

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}

Received 24 February 2023

Accepted 3 July 2023

Edited by J. Ilavsky, Argonne National Laboratory, USA

Keywords: small-angle scattering; *Shape2SAS*; geometrical shapes; teaching; simulation; virtual experiments; pair distance distributions; structure factors; form factors; multi-contrast particles.

Supporting information: this article has supporting information at journals.iucr.org/j

^aDepartment of Neuroscience, University of Copenhagen, Copenhagen, Denmark, ^bUniversity of Montana, Missoula, Montana, USA, ^cNiels Bohr Institute, University of Copenhagen, Copenhagen, Denmark, and ^dDepartment of Food Science, University of Copenhagen, Copenhagen, Denmark. *Correspondence e-mail: andreas.larsen@sund.ku.dk

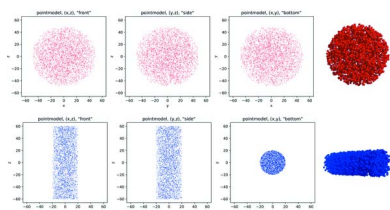
Shape2SAS is a web application that allows researchers and students to build intuition about and understanding of small-angle scattering. It is available at <https://somo.chem.utk.edu/shape2sas>. The user defines a model of arbitrary shape by combining geometrical subunits, and *Shape2SAS* then calculates and displays the scattering intensity and the pair distance distribution, as well as a visualization of the user-defined shape. Simulated data with realistic noise are also generated. Here, it is demonstrated how *Shape2SAS* can calculate and display the different scattering patterns for various geometrical shapes, such as spheres and cylinders. It is also shown how the effect of structure factors can be visualized. Finally, it is indicated how multi-contrast particles can readily be generated, and how the calculated scattering may be used to validate and visualize analytical models generated in analysis software for fitting small-angle scattering data.

1. Introduction

We introduce *Shape2SAS*, a website for simulating small-angle scattering (SAS) data and pair distance distributions from various shapes. *Shape2SAS* is readily available as a web application (<https://somo.chem.utk.edu/shape2sas>) implemented using the *GenApp* framework (Savelyev & Brookes, 2019) for constructing graphical user interfaces (GUIs) for scientific software. The user can construct particles of arbitrary shape by combining geometrical subunits, *e.g.* two spheres and a cylinder to form a dumbbell. The user can thus compare calculated scattering and pair distance distribution from different particle shapes, such as cylinders and spheres, or more elaborate particles shapes, such as dumbbells or spheres coated with smaller spheres. The dimensions and excess scattering-length density of the particle (*i.e.* contrast between particle and average solvent scattering-length density) can be adjusted to build intuition about how these affect the resulting scattering. The user can also investigate the effects of polydispersity or structure factors.

The website is particularly useful for educators when organizing introductory or advanced courses in SAS data analysis. Due to its nature as a stand-alone web application, the only requirement is a web browser. The program can likewise be used for testing analytical form factors, as demonstrated in this article.

Shape2SAS applies a Monte Carlo based method previously implemented in the program *McSim* (Hansen, 1990, 2014) (see the comparison between *Shape2SAS* and *McSim* in Table S1 of the supporting information).



Shape2SAS also outputs simulated data using an empirical error model. The simulated data provide an idea about how exposure time, concentration, contrast and particle size affect the noise in data. Furthermore, the simulated data can be used in tutorials on how to analyze SAS data, e.g. using programs for fitting analytical form factors, such as *SasView* (<https://www.sasview.org/>), *SASfit* (Breßler *et al.*, 2015) or *WillItFit* (Pedersen *et al.*, 2013), or programs for *ab initio* modeling (Franke & Svergun, 2009; Grant, 2018).

2. Applied small-angle scattering theory

In this section, we provide the theoretical and computational basis for *Shape2SAS*, and account for the implementation of this.

2.1. Normalized scattering from identical particles in solution

The normalized SAS from a diluted sample of randomly oriented identical particles can be calculated as a double sum over all scatterers in each particle (Debye, 1915):

$$I_{\text{norm}}(q) = \frac{\sum_{j,k=1}^N \Delta b_j \Delta b_k [\sin(qr_{jk})/qr_{jk}]}{\sum_{j,k=1}^N \Delta b_j \Delta b_k},$$

where Δb_j is the excess scattering length of the j th scatterer, r_{jk} is the distance between the j th and k th scatterers, and q is the momentum transfer [$q = 4\pi \sin(\theta)/\lambda$, where 2θ is the scattering angle and λ is the wavelength of the incoming X-rays or the de Broglie wavelength of the incoming neutrons]. N is the number of scatterers in each particle. We assume point scatterers, so the effect of atomic form factors is neglected. The user provides excess scattering-length densities as input, $\Delta \text{SLD}_i = \Delta b_i/V$, where V is the effective volume of one scatterer. The effective volume is given as the particle volume (V_p) divided by the number of scatterers: $V = V_p/N$. The point density is kept constant in all subunits, so V is also constant. This normalized scattering is the form factor, $P(q)$, of the particle.

