

GSC Refactoring

Generic Scattering Calculator

SLD Data File

Nuclear Data No File Loaded

Magnetic Data No File Loaded

Shape

Input Parameters

Polarisation Settings

In Polarisation (fraction up)

Out Polarisation (fraction up)

Up Polarisation Direction, θ °

Up Polarisation Direction, φ °

SLD/Geometry Settings

Background cm^{-1}

Scale

Solvent SLD \AA^{-2}

Total volume \AA^3

Q Range

No. of Qx (Qy) bins

Qx (Qy) Max \AA^{-1}

Qx (Qy) Min \AA^{-1}

Log Spacing

Radius of Gyration

Rg - Mass

RG - SLD

Plugin Models

Export Model

Fixed orientation

SLD Pixel Info

No. of Pixels

Mean SLD

Mx \AA^{-2}

My \AA^{-2}

Mz \AA^{-2}

Nucl. \AA^{-2}

Nodes

x

y

z

Step Size

x \AA

y \AA

z \AA

Coordinate System Info

Environment Coordinates (uvw)

Yaw °

Pitch °

Roll °

Sample Coordinates (xyz)

Yaw °

Pitch °

Roll °

sample



x y z

environment



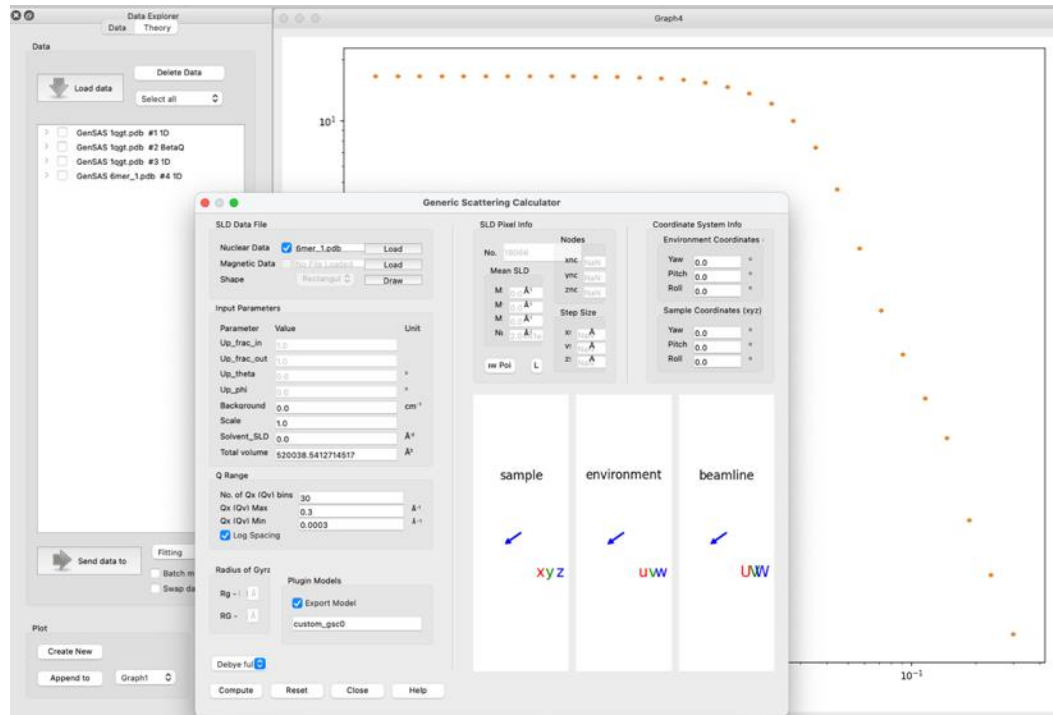
u v w

beamline



U V W

Generic Scattering Calculator

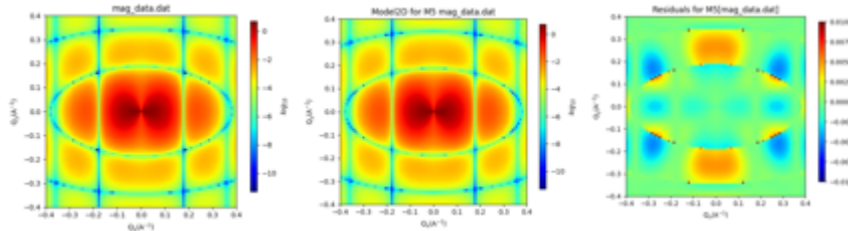


Magnetic/polarized SANS

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

Interacting bio-molecules

- Generate scattering profile from PDB file
- Save as a custom/plugin model
- Use for fitting with structure factors (including beta approximation)
- PDB reader and engine refactored



