

## What's new in GSAS-II

Brian H. Toby,<sup>a)</sup> and Robert B. Von Dreele

*Advanced Photon Source, Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, Illinois 60439-4814*

(Received 2 June 2014; accepted 22 July 2014)

The General Structure and Analysis Software II (GSAS-II) package is an all-new crystallographic analysis package written to replace and extend the capabilities of the universal and widely used GSAS and EXPGUI packages. GSAS-II was described in a 2013 article, but considerable work has been completed since then. This paper describes the advances, which include: rigid body fitting and structure solution modules; improved treatment for parametric refinements and equation of state fitting; and small-angle scattering data reduction and analysis. GSAS-II offers versatile and extensible modules for import and export of data and results. Capabilities are provided for users to select any version of the code. Code documentation has reached 150 pages and 17 web-tutorials are offered.

© 2014 International Centre for Diffraction Data. [doi:10.1017/S0885715614000736]

Key words: Powder diffraction, Crystallographic analysis, Rietveld analysis, Software

### I. INTRODUCTION

When Hugo Rietveld first developed his method for fitting powder diffraction data, it was implemented in the Algol computer language (Rietveld, 1969). He shortly after rewrote that code in Fortran, which greatly facilitated its use and distribution. That Fortran code was passed from laboratory to laboratory, acquiring additional capabilities and modifications and was incorporated in many much later Rietveld refinement codes. One of the first Rietveld codes to be written completely from scratch was included in the General Structure Analysis Software (GSAS) package, which was first distributed in the early 1980s (Larson and Von Dreele, 2004). GSAS was certainly the first package that combined Rietveld analysis with other crystallographic analysis capabilities, such as fitting single-crystal data. It also provided the ability to fit to all types of diffraction data, including constant-wavelength (CW) and time-of-flight (TOF) neutron diffraction data as well as both laboratory and synchrotron powder diffraction data. GSAS pioneered many advances in the fitting of powder diffraction data, such as the combined use of multiple diffraction datasets and offered a comprehensive set of post-refinement tools.

GSAS has seen wide adoption in the crystallographic community. Use can be simplified, particularly for novice users, when combined with the EXPGUI graphical-user-interface front end (Toby, 2001). The combination has also become a favorite platform for teaching Rietveld refinement. However, despite the successes of the GSAS/EXPGUI combination, it became clear many years ago that there are many reasons to explore creation of a new crystallographic package (Toby and Von Dreele, 2007). GSAS was developed within the memory limitations of computers readily available several decades ago; this now limits the ability for GSAS to handle the more complex analyses that researchers now wish. The continued support of GSAS and EXPGUI are not guaranteed. While GSAS is written in Fortran, which is presumed to be quite

portable, in truth the experience shows that changes in compilers and platforms require a revalidation of the results from the code and often an extensive debugging endeavor to track down the compiler-sensitive sections of code. EXPGUI depends on the Tcl/Tk script interpreter and the unsupported BLT computational and plotting package. It has been very difficult to support this on newer operating systems. Furthermore, the target audience for the Rietveld analysis has evolved over these many years. Powder diffraction crystallographic analysis has migrated from a technique utilized by a small number of experts to one widely used in many fields for many purposes. Software needs to adapt to this, since one cannot assume the knowledge base of an expert.

For these reasons, the GSAS-II project was initiated to develop a new and modern crystallographic analysis code that expanded the scope of the original package to also include pre-refinement analysis steps such as image integration, pattern indexing, and structure solution. The GSAS-II package was previously described in the literature (Toby and Von Dreele, 2013). However, this package was not complete at the time when that was written, but it is fair to say that no actively utilized scientific package is ever completed.

GSAS-II is a graphically oriented program, both with respect to how users interact with the program, and with respect to interpretation of input and derived constants, as shown in Figure 1. GSAS-II is intended to perform all commonly needed crystallographic computations for materials crystallography as well as many needed for small-molecule and protein analysis. This includes structure solution, structure refinement, Fourier maps computation, visualization, and distance-angle computation. For powder diffraction, area detector data reduction and powder diffraction indexing routines are provided. The program will fit both single-crystal and powder data from constant wavelength X-ray and neutron sources, including laboratory instruments with both  $K\alpha_1$  and  $K\alpha_2$  radiation present. It is expected that the implementation for TOF powder and single-crystal refinement will be completed in the near future.

