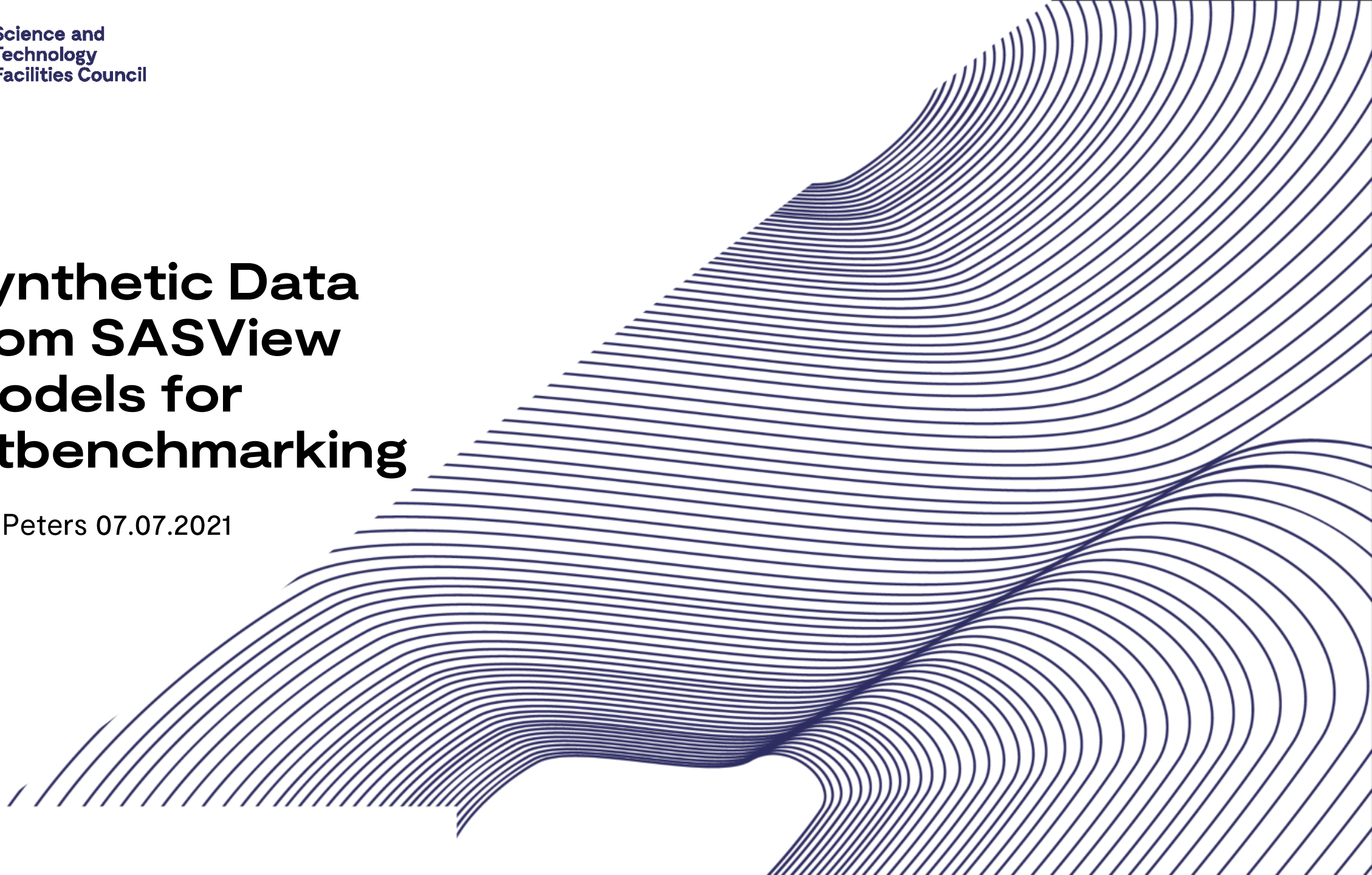
Science and Technology Facilities Council

Synthetic Data from SASView Models for Fitbenchmarking

Ben Peters 07.07.2021

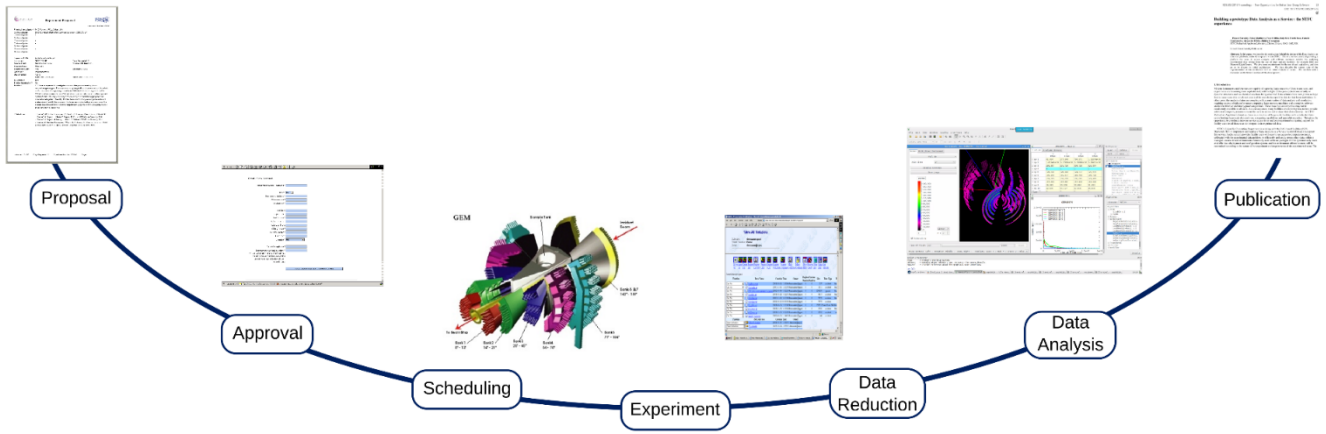


Introduction to FitBenchmarking

- Optimisation is an important and reoccurring step in the workflow from data gathering to analysis.
- FitBenchmarking is an open-source software by STFC for benchmarking optimisation algorithms. It evaluates a set of problems with a set of minimizers. It records, for each minimizer across all problems:
  - Any encountered errors,
  - the runtime,
  - and the accuracy in terms of residuals

to *benchmark* the performance of optimisation software for given problem types.