Workflow Summary Table

#	Workflow Name	Data Input	Mode	Output
1	Fixed-Nuclear	Nuclear only Grid/Elem	Mode 0	2D I(qx, qy)
2	Fixed-Magnetic	Magnetic only Grid/Elem	Mode 0	2D I(qx, qy) + Magnetic
3	Fixed-Both	Nuclear + Magnetic Grid/Elem	Mode 0	2D I(qx, qy) + Both contrib
4	Fixed-Default	No files Grid only	Mode 0	2D I(qx, qy)
5	Debye-Average	Nuclear only Grid only	Mode 1	1D I(Q) + Plugin*
6	Debye-Beta	Nuclear only Grid only	Mode 2	1D I(Q) + $\beta(Q)$ + Plugin*

Calculation engines

Calculation modes (cbOptionsCalc)

Mode	Index	is_avg	is_beta	Output	Available For
Fixed Orientation	0	False	False	2D I(qx, qy)	All data types
Debye Full Average	1	True	False	1D I(Q)	Nuclear only, grid type
Debye + $\beta(Q)$	2	True	True	1D I(Q) + $\beta(Q)$	Nuclear only, grid type

Constraint matrix

Data Input	Fixed Orientation	Debye Average	Debye + $\beta(Q)$
Nuclear Only	✓	✓*	✓*
Magnetic Only	✓	✗	✗
Nuclear + Magnetic	✓	✗	✗
Default Grid	✓	✓	✓

Shape2SAS



The screenshot displays the Shape2SAS (Experimental) web application interface. It is divided into several sections:

- Shape definition:** A table with 15 rows and 3 columns for defining the model components. Row 1 lists the shapes: ellipsoid, hollow_sphere, and cylinder. Rows 2-15 define various parameters such as ΔSLD , radii (a , b , c), center of mass (COM), rotation (RP), and orientation (α , β , γ).
- Shape visualization:** A 3D visualization of a green, porous, cube-like structure composed of small spheres. An arrow points to it with the label "Shape visualization".
- Structure factor:** A plot of $P(q)$ vs q on a log-log scale. An arrow points to it with the label "Structure factor".
- Scattering parameters:** A panel for adjusting simulation parameters. The "Structure factor" is set to "Hard Sphere" with a radius of 50.0. Other parameters include interface roughness (0.0), relative polydispersity (0.0), volume fraction (0.02), and relative exposure time (500).
- Scattering profile:** A plot of $I(q)$ vs q on a log-log scale, showing the simulated SAXS data for Model_1. An arrow points to it with the label "Scattering profile".
- Simulation parameters:** A panel for setting simulation details: q min (0.001), q max (0.5), number of points in $p(r)$ (100), number of simulated points (3000), and number of points in q (400). The model name is "Model_1".
- Profile saving:** Buttons for "Create GANGLI file", "SAXS", "Reset", "Close", and "Help".

Shape definition

Shape visualization

Structure factor

Scattering profile

Profile saving

Shape2SAS: a web application to simulate small-angle scattering data and pair distance distributions from user-defined shapes.

Andreas Haahr Larsen,^{a*} Emre Brookes,^b Martin Cramer Pedersen,^c and Jacob Judas Kain Kirkensgaard^{c,d}

Features beyond 6.0.0 - particle editor



Define functions
sld and (optionally)
magnetism

The screenshot shows the Particle Editor interface with the following components:

- Code Editor:** Contains Python code for defining SLD functions. The code includes a docstring, a description of the SLD function, and a simple example of a cube with a 100 Angstrom side length.
- Log Panel:** Shows the output of the build process, indicating a successful build at 2023-05-16 02:43:28.
- Visualization Controls:** Includes buttons for 'Build' and 'Scatter', a 'View Radius' slider set to 100.00 Å, and radio buttons for 'SLD' and 'Magnetism'. Below these are sliders for 'Depth' (0%), 'B Field (display)' (0°), and 'θ' (0°), along with 'φ' (0°) and 'X', 'Y', 'Z' view buttons.
- Visualizations:** Two 3D plots are shown: the top one is a total projected density, and the bottom one is a slice of the density.