It should be noted that GSAS-II is being developed in an agile fashion where new functionality is prototyped and then

<sup>a)</sup> Author to whom correspondence should be addressed. Electronic mail: [toby@anl.gov](mailto:toby@anl.gov)

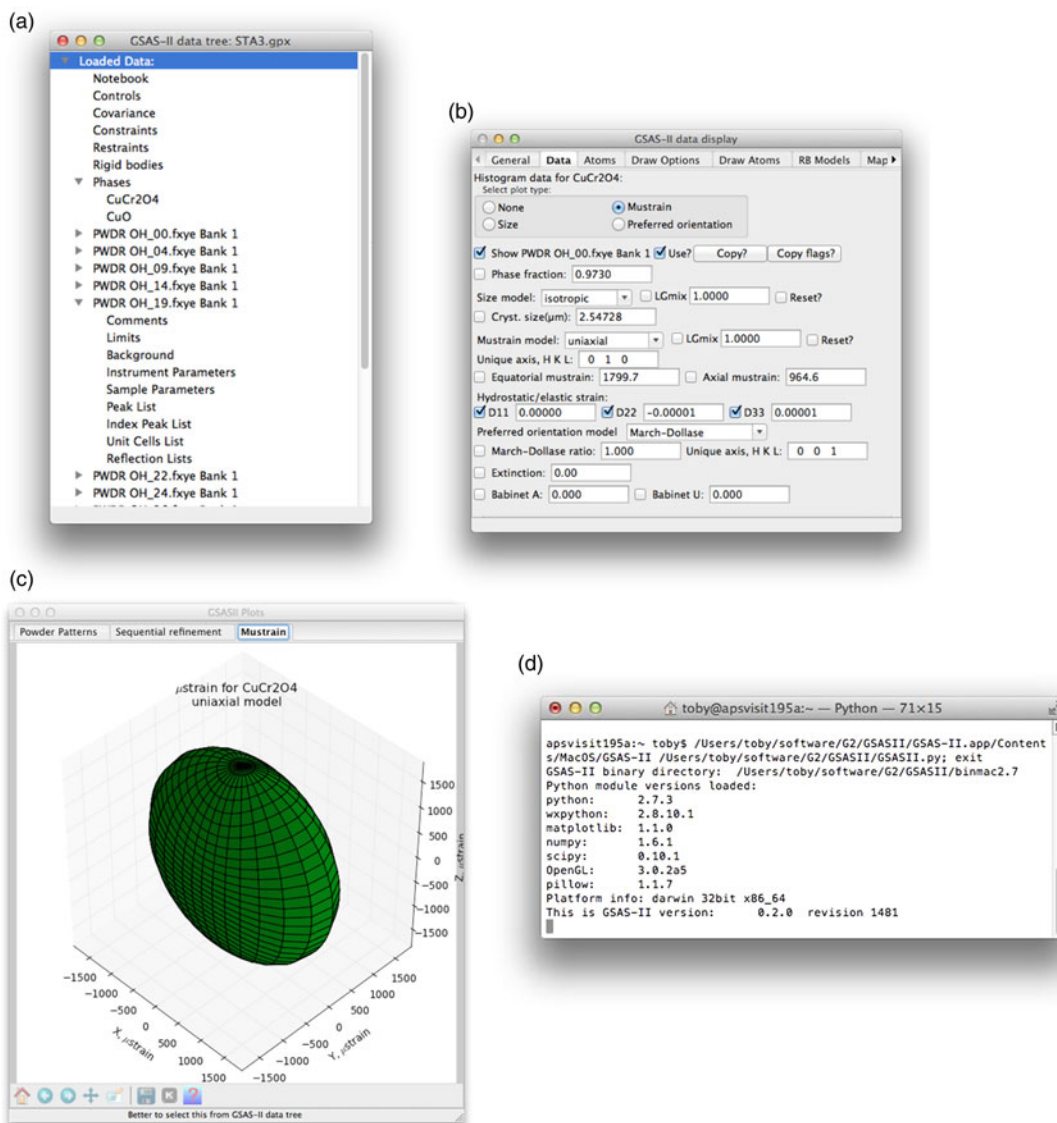


Figure 1. The GSAS-II user interface. GSAS-II runs with four open windows. In the case shown here, two-phase sequential fits have been made to 17 measurements made as a function of temperature. The main window (a) shows the contents of the project file as a hierarchical “tree,” which includes the parameters associated with the overall refinement, the 17 datasets and two phases. The “data display” window (b), responds to selection of a tree item in the main window and shows its contents. The third, the graphics window (c), plots results from the data display window, which in this example shows the anisotropic microstrain (residual stress) coefficients as a 3D surface. The fourth window (d) is a console (terminal) window that shows status and error messages, but is not used for any input. Note that main and data display windows have associated menu bars. For Windows and Linux, these are displayed on the windows, but for the Mac, they are displayed at the system menubar location.

