

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 responsibility ^{^1}.

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.

To enable the production and maintenance of software that meets users expectations, a greater level of resources needs to be applied to the problem. This has been recognised in the neutron scattering community through projects such as DANSE ^{^2}, Mantid ^{^3}, CCPSAS^{^4} and projects within the European Union Horizon 2020 programme such as SINE2020^{^5}. While each facility or scientist may not be able to commit the necessary resources, pooling of expertise between facilities and with the user community into a single project can amplify the effort and provide gains in quality, functionality, longevity and supportability with obvious benefits to both the facilities and the users.

The aim of the SasView project is to provide open source, *community and collaboratively developed software for the analysis of small angle scattering data*

The community collaborative development model is designed to encourage and enable contribution from a range of experts in small angle scattering, computer science and software development. Users of the software are encouraged to participate through bug reports, submitting code and by integrating their own scattering models as well as by joining the team which does not require any specific commitment. See the SasView website at <https://www.sasview.org/contribute/> about how to get involved with the SasView community.

This is a regularly updated document, reviewed at each Code Camp. Document history is maintained and previous iterations can be found in the SasView GitHub repository^{^6}

Development Model

The SasView project is co-ordinated by a small management team who are also members of the broader development team. The active development team currently consists mostly of scientists and software engineers from scattering facilities around the world. Progress is monitored and project direction discussed at fortnightly video conferences. The project aims for a major release each year, with minor bug fix releases as required in between.

The source code for SasView is hosted on GitHub, allowing anyone to fork the code base, add functionality or fix bugs, and easily request that the code be merged into the project. The use of unit tests and code quality metrics is employed to help manage the contribution of code from a varied and dispersed developer base.

As of 2011, few if any of the developers have SasView as a major component of their job assignments. Given the nature of software development this currently means most development occurs during annual code camps. While this has proven to be a highly successful approach, it limits the rate of progress. The Roadmap presented below is intended to give guidance to the development team and to our stakeholders as to the direction we aim to take over the next 5 years. It is an optimistic one predicated on more resources becoming available, either through new developers joining the project or work on SasView becoming a greater part of the day-to-day assignment for the current developers. It also should be used as a source of ideas for projects that could be undertaken with short-term resources or to meet specific requirements of stakeholders if they have the necessary resources available.

Roadmap

Every release should include bug fixes and new models as requests come in. These are assumed with each release and not included specifically below. Likewise general robustness and ease of use issues will be addressed in each release cycle.

Late 2018 to late 2021 - SasView 5

Due to an oversight the last update to the Roadmap failed to record a number of SasView releases which we correct here:

- Oct 2018: <https://github.com/SasView/sasview/releases/tag/v4.2.0>
- Feb 2019: <https://github.com/SasView/sasview/releases/tag/v4.2.1>
- May 2019: <https://github.com/SasView/sasview/releases/tag/v4.2.2>
- May 2019: <https://github.com/SasView/sasview/releases/tag/v5.0.0>
- Feb 2020: <https://github.com/SasView/sasview/releases/tag/v5.0.1>
- Apr 2020: <https://github.com/SasView/sasview/releases/tag/v5.0.2>
- Jul 2020: <https://github.com/SasView/sasview/releases/tag/v5.0.3>
- Apr 2021: <https://github.com/SasView/sasview/releases/tag/v5.0.4>

Most notable here was the move from Python2/wxPython in SasView version 4.x to Python3/pyQt in SasView version 5.x.

Late 2021 to mid 2022 - Release 5.0.5

Due to the Covid pandemic the Code Camp scheduled for March 2020 in Caltech was cancelled and another was not scheduled for another 2+ years. With the developer community isolated, an attempt was

made to pivot to short virtual Hackathons with some success, but progress was clearly impacted. Nonetheless, release 5.0.5 provided some new features and resolved a number of issues. In particular, the generic scattering calculator got its first much needed overhaul, this one focused on the needs of the magnetic scattering community. Work will now begin on refactoring the simultaneous/constrained fitting workflow interface.

Task Summary:

- A sasmodels interface for custom re-parameterization of models is provided
- Continued work on refactoring constrained/simultaneous fits
- The generic scattering calculator was significantly overhauled particularly for use with magnetic materials
- Started developing ideas for community interactions and hosted several training workshops (some virtual)
- Work on incorporating polyhedral form factors continued
- Work on graphical representation of models continued
- Work on defining and incorporating generic resolution functions was started
- Work to update tutorials to support 5.x continued
- Work on the SasView paper continued
- Model marketplace was upgraded
- Begin work on simplifying dependencies/requirements
- As usual, a large number of bug fixes and other minor improvements were included

SasView 5.0.5 was released on June 3, 2022.

