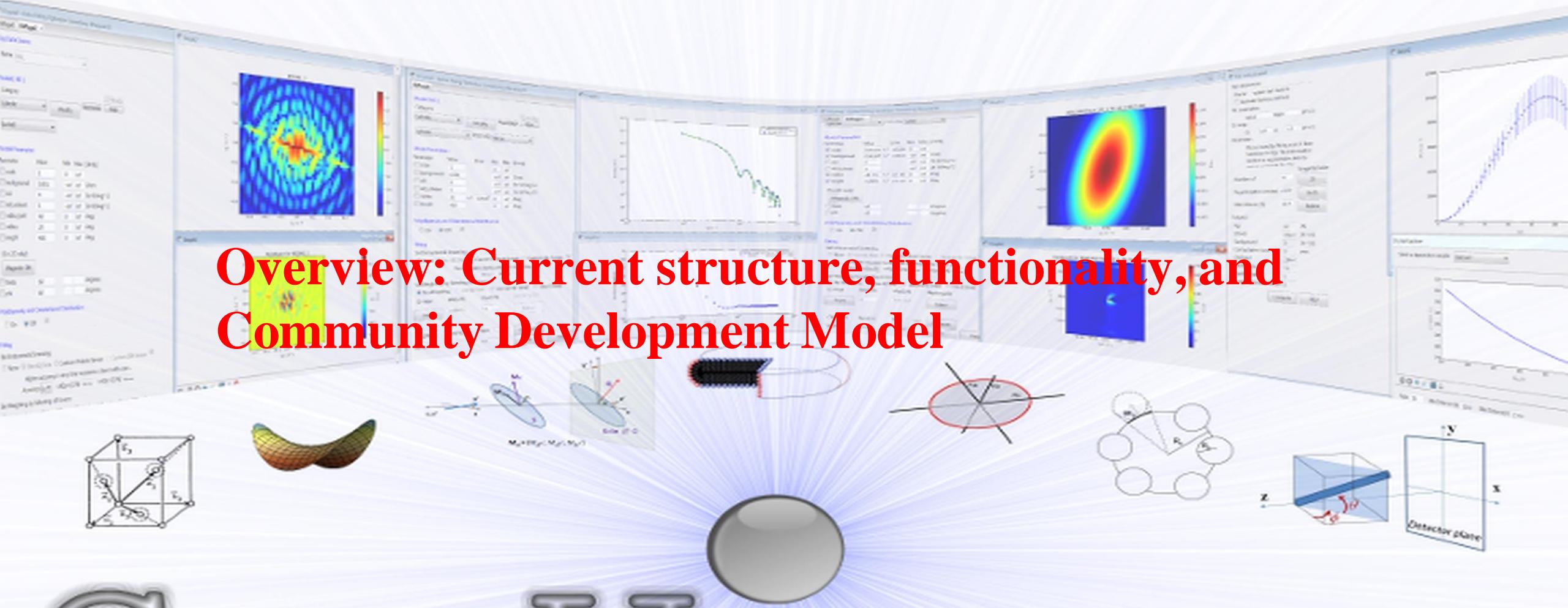


Overview: Current structure, functionality, and Community Development Model



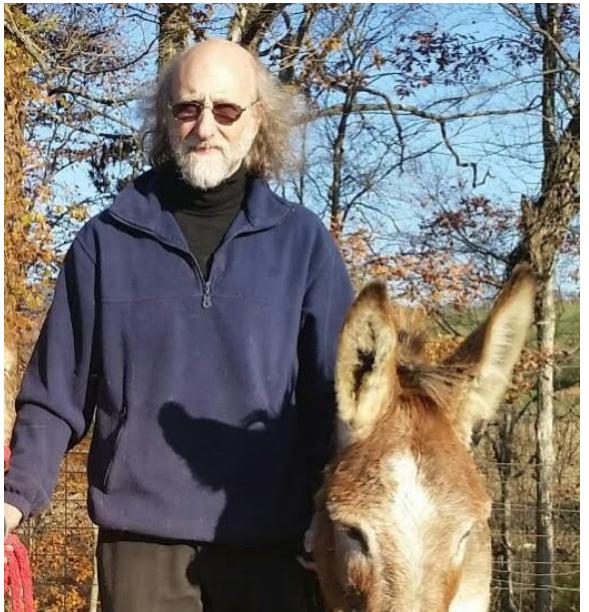
SasView



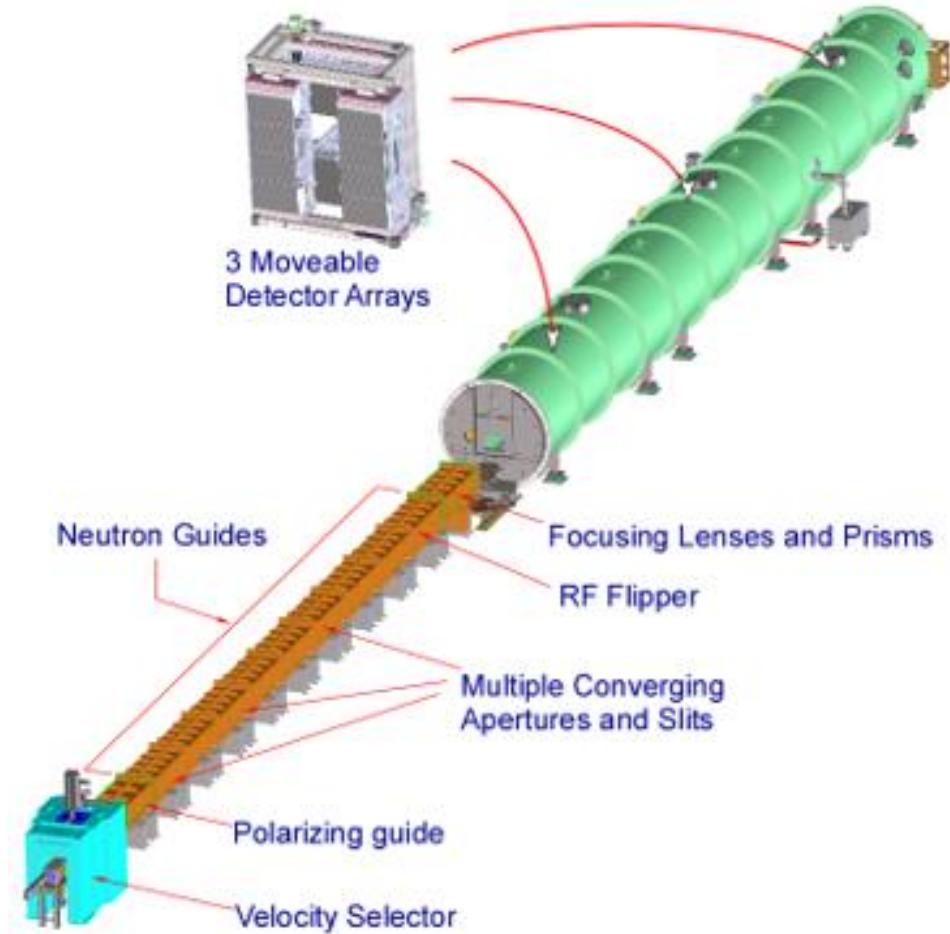
ansto



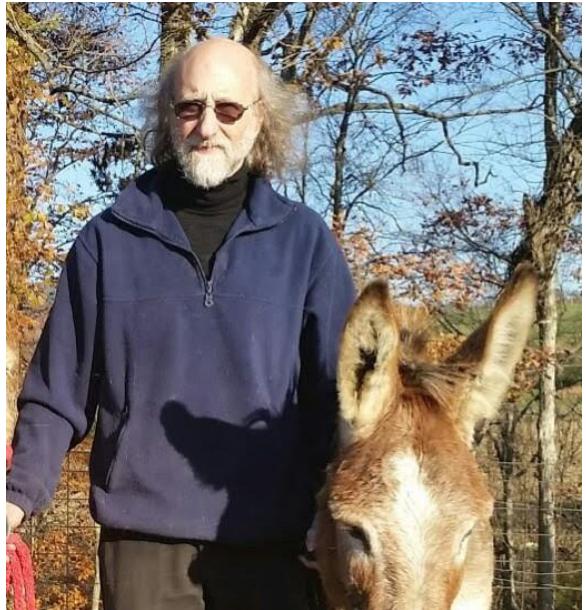
Notes from the Speaker (my 5 min)



- Scattering since last millennium
- Leader of the NIST small angle group since 2004

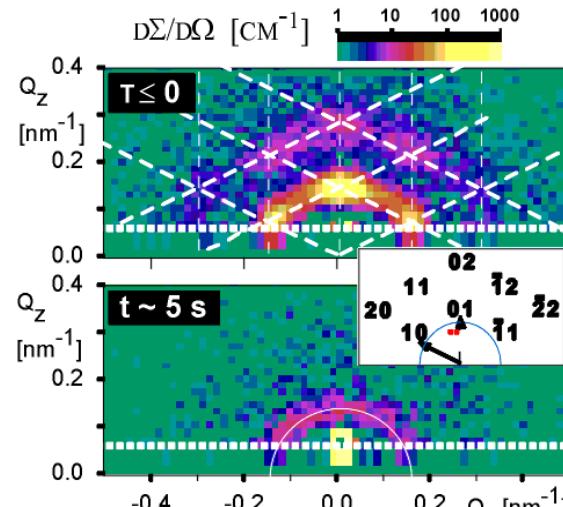


Notes from the Speaker (my 5 min)



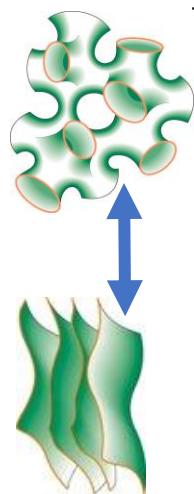
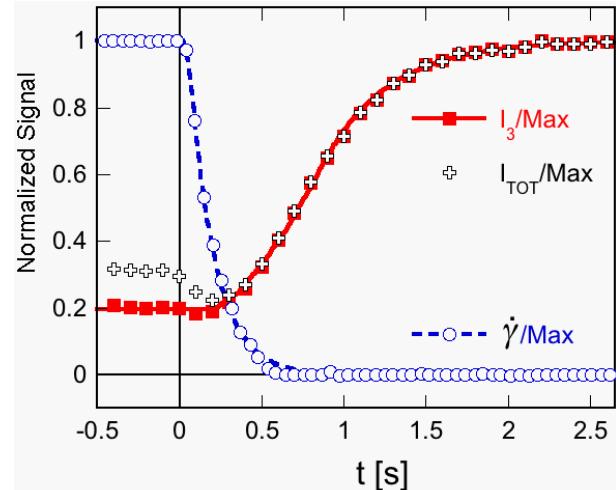
- Colloid/self-assembly scientist

Ordering of wormlike
micelles near a surface
and under flow

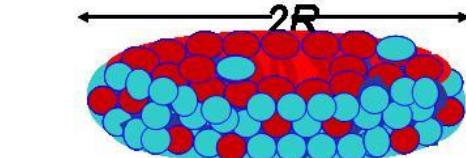
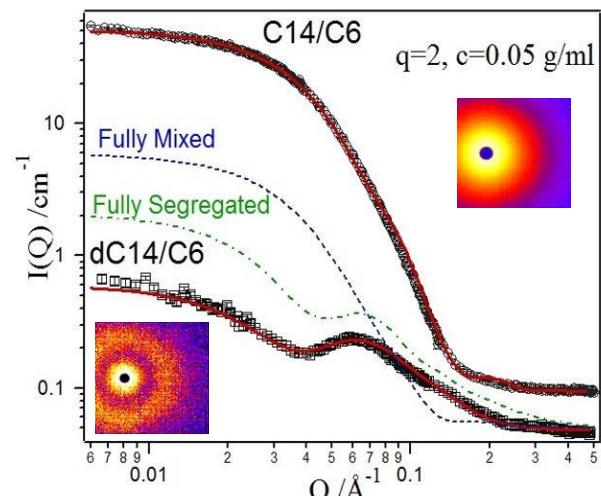
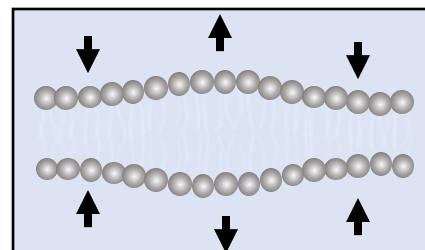
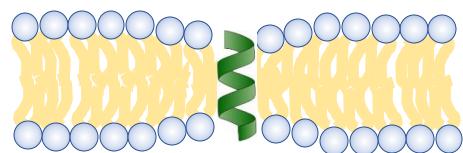


Flow induced sponge
to lamellar transition
and relaxation

Energetics of fusion



Lipid membrane
structure and dynamics



Segregation of lipids
in mixed lipid system

Notes from the Speaker (my 5 min)



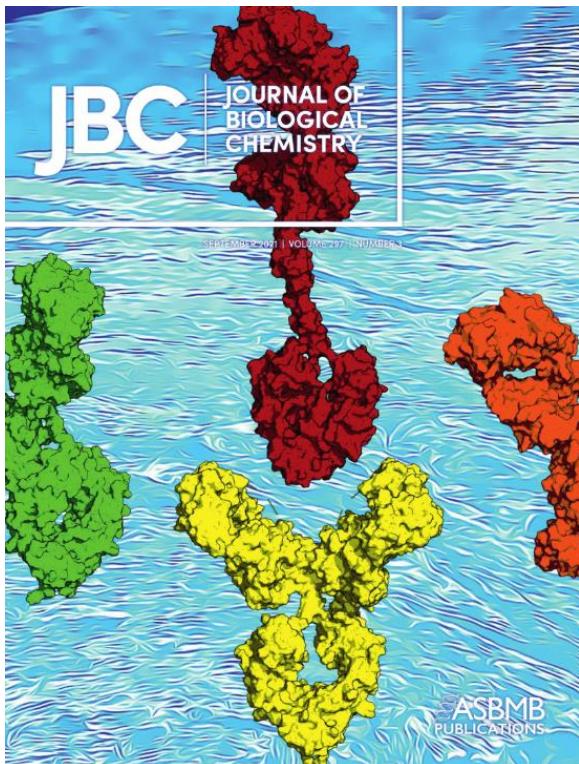
- Long standing interest in building SAS and scattering community and collaboration
 - SasView
 - canSAS
 - CCP-SAS



canSAS V

Workshop Report

The Fifth canSAS meeting was held on October 29th to 31st on the NIST campus in Gaithersburg, Maryland, hosted by the NCNR SANS group. The aim of the international organizing committee was to keep the group to a reasonable size of motivated people to have frank and open discussion of issues and work on solutions that would be outcomes of the meeting. The format included oral presentations, posters, and software demos as well as plenty of discussion time from moderated discussions, to open discussions, to various informal discussion opportunities. The workshop was successful at stimulating significant levels of interactions and lively discussions.