improved. The chosen language, Python, is particularly suited for this style of code development as well as capable of covering all aspects of the project. Code is added to the package on an on-going basis. In the ~18 months since the initial paper was written, there have been nearly 600 sets of changes to the GSAS-II codebase. Some of these changes have made mere minor spelling corrections and others represent bug fixes, but still others incorporate completely new capabilities. The goal of this paper is to outline what features have been added to GSAS-II since the previous paper was completed.

## II. RIGID BODY REFINEMENT AND SIMULATED ANNEALING

The significant strength of the parent GSAS package was its pioneering flexible treatment of molecular fragments as rigid units for all scales of crystallographic refinements. This

has been implemented in GSAS-II for refinements and it has been extended to structure solution. Rigid bodies are valuable in refinements where less complete data are available than what might be desired, which perhaps includes most powder diffraction analysis. Rigid bodies describe the relative arrangement of a group of atoms, but allow the orientation and location of that group of atoms to be refined. This will reduce the number of positional parameters needed to describe that group of atoms to a maximum of six and is ideal to use in refinements for materials with moieties that have very well-understood configurations, such as a perchlorate anion or a phenyl ring.

A module has also been placed into GSAS-II for solution of structures using a simulated annealing Monte Carlo fitting. In this method, atoms or molecular fragments are placed in the asymmetric unit with random locations and orientations and are then moved to improve the fit or probabilistically to sometimes degrade the fit to escape from local minima (Dmowski

*et al.*, 1988; Favre-Nicolin and Cerny, 2002; Stephens and Huq, 2002; David *et al.*, 2006). Since the goal in structure solution is to find an approximate and possibly incomplete structural description of the material, it can be appropriate to solve the structure with idealized rigid body units, such as octahedral  $MO_6$  units, which will be modeled more accurately once the approximate structure is determined. For this process in GSAS-II, reflection intensities are estimated using Pawley refinement (Pawley, 1981). These intensities as well as their covariance terms are utilized in the structure solution, thus treating intensity partition for overlapped reflections as well as offering the improved speed in comparison to a full profile fit (David *et al.*, 2006).

### III. STATE VARIABLES, SEQUENTIAL REFINEMENT, AND PARAMETRIC FITTING

It is becoming increasingly common for researchers to collect large numbers of diffraction datasets as a function of one or more system variables, e.g. physical parameters such as temperature, chemical potential, etc. Every histogram (dataset) in GSAS-II may be labeled with five different system variables. Two of these are temperature and pressure. The other three defaults to values for humidity, applied voltage, and applied stress, but these latter three parameters can be relabeled for a project to anything that might be varied for measurement during that set of experiments.

The sequential refinement capability described in the previous GSAS-II publication has been extended for these types of analyses. In this process, an unlimited number of datasets can be read into a project and then at key points in the

sequence, where there are phase changes or other changes in parameterization, manual refinements are performed. Once that is done, the sequential refinement process is started, and refinements using the remaining datasets are performed using each manually determined fit as a starting point for a group of similar datasets. The results from each successive fit can “cascade”, where the results from one refinement are used for the next, or all refinements are initiated from exactly the same parameters, or the user can designate the specific starting parameters for each fit. In GSAS-II, sequential refinements can be continued, optionally with addition of extra variables once the fit is close to a minimum, but the parameterization must be the same for each group of fits performed together. Figure 2 shows how the results from a sequential refinement may be presented.

