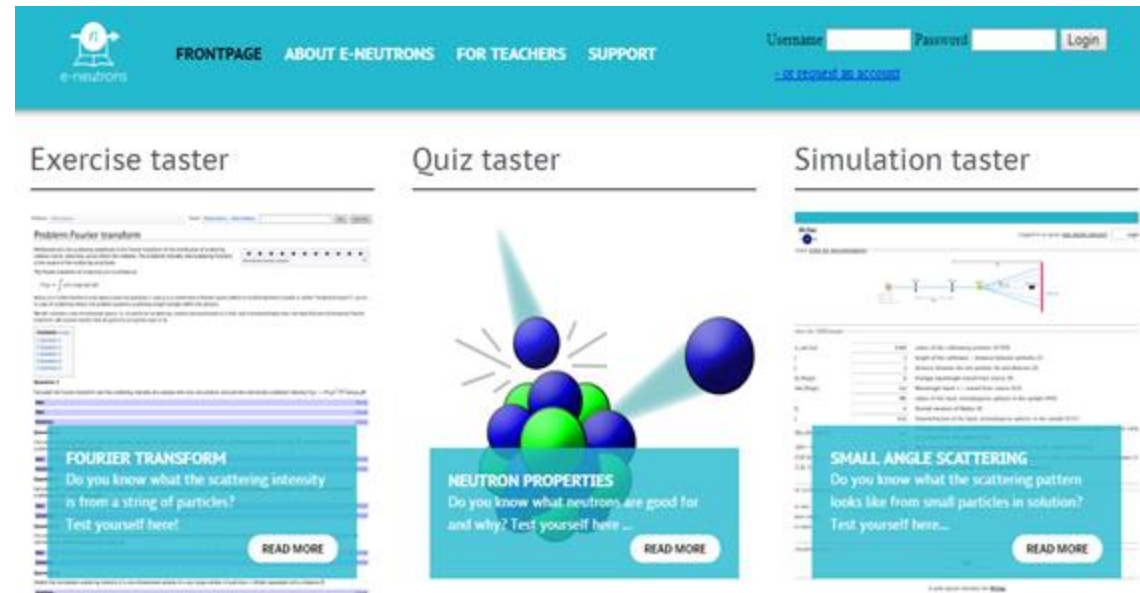
HOME VIDEOS PLAYLISTS CHANNELS DISCUSSION ABOUT

Uploads ▶ PLAY ALL

- Using the P(r) calculator in SasView 65 views • 5 months ago
- 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