sld / magnetism
“x-ray” projection

sld / magnetism
cross section

Magnetic field
controls for display

“Recompile” and
update display

Compute
scattering

Feedback on
code and
calculations

GSC on GitHub

- [#3676 — Resolve weakness in the Generic SANS Calculator tool](#)
- [Discussion #2351 — Weakness in the Generic SANS Calculator tool](#)
- [#1847 — Weakness in the Generic SANS Calculator tool](#)
- [#1400 — Generic scattering calculation is incorrect for orientation-averaged asymmetric particles](#)
- [#596 — Generic Scattering Calculator Enhancements: make a theory curve](#)
- [#597 — Generic Scattering Calculator Enhancements: integrate into fitting](#)
- [#598 — Generic Scattering Calculator Enhancements: accept MD trajectory files](#)
- [#2011 — Extend generic scattering calculator with tabs for other simulators](#)
- [#2552 — Generic Scattering Calculator GUI needs redesign](#)
- [#2734 — GSC not drawing the structure](#)
- [#2798 — Saving Project with the general scattering calculator throws an error](#)
- [#2537 — Log spacing for Generic Scattering Calculator](#)
- [#1886 — “Save SLD Data” saves rows/columns the wrong way around](#)

A few options moving forward

- Refactor the current interface
- Move out from Tool to Perspective
- Send any data to any Tool or Perspective

Potential next step

- Identify the working group
- Develop strategy/roadmap moving forward
- Engage Users for early feedback
- Implementation loop (including users' feedback)

GSC Refactoring

Part II

Generic Scattering Calculator

SLD Data File

Nuclear Data No File Loaded

Magnetic Data No File Loaded

Shape

Input Parameters

Polarisation Settings

In Polarisation (fraction up)

Out Polarisation (fraction up)

Up Polarisation Direction, θ °

Up Polarisation Direction, φ °

SLD/Geometry Settings

Background cm⁻¹

Scale

Solvent SLD Å⁻²

Total volume Å³

Q Range

No. of Qx (Qy) bins

Qx (Qy) Max Å⁻¹

Qx (Qy) Min Å⁻¹

Log Spacing

Radius of Gyration

Rg - Mass

RG - SLD

Plugin Models

Export Model

Fixed orientation

SLD Pixel Info

No. of Pixels

Mean SLD

Mx Å⁻²

My Å⁻²

Mz Å⁻²

Nucl. Å⁻²

Nodes

x

y

z

Step Size

x Å

y Å

z Å

Coordinate System Info

Environment Coordinates (uvw)

Yaw °

Pitch °

Roll °

Sample Coordinates (xyz)

Yaw °

Pitch °

Roll °

sample



x y z

environment



u v w

beamline



U V W

Conclusion from the previous meeting

We will try to look at:

- 1.Entry layer (molecule type, magnetic data, volume/normalization)
- 2.Modality (magnetic soft matter, bioSAS, etc.)
- 3.Calculation engines (pluggable backends)

What is the target group?

What information/functionality are they looking for?

Why do they want to find this information?

How do we know it?

What is the target group?

PhD Students, Postdocs, Pls
Instrument (Data) Scientists
Industrial Users?

What information/functionality are they looking for?

How the scattering pattern of my system look like?

Why do they want to find this information?