A parametric fitting module has also been added. This is used after a sequential fit and has been performed to fit an equation of state to a refined parameter or a quantity computed from refined parameters (pseudo-variable). This equation is

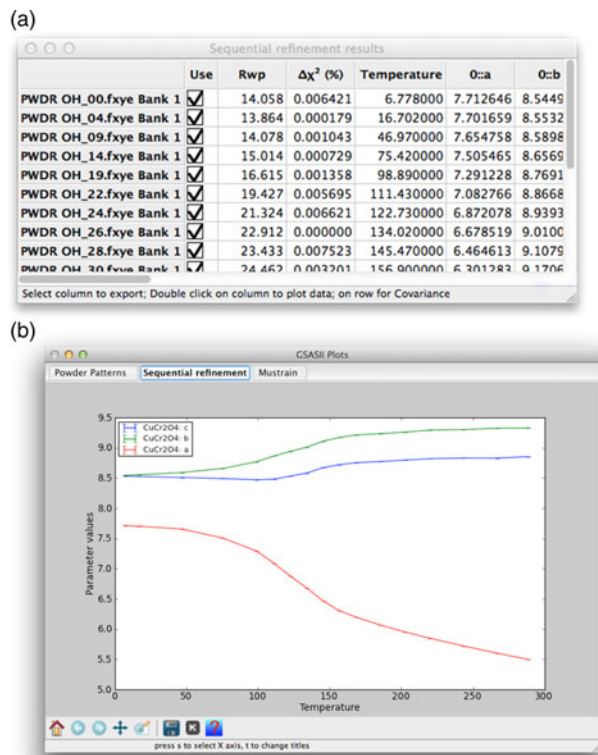


Figure 2. The data display (a) and graphics window (b) showing output from a sequential fit of multiple datasets, as demonstrated in the sequential refinement tutorial distributed in GSAS-II.

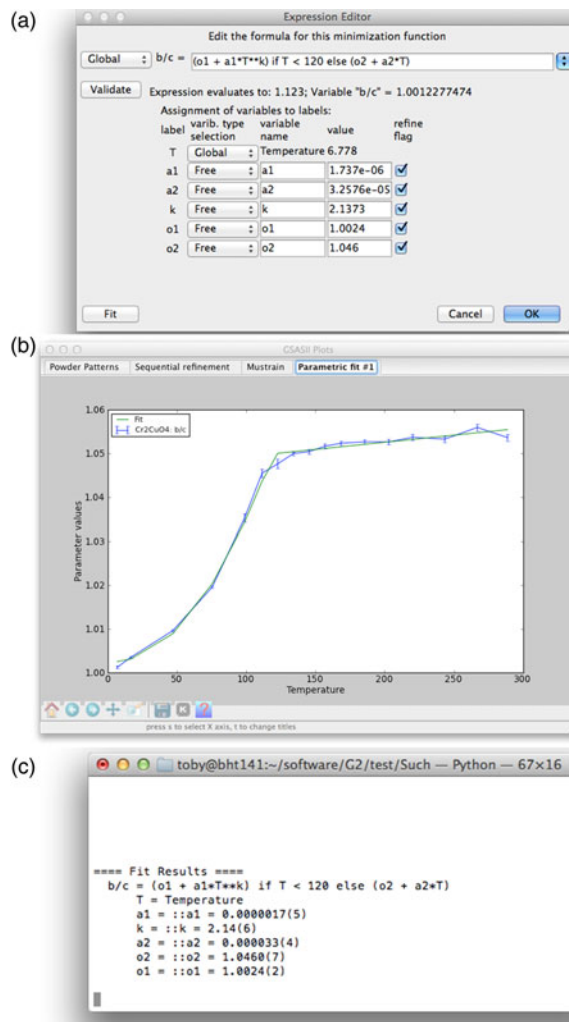


Figure 3. Plotting and fitting of the ratio of two lattice constants from the sequential refinement as a tutorial. The user-supplied equations  $o_1 + a_1 T^k$  and  $o_2 + a_2 T$  are used to fit different parts of the data. The equation and fit parameters are shown in the parametric fitting window (a) and plotted in the graphics window (b). Note that the uncertainties shown for the b/c ratio are computed directly from the covariance matrix at each temperature. Uncertainties for the fitted parameters are shown in the console window (c).



supplied by the user as a line of Python code and it may be non-linear and may even contain the Python “if” operator or reference user-supplied functions. The user designates which variables are free parameters and should be fit, as is shown in Figure 3. Through user-supplied functions or use of the “if” operator, different equations can be applied to different subsets of the datasets. Pseudo-variables are sets of values computed from the user-supplied function of the fit parameters. These functions are also supplied by the user, also in Python and may also be non-linear. The uncertainties for pseudo-variables are computed incorporating parameter covariance. This module thus allows for parametric fitting with proper statistical treatment, but also does so in a manner that makes it clear when the designated function is not a good fit to the equation of state, which does not happen when an equation is built into the original model as a constraint.

