The Journal of Open Source Software

Cont2SAS: A python package for calculating SAS parameters from continuum nanostructures

Arnab Majumdar [1,3], **Martin Müller**[1,2,3], **and Sebastian Busch** [1¶]

1 German Engineering Materials Science Centre (GEMS) at Heinz Maier-Leibnitz Zentrum (MLZ), Helmholtz-Zentrum Hereon GmbH, Lichtenbergstr. 1, 85748 Garching, Germany **2** Institute of Materials Physics, Helmholtz-Zentrum Hereon GmbH, Max-Planck-Str. 1, 21502 Geesthacht, Germany **3** Institut für Experimentelle und Angewandte Physik (IEAP), Christian-Albrechts-Universität zu Kiel, Leibnizstr. 19, 24098 Kiel, Germany ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Cont2SAS is a software tool built around the existing software solution Sassena – known for calculating scattering patterns from simulated atomic structures. The goal of Cont2SAS is to provide a software similar to Sassena, but for calculating Small Angle Scattering (SAS) data from simulated Continuum (Cont) nanostructures. Cont2SAS can calculate SAS patterns, i.e. SAS intensity (I) vs. scattering vector magnitude (Q) using a novel numerical method. Cont2SAS can also calculate effective cross-section (σ_{eff}), i.e. the count rate of scattered radiation per incident unit flux. σ_{eff} is calculated by integrating instrument-agnostic SAS patterns taking instrument geometry into account. Cont2SAS can be used for different purposes, such as validating simulations, tuning simulation parameters, and analyzing SAS data.

Statement of need

Small Angle Scattering (SAS) experiments with neutrons (SANS) and X-rays (SAXS) are useful techniques for probing material nanostructures (Chen et al., 2012). With the Time Resolved (TR) variant of SAS, the time evolution of nanostructures can also be probed (Hollamby, 2013). However, a direct retrieval of nanostructure from SAS data is not possible due to the so-called phase problem (Billinge & Levin, 2007). Therefore, it is fruitful to combine SAS data with simulations to study nanostructures (Majumdar et al., 2024). This approach is also useful for validating theories underlying simulations (Reich et al., 2022).

Simulated structures in real space can be generated using atomistic and continuum simulation. However, continuum simulation is the only option when the nanostructure evolution is influenced by bigger features or the SAS data is recorded for long time period, as atomistic simulation can not simulate large structures for long time (Ghavanloo et al., 2019). Despite this indispensability, there is no software available for calculating SAS data from continuum simulations. Cont2SAS aims to fill this void by providing a tool that offers fast and error-free comparison of continuum simulations and SAS data. The ultimate goal is to study nanomaterials and validating theories using SAS data, particularly when length-scales and time-scales, inaccessible to atomistic simulations, need to be simulated.

State of the field

From atomistic simulations, different software such as nMoldyn (Hinsen, 2023; Róg et al., 2003), MDANSE (Goret et al., 2017; Neutron & Facility, 2023), LiquidLib (Walter et al., 2018; Z-laboratory, 2023), and Sassena (Lindner, 2012, 2017; Lindner & Smith, 2012; Majumdar et

The Journal of Open Source Software

al., 2024; Majumdar & Lindner, 2023) can calculate SAS patterns but they can not calculate the same from continuum simulations. An in-house code is available for conitnuum simulations but it was built for a 2D membrane system lacking generality (Dorrell et al., 2020). Hence, `Cont2SAS` is built using `Sassena` as a backend calculator. `Sassena` was chosen over other software due to its notable computation speed and robustness (Majumdar et al., 2024).