- <https://github.com/SasView/sasview/releases/tag/v5.0.5>

Late 2022 - Mid 2023 - Release 5.1

The primary goal, and therefore focus of work, for release 5.1 is to achieve full feature parity with the 4.x versions. While 5.x has many new features and advances there remain a number of outstanding issues tagged with the "For Feature Parity" label in Github such as the orientation viewer. Subject to the availability of sufficient resources, release 5.1 will also begin work on a number of much needed infrastructure refactoring work such as pulling out the data loader, cleaning up the slicer code base while adding features, refactoring the plotting design, and separating out the GUI package from the calculation package; add new scattering calculators to the generic scattering calculator; and add improved functionality to the P(R) perspective. Work will continue on an advanced data fitting with SasView tutorial. On the documentation and community development fronts, work will continue to expand and diversify the contributor community, several new tutorials, both written and video, will be started, and the SasView paper will be submitted.

Task Summary (Subject to the availability of sufficient resources):

- Finish overhaul of the Corfunc module
- Finish work on (some rudimentary) tools for SAXS/SANS co-refinement
- Finish initial simultaneous/constrained workflow upgrades
- Finish refactoring of slicers
- Begin work on advanced model fitting tutorial
- Begin work on writing custom models tutorial

- Create first in series of video tutorials
- Begin work on tutorials (written and video) for scripting
- Complete work on separating out a generic dataloader package (including multiple data type support e.g. multiple detectors)
- Complete SasView paper
- Continue work to expand the size and diversity of the contributor community
- Begin adding other algorithms to the generic scattering calculator such as FFT-based, SPONGE, McSim, Golden Vector approach etc., both isotropic and oriented systems
- Begin work on including the beta approximation calculation into generic scattering calculator
- Add tabbed and batch functionality to P(R) inversion to match fitting perspective functionality
- Reimplement the orientation viewer
- Continue discussions on how to address multimodal fitting
- Continue work on developing more advanced resolution calculations in co-operation with NXcanSAS group (e.g. generic resolution functions etc)
- Refactor the way plotting is tied to data and fitting
- Refactor reporting and exporting of perspective data
- Refactoring of project/analysis Saving and Loading
- Begin work on separating out sascalc package
- Begin work improving magnetic SANS workflows
- Begin adding support for handling multiple scattering calculation
- Begin work on external databases integration for final results deposition
- Reach out to BNL about maintaining conda forge as the official Linux distribution for facilities.
- Create a user forum (like reddit or stackoverflow for community discussion)
- Complete work on simplifying dependencies/requirements
- Add the usual bug fixes and other minor improvements as time and interest permit

Late 2023 - Mid 2024 - Release 5.2

Subject to the availability of sufficient resources, release 5.2 will focus on adding major new functionality along with the usual bug fixes and minor improvements. Work should begin on new fitting functionality such as allowing for the reading in of an array representing either $P(Q)$ or $S(Q)$ for P^*S and fitting oriented models to 1D slices, etc.; adding PRISM⁹ (polymer reference interaction site model), McSAS and FFSAS (providing users with better approaches for obtaining particle size distributions) fitting modules. New real-space to Fourier-space calculators may continue to be added to the general scattering calculator and some new tools for multimodal analysis may begin to show up in this release as well. Final separation of the GUI package from the calculation package should be completed with this release. Use cases and design development will commence on a web interface (possibly including smartphone app capabilities). Meanwhile, tutorial and community development efforts will continue while an effort will be made to develop sustainable plans for documentation maintenance and development writ large.

Task Summary (Subject to the availability of sufficient resources):

- Begin model fitting refactoring work to allow reading in an array representing either $P(Q)$ or $S(Q)$ for P^*S fits, fitting oriented models to 1D cuts, etc.
- Complete work on separating out sascalc package (including deployment on pypi)
- Finish work to integrate McSAS and FFSAS

- Begin work to refactor/improve P(R) inversion by adding full automation of parameter choices, 2D as well as 1D P(R), and/or other algorithms
- Begin work on architecture manual
- Finish work on advanced model fitting tutorial
- Finish work on writing custom models tutorial
- Begin work on inclusion of PRISM⁹ functionality
- Begin work to integrate McSAS and FFSAS
- Begin work on writing scripting tutorial
- Create at least one more tutorial as appropriate
- Continue work to expand the size and diversity of the contributor community
- Develop training workshop strategy
- Create more video tutorials.
- Begin use case and design on Web interface (with possible smartphone app feature); initial version can have minimal features but would be useful for demos?
- Continue adding other algorithms to the generic scattering calculator such as FFT-based, SPONGE, McSim, Golden Vector approach etc.
- Start implementing further tools/algorithms (e.g. different cost functions) to aid multimodal fitting as appropriate
- Continue work on developing more advanced resolution calculations in co-operation with NXcanSAS group (e.g. generic resolution functions etc)
- Finish multiple scattering calculator implementation
- Develop documentation maintenance and development strategy
- Add the usual bug fixes and other minor improvements as time and interest permit
- Extend available integration options such as adaptive, MC integration
- P * S computation enhancements (adding locally monodisperse approx.; fix volume fraction issue, etc.)
- Finish work on simplifying dependencies/requirements
- Finish work on incorporating beta approximation calculation into generic scattering calculator

