# The Parameter File

An example of a *sas\_temper* configuration file, which is a YAML file, is shown below that presents many of the various options that can be used when configuring the program. A full description of all options for configuring the model is also provided.

file:

name: input\_data.txt

qmin: 0.01

qmax: 0.30

output\_files: cs\_fit\_sphere

sa\_parameters:

temperatures: 100

temperature\_rate: 0.90

parameter\_rate: 0.95

iterations: 1000

models\_to\_generate: 3

model:

name: core\_shell\_sphere

Category: shape-sphere

scale:

fixed: [1.0]

background:

fixed: [0.001]

sld\_core:

linear: [-0.56, 8.00]

sld\_shell:

linear: [-0.56, 8.00]

sld\_solvent:

linear: [-0.56, 6.38]

radius:

log: [1.0, 3.0]

polydispersity:

SchulzDispersion: [0.0, 0.40]

thickness:

log: [0.0, 2.0]

coupled: [radius]

polydispersity:

SchulzDispersion: [0.0, 0.40]

Structure\_Factor:

name: hardsphere

volfraction:

linear: [0.0, 0.50]

The first block of the sas\_temper configuration file specifies the data to be modeled as well as the q-range over which to perform the fitting.

file:

name: input\_data.txt

qmin: 0.01

qmax: 0.30

The data is assumed to be 3-, 4-, or 6-column ASCII text. Any metadata should be demarked as such with a non-numerical character at the start of the line. The program does not use any metadata present int he data files. The program will not accept data with other numbers of columns, or data in a different format. The columns, which are tab or space speparated contain the following information. The 6-column format contains the data provided in the format that is output by the NIST Igor data reduction routines. The ORNL data reduction routines, *drt-sans*, produce data in the 4-column format.

**column 1 column 2 column 3 column 4 column 5 column 6**

Q I dI dQ () Q-bar shadow factor

It is generally assumed that the units of Q are Angstroms-1, while I is assumed to be provided in absolute units of cm-1. Data that have not been scaled into absolute units can still be fit, but the scale factor cannot be fixed to 1.0. Data provided with Q specified in nm-1 can also be fit as long as length parameters are specified in nm and understood to be returned in the same.

The second block of parameters, which is the simplest of the set, specifies the name of the files that are output by the program.

output\_files: fit\_cs\_sphere

All of the results that are output by sas\_temper will be found in the directory in which the program is run and will be named “fit\_cs\_sphere\*.\*”. The results will either be ASCII text, png images. The model profiles resulting from the fits are output as 2-column, tab separated ASCII text. Comments and other metadata in the output data are denoted by having the line in the file start with the character “#”.

In the third block of the configuration file, the parameters for configuring the simulated annealing engine used in *sas\_temper* are specified.

sa\_parameters:

temperatures: 100

temperature\_rate: 0.90

parameter\_rate: 0.95

iterations: 1000

models\_to\_generate: 3

The temperatures parameter specifies the number of times that the program will decrease the temperature in the simulated annealing process. Each time the temperature decreases, it is multiplied by the temperature\_rate parameter, which must be a number 0.0 < temperature\_rate < 1.0. The parameter\_rate parameter controls how the range of possible values of the various model parameters constricts during the five refinement fittings performed automatically by *sas\_temper*. Acceptable values are 0.0 < parameter\_rate ≤ 1.0. The iterations parameter specifies how many models are tested per temperature setting for a given fitting performed. In the example above, a total of 100,000 models will be tested during each of the five fittings performed (i.e. a total of 500,000 intensity profiles will be calculated) to produce a single fit to the data that is returned to the user as a result. The final parameter above, models\_to\_generate, specifies how many different results are returned to the user before *sas\_temper* exits. The number has to be at least 1, while the upper limit is mostly dictated by the patience of the user. With a sufficiently large number (> 20), comparing the results from two different data sets becomes easier and can be subjected to statistical tests to provide a measure of whether the results truly differ or not.

The final block of parameters is specific to the model that the data will be fit with. A complete list of all models implemented in the *sasmodels* package and possible parameters are provided in a separate document (models\_and\_parameters.yaml) that can be found in the same directory of the code repository as this document. Rather than describing each parameter, the various keywords for the parameters are described below.

Parameters that are numbers are specified using four possible keywords: fixed, linear, log and integer.

fixed: [*value*]

The value of the parameter is always set to *value* throughout the modeling process.

linear: [*min\_value*, *max\_value*]

The parameter is selected from a uniform distributrion from *min\_value* ≤ parameter ≤ *max\_value*. Using a linear range is best when the value is reasonably well known and can be constrained, such as for a scattering length density.

log: [*min\_value*, *max\_value*]

First, a value *X* is selected from a uniform distributrion from *min\_value* ≤ *X* ≤ *max\_value*. Then, the value of the parameter is set to 10X. Using a logarithmic range for an initial fitting of data with an unknown structure is a good way to avoid spending a great deal of time engaging in trial-and-error searches for a good starting point.

integer: [*min\_value*, *max\_value*]

There are parameters within the sasmodels model set that use integers to pick from a set of options for the model profile calculation. Here, an integer value is selected from a uniform distribution within *min\_value* ≤ parameter ≤ *max\_value*.

There are other options that are modifiers on the parameters used for the calculations.

coupled: [*paramerer\_name*]

While not intrinsic to *sasmodels* or *SasView*, *sas\_temper* makes it possible to couple the value of one parameter to another parameter. Here, the value of the parameter is coupled to another that has the name *parameter\_name*. When coupled, the value *V* selected for the parameter is used to calculate the value *Z* that is used in the model intensity profile calculation using parameter\_name’s value, denoted here as W. Specifically, *Z* = *VW*.

polydispersity:

*Distribution\_name*: [*min\_value*, *max\_value*]

Many of the parameters of the models implemented in sasmodels can be made polydisperse. These parameters can also have a polydispersity applied to them in *sas\_temper*. The built-in mechanisms for the calculations implemented in *sasmodels* are employed in *sas\_temper*. Four options are available for the kind of polydispersity distribution employed when modeling with *sas\_temper* that are specified as the *Distribution\_name* parameter: *SchulzDistribution*, *GaussianDistribution*, *LogNormalDistribution* and *RectangleDistribution*. The width of the distribution, *w*, applied during the calculation is selected from a uniform distribution *min\_value* ≤ *w* ≤ *max\_value*.

Structure\_Factor:

The structure factors that are implemented in *sasmodels*, which can be found in the documentation at <http://www.sasview.org/docs/user/qtgui/Perspectives/Fitting/models/index.html>, are implemented in *sas\_temper* as a model within a model. Parameters for the structure factors function exactly as they do for any parameter of a regular model.