- The default problem sets in FitBenchmarking are from the *nonlinear regression* problems of the *NIST Standard Reference Database*.



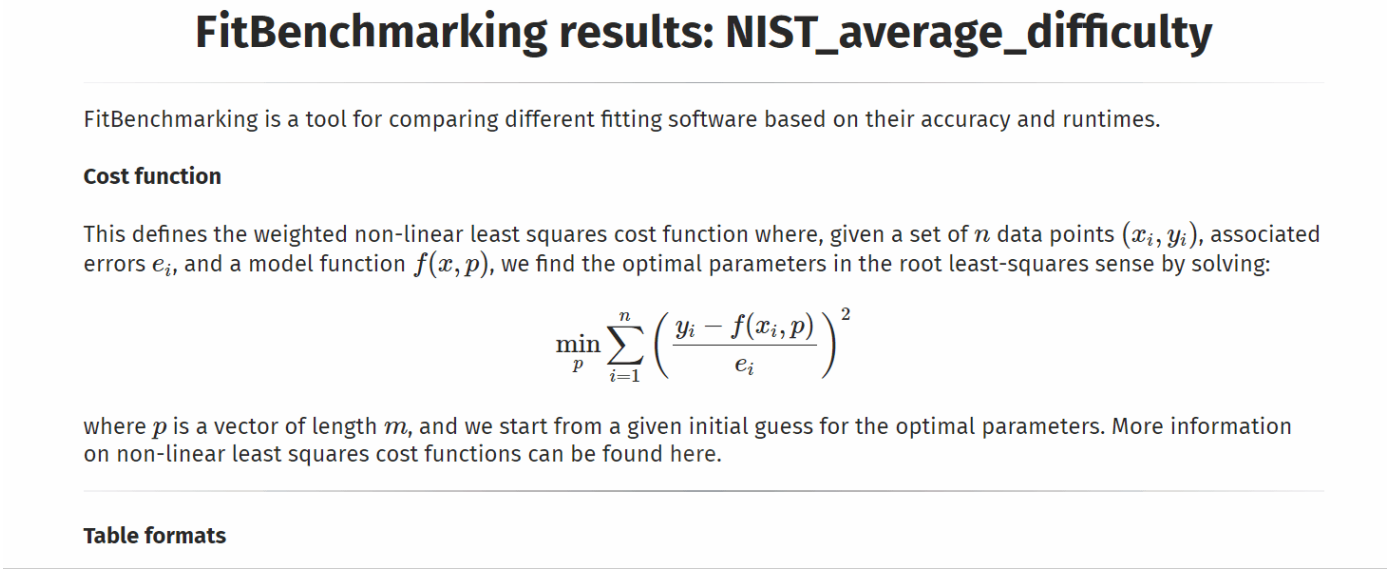
Science and Technology Facilities Council

Introduction to FitBenchmarking

- Optimisation is an important and reoccurring step in the workflow from data gathering to analysis.
- FitBenchmarking is an open-source software by STFC for benchmarking optimisation algorithms. It evaluates a set of problems with a set of minimizers. It records, for each minimizer across all problems:
  - Any encountered errors,
  - the runtime,
  - and the accuracy in terms of residuals

to *benchmark* the performance of optimisation software for given problem types.

- The default problem sets in FitBenchmarking are from the *nonlinear regression* problems of the *NIST Standard Reference Database*.



FitBenchmarking results: NIST_average_difficulty

FitBenchmarking is a tool for comparing different fitting software based on their accuracy and runtimes.

Cost function

This defines the weighted non-linear least squares cost function where, given a set of n data points (x_i, y_i), associated errors e_i, and a model function $f(x, p)$, we find the optimal parameters in the root least-squares sense by solving:

$$\min_p \sum_{i=1}^{n} \left(\frac{y_i - f(x_i, p)}{e_i} \right)^2$$

where p is a vector of length m, and we start from a given initial guess for the optimal parameters. More information on non-linear least squares cost functions can be found here.

Table formats

Introduction to FitBenchmarking

- Optimisation is an important and reoccurring step in the workflow from data gathering to analysis.
- FitBenchmarking is an open-source software by STFC for benchmarking optimisation algorithms. It evaluates a set of problems with a set of minimizers. It records, for each minimizer across all problems:
  - Any encountered errors,
  - the runtime,
  - and the accuracy in terms of residuals

to *benchmark* the performance of optimisation software for given problem types.

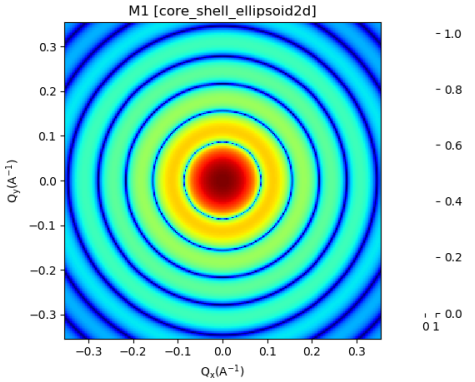
- The default problem sets in FitBenchmarking are from the *nonlinear regression* problems of the *NIST Standard Reference Database*.