Late 2024 - Mid 2025 - Release 5.3 / Release 6.0 alpha/beta

Subject to the availability of sufficient resources, release 5.3 will focus on wrapping up some new features and preparing for more distributed operation. In particular, the new fitting functionality such as allowing for the reading in of an array representing either P(Q) or S(Q) for P*S, fitting oriented models to 1D slices, PRISM, etc. should be ready for this release along with some more advanced resolution calculation options, while continuing work on a web app, allowing computational code to run on a cluster with an intelligent launcher/scheduler for the GUI frontend which will make the use of the a cluster backend transparent to the user, data pipelining etc as well starting work on providing intelligent feedback. However, while the web UI and remote job operation work will continue, is not expected to be ready for this release. Meanwhile, tutorial and community development efforts will continue and documentation tasks as envisioned in the documentation strategy exercises will be worked on. along with work on some infrastructure projects such as providing more robust parallel processing support.

Task Summary (Subject to the availability of sufficient resources):

- Finish model fitting refactoring work to allow reading in an array representing either P(Q) or S(Q) for P*S fits, fitting oriented models to 1D cuts, etc.

- Enable computational code to run on clusters and refactor GUI to add an intelligent launcher/scheduler that makes the use of a cluster back end transparent to the user
- Begin work on implementing headless operation/realtime analysis workflows
- Finish work to refactor/improve P(R) inversion by adding full automation of parameter choices, 2D as well as 1D P(R), and/or other algorithms
- Finish work on architecture manual
- Finish work on inclusion of PRISM⁹ functionality
- Expand parallel processing support for multi-GPU, multi-CPU and CPU/GPU computation
- Create at least one more tutorial as appropriate
- Continue work to expand the size and diversity of the contributor community
- Continue developing training workshop strategy and begin implementation
- Create at least one more video tutorials
- Continue implementing further tools/algorithms to aid multimodal fitting as appropriate
- Start including intelligent limits/help (possibly include switch between enforcement and warning only) and explore the use of wizards and AI
- Continue work on web UI and/or smartphone app
- Complete implementation of first advanced resolution calculations subject to sufficient community agreement
- Begin work on generic O-Z solver
- Documentation maintenance and development
- Add proper Qz support to SasView analysis
- Finish implementation on tools/algorithms for multimodal fitting
- Begin work on full history stack (Redo/Undo functionality)
- Add the usual bug fixes and other minor improvements as time and interest permit

Late 2025 - Mid 2026 - Release 6.0

Subject to the availability of sufficient resources, release 6.0 will allow running compute intensive portions of SasView computation on a cluster back end with a transparent access from the desktop GUI. It will also allow deployment as a webservice with a web based front end which will have limited functionality in this first instance. Integration of SasView into a realtime analysis workflow on actual beamlines should now be possible. The use of wizards and intelligent user guidance will be expanded and new workflows/interfaces may be added as appropriate. Meanwhile, tutorial and community development efforts will continue and documentation tasks as envisioned in the documentation strategy exercises will be worked on.

Task Summary (Subject to the availability of sufficient resources):

- Deploy computational code on clusters
- Complete headless operation and realtime analysis workflows
- Deploy Web application
- Finish work on O-Z solver
- Continue work on smartphone app
- Create at least one more tutorial as appropriate
- Continue work to expand the size and diversity of the contributor community
- Create at least one more video tutorials
- Expand use of wizards and intelligent user guidance

- Add new workflow interfaces as appropriate (e.g. contrast match point, nuclear/magnetic separation, etc)
- Documentation maintenance and development
- Finish work on full history stack (Redo/Undo functionality)
- Finish work on external database integration
- Add the usual bug fixes and other minor improvements as time and interest permit

Late 2026 - Mid 2027 - Release 6.x

Subject to the availability of sufficient resources, release 6.x will try place an emphasis on addressing requests for smaller feature enhancements and improvements to the interface and workflow. It will also continue to expand on intelligent guidance and include more functionality on web app and see the deployment of a smartphone app. SAXS specific workflows will be included. Work on a smartphone app interface to the webservice will continue but likely will not be ready for this release. Meanwhile, tutorial and community development efforts will continue and documentation tasks as envisioned in the documentation strategy exercises will be worked on.

- Begin UI usability review and testing. Consider Full external review such as from the Software Sustainability Institute or NSF equivalent institute etc. This should also include cyber security review. End product should include design guidelines for the project.
- Continue to expand use of wizards and intelligent user guidance
- Continue work on smartphone app
- Create at least one more tutorial as appropriate
- Continue work to expand the size and diversity of the contributor community
- Create at least one more video tutorials
- Expanded functionality of web app
- Add new workflow interfaces as appropriate (e.g. contrast match point, nuclear/magnetic separation, etc)
- Documentation maintenance and development
- Add the usual bug fixes and other minor improvements as time and interest permit

Revision History

- 2015-11-24 : First release
- 2016-10-11 : Updated after Code Camp V discussions
- 2018-09-07 : Updated after Code Camp VI & VII discussions
- 2024-02-18 : Updated after Code Camp X discussion (November 2022)