2.2. The pair distance distribution

By binning the scattering pairs after their pair distances, r_{jk} , the double sum can be reduced to a single sum over the number of bins:

$$I_{\text{norm}}(q) = \frac{\sum_{i=1}^{N_{\text{bins}}} p_i [\sin(qr_i)/qr_i]}{\sum_{i=1}^{N_{\text{bins}}} p_i},$$

where $r_i = [i - (1/2)]dr$ and dr is the bin width. The term p_i is the number of pairs in each bin, weighted by the product of their excess scattering lengths:

$$p_i = \sum_{j,k=1}^N f(r_{jk}) \Delta b_j \Delta b_k,$$

where $f(r_{jk}) = \begin{cases} 1, & \text{if } (i-1)dr \leq r_{jk} < idr, \\ 0, & \text{otherwise.} \end{cases}$

This is also a double sum but $\sin(qr)/qr$ is not evaluated for each distance.

The pair distance distribution $p(r)$ is the continuous limit of $p = \{p_1, p_2, \dots, p_{N_{\text{bins}}}\}$, as $N_{\text{bins}} \rightarrow \infty$ and $N \rightarrow \infty$. In this limit, the contributions from the self terms ($j = k$) are negligible. Therefore, the self terms are excluded from the sums in *Shape2SAS*, and $p(0) = 0$ is added.

The largest distance in the particle, D_{max} , is given as output, along with the radius of gyration (Glatter, 1977; Guinier & Fournet, 1955):

$$R_g^2 = \frac{1}{2} \frac{\int_0^{D_{\text{max}}} r^2 p(r) dr}{\int_0^{D_{\text{max}}} p(r) dr}.$$

The output $p(r)$ is normalized, so the maximum is unity.

2.3. Polydispersity

Polydispersity implies a distribution of sizes of the particles (Pauw *et al.*, 2013). *Shape2SAS* allows for inclusion of a simple Gaussian size distribution. The size distribution is implemented by introducing a size-scaling parameter, s , which scales all distances in the user-defined shape ($r \rightarrow sr$). The parameter s is normally distributed, with a mean of unity ($\mu_s = 1$) and standard deviation (σ_s) set by the user.

$$p_{\text{poly}}(r, \sigma_s) = \frac{\int_0^\infty p_s(r) \exp\left\{(-1/2)\left[(s-1)/\sigma_s\right]^2\right\} v_s^2 ds}{\int_0^\infty \exp\left\{(-1/2)\left[(s-1)/\sigma_s\right]^2\right\} v_s^2 ds},$$

where $p_s(r)$ is the pair distance distribution after scaling all distances with s and $v_s = s^3$ is a relative (unitless) volume, which is included to account for the fact that the scattering from a particle is proportional to the square of the molecular volume. The relative volume is an approximation and is only exact for spheres. The integrals are evaluated numerically with upper and lower limits of $1 \pm 3\sigma_s$.

While this is a simplification of the many ways in which a sample can exhibit polydispersity, the implementation is simple and efficient, and allows users and students to explore the consequences of polydispersity in SAS data.

2.4. Structure factors

Interparticle interactions can be expressed in the form of structure factors, and the normalized scattering intensity can be expressed as a product of the form factor and the structure factor (Breßler *et al.*, 2015):

$$I_{\text{norm}}(q) = P(q)S(q).$$

So far, two structure factors are implemented in *Shape2SAS*: a hard-sphere structure factor (Kinning & Thomas, 1984), describing interparticle repulsion, and a 2D fractal structure factor, describing particle aggregation (Larsen *et al.*, 2020). The decoupling approximation is also applied to account for non-spherical or polydisperse particles (Kotlarchyk & Chen, 1983; Larsen *et al.*, 2020). Although such interparticle interactions would affect the effective pair distance distribution, the program only applies the structure factor on the calculated scattering intensity. That is, if a structure factor is opted for,

the output $p(r)$ is from non-interacting particles, whereas $I(q)$ is from a sample of interacting particles. The $p(r)$ of the interacting particles can *e.g.* be generated in *BayesApp* (Hansen, 2012), which is likewise available as a web application in *GenApp*.

2.5. Interface roughness

A particle can be modeled by combining geometrical subunits. If the subunits have different contrasts, there may be discrepancies between the sharp boundaries between subunits of the model and the softer and more fluent boundaries between components in an actual particle. The interface roughness can be modeled by including a surface roughness factor (Als-Nielsen & McMorro, 2011), which effectively smears the interfaces:

$$I_{\text{norm}}(q) = P(q)S(q) \exp\left[-\frac{1}{2}(q\sigma_{\text{rough}})^2\right],$$

where σ_{rough} is a smearing parameter. In *Shape2SAS*, surface roughness only affects $I_{\text{norm}}(q)$, not $p(r)$. Such terms have been applied to fit scattering from liquid–vapor interfaces (Als-Nielsen, 1986), microemulsions (Skar-Gislinge & Arleth, 2011; Foster *et al.*, 2008) and protein–lipid nanodiscs (Skar-Gislinge & Arleth, 2011; Barclay *et al.*, 2022).

2.6. Simulation of data with an empirical noise model

Besides the pair distance distribution and the normalized scattering, *Shape2SAS* also outputs simulated data, $I_{\text{sim}}(q)$.

First, the forward scattering is calculated, so that it scales with particle volume, V_p , and volume fraction (concentration of the sample), η :

$$I(0) = \eta V_p \sum_{i=1}^{N_{\text{bins}}} p_i.$$

The last factor depends on the contrast, and if all subunits have the same contrast then $\sum_i p_i = \Delta\text{SLD}^2$. The scaled scattering is given as $I(q) = I(0)I_{\text{norm}}(q)$.