#### IV. IMPORT AND EXPORT MODULES

Based on the experience gained with previous codes, GSAS-II uses a modular approach for reading diffraction data; when the program is started it reads in all Python files matching a specified naming convention. These files define different protocols for reading input files. A similar mechanism has now been put in place for exporting results, which include structural models (coordinates), observed and computed histograms, Fourier and charge-flipping maps, etc. from refinements. By implementing this modular approach for imports and exports, the program can be extended by the user to support new formats through the development of a short set of routines (implemented as a Python class) that can be modeled after existing code. Once this has been written, *no changes are needed to GSAS-II* to incorporate it into the package or place that format in appropriate menus.

Some significant import and export modules include both input and output support for the International Union of Crystallography crystallographic information framework standard (Hall *et al.*, 1991; Brown and McMahon, 2002) and the PDB format (Berman *et al.*, 2000) for protein structures. GSAS-II now can also import coordinates combined with irreducible representational modes that have been computed using the ISODISPLACE web site, <http://stokes.byu.edu/isodistort.html> (Campbell *et al.*, 2006). The modes are used to create linear constraints on the coordinates.

TABLE I. List of currently supplied GSAS-II tutorial web pages.

Starting GSAS-II
Powder pattern peak picking, fitting, and indexing
Lab X-ray powder pattern Rietveld refinement
CW Neutron powder pattern Rietveld refinement with constraints
Combined neutron X-ray Rietveld refinement and Fourier maps
Two-dimensional (2D) Image calibration
2D Image integration
2D Strain data fitting
Sequential refinement of crystallographic data
Parametric fitting and pseudo-variables
Synchrotron X-ray powder pattern Rietveld refinement
Charge-flipping structure solution (two examples)
Small-angle image processing
Small-angle X-ray data size distribution
Fitting small-angle scattering data
Sequential refinement of small-angle scattering data

These import/export routines allow one to “close the loop” when using GSAS-II for the protein structure refinement in tandem with omit map creation (Bhat, 1988) and Coot (Emsley *et al.*, 2010) for model visualization and building.

#### V. UPDATING AND REGRESSION

The GSAS-II package is written in Python with a small amount of speed-sensitive coding performed in Fortran. All source code is supplied, as well as shared-library versions of the compiled Fortran code for most common computing platforms. These codes are placed in a Subversion version control system, so that every change to any part of the code is tagged with a version number (Pilato *et al.*, 2008). If GSAS-II is installed by downloading the code directly from the Subversion repository, then a menu option can be used to check if the latest version of the code is installed. This same menu also provides an option to regress, which will return the source code to a previous version. This is provided in case an update introduces a bug or so that if there is a question, results from a previous version of the code can be compared to the results from a later version.

#### VI. SMALL-ANGLE SCATTERING

Capabilities for reduction analysis of isotropic small-angle scattering data have been added to GSAS-II, modeled on the Irena code (Ilavsky and Jemian, 2009). A full description of the small-angle scattering analysis capabilities has been written (Von Dreele, *in press*).

#### VII. TUTORIALS

An important aspect of the documentation for GSAS-II is a set of tutorials that show how different aspects of the program are used. These tutorials are prepared as web pages that are distributed along with the source code. At present, 17 tutorials of varying length have been created. These are described in Table I.

#### VIII. SPHINX DOCUMENTATION

As GSAS-II has matured, the code has been reorganized to keep modules small and in some cases to provide better groupings for related routines. It has also grown significantly in size, although it is still much more compact than comparable ones written in other languages. To better allow developers to follow the code and locate routines, comments for most classes and higher-level routines have been provided using “restructured text”. The Sphinx documentation generator can then read these comments, which provides as output a series of web pages that include cross-referenced indices and even links showing the code (Brandl, 2010). The documentation from Sphinx is also available as an Adobe portable document file. This developer’s documentation is expected to expand as the code does, but is currently >150 pages in length if printed.