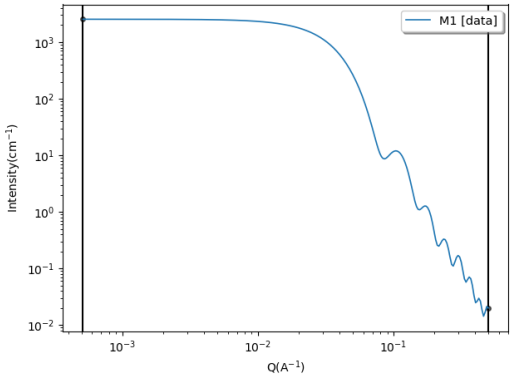
Dataset Name	Level of Difficulty	Model Classification	Number of Parameters	Number of Observations	Source
Misra1a	Lower	Exponential	2	14	Observed
Chwirut2	Lower	Exponential	3	54	Observed
Chwirut1	Lower	Exponential	3	214	Observed
Lanczos3	Lower	Exponential	6	24	Generated
Gauss1	Lower	Exponential	8	250	Generated
Gauss2	Lower	Exponential	8	250	Generated
DanWood	Lower	Miscellaneous	2	6	Observed
Misra1b	Lower	Miscellaneous	2	14	Observed
Kirby2	Average	Rational	5	151	Observed
Hahn1	Average	Rational	7	236	Observed
Nelson	Average	Exponential	3	128	Observed
MGH17	Average	Exponential	5	33	Generated
Lanczos1	Average	Exponential	6	24	Generated
Lanczos2	Average	Exponential	6	24	Generated
Gauss3	Average	Exponential	8	250	Generated
Misra1c	Average	Miscellaneous	2	14	Observed
Misra1d	Average	Miscellaneous	2	14	Observed
Roszman1	Average	Miscellaneous	4	25	Observed
ENSO	Average	Miscellaneous	9	168	Observed
MGH09	Higher	Rational	4	11	Generated
Thurber	Higher	Rational	7	37	Observed
BoxBOD	Higher	Exponential	2	6	Observed
Rat42	Higher	Exponential	3	9	Observed
MGH10	Higher	Exponential	3	16	Generated
Eckerle4	Higher	Exponential	3	35	Observed
Rat43	Higher	Exponential	4	15	Observed
Bennett5	Higher	Miscellaneous	3	154	Observed

Science and Technology Facilities Council

SasView in FitBenchmarking

- The capability for parsing Sasview problems was introduced to FitBenchmarking before I joined.
- However, there were no example Sasview problem sets included in FitBenchmarking.
- I chose to introduce the following datasets:
  - Simple
  - Moderate
  - Complex
  - Shape Independent





Model

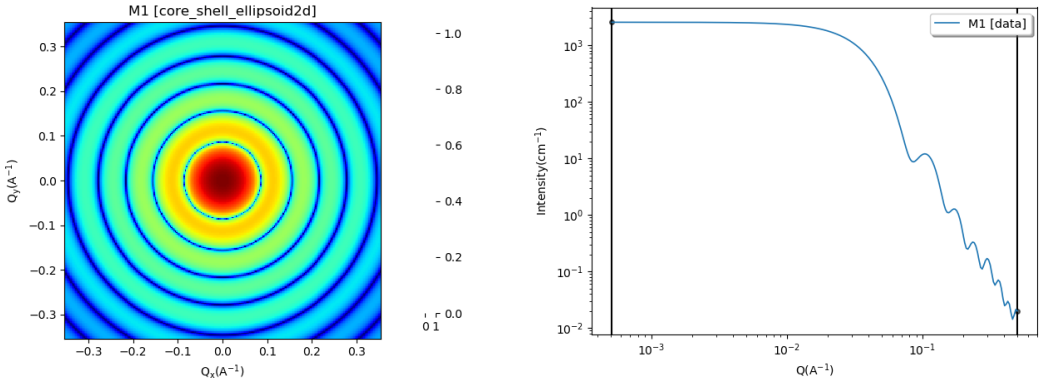
Category: Ellipsoid | Model name: core_shell_ellipsoid | Structure factor: None

Parameter	Value	Min	Max	Units
scale	1.0	0.0	∞	
background	0.001	-∞	∞	cm^{-1}
core_shell_ellipsoid				
radius_equat_core	20	0.0	∞	Å
x_core	3	0.0	∞	None
thick_shell	30	0.0	∞	Å
x_polar_shell	1	0.0	∞	
sld_core	2	-∞	∞	10^{-6}/Å^2
sld_shell	1	-∞	∞	10^{-6}/Å^2
sld_solvent	6.3	-∞	∞	10^{-6}/Å^2

Science and Technology Facilities Council

SasView in FitBenchmarking

- The capability for parsing Sasview problems was introduced to FitBenchmarking before I joined.
- However, there were no example Sasview problem sets included in FitBenchmarking.
- I chose to introduce the following datasets:
  - Simple
  - Moderate
  - Complex
  - Shape Independent



Model

Category	Model name	Structure factor
Ellipsoid	core_shell_ellipsoid	None

Parameter	Value	Min	Max	Units
scale	1.0	0.0	∞	
background	0.001	-∞	∞	cm^{-1}
core_shell_ellipsoid				
radius_equat_core	20	0.0	∞	Å
x_core	3	0.0	∞	None
thick_shell	30	0.0	∞	Å
x_polar_shell	1	0.0	∞	
sld_core	2	-∞	∞	10^{-6}/Å^2
sld_shell	1	-∞	∞	10^{-6}/Å^2
sld_solvent	6.3	-∞	∞	10^{-6}/Å^2