Software features

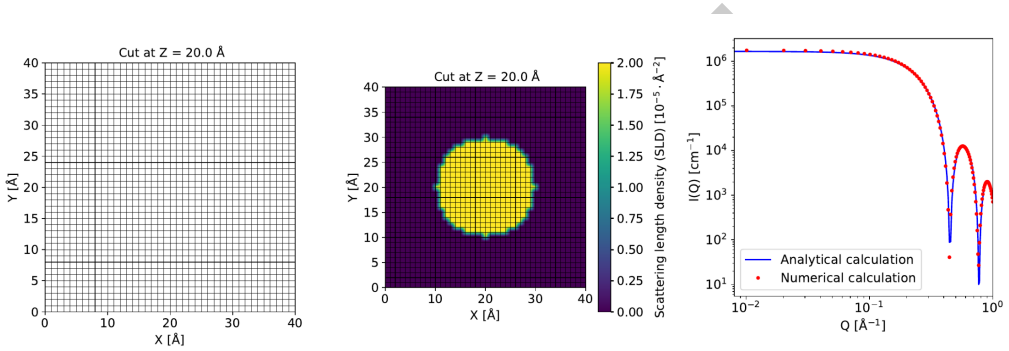

Figure 1: Workflow of SAS pattern calculation: [left] Mesh generation, [middle] SLD assignment, [right] numerical calculation of SAS pattern. The numerical calculation matches well with the known analytical formula for spherical nanoparticles (Guinier et al., 1955).

`Cont2SAS` calculates SAS patterns taking simulated nanostructures as an input. The simulated structure must provide either Scattering Length Density (SLD) (β) values or a set of variables from which SLD values can be calculated. The simulated input is processed to a data taylor-made for `Sassena`. `Sassena` calculates the SAS intensity (I) as a function of scattering vector magnitude (Q), i.e. SAS pattern, for different time steps (see Figure 1).

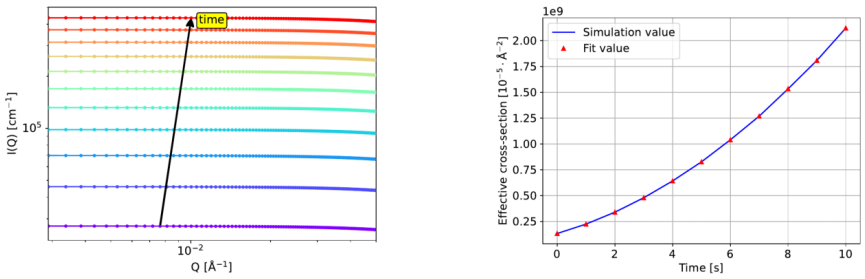

Figure 2: Workflow of effective cross-section (σ_{eff}) calculation: [left] SAS patterns and [right] σ_{eff} calculated from SAS patterns at different time steps. With time, the SLD of a spherical nanoparticle increases linearly. The calculated σ_{eff} values are proportional to the square of SLD difference between the particle and its environment.

`Cont2SAS` can also calculate the time evolution of effective cross-section (σ_{eff}), which is defined as the count rate per incident unit flux. This feature was not available in the in-house code (Dorrell et al., 2020) or `Sassena` (Majumdar & Lindner, 2023). The time evolution of count rate is useful when the chemical composition changes over time instead of the nanostructure, e.g. while storing hydrogen in ball-milled powder sample (Aslan et al., 2019). Figure 2 demonstrates the workflow of such calculation from a series of SAS patterns. The calculated σ_{eff} must be multiplied by an empirical factor before comparing with measured count rate.

Conclusion

Cont2SAS provides a software platform for calculating SAS pattern from continuum simulations of nanostructures. The addition of effective cross-section in the software package further enables the analysis of count rate. Cont2SAS can be used for analyzing SAS data and validating simulations to study nanomaterials.

Acknowledgements

This publication was written in the context of the work of the consortium DAPHNE4NFDI in association with the German National Research Data Infrastructure (NFDI) e.V. NFDI is financed by the Federal Republic of Germany and the 16 federal states and the consortium is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - project number 460248799. The authors would like to thank for the funding and support. Furthermore, thanks go to all institutions and actors who are committed to the association and its goals.