[SAS 2022 Campinas, Brazil: September 11-16, 2022](#)

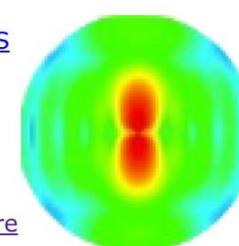
Mark your calendars

[Learn About SAS](#)

[News & Events](#)

[Resources](#)

[Software](#)



[SAS Calendar](#)

[Instruments & Facilities](#)

[Organisations](#)

[Forums & Listservers](#)

[About the Portal](#)

canSAS V (2007)

- Reduced formats ([NXcanSAS](#))
- Inter-facility standards
- SAS portal ([smallangles.org](#))
- IGOR collaborations
- Facility rep quarterly meetings

SasView: A SAS data analysis platform

Data Analysis eh?

.... So what exactly does that mean?

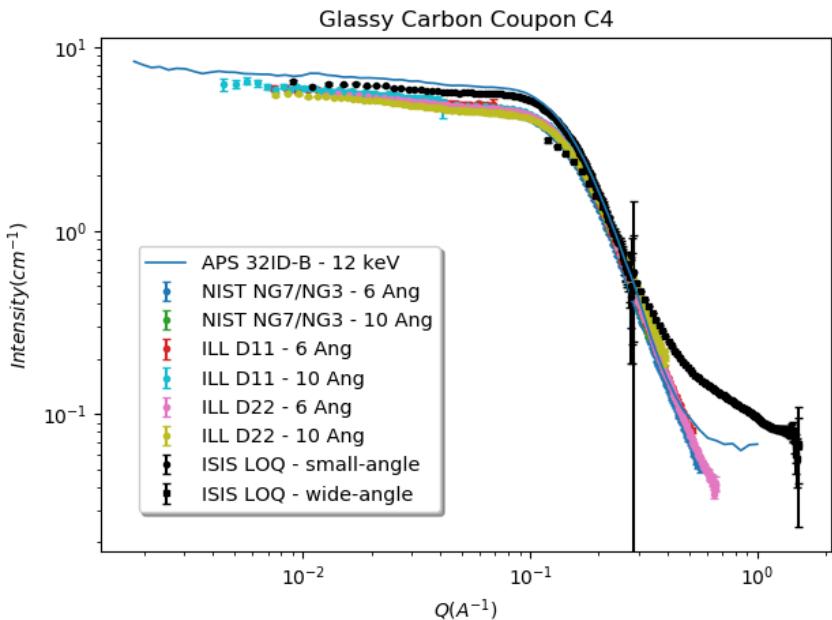
Only works on Reduced data

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

....
Sorta

Experimental Data Flow

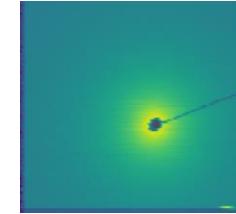
Reduced data should be reproducible anywhere!



<http://www.cansas.org>

Instrument-
dependent
data

Instrument-
independent
data

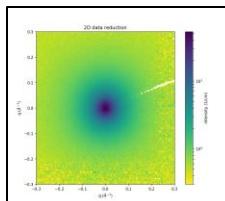


Data Acquisition System

'raw' data file
(NXS)

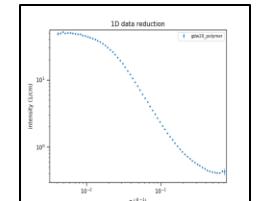
Data Reduction Programs
(e.g. *Mantid*)

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



Data Analysis Programs

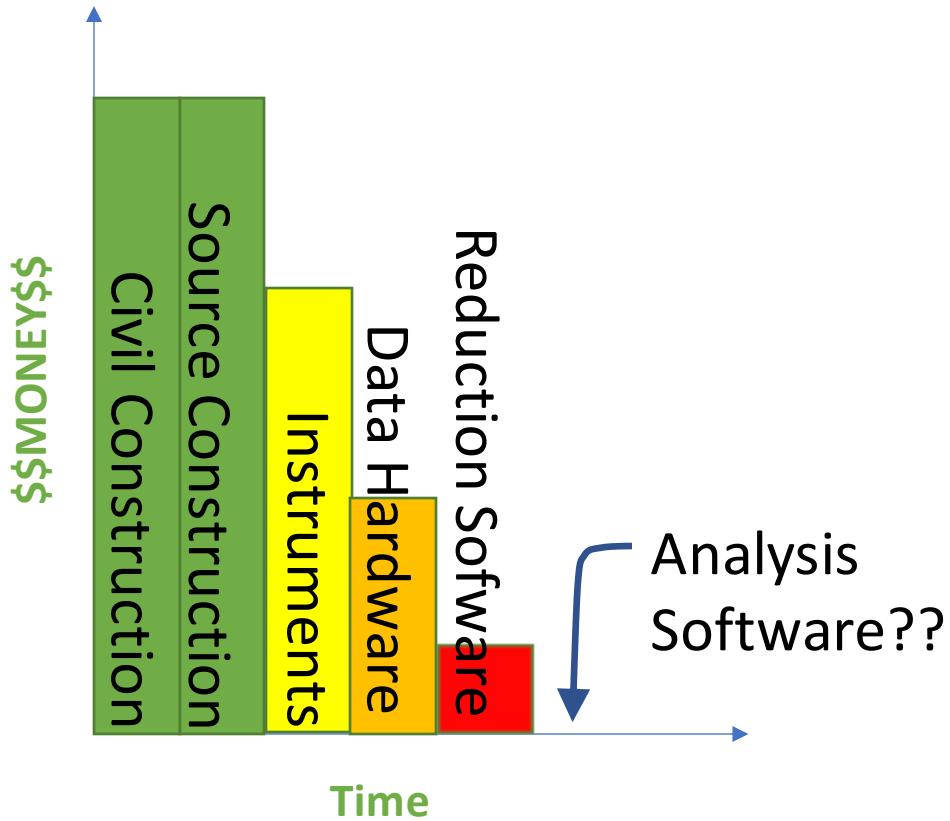
Paper / Thesis



Analysis Software – Who's job is it anyway

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

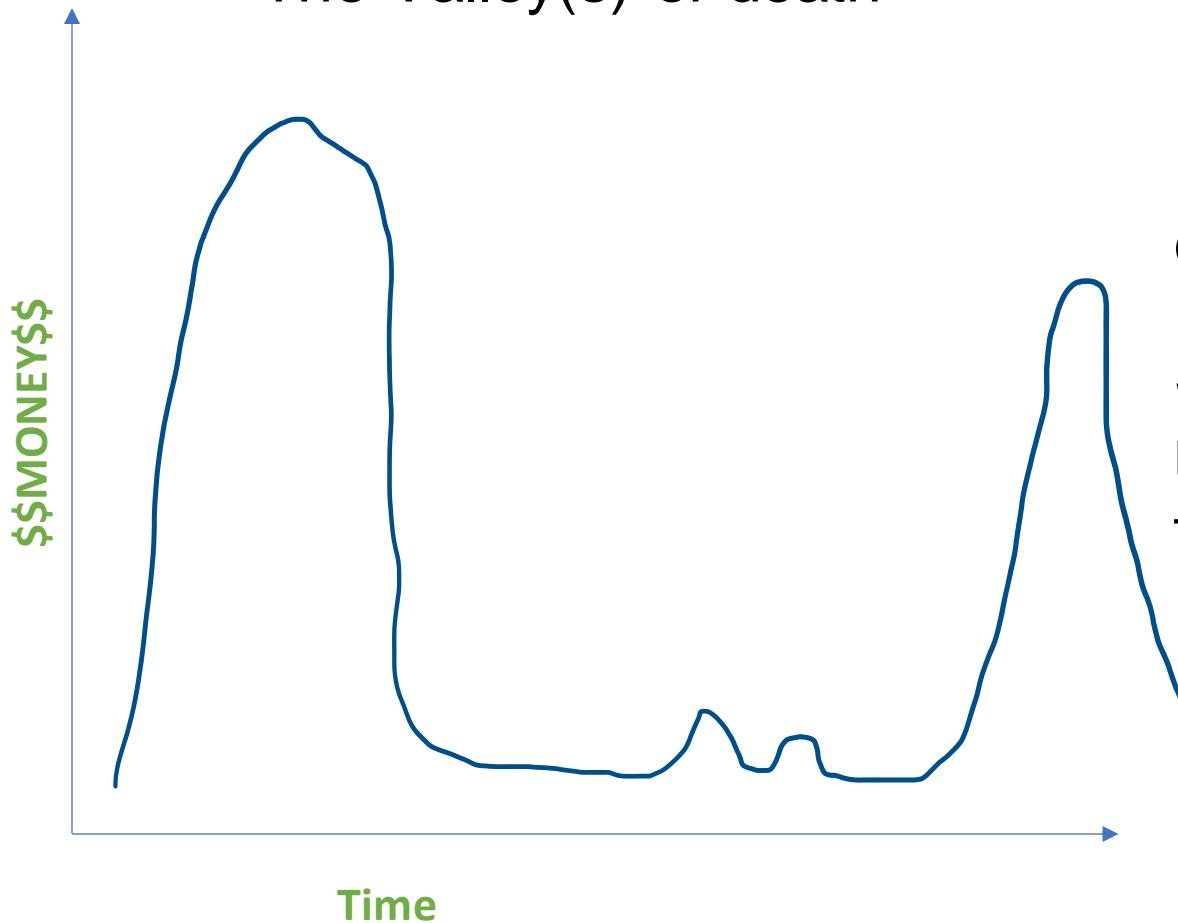
Or maybe
→ The Domain Science funding agencies?

The resource Problem Part II

The feast and famine roller coaster

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

The Valley(s) of death

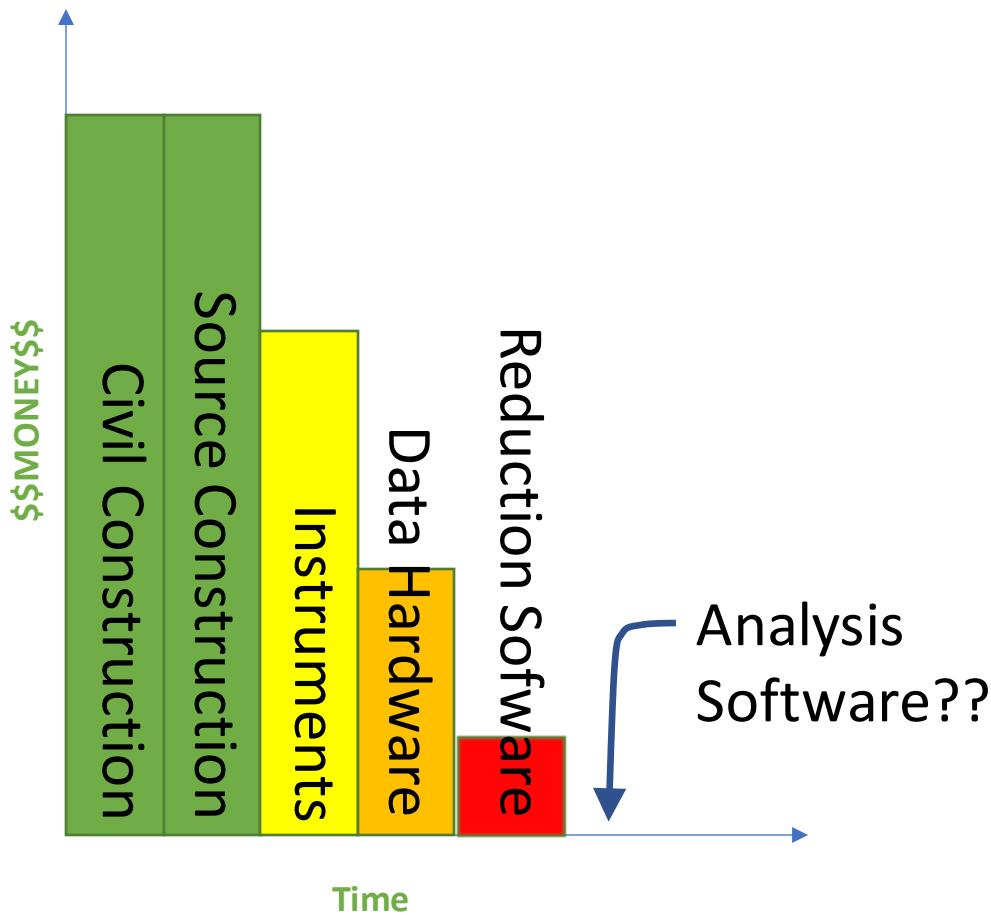


Facility directors discretion, NIH, NSF, DOE, etc.