# Resources, Education & Outreach

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



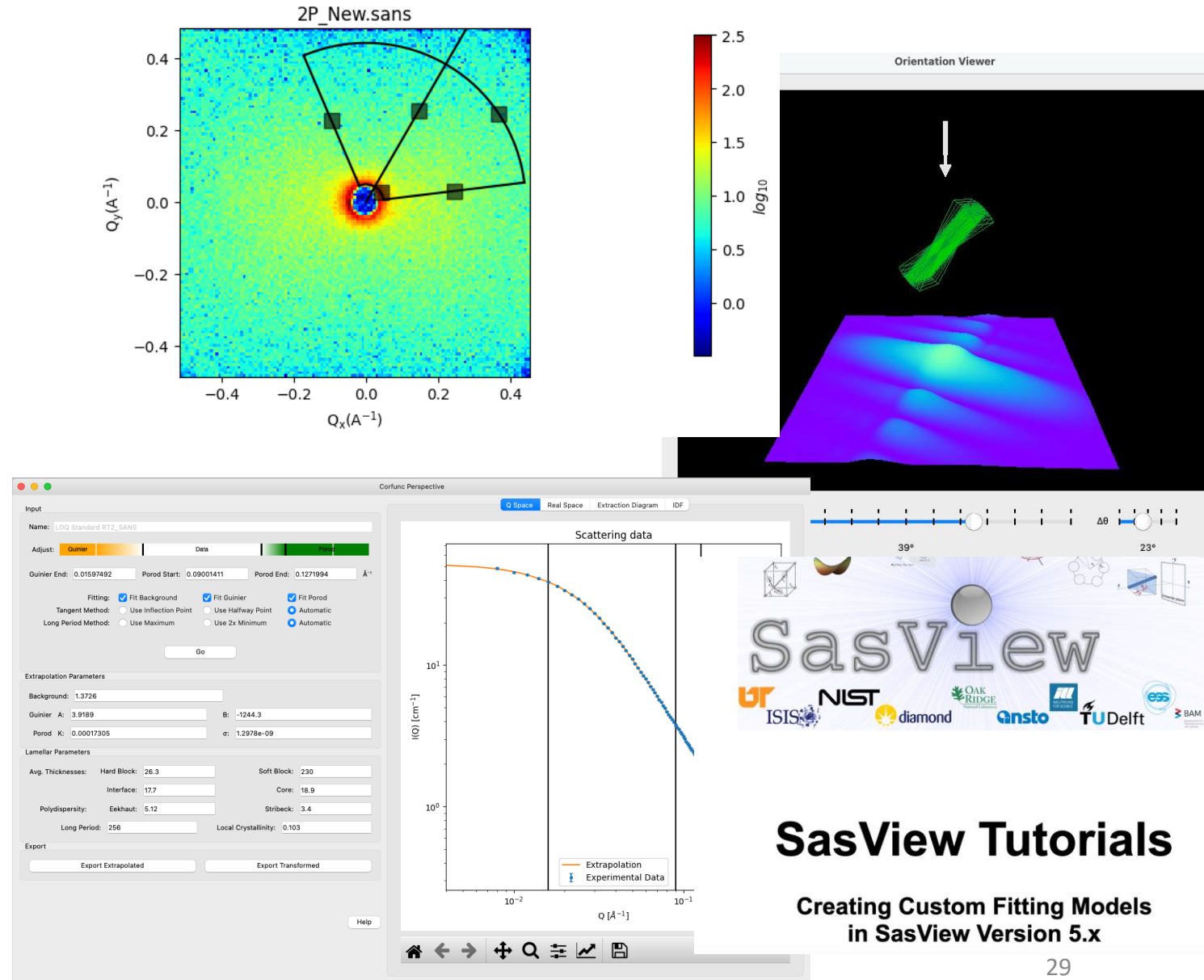
*All the work of ISIS Sandwich Student Michael Oakley*



# A taste of things to come – SasView 6.0.0-alpha

*Use at your own risk...*

- 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



## SasView Tutorials

Creating Custom Fitting Models  
in SasView Version 5.x



# Towards Generic Resolution Functions

```
<ldata>  
<Q unit="1/A">0.00714</Q>  
<l unit="1/cm">226.539</l>  
<ldev unit="1/cm">1.93973</ldev>  
<Qdev unit="1/A">0.00144239</Qdev>  
</ldata>
```

Data reduction  
(resolution  
values)



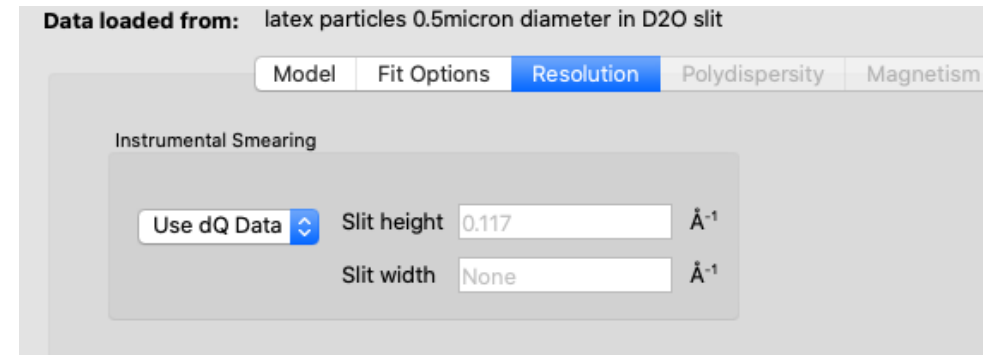
File with  
 $q$ ,  $l(q)$ ,  $dl$ ,  $dQ$