References

Aslan, N., Horstmann, C., Metz, O., Kotlyar, O., Dornheim, M., Pistidda, C., Busch, S., Lohstroh, W., Müller, M., & Pranzas, K. (2019). High-pressure cell for in situ neutron studies of hydrogen storage materials. *Journal of Neutron Research*, *21*(3-4), 125–135. https://doi.org/10.3233/JNR-190116

Billinge, S. J., & Levin, I. (2007). The problem with determining atomic structure at the nanoscale. *Science*, *316*(5824), 561–565. https://doi.org/10.1126/science.1135080

Chen, Z. H., Kim, C., Zeng, X., Hwang, S. H., Jang, J., & Ungar, G. (2012). Characterizing size and porosity of hollow nanoparticles: SAXS, SANS, TEM, DLS, and adsorption isotherms compared. *Langmuir*, *28*(43), 15350–15361. https://doi.org/10.1021/la302236u

Dorrell, M. W., Beaven, A. H., & Sodt, A. J. (2020). A combined molecular/continuum-modeling approach to predict the small-angle neutron scattering of curved membranes. *Chemistry and Physics of Lipids*, *233*, 104983. https://doi.org/10.1016/j.chemphyslip.2020.104983

Ghavanloo, E., Rafii-Tabar, H., & Fazelzadeh, S. A. (2019). *Computational continuum mechanics of nanoscopic structures*. Springer. https://doi.org/10.1007/978-3-030-11650-7

Goret, G., Aoun, B., & Pellegrini, E. (2017). MDANSE: An interactive analysis environment for molecular dynamics simulations. *Journal of Chemical Information and Modeling*, *57*(1), 1–5. https://doi.org/10.1021/acs.jcim.6b00571

Guinier, A., Fournet, G., Walker, C. B., & Yudowitch, K. L. (1955). *Small-angle scattering of x-rays*. Wiley New York.

Hinsen, K. (2023). *nMOLDYN3–version 3.0.12*. https://github.com/khinsen/nMOLDYN3/.

Hollamby, M. J. (2013). Practical applications of small-angle neutron scattering. *Phys. Chem. Chem. Phys.*, *15*, 10566–10579. https://doi.org/10.1039/C3CP50293G

Lindner, B. (2012). *Towards a unification of supercomputing, molecular dynamics simulation and experimental neutron and x-ray scattering techniques* [PhD thesis]. University of Tennessee, United States.

Lindner, B. (2017). *Sassena*. https://github.com/camm/sassena.

Lindner, B., & Smith, J. C. (2012). Sassena—x-ray and neutron scattering calculated from molecular dynamics trajectories using massively parallel computers. *Computer Physics*

Communications, *183*(7), 1491–1501. https://doi.org/10.1016/j.cpc.2012.02.010

Majumdar, A., & Lindner, B. (2023). *Sassena*. https://codebase.helmholtz.cloud/DAPHNE4NFDI/sassena. https://doi.org/10.5281/zenodo.10037485

Majumdar, A., Müller, M., & Busch, S. (2024). Computation of x-ray and neutron scattering patterns to benchmark atomistic simulations against experiments. *International Journal of Molecular Sciences*, *25*(3), 1547. https://doi.org/10.3390/ijms25031547

Neutron, U. S. I., & Facility, M. (2023). *MDANSE–version 1.5.2*. https://github.com/ISISNeutronMuon/MDANSE/.

Reich, V., Majumdar, A., Müller, M., & Busch, S. (2022). Comparison of molecular dynamics simulations of water with neutron and x-ray scattering experiments. *EPJ Web of Conferences*, *272*, 01015. https://doi.org/10.1051/epjconf/202227201015

Róg, T., Murzyn, K., Hinsen, K., & Kneller, G. R. (2003). nMoldyn: A program package for a neutron scattering oriented analysis of molecular dynamics simulations. *Journal of Computational Chemistry*, *24*(5), 657–667. https://doi.org/10.1002/jcc.10243

Walter, N. P., Jaiswal, A., Cai, Z., & Zhang, Y. (2018). LiquidLib: A comprehensive toolbox for analyzing classical and ab initio molecular dynamics simulations of liquids and liquid-like matter with applications to neutron scattering experiments. *Computer Physics Communications*, *228*, 209–218. https://doi.org/10.1016/j.cpc.2018.03.005

Z-laboratory. (2023). *Liquidlib–version 1.0*. https://github.com/Z-Laboratory/LiquidLib/tree/master.