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 resource Problem Reframed – The Need

- 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



EUROPEAN
SPALLATION
SOURCE



The SasView Approach

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



Science & Technology Facilities Council



National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

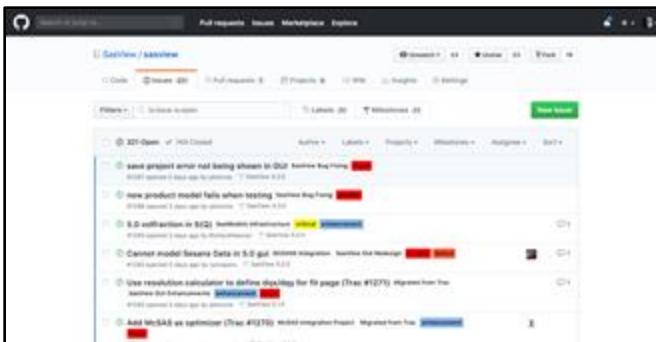


Open, Collaborative, Community Development

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

(Zenodo) DOI for each release

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



<https://github.com/SasView>

- Website
- Documentation: in-program & online
- Written Tutorials
- Video Tutorials (YouTube)
- scattering schools/workshop
- university courses
- Bootcamps & regional workshops
- Twitter
- help@sasview.org
- users@sasview.org

Rolling 5 Year Roadmap

Model Marketplace for Users to share their models



<http://marketplace.sasview.org>



<https://www.sasview.org>



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 ...

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 ...

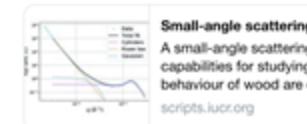
- New dependencies vs long term maintenance (sustainability)
 - Code quality vs long term sustainability
 - Changing/degrading the current user experience for the existing user base



Paavo A. Penttilä
@PaavoPenttilä

Follow

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!



11:33 PM - 26 Mar 2019

1 Retweet 8 Likes



1



8



Paavo A. Penttilä
@PaavoPenttilä

Follow

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

Paavo A. Penttilä @PaavoPenttilä

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...

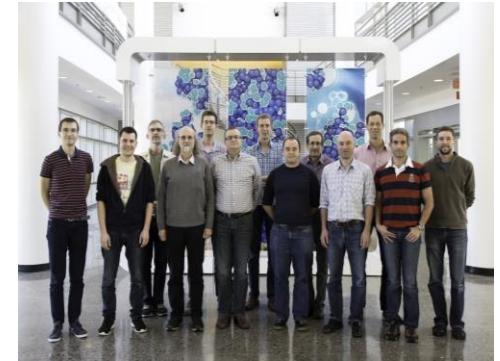
12:02 AM - 27 Mar 2019



History and Status of SasView

- 2006; originates in NSF DANSE project
- 2013; transitions into a community project
- 2016; Sine2020 project funded
- 2022; 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?"
- Publications? > 100/year



SasView Model: Some Current Problems

The price of success:

- Many people view the project as a well-funded group of professionals → barrier to contribution – e.g: Old website and no graphics = not serious; Nice graphic slick website = professional → way beyond my skills – how to navigate?
- Large project with many moving pieces is a barrier to new volunteer coders
- Heavily geared towards colloid science

- The tragedy of the commons: somebody else will take care of the problem.
- Hard to get non-coders to believe they can contribute equally – it is unfortunately in the name “code camp” ... words matter → SasView camp?
- Building community, especially during a pandemic is HARD WORK
- Lack of funds for small things ... but beware too much money?
- Hard to keep up with increasing security issues with these resources so far.

The resource Problem Reframed – The Solution

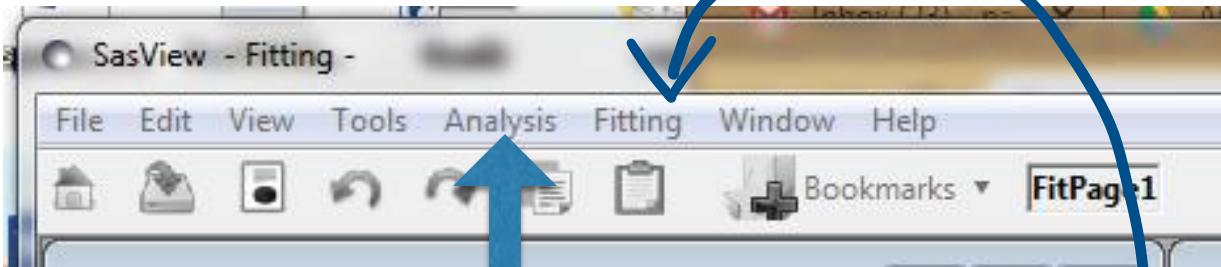
The SasView approach

- 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)

So ...

What Can SasView Do Currently?

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

Fitting Perspective: 1D Analysis

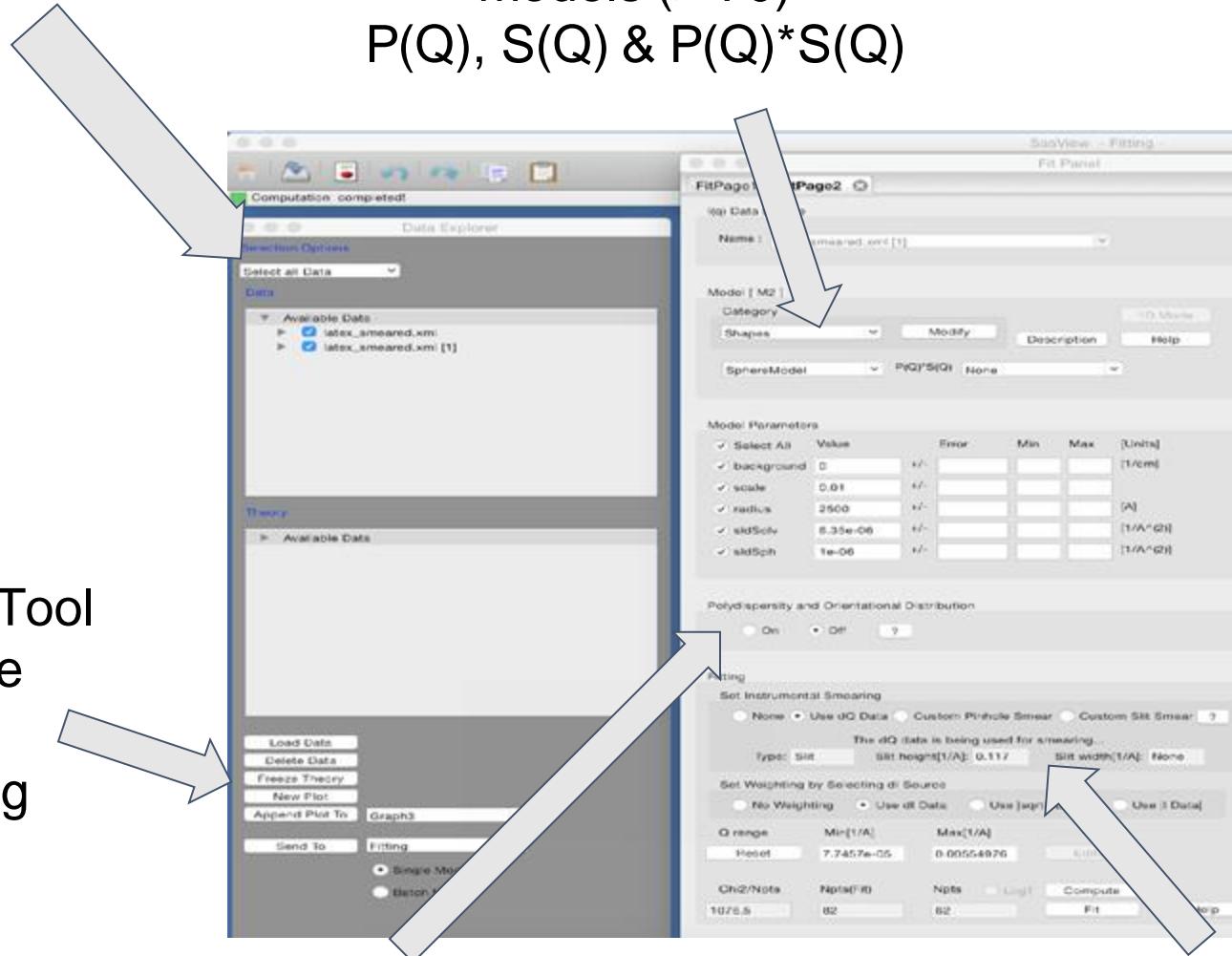
Data management

Common data formats

supported, including

NXCansas & *cansas1D*

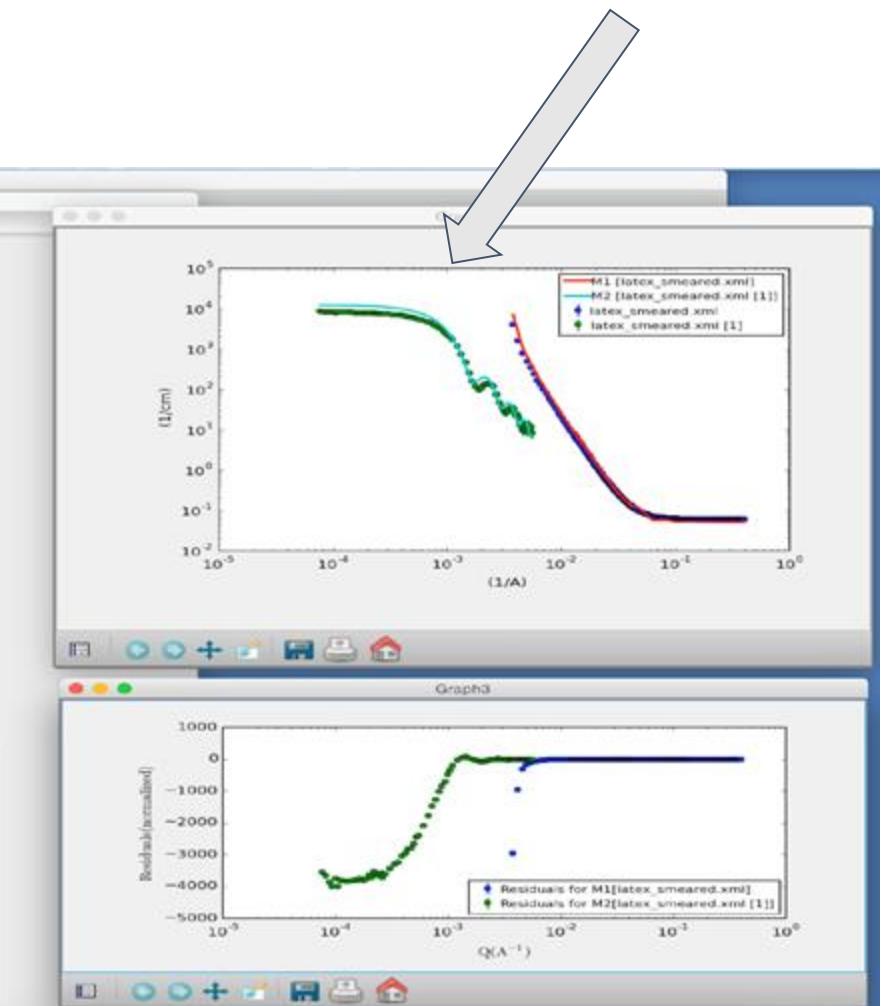
Wide choice of built-in
models (> 70)
 $P(Q)$, $S(Q)$ & $P(Q)^*S(Q)$



Analysis Tool
Choice
&
Plotting

Generic parameter polydispersity
Choice of distribution and distribution parameters