The errors, $\sigma_{\text{sim}}(q)$, are estimated using an empirical model based on small-angle X-ray scattering (SAXS) data (Sedlak *et al.*, 2017), using the scaled scattering, $I(q)$, and the relative exposure time as input (Appendix A). When using the default parameters, the errors resemble those from a typical synchrotron SAXS experiment. To simulate laboratory-source data, the exposure time should be reduced, *e.g.* by an order of magnitude (Sedlak *et al.*, 2017). Future efforts are needed to simulate realistic small-angle neutron scattering (SANS) errors (see *Discussion*). The simulated data, $I_{\text{sim}}(q)$, are sampled stochastically from normal distributions with means $I(q)$ and standard deviations $\sigma_{\text{sim}}(q)$.

3. Implementation

3.1. Architecture

The program is designed for ease of use and ease of maintenance. Input parameters are provided via an online

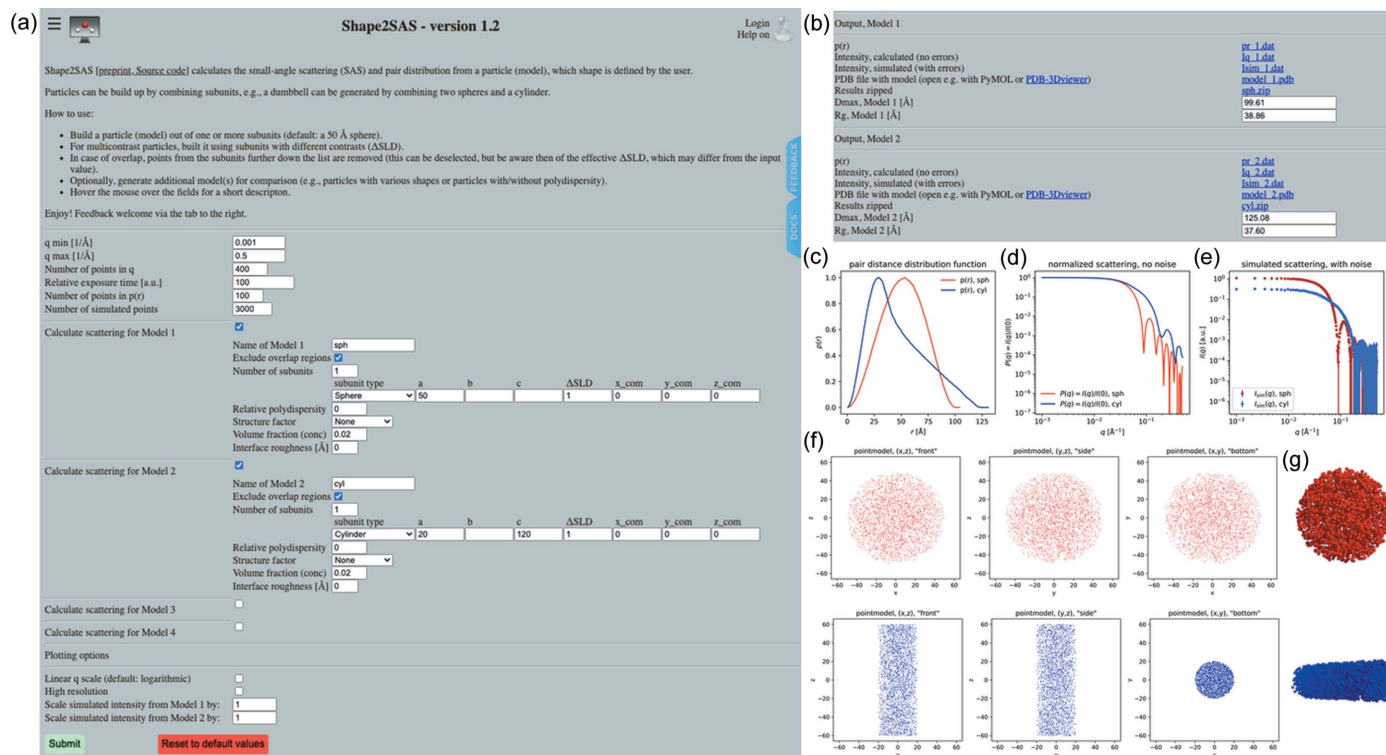


Figure 1 Simulating scattering from a sphere (red) and a cylinder (blue) using *Shape2SAS*. (a) GUI input parameters. (b) GUI output parameters. (c) Pair distance distributions. (d) Calculated normalized scattering. (e) Simulated scattering intensities with noise. (f) 2D projections of the models. (g) 3D models rendered in *PyMOL* (<http://www.pymol.org>).

GUI (Fig. 1). Inputs are read by a Python wrapper script. The wrapper calls Python functions that perform the calculations and return output files, plots and values to the wrapper. Finally, the wrapper returns the output to the GUI, which displays them for the user.