#### ACKNOWLEDGEMENTS

The authors thank the many users who have taken the time to let the authors know when GSAS-II has not worked properly for them and provided the authors with enough detail to

track down their problems. Many users have also provided valuable suggestions on how GSAS-II could be more useful for their work or could be more convenient; not all good suggestions have yet been followed, but in time the authors hope to get to them. Use of the Advanced Photon Source, an Office of Science User Facility operated for the US Department of Energy (DOE) Office of Science by Argonne National Laboratory, was supported by the US DOE under Contract no. DE-AC02-06CH11357.

## SUPPLEMENTARY MATERIAL

The supplementary material for this article can be found at <http://www.journals.cambridge.org/PDJ>

- Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). "The protein data bank," *Nucleic Acids Res.* **28**, 235–242.
- Bhat, T. N. (1988). "Calculation of an OMIT map," *J. Appl. Crystallogr.* **21**, 279–281.
- Brandl, G. (2010). Sphinx: python documentation generator. Available at [sphinx-doc.org](http://sphinx-doc.org).
- Brown, I. D., and McMahon, B. (2002). "CIF: the computer language of crystallography," *Acta Crystallogr. B* **58**, 317.
- Campbell, B. J., Stokes, H. T., Tanner, D. E., and Hatch, D. M. (2006). "ISODISPLACE: a web-based tool for exploring structural distortions," *J. Appl. Crystallogr.* **39**, 607–614.
- David, W. I. F., Shankland, K., van de Streek, J., Pidcock, E., Motherwell, W. D. S., and Cole, J. C. (2006). "DASH: a program for crystal structure determination from powder diffraction data," *J. Appl. Crystallogr.* **39**, 910–915.
- Dmowski, W., Toby, B. H., Egami, T., Subramanian, M. A., Gopalakrishnan, J., and Sleight, A. W. (1988). "Short-range ordering due to displacements of thallium and oxygen atoms in superconducting  $Tl_2Ba_2CaCu_2O_8$  observed by pulsed-neutron scattering," *Phys. Rev. Lett.* **61**, 2608–2611.
- Emsley, P., Lohkamp, B., Scott, W. G., and Cowtan, K. (2010). "Features and development of Coot," *Acta Crystallogr. D* **66**, 486–501.
- Favre-Nicolin, V., and Cerny, R. (2002). "FOX, 'free objects for crystallography': a modular approach to ab initio structure determination from powder diffraction," *J. Appl. Crystallogr.* **35**, 734–743.
- Hall, S. R., Allen, F. H., and Brown, I. D. (1991). "The crystallographic information file (CIF): a new standard archive file for crystallography," *Acta Crystallogr. A* **47**, 655.
- Ilavsky, J., and Jemian, P. R. (2009). "Irena: tool suite for modeling and analysis of small-angle scattering," *J. Appl. Crystallogr.* **42**, 347–353.
- Larson, A. C., and Von Dreele, R. B. (2004). Report LAUR 86-748. Los Alamos National Laboratory.
- Pawley, G. S. (1981). "Unit-cell refinement from powder diffraction scans," *J. Appl. Crystallogr.* **14**, 357–361.
- Pilato, C. M., Collins-Sussman, B., and Fitzpatrick, B. W. (2008). *Version Control with Subversion* (O'Reilly Media, Inc., Sebastopol, CA).
- Rietveld, H. M. (1969). "A profile refinement method for nuclear and magnetic structures," *J. Appl. Crystallogr.* **2**, 65–71.
- Stephens, P. W., and Huq, A. (2002). "PSSP: an open source powder structure solution program for direct space simulated annealing," *Trans. Am. Crystallogr. Assoc.* **37**, 125–142.
- Toby, B. H. (2001). "EXPGUI, a graphical user interface for GSAS," *J. Appl. Crystallogr.* **34**, 210–213.
- Toby, B. H., and Von Dreele, R. B. (2007). Report from the GSAS-II Workshop: 10–11 May 2007, IUCr Computational Commission Newsletter, edited by L. Cranswick.
- Toby, B. H., and Von Dreele, R. B. (2013). "GSAS-II: the genesis of a modern open-source all-purpose crystallography software package," *J. Appl. Crystallogr.* **46**, 544–549.
- Von Dreele, R. B. (in press). "Small angle scattering data analysis in GSAS-II," *J. Appl. Crystallogr.*