Simultaneous fitting



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

Fitting Perspective: 2D Analysis

Fit panel - Active Fitting Optimizer: Levenberg-Marquardt
FitPage1

No data loaded

Model

Category	Model name	Structure factor
Cylinder	cylinder	None

Parameter

Parameter	Value	Min	Max	Units
scale	1.0	0.0	∞	
background	0.001	$-\infty$	∞	cm^{-1}
cylinder				
sld	4	$-\infty$	∞	$10^{-6}/\text{\AA}^2$
sld_solvant	1	$-\infty$	∞	$10^{-6}/\text{\AA}^2$
radius	50	0.0	∞	\AA

Polydispersity

Length: 400 \AA

theta: 60 degrees

phi: 85 degrees

Options

Polydispersity: Min range 0.001 \AA^{-1} , Max range 0.3 \AA^{-1} , Smearing: None

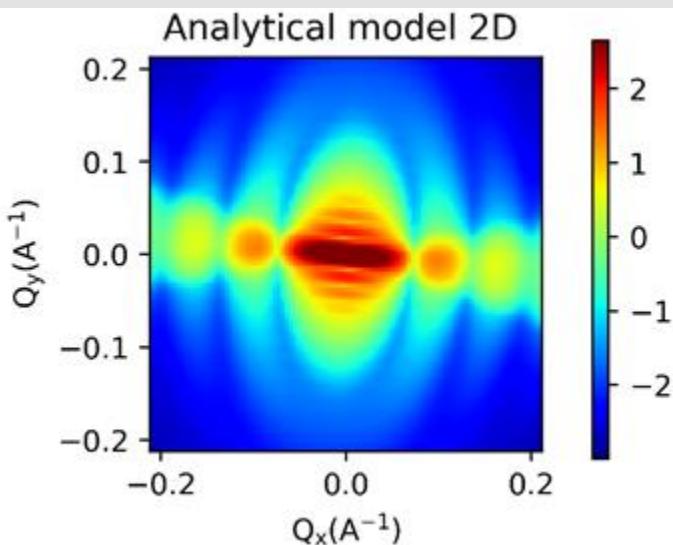
2D view

Magnetism

Fitting details

χ^2 : 24895

Show Plot Fit Help



FitPage1

No data loaded

Model

Category	Model name	Structure factor
Sphere	core_shell_sphere	None

Parameter

Parameter	Value	Min	Max	Units
scale	1.0	0.0	∞	
background	0.001	$-\infty$	∞	cm^{-1}
core_shell...				
radius	60.0	0.0	∞	\AA
thickness	10.0	0.0	∞	\AA
sld_core	1.0	$-\infty$	∞	$10^{-6}/\text{\AA}^2$
sld_shell	2.0	$-\infty$	∞	$10^{-6}/\text{\AA}^2$
sld_solv...	3.0	$-\infty$	∞	$10^{-6}/\text{\AA}^2$

Options

Polydispersity: Min range 0.0005 \AA^{-1} , Max range 0.5 \AA^{-1} , Smearing: None

2D view

Magnetism

Fitting details

χ^2 : ---

Calculate Fit Help

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.

Graph3

P123_D2O_40_percent.dat

Q_y(\AA^{-1})

Q_x(\AA^{-1})

0 0.05 -0.05

0 -5 -10

FitPage1

No data loaded

Model

Polarisation/Magnetic Scattering

Parameter	Value	Min	Max	Units
up_frac_i	0.0	0.0	1.0	
up_frac_f	0.0	0.0	1.0	
up_angle	0.0	0.0	360.0	degrees
up_phi	0.0	0.0	180.0	degrees
sld_core_M0	1.0	$-\infty$	∞	$10^{-6}/\text{\AA}^2$
sld_core_mtheta	0.0	-90.0	90.0	degrees
sld_core_mphi	0.0	-180.0	180.0	degrees
sld_shell_M0	1.5	$-\infty$	∞	$10^{-6}/\text{\AA}^2$
sld_shell_mtheta	0.0	-90.0	90.0	degrees
sld_shell_mphi	0.0	-180.0	180.0	degrees

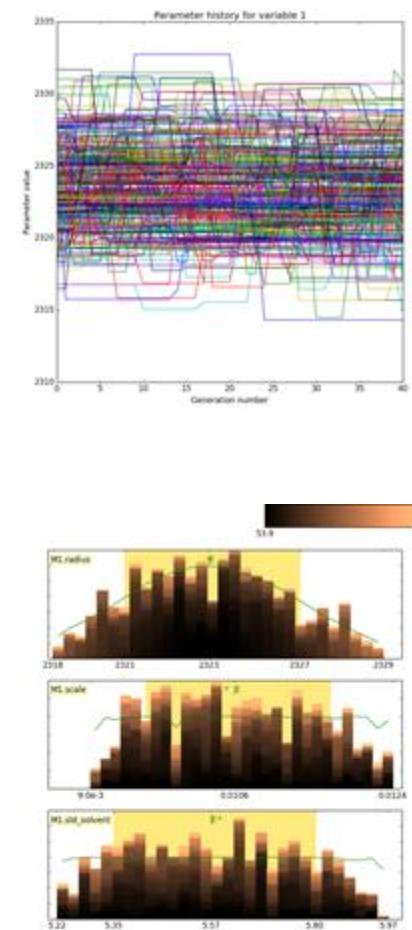
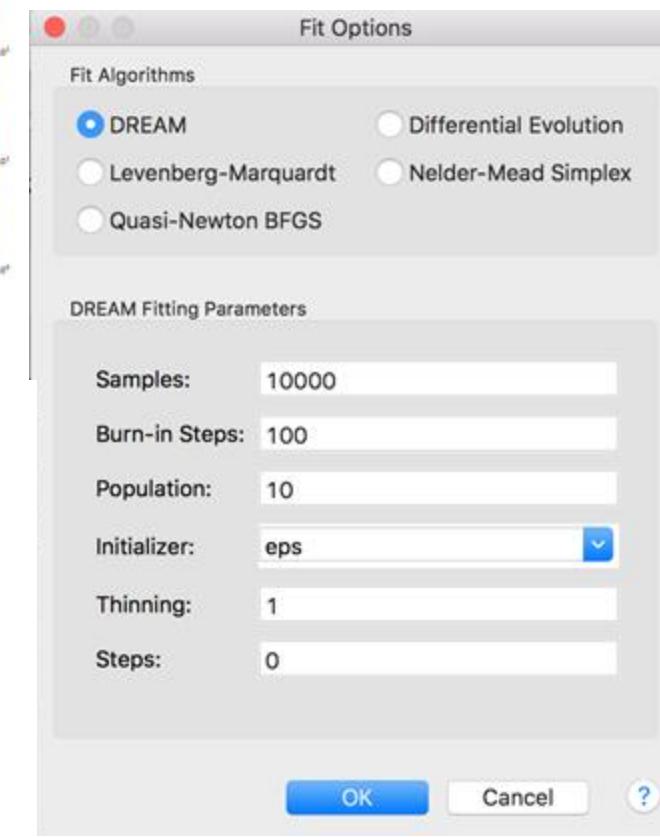
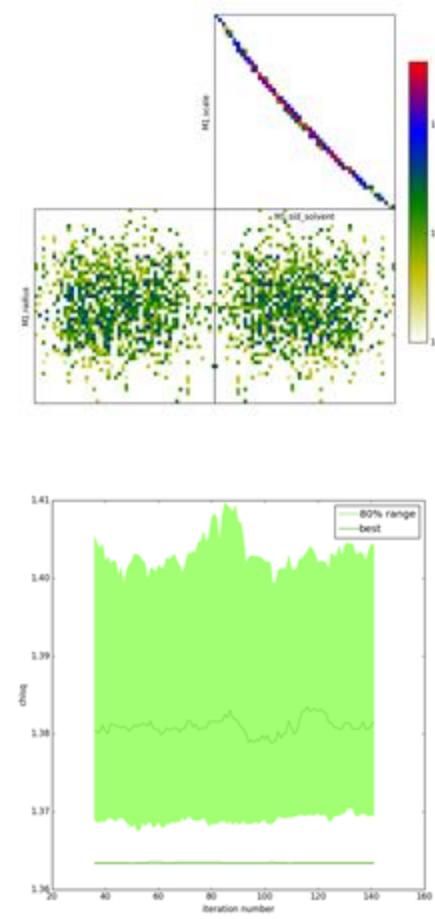
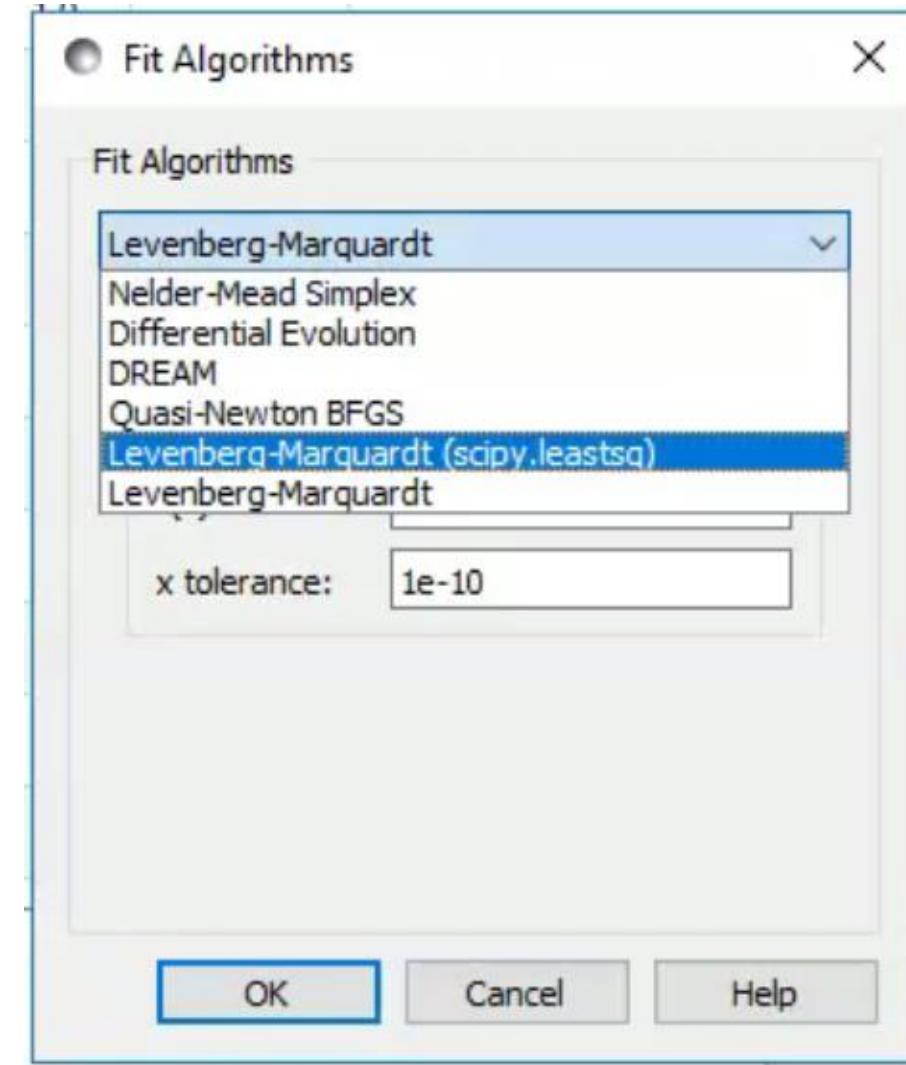
Display magnetic angles