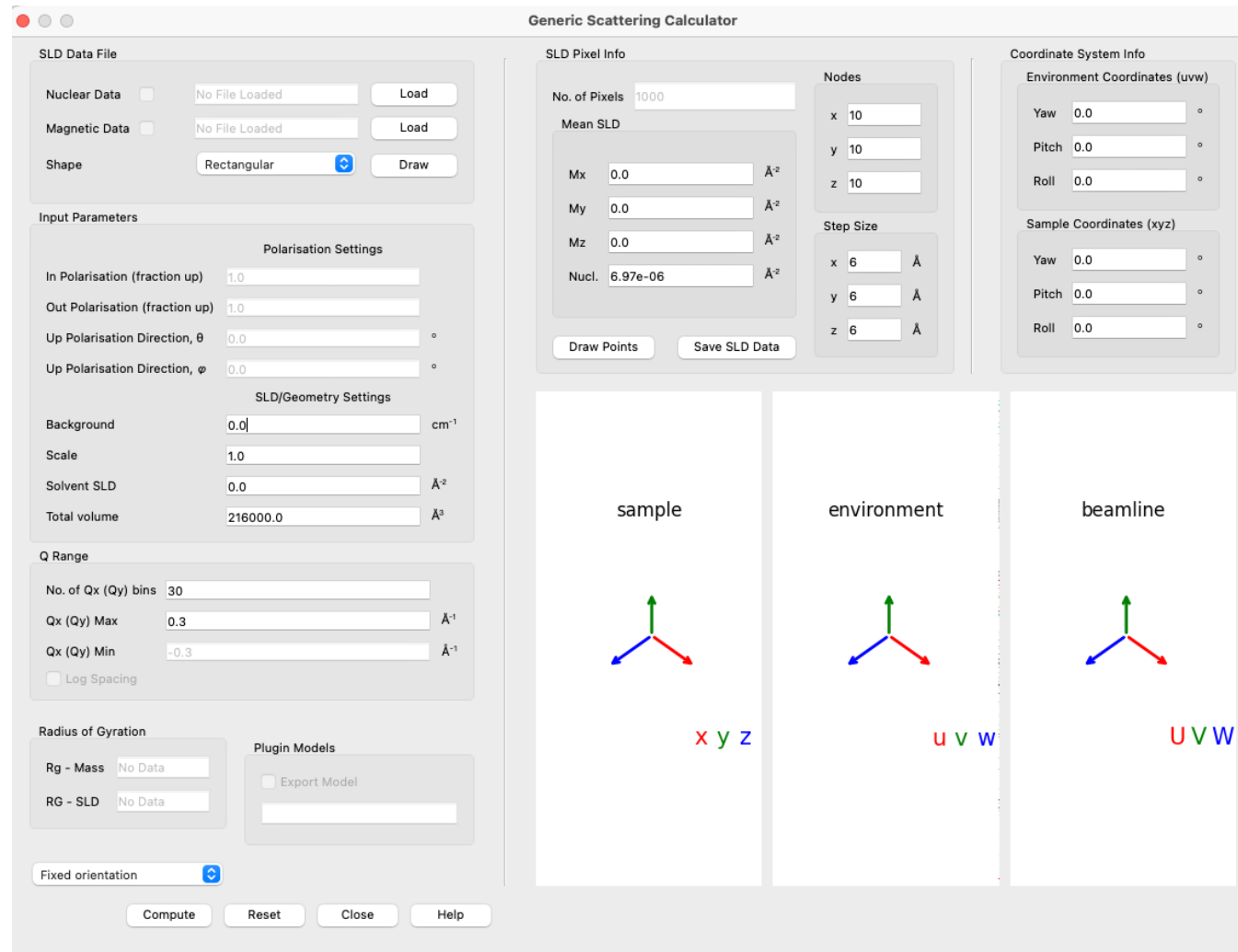
To explain their data
To better plan the experiment

How do we know it?