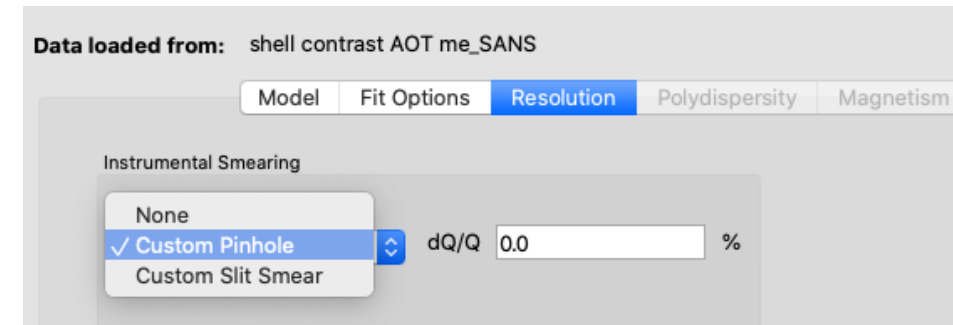
Data Analysis  
(reads in  $dQ$   
from file)

In SasView:

File with  $dQ$



File without  $dQ$



# Towards Generic Resolution Functions

*Working with canSAS to extend NXCanSAS format with defined labels*

|                            | <b>ResName</b> | <b>ResData</b>                                     |
|----------------------------|----------------|----------------------------------------------------|
| Analytic Gaussian          | GaussFn        | Qdev                                               |
| General, equally spaced    | CurveEq        | Ndata, V0, V1,V2, ..., Vmid, ... Vend              |
| General, arbitrary spacing | CurveGen       | X0, V0, X1, V1, X2, ..., Xj, Vj, ....., Xend, Vend |
| User supplied              | UserRes        | FnName, FnData                                     |

NXCanSAS placeholders:

/entry/data/q/resolutions

Dataset name corresponding to Q resolution data

/entry/data/q/resolutions\_description

Can be a simple description (i.e. Gaussian) or metadata on Q resolution data

# SasView Model: Some Current Challenges

- The tragedy of the commons: somebody else will take care of the problem – New features are more interesting than fixing fragility issues
- Hard to get non-coders to believe they can contribute equally – it is unfortunately in the name “code camp” ... words matter → “Contributor camp”
- Building community, especially during a pandemic is HARD WORK
- Lack of funds for small things ... but beware too much money?
- Cost of writing sustainable/maintainable code is surprisingly high and can be a deterrent to new contributors
- Hard to keep up with increasing security issues with the resources so far.
- Heavily geared towards colloid science
  - NEW: Magnetic SANS effort

## The price of success:

- Many people view the project as a well-funded group of professionals → barrier to contribution
- Large project with many moving pieces is a barrier to new volunteer coders

# Contributing to SasView

- *You don't have to be coding ninja to help out!*
- Respond to queries posted to [help@sasview.org](mailto: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 PySide2/Matplotlib
- Server and Github admin task
- Develop/improve databases (e.g. marketplace) etc.




# Contributing Student Projects – Summer 2022 and 2023

- Simultaneous SAXS and SANS fitting- **Iestyn Cadwallader-Jones**, supervisor: Miguel Gonzales (ILL)
- Plotting improvements, benchmarking for hypergeometric functions- **Nouhalia Agouzal** and **Dorian Lozano**, supervisor: Miguel Gonzales (ILL)
- Wedge slicer and improving slicer code - **Ellis Hewis**, supervisor Dirk Honecker (ISIS)
- Extending scattering calculator for magnetic systems – **Ruben Lopes**, supervisor Dirk Honecker (ISIS)
- Extending scattering calculator for Interacting bio-molecules – **Alex Zheng**, supervisor Yun Liu (NIST)
- Local documentation generator and editor (plugin models) – **Brayden Miller**, supervisor Paul Butler (NIST)
- Web-based API, **Xael Shan**, supervisor Jeff Kryzwon (NIST)

# Resources for Contributors


- Reducing entry barrier and improving release cycles frequency
  - Faster release cycles
  - E-learning course
  - Contributor Camp

**SasView for new contributors**



Course Settings Participants Grades Reports More ▾

▼ **General** Collapse all

 **FORUM**  
Announcements


▼ **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!)