Calculate Fit Help

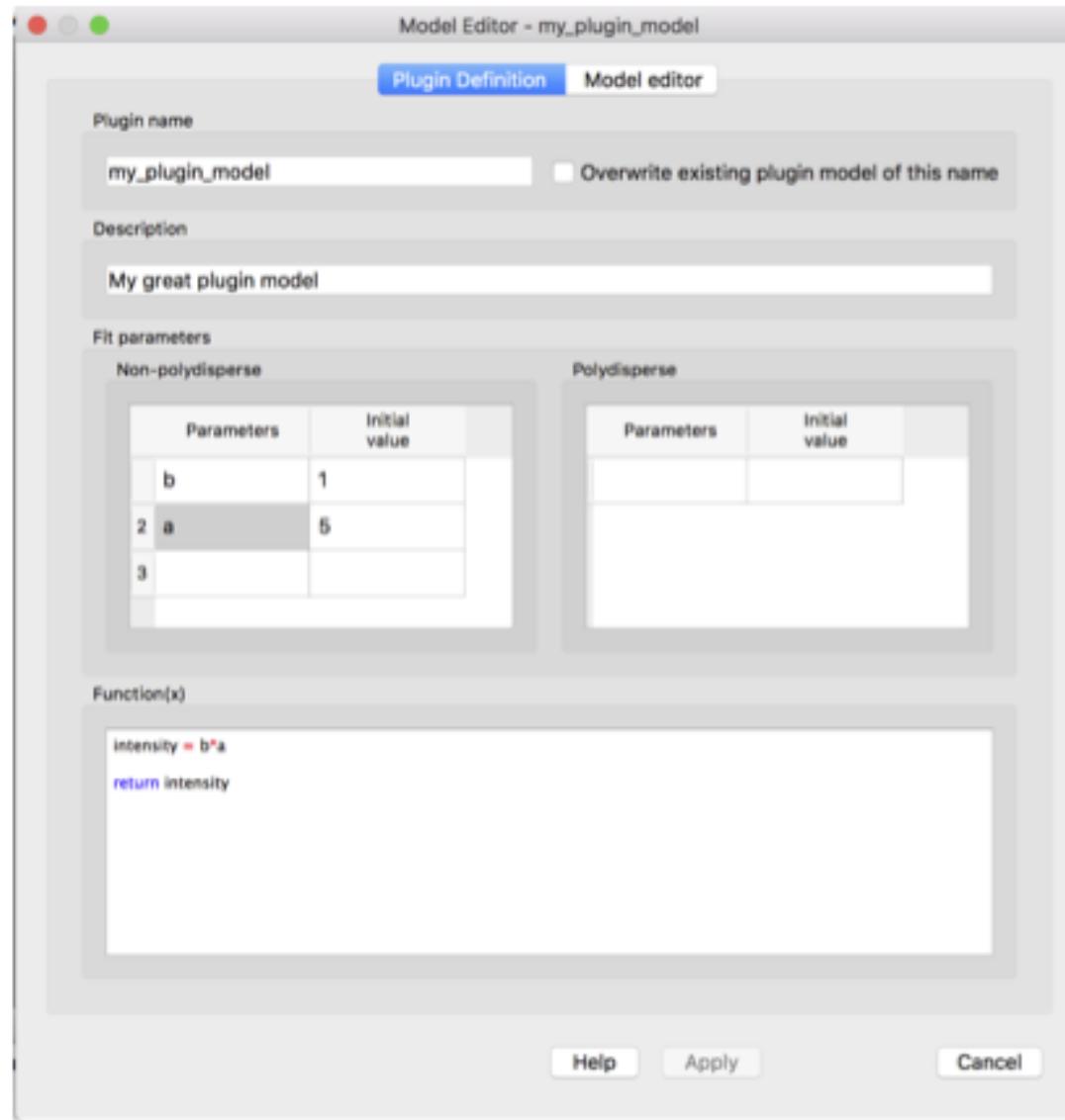
Turning on GPU Option highly recommended for fitting

Choice of Fitting Algorithms

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



Plugin Model Editor



The screenshot shows the 'Model Editor - my_plugin_model' window with the 'Model editor' tab selected. It displays a 'Model' section with a summary box containing the text 'Calculates my_plugin_model.' and 'My great plugin model'. Below this are sections for 'References' and 'Authorship and Verification'. A large code editor area shows the generated Python code:

```
"""Author:"" ---- ""Date:"" 2019YYY-03m-19d
"""Last Modified by:"" ---- ""Date:"" 2019YYY-03m-19d
"""Last Reviewed by:"" ---- ""Date:"" 2019YYY-03m-19d

from math import *
from numpy import inf

name = "my_plugin_model"
title = "User model for my_plugin_model"
description = """My great plugin model"""

parameters = [
    # "name": "units", default: [lower, upper], "type": "description",
    {"b": 1.0, [-inf, inf], "type": "description"}, #>
    {"a": 5.0, [-inf, inf], "type": "description"} #>

def iq(x, b, a):
    """Absolute scattering"""
    intensity = b*a

    return intensity
    ## uncomment the following if iq works for vector x
    #iq.vectorized = True

#def iqxy(x, y, b, a):
#    """Absolute scattering of oriented particles."""
#    ...
#    # return oriented_form(x, y, args)
#    ## uncomment the following if iqxy works for vector x, y
#    #iqxy.vectorized = True
```

Annotations highlight specific parts of the code: 'Calculates my_plugin_model.' and 'My great plugin model' are highlighted in yellow; the parameter definitions ('parameters' block) and the 'iq' function definition are also highlighted in yellow. To the right of the code editor, four cyan-colored callout boxes provide context: 'Model Documentation built in' points to the summary text; 'Parameter definition' points to the 'parameters' block; 'Model calculation' points to the 'iq' function definition; and 'Tests etc' points to the commented-out code at the bottom.

Plugin Model Editor: Python & C model files

```
cylinder.py x
182 import ...
184
185 name = "cylinder"
186 title = "Right circular cylinder with uniform scattering length density."
187 description = """
188     f(q, alpha) = 2*(sld - sld_solvent)*V*sin(q*cos(alpha)/2)
189     /[q*cos(alpha)/2]*J1(q*sin(alpha))/[q*sin(alpha)]
190
191     P(q, alpha) = scale/V*f(q, alpha)^2+background
192     V: Volume of the cylinder
193     R: Radius of the cylinder
194     L: Length of the cylinder
195     J1: The bessel function
196     alpha: angle between the axis of the
197     cylinder and the q-vector for 1D
198     :the output is P(q)=scale/V=integral
199     from pi/2 to zero of...
200     f(q, alpha)^2*sin(alpha)*dalpha + background
201
202     """
203 category = "shape:cylinder"
204
205     [{"name": "name", "units": default, [lower, upper], "type": "description"}, "parameters": [{"sld": "1e-6/Ang^2", 4, [-inf, inf], "sld": "Cylinder scattering length density"}, {"sld_solvent": "1e-6/Ang^2", 1, [-inf, inf], "sld": "Solvent scattering length density"}, {"radius": "Ang", 20, [0, inf], "volume": "Cylinder radius"}, {"length": "Ang", 400, [0, inf], "volume": "Cylinder length"}, {"theta": "degrees", 60, [-360, 360], "orientation": "cylinder axis to beam angle"}, {"phi": "degrees", 60, [-360, 360], "orientation": "rotation about beam"}]
206
207 source = ["lib/polevi.c", "lib/sas_J1.c", "lib/gauss76.c", "cylinder.c"]
208
209 def ER(radius, length):
210     """
211         Return equivalent radius (ER)
212     """
213     ddd = 0.75 * radius + (2 * radius * length + (length + radius) * (length + pi * radius))
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
```

```
cylinder.c x
1 #define INVALID(v) (v.radius<0 || v.length<0)
2
3 static double
4 form_volume(double radius, double length)
5 {
6     return M_PI*radius*radius*length;
7 }
8
9 static double
10 fq(double qab, double qc, double radius, double length)
11 {
12     return sas_2J1x_x(qab*radius) * sas_sinx_x(qc*0.5*length);
13 }
14
15 static double
16 orient_avg_1D(double q, double radius, double length)
17 {
18     // translate a point in [-1,1] to a point in [0, pi/2]
19     const double zm = M_PI_4;
20     const double zb = M_PI_4;
21
22     double total = 0.0;
23     for (int i=0; i<GAUSS_N ;i++) {
24         const double theta = GAUSS_Z[i]*zm + zb;
25         double sin_theta, cos_theta; // slots to hold sincos function output
26         // theta (theta,phi) the projection of the cylinder on the detector plane
27         SINCOS(theta , sin_theta, cos_theta);
28         const double form = fq(q*sin_theta, q*cos_theta, radius, length);
29         total += GAUSS_W[i] * form * form * sin_theta;
30     }
31     // translate dx in [-1,1] to dx in [lower,upper]
32     return total*zm;
33 }
34
35 static double
36 Iq(double q,
37     double sld,
38     double solvent_sld,
39     double radius,
40     double length)
41 {
42     const double s = (sld - solvent_sld) * form_volume(radius, length);
43     return 1.0e-4 * s * s * orient_avg_1D(q, radius, length);
44 }
45
```

Complex numbers not originally supported – a bit in C now

BATCH in Fitting

Data Explorer

- Data Theory

Data

- Load data Delete Data Unselect all
- SILIC010_noheader_3col.DAT
- SILIC010_noheader.DAT
- SILIC010.DAT
- P123_D2O_40_percent.dat
- MyNewDataName
- apo ferritin.txt

Send data to Fitting Batch mode Swap data

Plot Create New Append to Help

FitPage1

Data loaded from: SILIC010_noheader_3col.DAT

Model Fit Option Magnetism

Model name: sphere Structure factor: None

Parameter	Value	Min	Max	Units
scale	1.0	0.0	∞	
background	0.001	$-\infty$	∞	cm^{-1}
sphere				
sls	1	$-\infty$	∞	$10^{-6}/\text{\AA}^2$
sls_solv	6	$-\infty$	∞	$10^{-6}/\text{\AA}^2$
radius	50	0.0	∞	\AA

Options Fitting details Fitting error

- Polydispersity
- Magnetism
- Chain fit

Min range 0 \AA^{-1} Max range 0.12083 \AA^{-1} Smearing: χ^2 7458.9

Fit Help

Batch Fitting Results

File

Chi2	Data	scale	scale (Err)	background	background (Err)	radius	radius (Err)	thickness	thickness (Err)	sls_core	sls_core (Err)	sls_shell	sls_solv
10.675	shell contrast AOT me_SANS	7.5704	90.49	0.20153	0.0035153	33.188	7.5401	1.8981	13.11	2.8355	1.0192	2	3
47.213	drop contrast AOT me_SANS	13.908	$9.9988e+07$	0.15359	0.0033333	32.166	93.014	0	39.688	2.5733	$1.5341e+06$	2	3
4.3378	core contrast AOT me_SANS	64.908	49991	0.029773	0.03639	23.867	100.43	0.45126	187.17	2.8903	42.867	2	3

< >

BatchPage303 Plot Close Help

Graph9

Intensity(cm^{-1})

Graph8

Residuals(normalized)

Residuals for M303[shell contrast AOT me_SANS]

Constrained Fitting

BatchPage2 FitPage3 FitPage4 FitPage5 Const. Simul. Fit +

Source choice for simultaneous fitting

Single fits Batch fits

FitPage	Model	Data	Mnemonic
1 <input checked="" type="checkbox"/> FitPage3	core_shell_sphere	AOT_Microemulsion...	M3
2 <input checked="" type="checkbox"/> FitPage4	core_shell_sphere	AOT_Microemulsion...	M4
3 <input checked="" type="checkbox"/> FitPage5	core_shell_sphere	AOT_Microemulsion...	M5

Constraints

Add constraints

Constraint
1 <input checked="" type="checkbox"/> M3:scale = M2.scale * sin(90)

Complex Constraint

2 parameter constraint

M3 sld_core = M3 background

Edit complex constraint:

M3:sld_core = M3.sld_shell

Add Help

Fitting Perspective: Invariant Calculation

Invariant

I(q) Data Source
For more information, click on Details button.

Name: **latex_smeared.xml [1]**

Total Q Range (1/ \AA): Min : **7.7457e-05** Max : **0.00554976**

Outputs

Volume Fraction **2.26e-13** +/- **1.67e-18**
Specific Surface [1/ \AA]
Invariant Total [Q*] **0.000445** +/- **7.4e-06** [1/(cm \cdot A 3)]

Customized Inputs

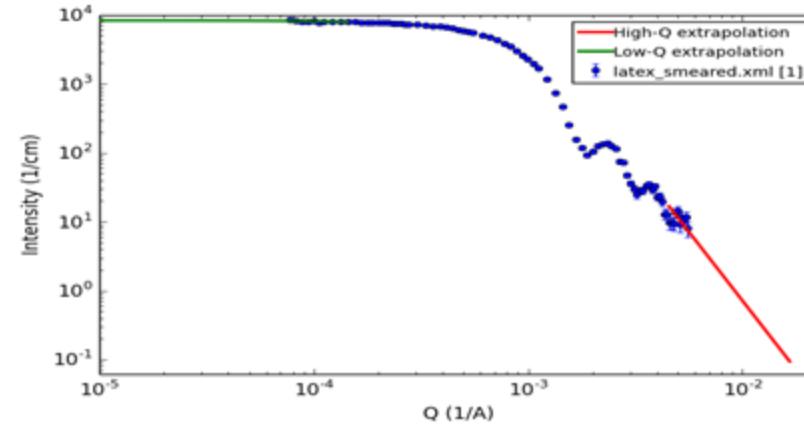
Background: **0.0** [1/cm] Scale: **1.0**
Contrast: **1.0** [1/] Porod Constant: [1/(cm \cdot A 4)
(optional)]

Extrapolation

Extrapolation Maximum Q Range [1/ \AA]: Min: **1e-05** Max: **10**

Low Q High Q

Enable Extrapolate Low Q Enable Extrapolate high-Q
Npts **10** Npts **10**
 Guinier Power Law
 Fix Fit
Power **4**