Science and Technology Facilities Council

SasView in FitBenchmarking

- The capability for parsing Sasview problems was introduced to FitBenchmarking before I joined.
- However, there were no example Sasview problem sets included in FitBenchmarking.
- I chose to introduce the following sasmodel problem sets to FitBenchmarking:
  - Simple
  - Moderate
  - Complex
  - Shape Independent

Problem Set	Sasmodels
Simple Shapes	Cylinder, Sphere, Ellipsoid
Moderate Shapes	Core Shell Cylinder, Core Shell Sphere, Stacked Disks, Core Shell Ellipsoid
Complex Shapes	Core Shell Parallelepiped, Flexible Cylinder Elliptical, Parallelepiped, Triaxial Ellipsoid
Shape Independent	Broad Peak, DAB, Line, Lorentz

ERU

Science and Technology Facilities Council

Making Problems from Sasmodels

Types of data:

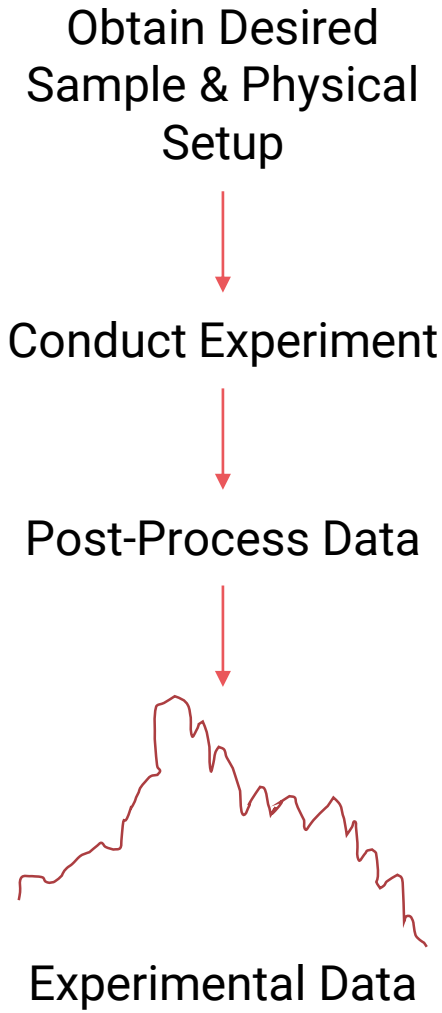
- Real
  - From an experiment including processed experimental data.
- Simulated
  - Generated following the physics of a real experiment.
- Model
  - Idealistic data generated from an equation based on an idea of how a real dataset might behave.
- Synthetic
  - Perturbed model data or otherwise "realistic" data.

Why Synthetic Data?

- Ideally there would be *real data* for each model, but it is not feasible to use beam time just to create FitBenchmarking problem sets.

- Simulated data may be preferable to synthetic but require more physics expertise and set-up effort; especially since 15 models were to be used (setting up 15 different simulations).

- Synthetic data are simple because they start with the underlying model and apply a noise filter, a process which can be done very quickly to a large number of models.

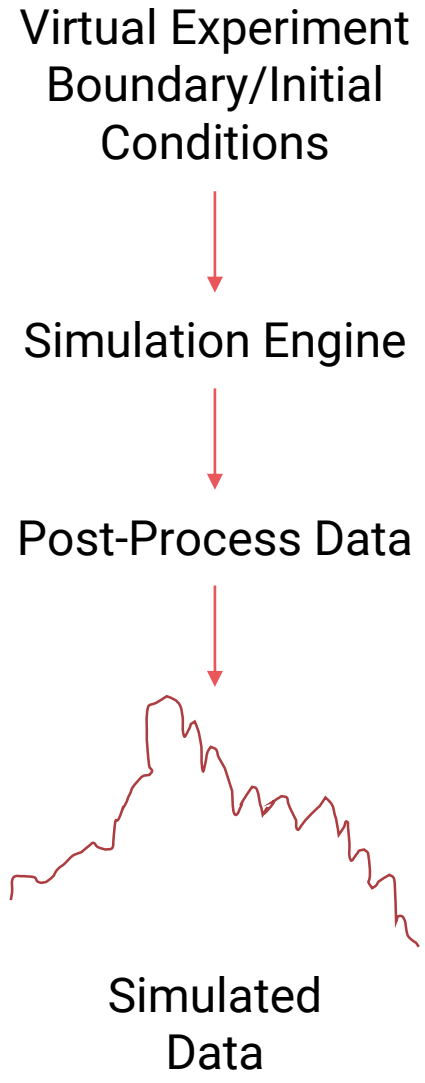


Why Synthetic Data?

- Ideally there would be experimental data for each model, but it is not feasible to use beam time just to create FitBenchmarking problem sets.

- Simulated data can be the next most physical data but require more physics expertise and set-up effort; especially since 15 models were to be used (setting up 15 different simulations).

- Synthetic data are simple because they start with the underlying model and apply a noise filter which is based on experimental data, a process which can be done very quickly to a large number of models.

