Writing Models in SasView

First Contributing to SasView workshop

Paul Butler
Virtually May 19, 2021

What to expect

We will be using SasView 5.0.4.

No prior experience is required. However a knowledge of SAS would be helpful along with a knowledge of enough maths to write the necessary functions

These few slides are intended to provide a quick orientation on a couple of fundamental aspects and on the outline of the demo tutorial. They are intended to be followed immediately by a live, hands-on, demo/tour using SasView to write models.

SOME FUNDAMENTALS

POSTULATE: creating a new model in SasView is EASY

- There is ZERO difference between built in model files and custom/plugin model files.
  - But there are some differences in how the GUI handles them…
- You do NOT need anything but SasView to create a model
  - Though it may help for complicated things ….
- You must know:
  - The equation you want to use (I(Q) = ….. what?)
  - What are the adjustable parameters in that model
  - …. And then the complexities of polydispersity, orientation, integer parameters, “multiplicity” etc.

MORE FUNDAMENTALS

POSTULATE: creating a new model in SasView is EASY

NOTE: SasView provides “magic” and tools specific to SAS data, and the models and GUI make assumptions about the data being SAS data. *HOWEVER*, fundamentally, SasView can fit/model any data with any equation that can be given analytically (NSE maybe?)
NOTE2: SasView 4.2 is no longer being worked on and will soon be obsolete.

WHAT WE WILL COVER

POSTULATE: creating a new model in SasView is EASY

- The three types of model files that can be used (ALL include *mymodel.py*)
- How to use the built in tools in SasView to do most of the work
- How to create 1D and 2D models and the concept of orientation
- Structure factors vs form factors
- How to make a form factor available to P*S and/or the beta appox.
- How to make the model available immediately to the community

THE PLAN

POSTULATE: creating a new model in SasView is EASY

- Write a simple python model (MODEL TYPE I) with no polydispersity using the Add model function
- Rewrite using special functions (and even other packages, e.g.scipy)
- Add polydispersity and the form_volume function
- Enable use with structure factor (effective radius - deprecated and new)
- Rewrite in C (MODEL TYPE II and III)
- Using (and creating) C library functions
- Using cylinder model to look at orientation.
- Use of various other flags
- MAYBE .. multiplicity and Reparameterization…. A powerful new tool