Fitting Manual

# Github Repository

A repository with all needed files and scripts can be found at [www.github.com/criosx/reflectometry](http://www.github.com/criosx/reflectometry).

# Running Fits

## Requirements and Initialization

The molgroups.cc libraries that support composition space modeling for reflectometry are written in C++ for garefl models. Hence, to use composition space modeling the model has to be compiled with garefl and the model has to be passed to refl1d for parameter optimization. Therefore, the list of files required for such a fit is:

* garefl setup.cc file containing the model
* garefl Makefile
* rs.py as a wrapper script for refl1d
* run.py start script for refl1d
* general.py for some post-processing tasks

After compiling setup.cc using

* make clean
* make

run the garefl optimizer for a few iterations using

* ./fit

This will initialize needed garefl .dat files providing rs.py with necessary information about the model. The fit can be aborted after a few generations

## Testing a garefl composition space model fit for running under refl1d

A fit can be tested either using the garefl optimizer (genetic algorithm) or the refl1d optimizer (DREAM, differential evolution Monte Carlo Markov Chain). For the first option use

* ./rs.py –a

The second option requires

* refl1d run.py –fit=dream –init=lhs –parallel –store=S1 –burn=1000 –steps=500

Burn and step numbers can be varied to adjust the run time. The above command uses the dream optimizer and parallel execution of the fit on all available cores. The save directory is S1.

The goal of testing a fit is to check whether the model is able to describe the data, and whether the boundaries of the fit parameters are appropriately set.

## Running a garefl composition space model fit under refl1d (MCMC statistics)

A refl1d Monte Carlo Markov Chain fit can be started using rs.py:

* ./rs.py –MCMC

The default number of steps in the MCMC chain is 64000, which is the default for composition space spline fits. For orientation fits 256,000 or 512,000 steps are recommended using

* ./rs.py –MCMC 512000

## Probability Plots Orientation Fits

After the MCMC fit finished, some post-processing is required to create probability plots of the protein orientations. Most likely the by rs.py automatically created statistical file sErr.dat is too sparse for orientation plots. Therefore, first a full statistical file has to be created using the following commands:

* cp –r MCMC\_\* MCMC
* rm sErr.dat
* ./rs.py –stat –conf -1

A sparse sErr.dat can be recreate at any time by using the same sequence of commands and appending the option –sparse 0.001 to the end of the rs.py call. This will reduce the number of MCMC steps included in sErr.dat by a factor of 1000.

After creating a full sErr.dat, orientation probability plot data can be extracted using general.py.

* ipython
* import general
* general.fnCreateProbabilityPlot()

This creates a number of files, including sErr.dat\_betapos\_prob.dat and sErr.dat\_betaneg\_prob.dat. The former contains plot data for 0º ≤ β ≤ 90º, and the latter for 90º < β ≤ 180º. Most conveniently the data is displayed by importing it into an existing Orientation.pxp Igor file. When importing the data, select ‘Load Waves > Load Waves …’. Select ‘General Text’, ‘Overwrite existing waves’, ‘Load columns into matrix’, and enter into the base field ‘betaposprob’ or ‘betanegprob’, respectively.