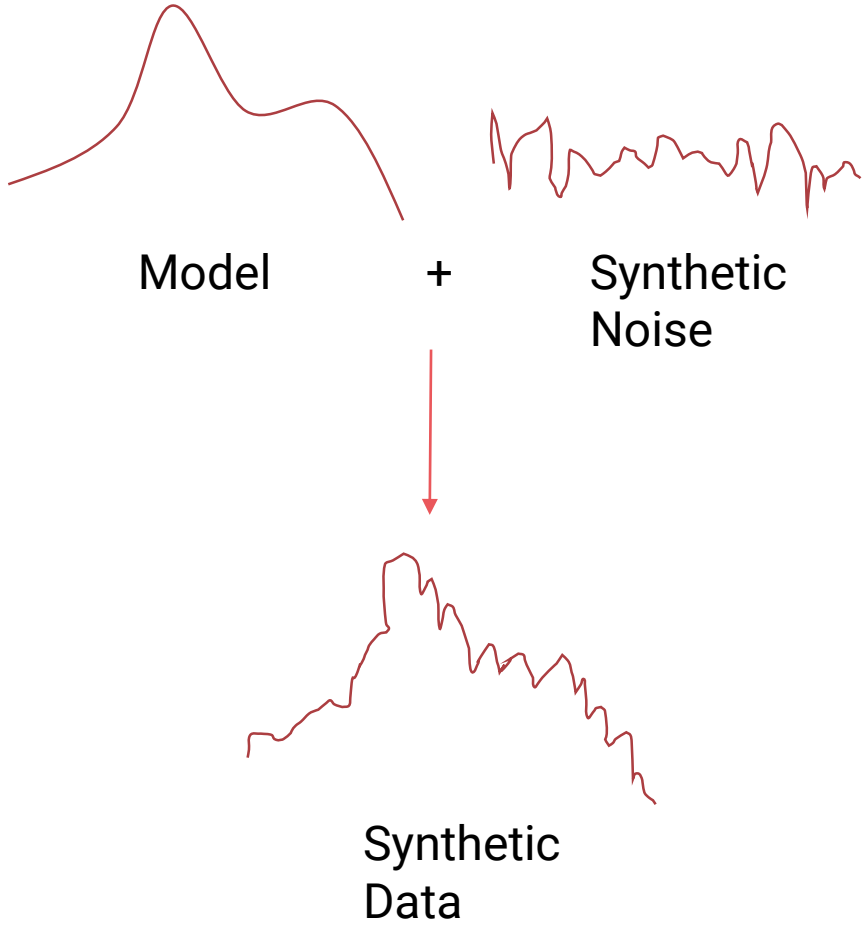


Why Synthetic Data?

- Ideally there would be experimental data for each model, but it is not feasible to use beam time just to create FitBenchmarking problem sets.

- Simulated data may be preferable to synthetic but require more physics expertise and set-up effort; especially since 15 models were to be used (setting up 15 different simulations).

- Synthetic data are simple because they start with the underlying model and apply a noise filter, a process which can be done very quickly to a large number of models.



Synthetic Problems from Sasmodels

1. Generate model data for 1D Q vs Intensity from sasmodels
   a) Set model parameters

   b) Set number of observations and spacing

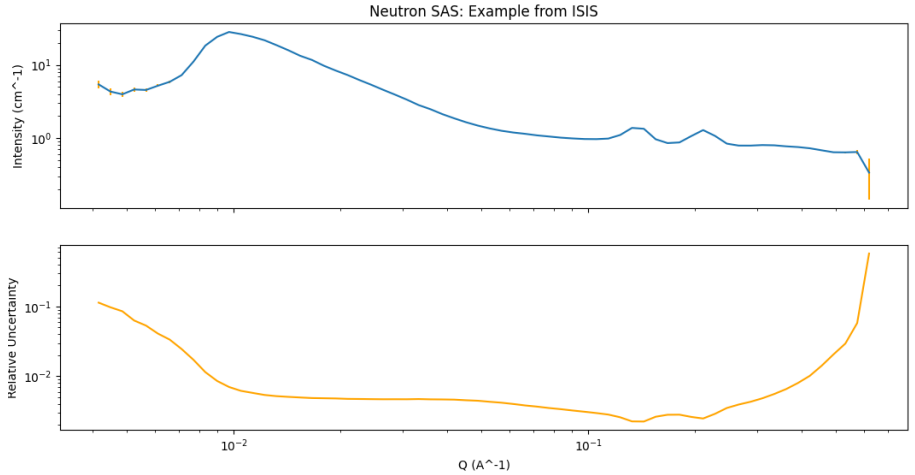
2. Deploy noise generation algorithm based on a real SAS experiment.
   a) From the sample experimental data, how can the uncertainty u be characterised?
      Find $u = f(x, y)$
   b) Gaussian sampling used at the model x positions, from a distribution with $\sigma(x) = u(x, y)$.
3. Write a problem definition file to be parsed by fitbenchmarking.
   a) Define initial parameter values

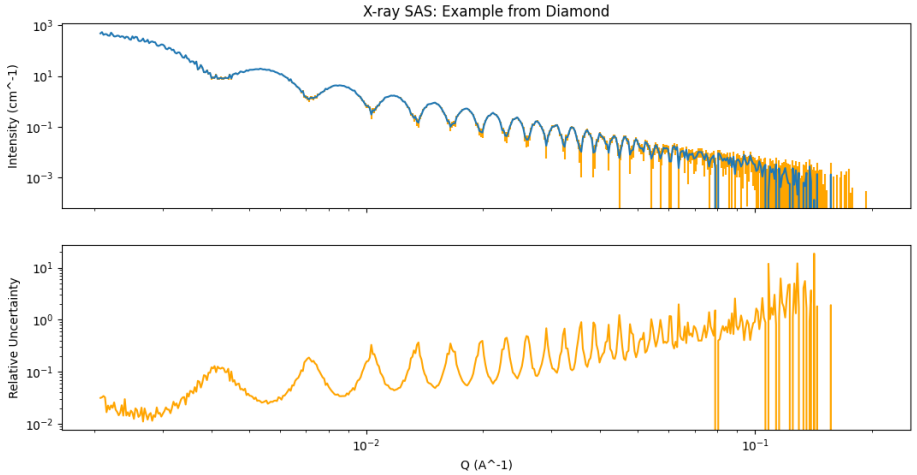
Synthesis for SAXS & SANS

- Small Angle Neutron Scattering (SANS):