Because we used it this way
Because we heard from the users

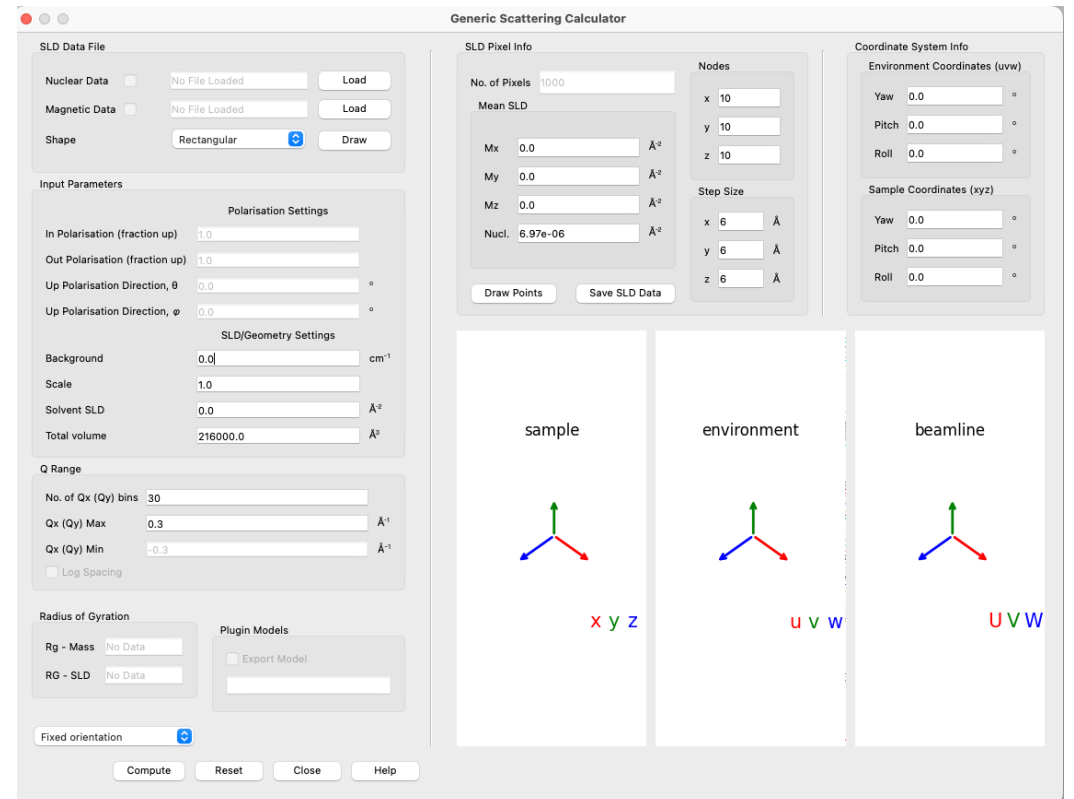
Map the current UI to layers

- 1.Entry layer (molecule type, magnetic data, volume/normalization)
- 2.Modality (magnetic soft matter, bioSAS, etc.)
- 3.Calculation engines (pluggable backends)



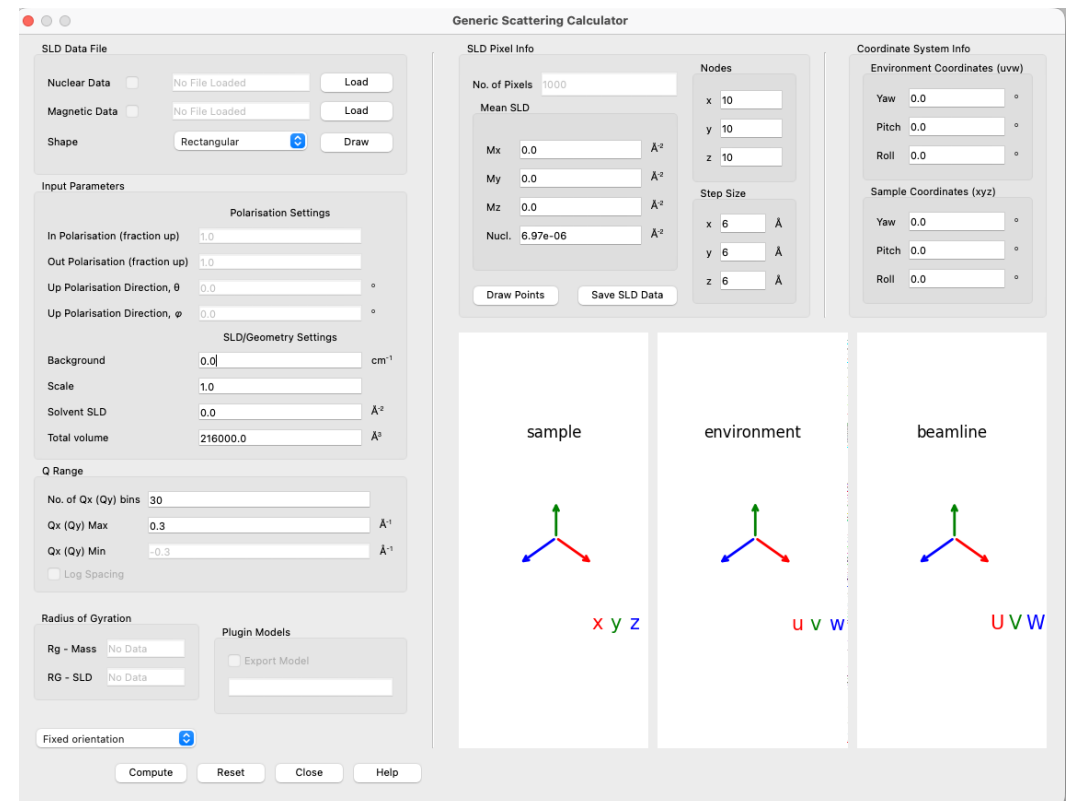
Entry Layer

- What are the mandatory fields to compute anything?
- What fields are optional but should be strongly recommended?
- What should be inferred from files vs explicitly set by user?



Modality

- What outputs does each modality need ($I(q)$, $I(q_x, q_y)$, 2D/1D)?
- Which controls are common across all modalities (q-range, scale/background?) and which are modality-only (polarization, spin channels)?



Calculation Engines

- What is the smallest common denominator engine interface?
- Which options belong to the engine vs modality vs entry?
- How do we represent engine capabilities so UI can enable/disable controls?