3.2. Description of core functions

A core element of *Shape2SAS* is the Python function that generates points from a given subunit. First, the limits of the geometric subunit are defined in either Cartesian, polar or spherical coordinates. Then, random uniformly distributed points are generated in the volume defined by the subunit. Lastly, if selected, the points are shifted to a new center of mass. The volumes of the subunits are calculated, and the number of points inserted in each subunit is adjusted, so that the point density is constant (see *Applied small-angle scattering theory*). The total number of points in the model (composed of one or more subunits) is $N = 3000$ by default, which provides a rapid result. For higher precision, N can be increased. The computational time goes as N^2 . For $N = 3000$ and two models, the results and plots are available within 6 s, whereas e.g. for $N = 9000$ and two models, the results and plots are available within 14 s. The SAS intensity calculated from the same particle will vary stochastically, with larger differences at higher values of q . By increasing N , the differences are decreased; depending on the q range of interest, N should be adjusted to balance computational time against precision. With 5000 points, the calculated intensity for a 50 Å sphere is precisely reproduced up to approximately $q = 0.2 \text{ \AA}^{-1}$ (Fig. S1 of the supporting information). In the case of overlap between subunits, the points from the subunits furthest down the list of subunits are removed, e.g. if the user chooses a 30 Å sphere with $\Delta\text{SLD} = -1$ as the first subunit and a 50 Å sphere with $\Delta\text{SLD} = 1$ as the second subunit, the resulting particle will be a core-shell particle with a 30 Å core with $\Delta\text{SLD} = -1$ and a shell with thickness of 20 Å and $\Delta\text{SLD} = 1$ (examples in Fig. S2). In the overlap region, only the points from the first subunit remain. Exclusion of points from overlapping subunits can be exploited in multi-contrast situations (see *Example 3*).

Another core element is the calculation of all distances in a model. This step is time and memory consuming, and one of the computational bottlenecks of the program. However, using the *NumPy* (Harris *et al.*, 2020) function `meshgrid()`, all distances between 3000 points are calculated in less than a second on the *GenApp* server. Another computational bottleneck is the calculation of $p(r)$. This is carried out using the *NumPy* function `histogram()`. When polydispersity is included, several histograms are calculated, and in that case *Shape2SAS* uses the `histogram1d()` function (<https://github.com/astrofrog/fast-histogram>), which is faster than *NumPy*'s `histogram()`. A polydisperse $p(r)$ is thus calculated in a few seconds.

3.3. User interface

A key goal of the project is accessibility and ease of use. Therefore, the program is implemented as a web application,

meaning that no installation is required. This is optimal for use in courses and tutorials. The program is part of the *GenApp* (Savelyev & Brookes, 2019) vision for making SAS software available for everyone.

The user builds up a model of a predefined set of geometric subunits, currently including sphere, triaxial ellipsoid, cylinder, disc, cube, cuboid, hollow sphere, hollow cube, cylindrical ring and discoidal ring. The geometrical parameters of the subunits can then be changed from the default values, along with their contrast and center of mass. A model consists of the collected points in the volume spanned by the subunits. By default, points are deleted from overlapping regions, but this is optional as mentioned above. If a structure factor is selected, the scattering contribution from the structure factor will be displayed along with the total scattering.

The user can choose to calculate $p(r)$ and $I(q)$ from additional models (up to four), and the procedure is the same as for the first model. The $p(r)$ and $I(q)$ from the models are plotted together in the GUI, and can thus be directly compared without having to plot the data in third-party software. The simulated data can be scaled in the plot for easy visual inspection.

All output data [$p(r)$, $I_{\text{norm}}(q)$ and $I_{\text{sim}}(q)$], as well as plots and a 3D model for visualization (in PDB format), can be downloaded from the web interface for further analysis, plotting *etc.*

3.4. Documentation and validation

All menus and input boxes in the GUI are described with help text, which is shown by hovering the mouse over the elements. The source code is documented with extensive comments in the code, including documentation of all functions. Central references are provided directly in the GUI. Each model is visualized as a 3D point model using a *Jmol* (<http://www.jmol.org/>) plugin and as 2D projections. *Shape2SAS* models and structure factors have been tested against analytical models using *SasView* (<https://www.sasview.org>).

Source code is available on GitHub (<https://github.com/ehb54/GenApp-Shape2SAS>) under the GNU General Public License v3.0.

4. Examples of use

These examples are designed to be relevant in the context of research as well as research-based teaching. The first example showcases how *Shape2SAS* can be used for generating intuition about SAS from different shapes, and the second example demonstrates the effect of inter-particle interactions. The last example demonstrates how more complex particles can be built and how *Shape2SAS* can be utilized to test analytical form factors.

4.1. Example 1: comparing scattering from particles of different shape

Shape2SAS can be used to quickly calculate and compare $p(r)$ and $I(q)$ from particles with various shapes. One example

is the scattering from a sphere with a radius of 50 Å, and a cylinder with a radius of 20 Å and a length of 400 Å (Fig. 1). Such an example could help a student build intuition about the scattering and pair distance distribution functions for *e.g.* spherical, elongated or hollow bodies. Moreover, R_g and D_{\max} are calculated and displayed in the GUI for quick comparison.

4.2. Example 2: hard-sphere structure factor

Shape2SAS can add inter-particle interactions to the scattering using built-in structure factors. One example is shown in Fig. 2, where the scattering from a sample of ellipsoids of revolution (minor axis 50 Å and major axis 100 Å) was calculated with and without interparticle interaction, described by the hard-sphere structure factor and the decoupling approximation, with a hard-sphere radius of 70 Å and a volume fraction of 0.2. Such an exercise could help students or researchers to understand the effect of structure factors and recognize interparticle interactions in measured SAS data.