Invariant Details

Invariant Chart

Q* from Low-Q : **0.638%**
Q* from Data : **93.1%**
Q* from High-Q : **6.24%**

Numerical Values

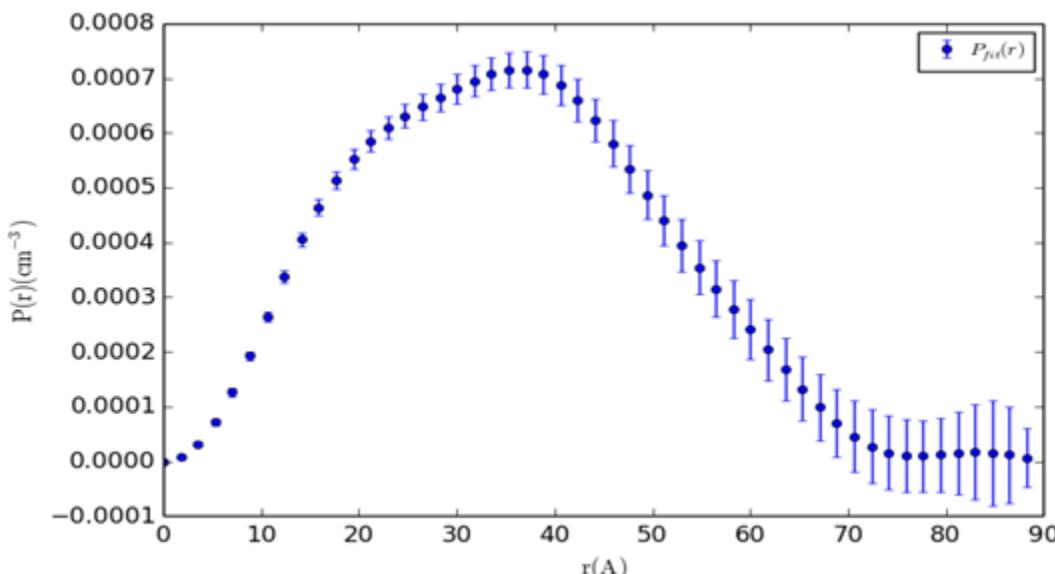
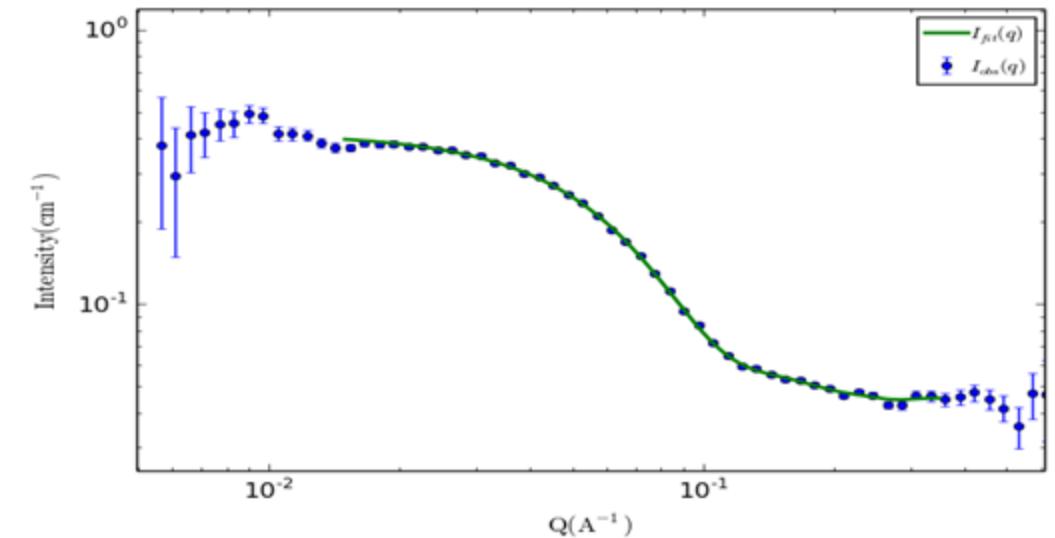
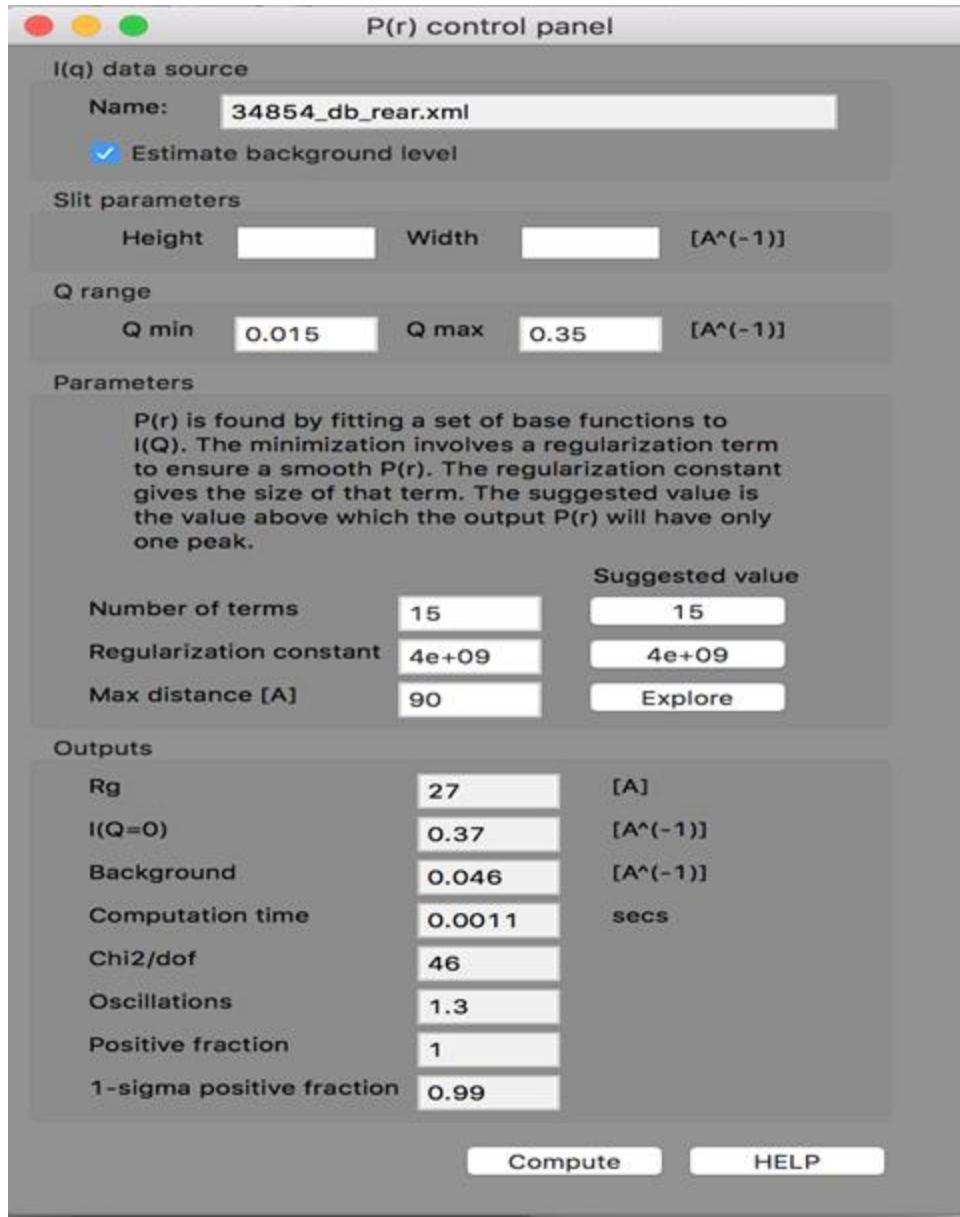
Q* from Low-Q **2.84e-06** +/- **1.11e-09** [1/(cm \cdot A 3)]
Q* from Data **0.000415** +/- **1.18e-06** [1/(cm \cdot A 3)]
Q* from High-Q **2.78e-05** +/- **7.3e-06** [1/(cm \cdot A 3)]

Warning

Extrapolated contribution at High Q is higher than 5% of the invariant.
The sum of all extrapolated contributions is higher than 5% of the invariant.
The calculations are likely to be unreliable!

Ok

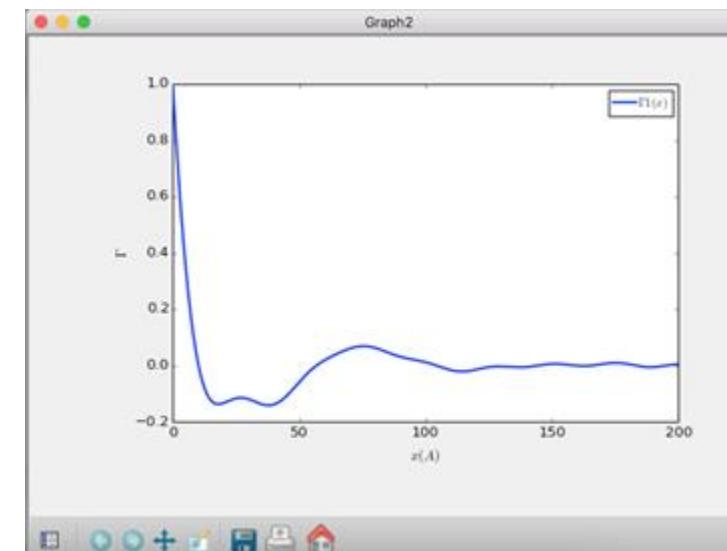
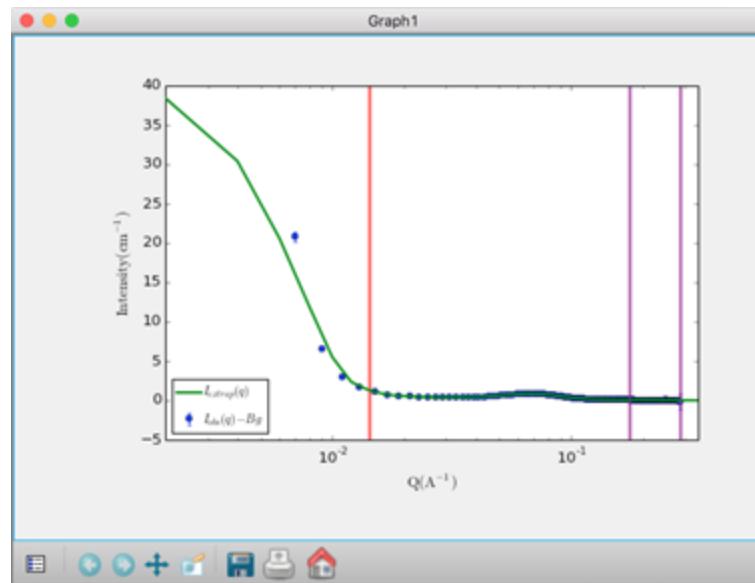
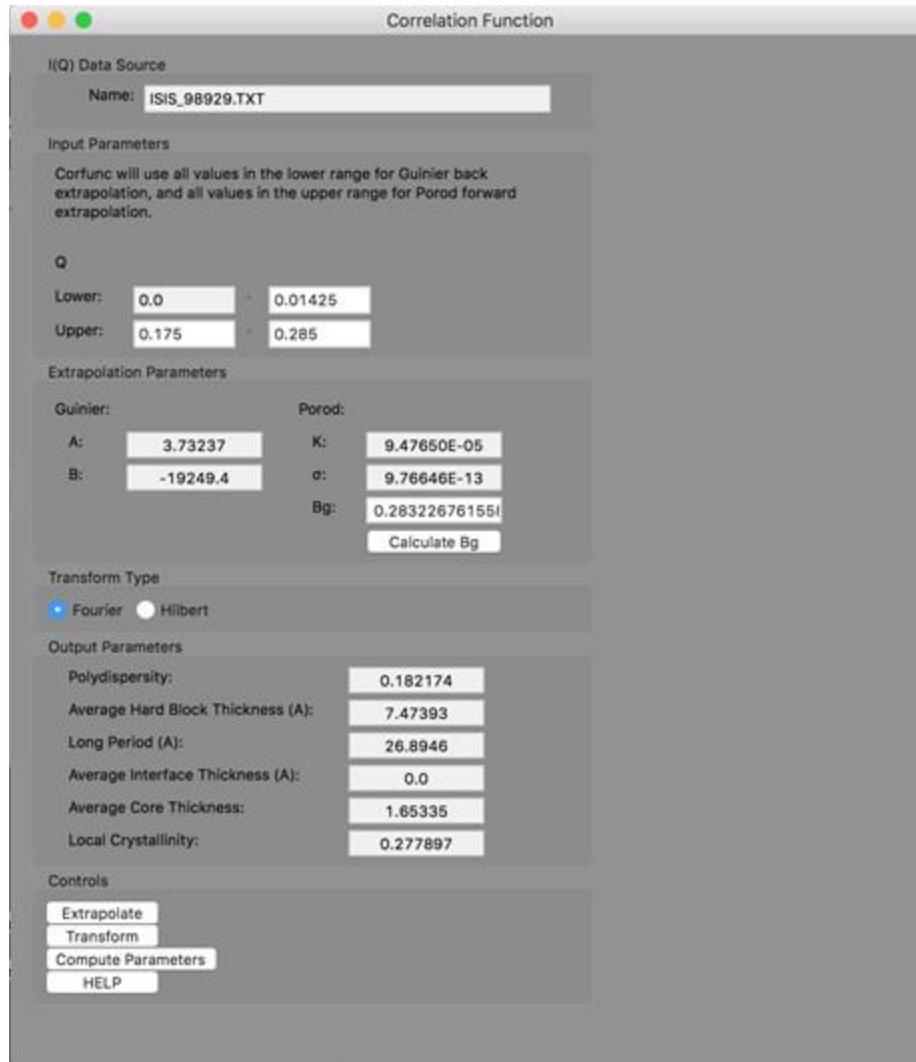
Fitting Perspective: P(r) Inversion



Fitting Perspective: Correlation Function Analysis (new!)