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 · d4e3a88

**New features**

- Orientation viewer
- Corfunc refactored

## CodeCampXII

Paul Butler edited this page on Sep 13 · 14 revisions

## Contributor Camp XII Planning [↗](#)

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 [↗](#)

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 2024  
University of Delaware, US

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

# What will SasView do in the future?



Whatever the community  
contributes.



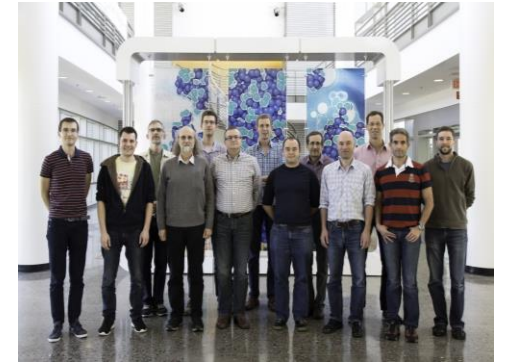
# The resource Problem – The SasView Solution

## vertical and horizontal **COLLABORATIONS**

- Facilities provide foundational support through participation of data and instrument scientists
- Grants and other projects provide “bold new functionality”
- Community, writ large, helps provide support and functionality
- New ideas tested and developed as before by individuals or larger groups (the community)
- Once validated and deemed ready for the larger community these groups provide resources (their labor) to integrate while active developer community helps with training on where things go and on parts of interest to them (collaborative)

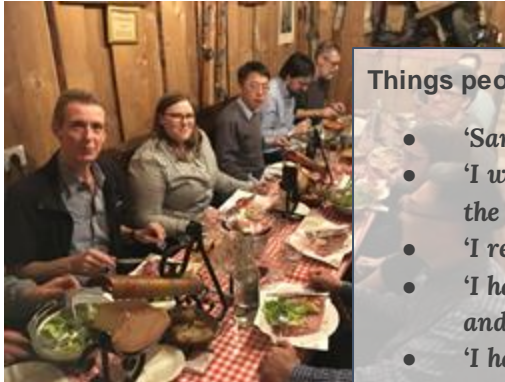
# Current Status of SasView

- 2006; originates in NSF *DANSE* project
  - 2013; transitions into a community project
  - 2016; Sine2020 project funded
  - 2023; Essentially a “volunteer army”
- 
- ~40 contributors from 9 organizations so far (~10-15 active at any one time)
  - 1 to 2 releases/year (5.0.5 JUST RELEASED)
  - Documentation/tutorial projects ongoing
  - Usage? Seems to be "everywhere?"
  - Lots of "complements" (a bit scary)
  - Publications? > 100/year



# Come and Join the Fun!

## *Contributor Camp XII Jan 16-22 University of Delaware*



### Things people are saying about SansView/SasView

- 'SansView is a very helpful tool, very complete and easy to use' - Niki
- 'I want to thank you for this amazing software. It's UI and options make the interpretation of spectra easier and faster' - Philippe
- 'I really like the SasView software' - Martin
- 'I have been using SasView as my software of choice for fitting SANS data, and I have been very happy with the software' - Greg
- 'I have found SasView very easy to use and the batch fit function is a wonderful time saving tool. I can finally stop making painful excel macros!' - Andrew
- 'I am a new user of SasView and I think it is a very useful and practical tool' - Arnaud
- 'Within 30 seconds...I am completely converted to SasView!' - Mike
- 'Thank you for creating and maintaining SasView. It is an incredibly helpful tool, and I use it regularly' - Pasha
- 'All the best and thank you again to carry on such a good job on SasView' - Niki
- 'Ooooh NICE PROGRAMME!! Hours of fun!' - Stuart
- 'I love such amazing software so much. It help our researches a lot.' - Po-Wei