4.3. Example 3: validating analytical form factor

Shape2SAS can be used when developing analytical form factors. By generating the same shape in *Shape2SAS* as that of the analytical model, the simulated data from *Shape2SAS* can be fitted. In this example, a core-shell cylinder was simulated (core radius 20 Å, core length 360 Å, shell thickness 20 Å, core contrast -1 and shell contrast $+1$). The example also showcases how multi-contrast particles can be generated in two ways in *Shape2SAS*. In the first approach, the core-shell cylinder is generated by combining non-overlapping subunits: a cylinder core, a hollow cylindrical shell and two small cylinders with shifted center of mass as end caps [Fig. 3(a)]. In the second approach, a large shell cylinder and a smaller core cylinder are combined, and points from the shell cylinder are removed from the overlapping region [Fig. 3(a)]. Both result in the same model, $p(r)$ and $I(q)$ [Figs. 3(b) and 3(c)]. If there is overlap, and exclusion of overlapping points is not opted for, then the contrast in the overlap region will effectively be the sum of contrasts of the overlapping subunits.

The simulated data were fitted with the analytical model `core_shell_cylinder` from *SasView* (https://www.sasview.org/sasmodels/model/core_shell_cylinder.html).

The data were well described by the model [Fig. 3(d)] and the model parameters were refined to values consistent with the input parameters (core radius 20.10 ± 0.07 Å, core length 363 ± 2 Å, shell thickness 20.1 ± 0.1 Å and shell contrast $+0.97 \pm 0.01$). The errors are standard deviations. The core contrast was fixed at -1 , since the combination of fitting core contrast and shell contrasts, and then scaling, gave high correlation between the parameters.

The demonstrated procedure can be valuable for testing new and more complex analytical models (Jakubauskas *et al.*, 2019), and for visualizing them as bead models.

This workflow is helpful when coding analytical models for SAS from particles in solution. While analytical models are often ideal for implementation in a model-refinement framework (such as *SasView*), *Shape2SAS* offers a simple, graphical and intuitive manner for testing the implementation and its accuracy – in real space.

5. Discussion

Shape2SAS is useful for demonstrations and tutorials, which has been showcased by three examples. *Example 1* compared the SAS from a sphere with the SAS from a cylinder; *Example 2* demonstrated the effect of interparticle repulsion via a hard-sphere structure factor; and *Example 3* showcased how more advanced shapes can be built, with various excess scattering-length densities, and how *Shape2SAS* can be used to test analytical form factors.

When carrying out analysis of the simulated data, discrepancies between input parameters and refined parameters may occur from the stochastic nature of *Shape2SAS* and the limited number of simulated points [Fig. 3(d)]. Moreover, if polydispersity is fitted, this is probably implemented differently in analytical models and may result in discrepancies.

Shape2SAS does not distinguish between SAXS and SANS. To make the distinction, the scattering-length densities should be provided in real units, reflecting either electron density or neutron scattering length density. Moreover, the empirical error model was derived from SAXS data. In SANS, the incoherent scattering should be considered along with

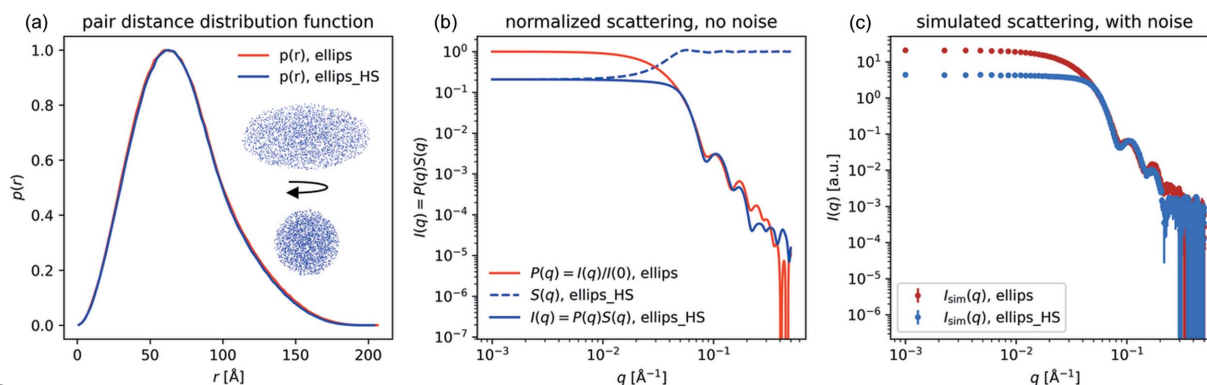


Figure 2

Scattering intensity from ellipsoids with and without interparticle repulsion. (a) Pair distance distributions from two instances of a prolate ellipsoid of revolution, without interparticle repulsion. Inset: 2D projections of the ellipsoid along the minor and major axes. (b) Normalized scattering intensity from the ellipsoids with and without interparticle repulsion as described by a hard-sphere structure factor. (c) Simulated scattering, with noise.

computer programs

resolution effects (Pedersen *et al.*, 1990). Furthermore, SANS data are often collected in several settings and subsequently merged (Pedersen, 2002a), which should be reflected in the error model.