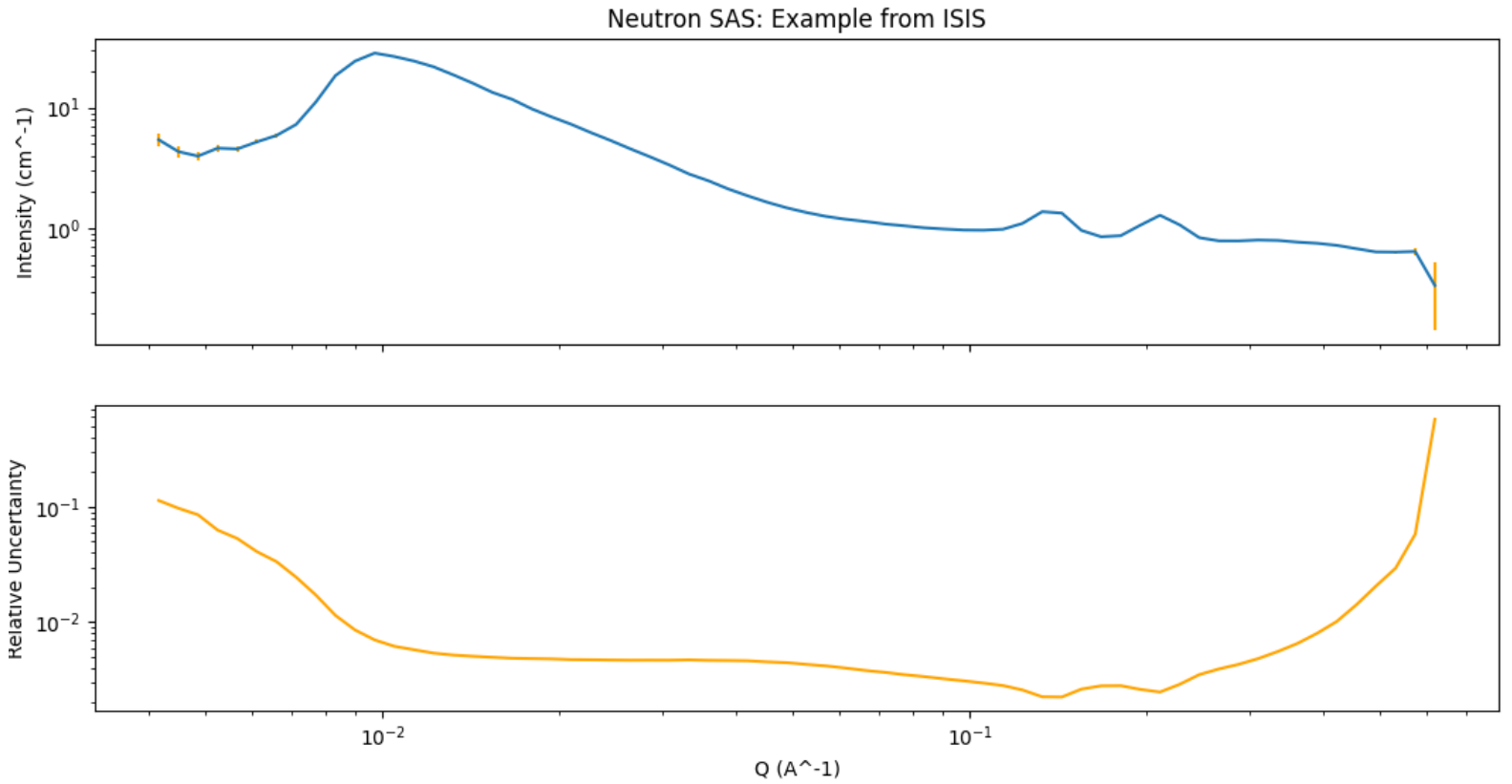
  33837rear_1D_1.75_16.5_NXcanSAS.h5

- Small Angle X-Ray Scattering (SAXS):