CCP13 (Fiber Diffraction) legacy code (Fortran)

Implemented by an ISIS summer student)



SESANS Analysis

Automatic Hankel Transform of SANS
models
(TU Delft & ISIS)

Fit panel - Active Fitting Optimizer: Levenberg-Marquardt

FitPage1

I(q) Data Source

Name : sphere2micron.ses

Model [M1]

Category

Sphere Description Help

sphere None

Model Parameters

Parameter	Value	Error	Min	Max	[Units]
<input checked="" type="checkbox"/> scale	0.005323	+/- 5.2808e-0	0	inf	
<input type="checkbox"/> background	0	+/-	-inf	inf	1/cm
<input type="checkbox"/> sld	1	+/-	-inf	inf	1e-6/Ang^2
<input type="checkbox"/> sld_solvent	6	+/-	-inf	inf	1e-6/Ang^2
<input checked="" type="checkbox"/> radius	10383	+/- 114.45	0	inf	Ang

Polydispersity and Orientational Distribution

On Off ?

Fitting

Set Instrumental Smearing

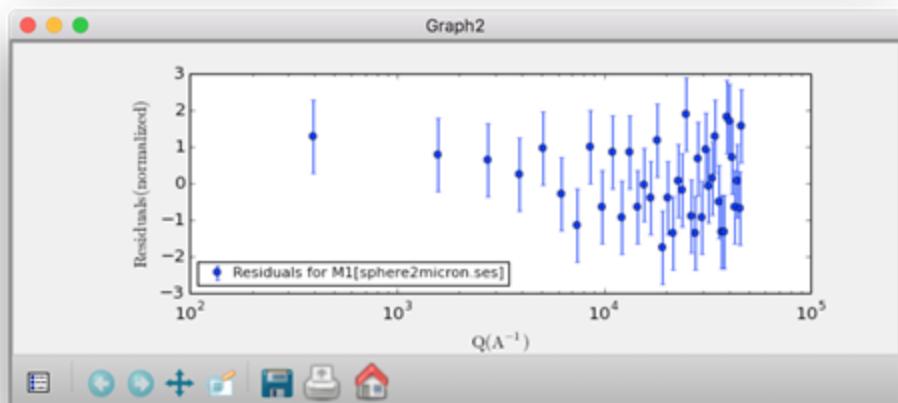
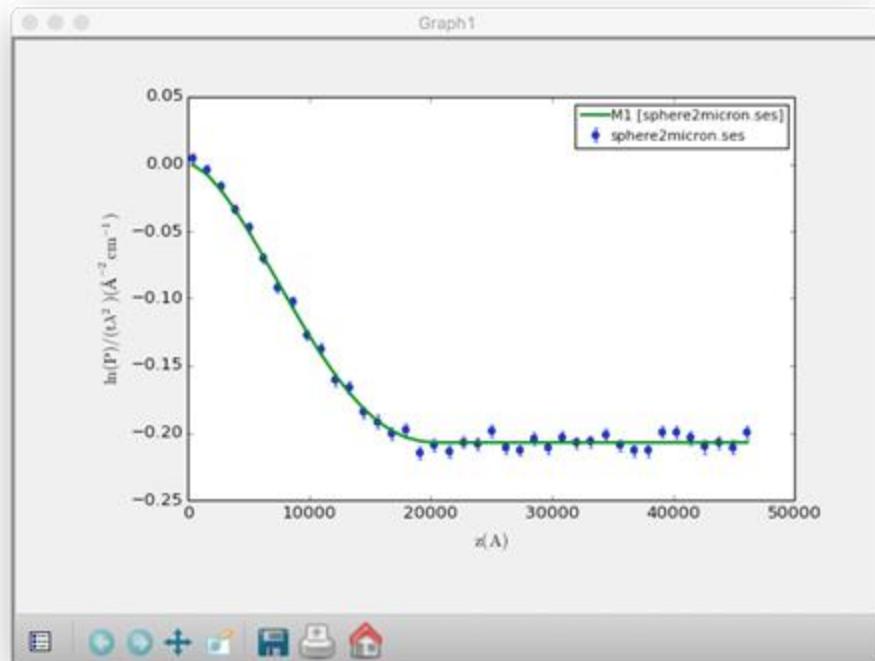
None Use dQ Data Custom Pinhole Smear Custom Slit Smear ?

The dQ data is being used for smearing...

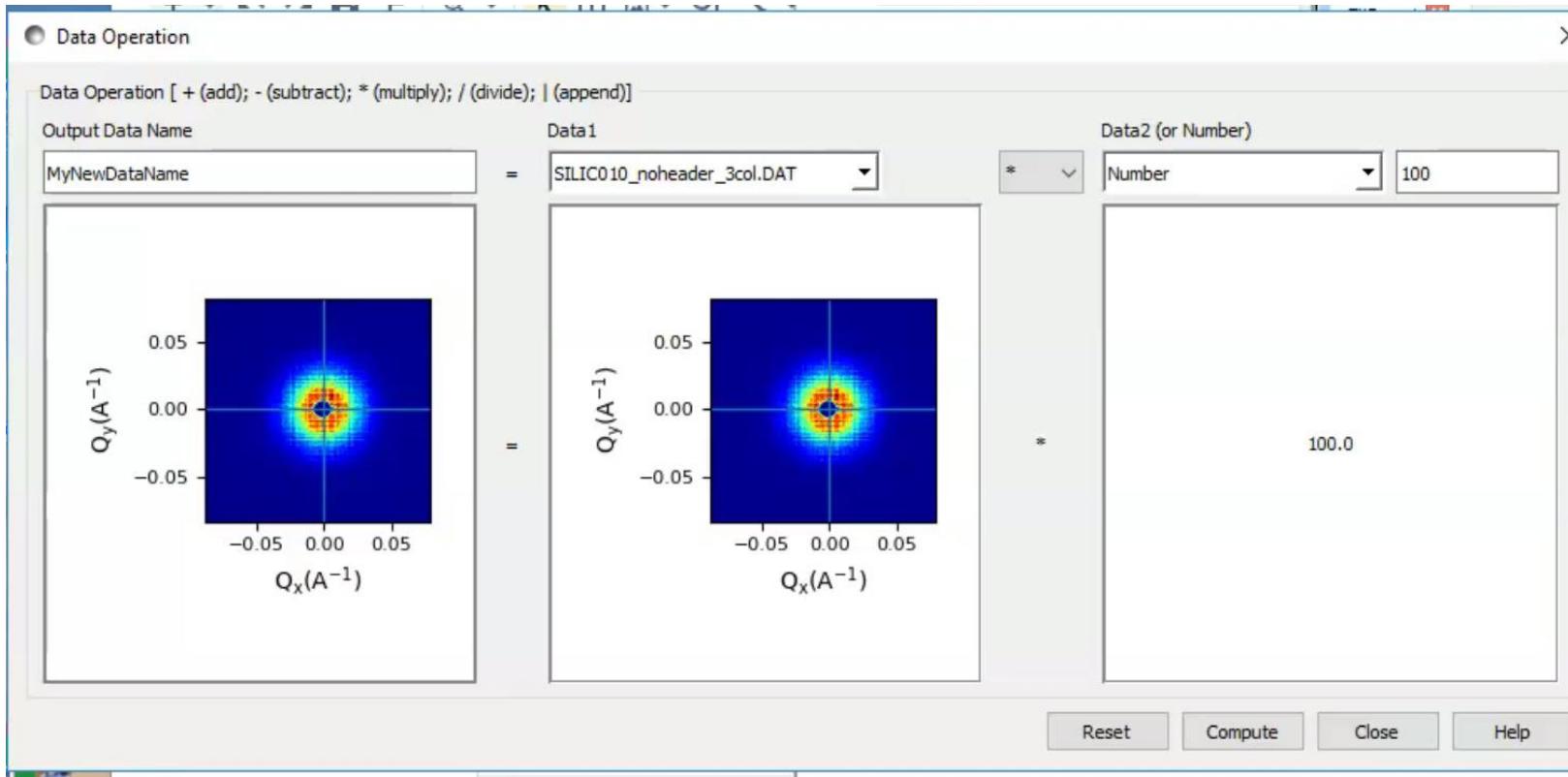
Type: Pinhole 19.578 dQ[%]: 2304.9

Set Weighting by Selecting dI Source

No Weighting Use dI Data Use |sqrt(I Data)| Use |I Data|



Data Operations and 1D stuff

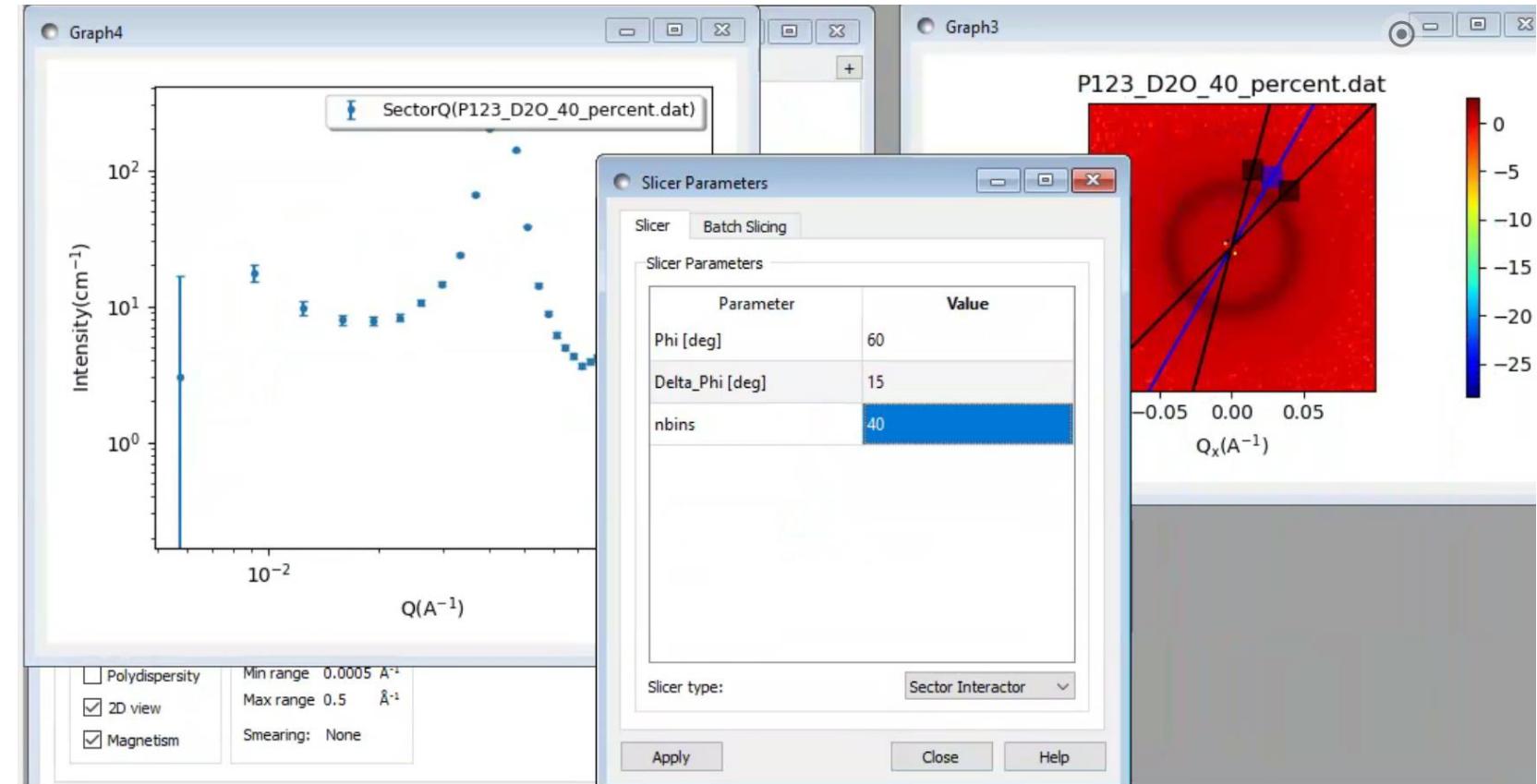
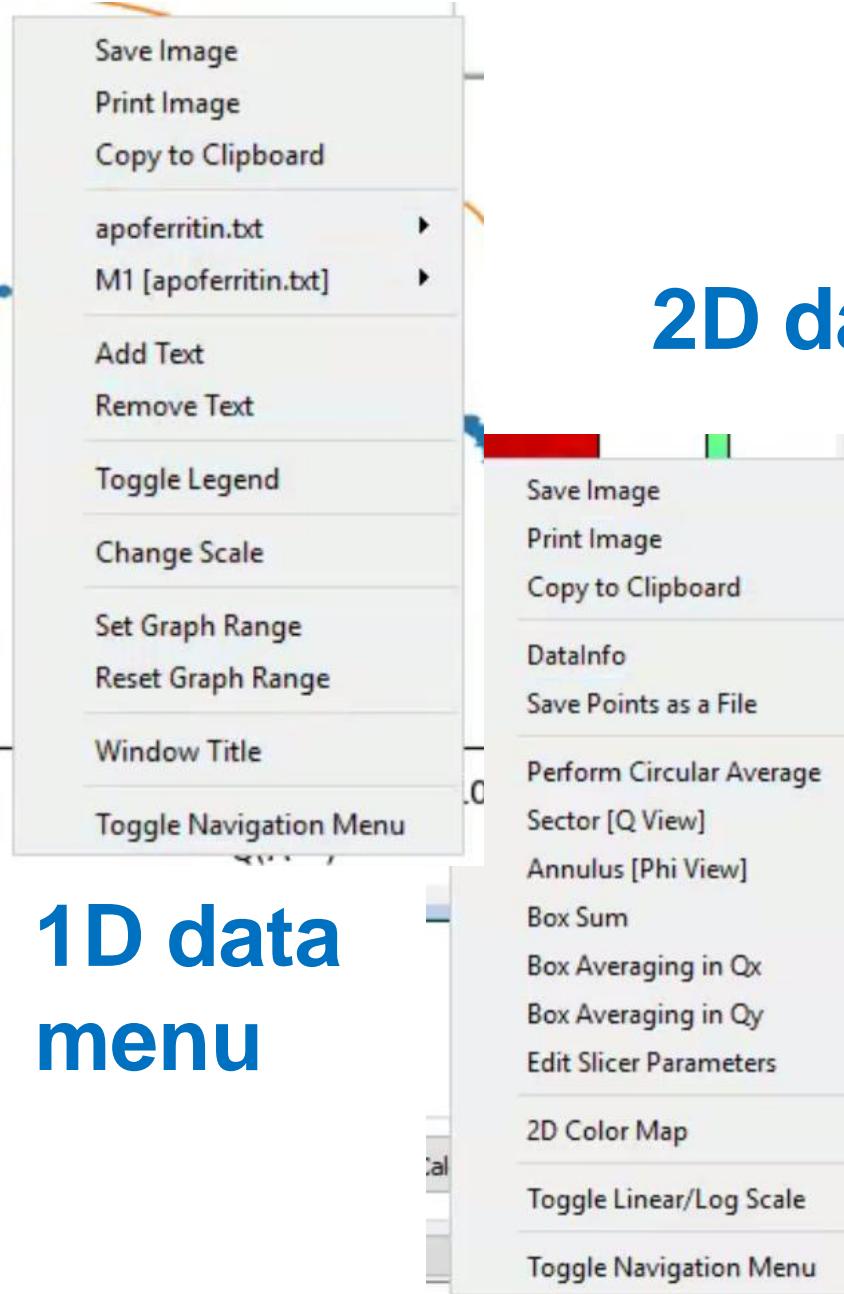