Other programs exist that can calculate the theoretical scattering from various shapes using form factors, including

SASfit (Kohlbrecher & Breßler, 2022), *Irena* (Ilavsky & Jemian, 2009) and *SasView* (<https://www.sasview.org/>). These programs can also be used to build intuition about the scattering from particles of various shapes, and the effect of changing model parameters or adding structure factors. *Shape2SAS* is complementary to these programs: (i) by being

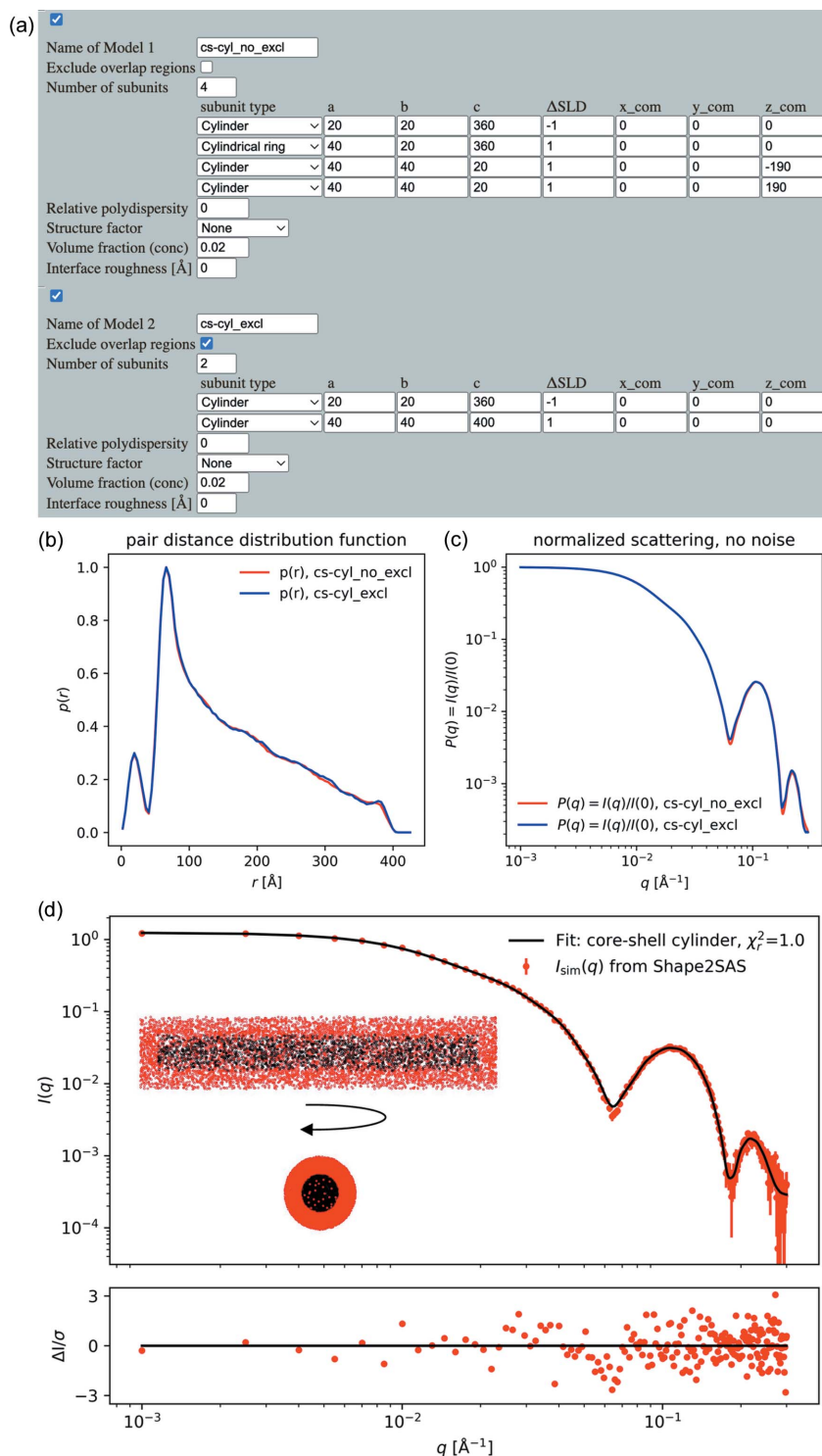


Figure 3

Core-shell cylinder data, simulated in *Shape2SAS* and fitted with an analytical model. (a) Core-shell cylinder input, without use of exclusion and with use of exclusion. (b) Pair distance distribution from *Shape2SAS*. (c) Normalized SAS intensity. (d) Simulated data fitted with the analytical model *core_shell_cylinder* from *SasView*. Inset: 2D projections of the core-shell cylinder.

available as a web application; (ii) by performing fast comparison of scattering and pair distance distributions from particles of various shapes; (iii) by automatically visualizing the particles; (iv) by generating simulated data with an empirical error model, which can be used as a virtual experiment for tutorials and to build intuition about some of the sources of noise in a SAS experiment; and (v) by being able to calculate scattering from particles that cannot be described by an analytical form factor.

Shape2SAS is one among many programs for making virtual experiments (Lefmann *et al.*, 2008), some of which can also generate 2D X-ray (Bergbäck Knudsen *et al.*, 2012; Franke *et al.*, 2020) or neutron (Willendrup & Lefmann, 2021) scattering data. *Shape2SAS* focuses on the data-analysis step, after reduction from 2D to 1D data, and after buffer subtraction. Common for such virtual-experiment software is the goal of preparing the user for best use of valuable beam time, and for helping the user to better understand and analyze the measured data. *Shape2SAS* excels in accessibility, meaning that virtual data can be generated in seconds, with no installation requirements.