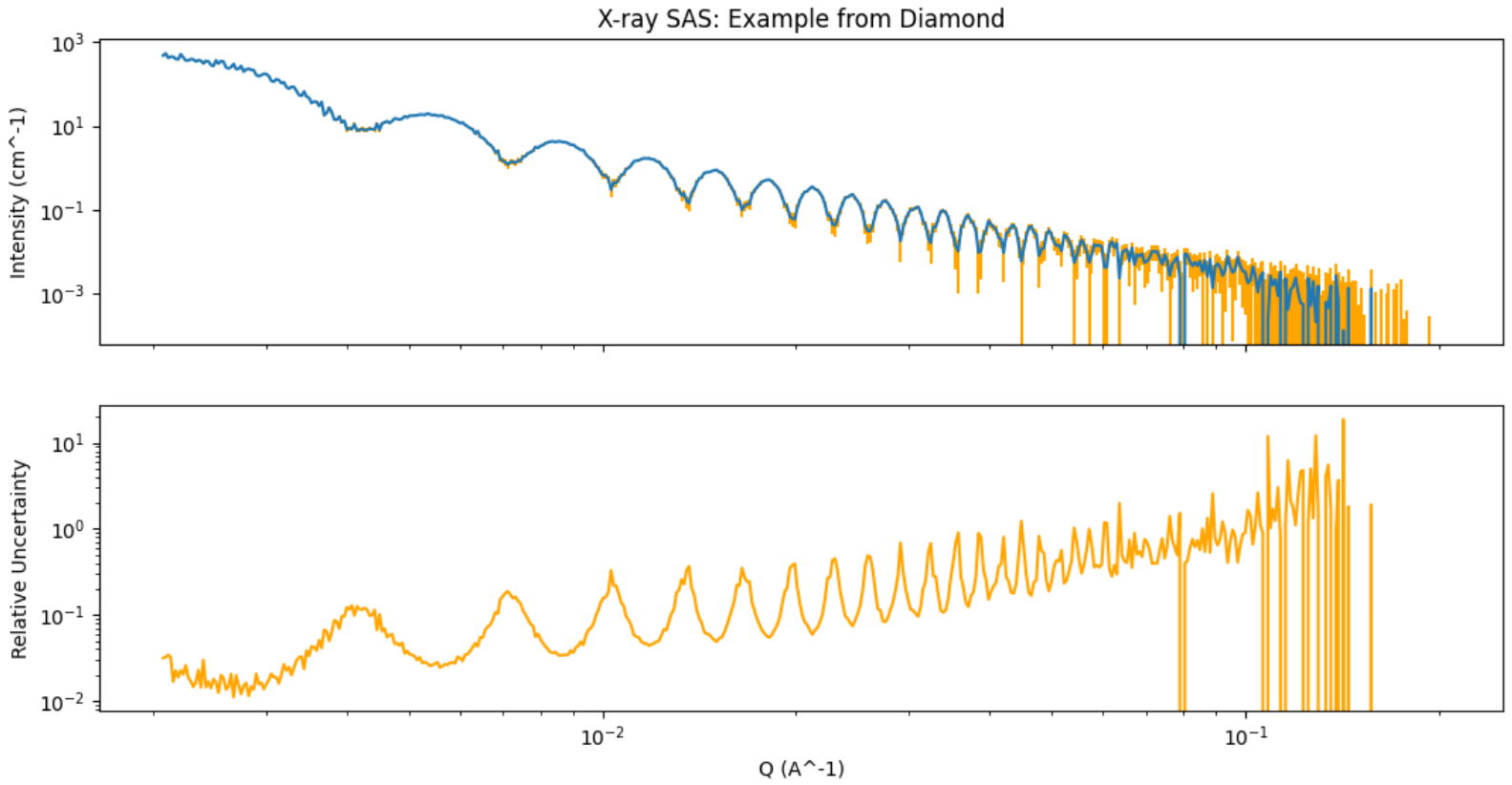
Synthesis for SANS



33837rear_1D_1.75_16.5_NXcanSAS.h5

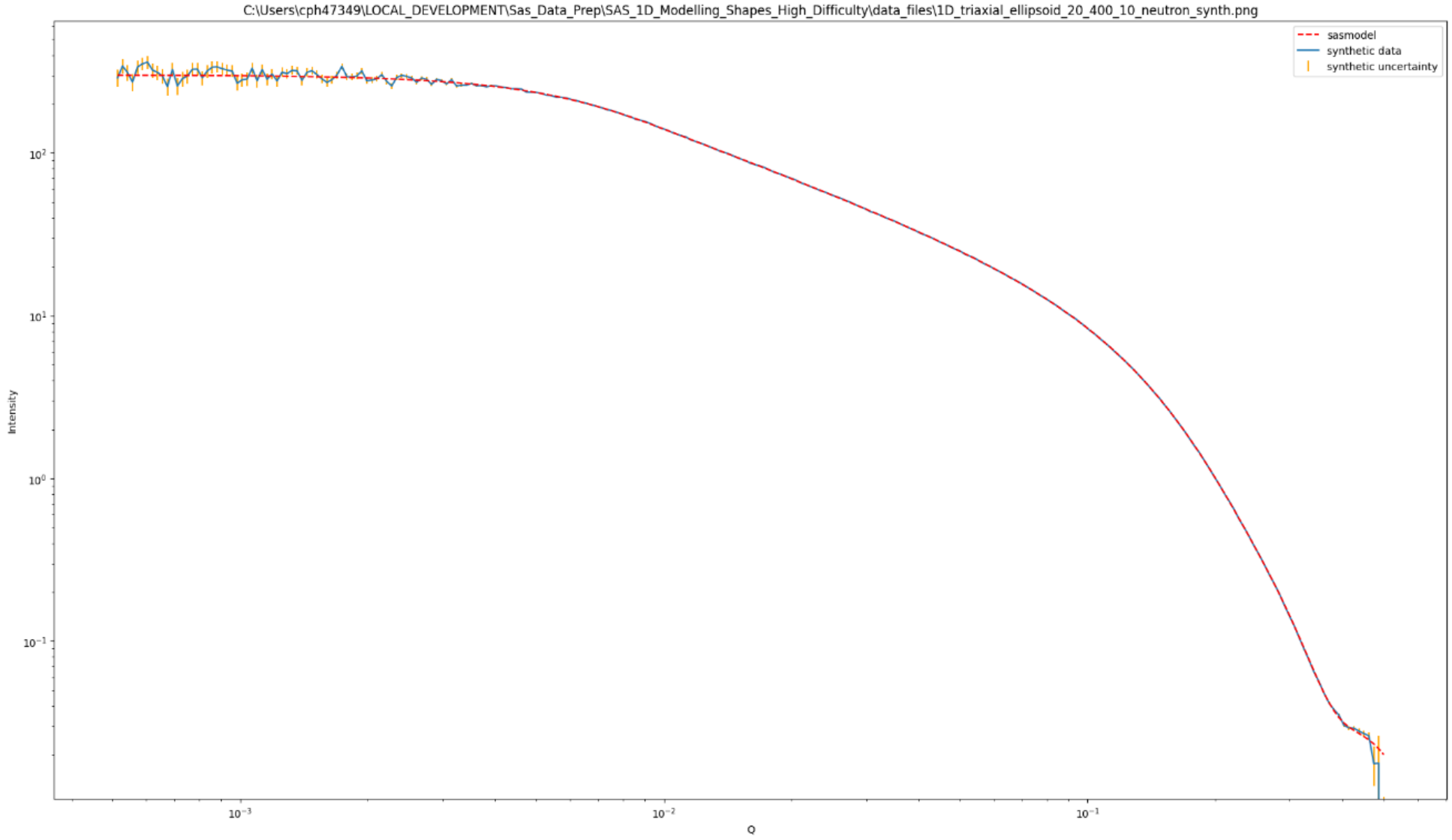
Science and Technology Facilities Council