PROBLEM: If using 2 data sets, ALL Q points must match (not interpolation etc)

DATA and Slicers

2D data menu and slicer example

1D data menu



Command Line and Jupyter Notebooks

SasCalc example

A simple example demonstrating pair distance distribution function $P(r)$ inversion. In SasView it is calculated using Moore formula (1980)

```
In [2]: from sas.sascalc.dataloader.loader import Loader
from sas.sascalc.pr.invertor import Invertor
import matplotlib.pyplot as plt
import numpy as np

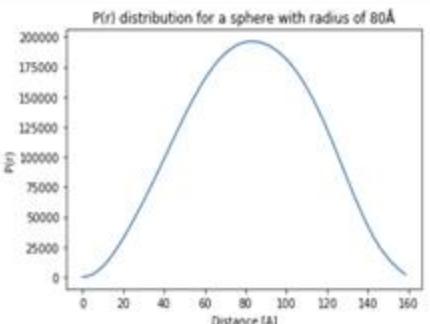
loader = Loader()
test_data = loader.load("sphere_80_err.txt")
x_data = test_data[0].x
y_data = test_data[0].y
z_data = test_data[0].dy

pr = Invertor()
pr.x = x_data
pr.y = y_data
pr.err = z_data

pr.alpha = 2.6e-5
pr.d_max = 160

#nfunc - number of base functions to use.
out, cov = pr.invert(nfunc=13)
pr_value = []
err_value = []
r = np.arange(0.0, pr.d_max, pr.d_max / pr.x.size)
for r_i in r:
    (value, err) = pr.pr_err(out, cov, r_i)
    pr_value.append(value)
    err_value.append(err)

plt.plot(r,pr_value)
plt.xlabel("Distance [Å]")
plt.ylabel("P(r)")
plt.title('P(r) distribution for a sphere with radius of 80Å')
plt.show()
```

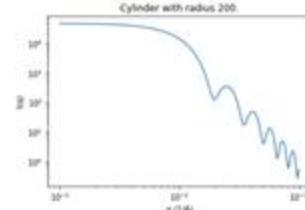


SasModels example

SasModels is a library of form and structure factor functions. The following example demonstrates how to generate a scattering pattern of a form factor of the cylinder model using sasmodels library. It requires sasmodels to be installed in the path.

```
In [35]: from numpy import linspace
from matplotlib import pyplot as plt
from sasmodels.core import load_model
from sasmodels.direct_model import call_kernel

model = load_model('cylinder')
q = linspace(-3., -1., 200)
kernel = model.make_kernel({q})
Iq = call_kernel(q, kernel, dict(radius=200))
plt.loglog(q, Iq)
plt.xlabel('q (Å-1)')
plt.ylabel('I(q)')
plt.title('Cylinder with radius 200.')
plt.show()
```



Fitting model function to data using bumps

The model functions from sasmodels can be used to fit experimental data. This can be done using bumps, which similar to sasmodels is a separate package and needs to be installed in your path.

```
In [36]: from sasmodels.core import load_model
from sasmodels.bumps_model import Model, Experiment
from sasmodels.data import load_data

from bumps.names import *
from bumps.fitters import fit
from bumps.formatatum import format_uncertainty

import pylab

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

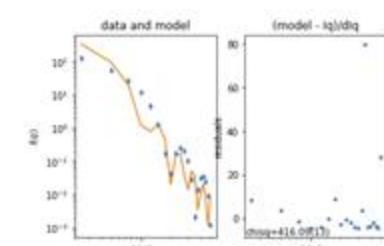
#We set some errors for demonstration
test_data.dy = 0.2*test_data.y

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

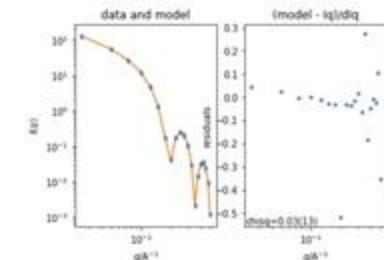
# SET THE FITTING PARAMETERS
model.radius.range(1, 50)
model.length.range(1, 500)

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, v, format_uncertainty(v, dv))
problem.plot()
pylab.show()
```



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



Resources maintained and 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)

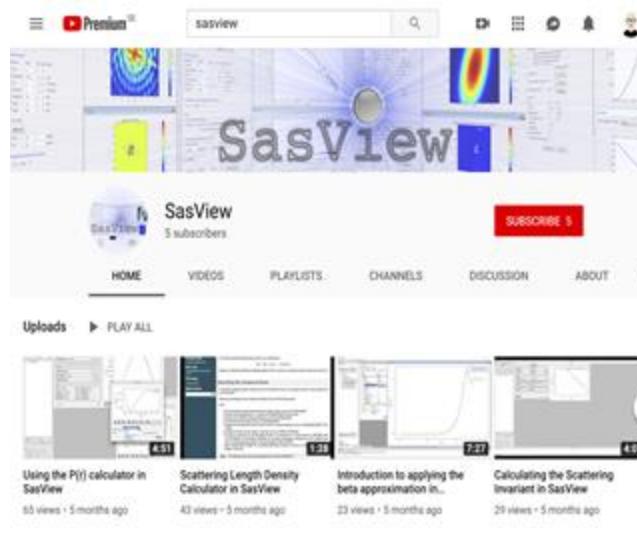
The screenshot shows a documentation page for SasView 4.1.2. The left sidebar contains a 'Table Of Contents' with sections like 'Loading Data', 'The data explorer', 'Loading data', and 'To load data, do one of the following:'. The main content area is titled 'Loading Data' and discusses the Data Explorer panel. It includes a note about loading data from the Data Explorer and how it interacts with other parts of the application.

The screenshot shows a Slack interface with a dark theme. The left sidebar lists channels: '#code_camp' (selected), '#general', '#github', '#jenkins', '#manuscript_chat', '#random', '#trac', and '#_training'. A message from user 'krzywon' at 2:26 PM states: 'I'm looking into the dataloader import issue. I changed associations.py to have a resolution today.' Another message from 'wpotrzebowski' at 4:40 PM discusses running SasView on OSX from GitHub Actions.

The top part of the screenshot shows the 'FAQ' page for SasView, which lists common questions about the software. The bottom part shows a GitHub repository page for 'mantidproject/mantid'. The page displays repository statistics, recent commits, and discussions. One commit by 'softwars.ac.uk' is highlighted, thanking Celine Gobin, Christopher Arnot, and John Wise for updating HEDMPODence.hedm.

Resources maintained and 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)



The image contains two side-by-side screenshots of the SasView software interface. The left screenshot shows a 'Fit Page' dialog box with several parameters listed, including 'Contract - I(B)' and 'M2 thickness'. The right screenshot shows a 'Constraints' dialog box where constraints like 'M2 thickness = M2 thickness' and 'M3 thickness = M3 thickness' are being set. Both screens show plots of scattering intensity versus scattering vector q .

A screenshot of the SasView website's 'Tutorials' section. The page has a dark header with the SasView logo and navigation links for 'ABOUT', 'LINKS & DOWNLOADS', 'CONTENT', and 'HELP'. Below the header, the word 'Tutorials:' is followed by a bulleted list of links:

- 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

Resources maintained and 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*

The screenshot shows a Moodle-based educational platform for neutrons. At the top, there's a navigation bar with links to FRONTPAGE, ABOUT E-NEUTRONS, FOR TEACHERS, and SUPPORT. On the right, there's a login form. Below the navigation, there are three sections: "Exercise taster", "Quiz taster", and "Simulation taster". Each section contains a brief description and a "READ MORE" button.

*Original put together by an SIS
Summer Student Lewis O'Driscoll*

The screenshot shows the SasView Marketplace interface. At the top, there's a search bar with a magnifying glass icon and a "Search" button. Below the search bar, it says "Categories / All Models". The main area is titled "All Models" and lists three entries:

Name	Description	Category	Upload Date	Author
correlated_spheres	Definition ----- The 1D scattering intensity of two correlated spherical particles can be written as: $S(q) = F_1^2 + F_2^2 + 2^2 F_1 F_2 \cdot \sin(qD)/qD$, 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

The RoadMap

SasView 5 Year Roadmap

The purpose of building and operating large scattering facilities is to provide unique tools to answer new scientific questions with the final presentation of results (usually in the form of a paper) as the output. The biggest obstacle to that output is often the analysis of the acquired data. Data analysis software has been variously viewed as being in the domain of the scientist using the facility, a service to be provided by scattering facilities, or as the individual responsibility of the scientists running the facility beamlines. The result has been a proliferation of packages and libraries, many written and supported by one key person, often not as their primary responsibility¹.

Over the past decade several trends have contributed to exacerbate the analysis bottleneck: 1) As the techniques have matured the user pool has broadened. This combined with an apparent decrease in the overall level of programming taught to scientists, means that fewer users are capable of building their own analysis tools. 2) With the increasing maturity of the field, a large amount of basic modeling is well understood and developed. Even those capable of coding their own should not be wasting their time re-inventing the wheel but focus on new science and perhaps new analysis developments to enable that new science. 3) The quantity of data being produced by instruments and the complexity of the experiments being performed have increased. 4) Finally, as the general software landscape has moved towards increased quality of usability and support, users of scattering facilities, in particular new users, have similar expectations of the software they use to operate the instrument and process and analyse their data.

Late 2018 to mid 2019 (from code camp VIII - ESS) - Release 4.2, Release 5.0

The focus in this period will be on development and release of version 5.0 of SasView. In parallel version 4.2 and possibly 4.3 will be released providing a maintained, stable, release for current users of SasView. This managed transition from the 4.x series to the 5.x series will allow for extensive user testing of the 5.0 version prior to release. We expect to continue maintenance of the final 4.x release beyond the release of 5.0, with an eventual end-of-life for 4.x occurring with the 5.2 release.

Full integration of the beta approximation work into 5.0 will be completed, with some limited beta approximation functionality being made available in 4.x.

The first SasView community meeting will be held at the SAS 2018 meeting in October 2018 providing SasView users and contributors with an introduction to the new functionality being made available in 5.0 and training on how to get involved in contributing to the SasView project. Building on this meeting a plan for expanding community interactions will be developed.

Release 4.2 and 5.0 will support separate plotting of the P(Q) and S(Q) components in a P*S analysis. Work will begin on integration of McSAS into SasView, primarily planning and design work.

The SasModels Marketplace will be updated to better support user needs and fix bugs in the deployment, including bringing the backend up to recent versions.

Living document

**Guides and tracks developers' work
Helps find candidate projects for funding.**

Discussed and updated at each Code Camp.

Hope to update after these discussions!!

Think about different Workflows and how to implement

Come and Join the Fun!

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 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
- ‘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