In principle, the *Shape2SAS* input parameters could be transformed into variable parameters, and by addition of a scaling and a constant background, such a modified program could be used for fitting. It has previously been shown that Monte Carlo bead modeling approaches are useful in SAS analysis (Pedersen, 2002*b*), especially for generating complicated models that are difficult to describe through analytical form factors [e.g. perforated vesicles (Pedersen *et al.*, 2012) or protein–lipid complexes (Pedersen *et al.*, 2022)]. However, other programs applying the same principles have been developed with fitting in mind, including *CDEF* (Deumer *et al.*, 2022) and *SPONGE* (<https://github.com/bamresearch/sponge>), and we recommend use of these for fitting. *CDEF* applies the same principle of binning scattering pairs as *Shape2SAS*, whereas *SPONGE* applies the Debye formula directly, making it more accurate at the cost of longer computational times. When fitting actual data, we also encourage parametrizing the model to reflect physical properties rather than geometrical, to be able to better constrain and validate the refined parameters, e.g. with biophysical assays [see e.g. Skar-Gislinge & Arleth (2011) and Larsen *et al.* (2018)]. This is not possible in *Shape2SAS* and would typically have to be adjusted from case to case, which is easier without a GUI. *Shape2SAS* may, however, be downloaded, and functions reused by developers, to accommodate specific needs.

In summary, *Shape2SAS* makes SAS theory intuitive, visual, playful and accessible for both new and experienced users.

APPENDIX A Empirical error model

The errors were generated following the previously outlined protocol (Sedlak *et al.*, 2017):

(1) Compute a theoretical SAS profile, $I(q)$, as described in the main text.

(2) Scale $I(q)$. Scaling is done such that the scattering from a 50 Å sphere with ΔSLD of unity (default parameters) has a forward scattering of approximately unity.

(3) Multiply the scaled $I(q)$ by the relative exposure time (GUI input) to get $I_s(q)$.

(4) Calculate the variance using $c = 0.85$ and $q_{arb} = 0.3 \text{ \AA}^{-1}$, which define the constant scattering intensity of the buffer, $I_{buf}(q) = cI_s(q_{arb})$ (Sedlak *et al.*, 2017):

$$\sigma^2 = \frac{I_s(q)}{N(q)} + \frac{2cI_s(q_{arb})}{1-c}.$$

In the original error model (Sedlak *et al.*, 2017), the number of pixels per q bin, $N(q)$, was approximated by kq . This is, however, only valid up to a certain q value, q_t , which depends on detector size and geometry. Therefore, we use a modified empirical expression for $N(q)$ that better describes the number of pixels per q bin at higher values of q :

$$N(q) = \begin{cases} kq, & \text{if } q \leq q_t, \\ kq_t \exp\{(-1/2)[(q - q_t)/b]^a\}, & \text{otherwise.} \end{cases}$$

Values of $k = 4500$, $q_t = 1.4 \text{ \AA}^{-1}$, $a = 3$ and $b = 0.6 \text{ \AA}^{-1}$ were used in *Shape2SAS* to approximate a synchrotron SAXS setup with centered beam center on a PILATUS 2M detector (Fig. S3).

Acknowledgements

The authors thank Steffen M. Sedlak and Jan Lipfert for support in expanding the error model to high- q data (Appendix A). The authors acknowledge Steen Hansen for valuable discussions and inspiration, and the students at various courses on scattering techniques at the University of Copenhagen, who have tested the program.

Funding information

This work used the Extreme Science and Engineering Discovery Environment (XSEDE) (Towns *et al.*, 2014), which is supported by National Science Foundation (NSF) grant No. ACI-1548562 and utilized Jetstream2 (Hancock *et al.*, 2021) at Indiana University through allocation TG-MCB17057 to EB. This work benefited from the *CCP-SAS* (Perkins *et al.*, 2016) software developed through a joint Engineering and Physical Sciences Research Council (EPSRC) (EP/K039121/1) and NSF (CHE-1265821) grant. AHL was funded by the Lundbeck Foundation grant R347-2020-2339. EB was funded by the National Institutes of Health, National Institute of General Medical Sciences grant GM120600 and NSF grant 1912444.

References

- Als-Nielsen, J. (1986). *Physica A*, **140**, 376–389.
- Als-Nielsen, J. & McMorrow, D. (2011). *Elements of Modern X-ray Physics*. Chichester: Wiley.
- Barclay, A., Tidemand Johansen, N., Tidemand, F. G., Arleth, L. & Pedersen, M. C. (2022). *Acta Cryst. D* **78**, 483–493.
- Bergbäck Knudsen, E., Prodi, A., Baltser, J., Thomsen, M., Kjær Willendrup, P., Sanchez del Rio, M., Ferrero, C., Farhi, E., Haldrup, K., Vickery, A., Feidenhans'l, R., Mortensen, K., Meedom Nielsen,