Synthesis for SAXS



Science and Technology Facilities Council

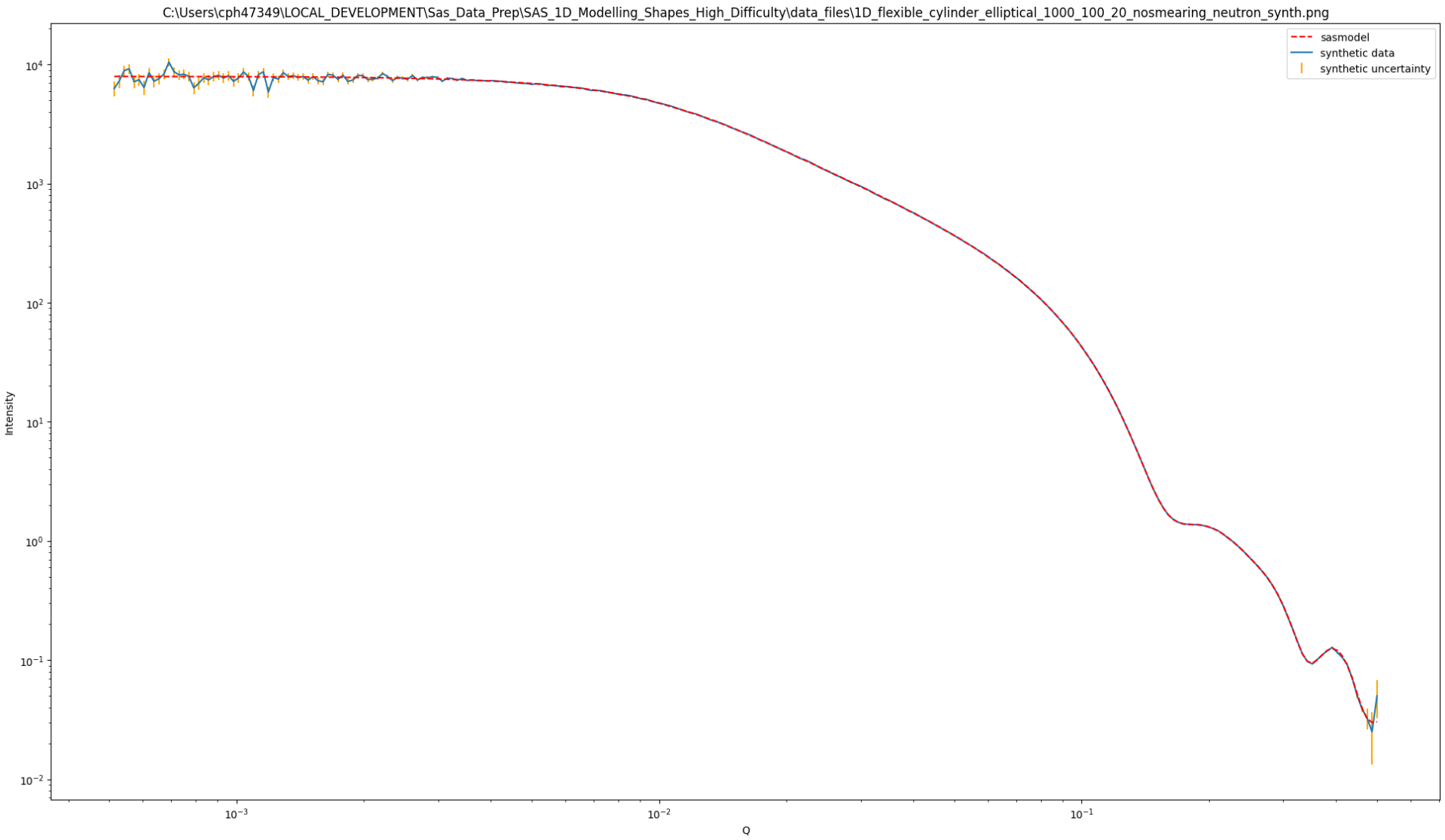
SANS Synthesis



Triaxial Ellipsoid

256 log-spaced datapoints

SANS Synthesis



Flexible cylinder elliptical

256 log-spaced datapoints

Science and Technology Facilities Council

SANS Synthesis - Results

Show fitbenchmarking results page for moderate problem set…

- It remains a work in progress, thanks for listening

Science and Technology Facilities Council

SAXS Synthesis