- M., Friis Poulsen, H., Schmidt, S. & Lefmann, K. (2013). *J. Appl. Cryst.* **46**, 679–696.
- Breßler, I., Kohlbrecher, J. & Thünemann, A. F. (2015). *J. Appl. Cryst.* **48**, 1587–1598.
- Debye, P. (1915). *Ann. Phys.* **351**, 809–823.
- Deumer, J., Pauw, B. R., Marguet, S., Skroblin, D., Taché, O., Krumrey, M. & Gollwitzer, C. (2022). *J. Appl. Cryst.* **55**, 993–1001.
- Foster, T., Sottmann, T., Schweins, R. & Strey, R. (2008). *J. Chem. Phys.* **128**, 064902.
- Franke, D., Hajizadeh, N. R. & Svergun, D. I. (2020). *J. Appl. Cryst.* **53**, 536–539.
- Franke, D. & Svergun, D. I. (2009). *J. Appl. Cryst.* **42**, 342–346.
- Glatte, O. (1977). *J. Appl. Cryst.* **10**, 415–421.
- Grant, T. D. (2018). *Nat. Methods*, **15**, 191–193.
- Guinier, A. & Fournet, G. (1955). *Small-Angle Scattering of X-rays*. New York: John Wiley and Sons.
- Hancock, D. Y., Fischer, J., Lowe, J. M., Snapp-Childs, W., Pierce, M., Marru, S., Coulter, J. E., Vaughn, M., Beck, B., Merchant, N., Skidmore, E. & Jacobs, G. (2021). *Proceedings of the Practice and Experience in Advanced Research Computing (PEARC '21)*, pp. 1–8. New York: Association for Computing Machinery.
- Hansen, S. (1990). *J. Appl. Cryst.* **23**, 344–346.
- Hansen, S. (2012). *J. Appl. Cryst.* **45**, 566–567.
- Hansen, S. (2014). *J. Appl. Cryst.* **47**, 1469–1471.
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C. & Oliphant, T. E. (2020). *Nature*, **585**, 357–362.
- Ilavsky, J. & Jemian, P. R. (2009). *J. Appl. Cryst.* **42**, 347–353.
- Jakubauskas, D., Kowalewska, Ł., Sokolova, A. V., Garvey, C. J., Mortensen, K., Jensen, P. E. & Kirkensgaard, J. J. K. (2019). *Sci. Rep.* **9**, 19405.
- Kinning, D. J. & Thomas, E. L. (1984). *Macromolecules*, **17**, 1712–1718.
- Kohlbrecher, J. & Breßler, I. (2022). *J. Appl. Cryst.* **55**, 1677–1688.
- Kotlarchyk, M. & Chen, S. H. (1983). *J. Chem. Phys.* **79**, 2461–2469.
- Larsen, A. H., Arleth, L. & Hansen, S. (2018). *J. Appl. Cryst.* **51**, 1151–1161.
- Larsen, A. H., Pedersen, J. S. & Arleth, L. (2020). *J. Appl. Cryst.* **53**, 991–1005.
- Lefmann, K., Willendrup, P. K., Udby, L., Lebeck, B., Mortensen, K., Birk, J. O., Klenø, K., Knudsen, E., Christiansen, P., Saroun, J., Kulda, J., Filges, U., Konnecke, M., Tregenna-Piggott, P., Peters, J., Lieutenant, K., Zsigmond, G., Bentley, P. & Farhi, E. (2008). *J. Neutron Res.* **16**, 97–111.
- Pauw, B. R., Pedersen, J. S., Tardif, S., Takata, M. & Iversen, B. B. (2013). *J. Appl. Cryst.* **46**, 365–371.
- Pedersen, J. S. (2002a). *Neutrons, X-rays and Light: Scattering Methods Applied to Soft Condensed Matter*, edited by P. Lindner & Th. Zemb, pp. 127–144. Amsterdam: Elsevier.
- Pedersen, J. S. (2002b). *Neutrons, X-rays and Light: Scattering Methods Applied to Soft Condensed Matter*, edited by P. Lindner & Th. Zemb, pp. 381–390. Amsterdam: Elsevier.
- Pedersen, J. S., Oliveira, C. L. P., Hübschmann, H. B., Arleth, L., Manniche, S., Kirkby, N. & Nielsen, H. M. (2012). *Biophys. J.* **102**, 2372–2380.
- Pedersen, J. S., Posselt, D. & Mortensen, K. (1990). *J. Appl. Cryst.* **23**, 321–333.
- Pedersen, M. C., Arleth, L. & Mortensen, K. (2013). *J. Appl. Cryst.* **46**, 1894–1898.
- Pedersen, M. C., Johansen, N. T., Roche, J., Järvå, M., Törnroth-Horsefield, S. & Arleth, L. (2022). *bioRxiv*, 2022.10.28.512841.
- Perkins, S. J., Wright, D. W., Zhang, H., Brookes, E. H., Chen, J., Irving, T. C., Krueger, S., Barlow, D. J., Edler, K. J., Scott, D. J., Terrill, N. J., King, S. M., Butler, P. D. & Curtis, J. E. (2016). *J. Appl. Cryst.* **49**, 1861–1875.
- Savelyev, A. & Brookes, E. (2019). *Future Generation Comput. Syst.* **94**, 929–936.
- Sedlak, S. M., Bruetzel, L. K. & Lipfert, J. (2017). *J. Appl. Cryst.* **50**, 621–630.
- Skar-Gislinge, N. & Arleth, L. (2011). *Phys. Chem. Chem. Phys.* **13**, 3161–3170.
- Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., Hazlewood, V., Lathrop, S., Lifka, D., Peterson, G. D., Roskies, R., Scott, J. & Wilkins-Diehr, N. (2014). *Comput. Sci. Eng.* **16**, 62–74.
- Willendrup, P. K. & Lefmann, K. (2021). *J. Neutron Res.* **23**, 7–27.