GP and Porod Python Analysis Scripts Guide

By Zachary Amato: SANDALS Instrument Scientist at the ISIS Neutron and Muon Source

The foundation of these analysis scripts was created by previous SANDALS instrument scientist Dr Sabrina Gärtner, to be used solely for the analysis of growing and heating ASW on NIMROD (made under interstellar conditions). I decided to alter them for general use by any user of NIMROD that looks at porous materials and wants to analyse the SANS region. These scripts will fit both a Porod and Guinier-Porod (GP) analysis for as many files as are indicated.

These are the files you should have:

1. Input_Parameters_Python_Analysis.csv
2. results_name_sequence/sample.txt
3. run_python_analysis.py
4. plot_python_analysis_results.py
5. plot_settings.py
6. CLASSES.py
7. analysis_input_file_handling.py
8. Guinier_Porod.py
9. Porod_Constant.py

1 Setting Up

All files must be put in the same directory that you are working from, along with added folders called 'Gudrun_results' and 'python_analysis'. The Gudrun_results folder must have all your mint files within it and the python_analysis folder should be empty as this is where these scripts will place all created output files.

To set up the results file and its name, you must ascertain whether your data evolves as a function of a certain parameter, or if it is separate samples. As in, when surface area is plotted, what do you want on the x axis? Do you want it either as a function of a parameter (e.g. time) or the sample names on the x axis. What you choose for this will determine what you have to call your results file (write sequence or sample accordingly, where indicated above). The example results file is for a sample called ASW and it is as a function of temperature so the file name is written as 'results_ASW_sequence.txt'. The contents of the results file must be in this order: file number (e.g. NIMROD00034903 is written only as 34903) and parameter/sample names. For example, the example results file has the file names alongside their respective temperature.

The script to run all of the analysis is run_python_analysis.py. If using IDaaS (ISIS Data Analysis as a Service), set off the scripts by typing "python3 run_python_analysis.py" in the terminal.

The Input_Parameters_Python_Analysis.csv excel sheet outlines important information to be used by the scripts, as well as lets you modify the model fittings (explained further later on). You will see at the top left that the working and results directories, run, sample and action are defined. The working directory is set as where you are calling the analysis scripts from. The sample and action parameters are important as they set out how the python scripts find your results file and whether you are analysing scripts as independent samples or as a function of a parameter. If you are analysing as a function of a parameter, set action as 'sequence', and 'sample' if not. When writing which files for the scripts to manually set start parameters, you only need to write the numbers of the file (e.g. NIMROD00034903 is written only as 34903).

For the GP fitting, the scripts will use the output of the previous fit as the starting point for the next. There is the option to start fitting from either the first files or the final. This is done by setting reverse_T to either

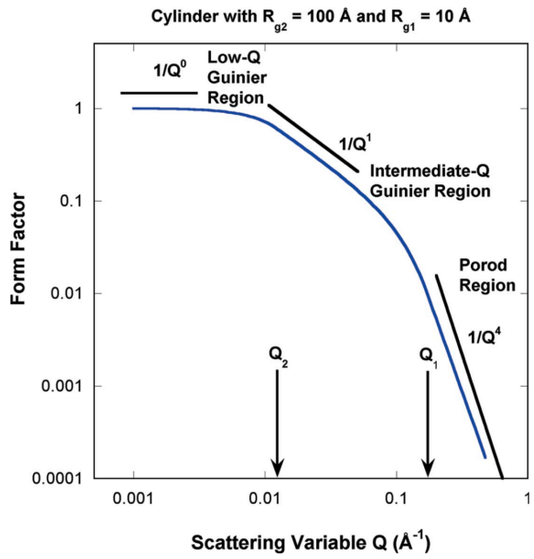

Figure 1: Form factor for a cylinder showing the low-Q Guinier region, the intermediate-Q Guinier region and the high-Q Porod region. Taken from[3].

true or false at line 780 of the Guinier_Porod.py script. Setting it as true means that the scripts will start at the last file. This is useful if your statistics are much better with later samples, as in when you may be growing a sample *in situ* for example.

If you have chosen sequence (results as a function of a parameter), you will need to change the axes labels of the output plots to the name of your parameter. This is done in plot_python_analysis_results.py at line 99, 101 and 247 where you can see "Parameter". Script plot_settings.py also needs to be changed at line 86 where the colorbar label is set to "Parameter" by default (can change the colorbar range and tick numbers here).

2 Porod Model

Scattering measurements at large values of Q in the SANS region provide information on a surface's fractal dimension. The fractal dimension D characterizes the complexity of a geometric shape's structure (pores in this case)[5]. It was found by[4] that for large values of Q where the probed length scales are much smaller than the size of the particle, the scattering intensity should decay as $\mathrm{I(Q)} \sim Q^{-4}$ due to scattering from the sharp, smooth interface that separates the particles from its environment[1,3]. This 'Porod region' is seen in Figure 1.

It can be generalised to rough or broad interfaces whereby $I(Q) \sim Q^{-(6-D_s)}$ (D_s is the fractal dimension of the surface)[5]. For a smooth surface, $D_s = 2$ and this allows Porod's original Q^{-4} to be recovered. If the surfaces are rough, D_s is between 2 and 3 and that leads to the scattering intensity decaying with an exponent between -3 and -4 in the surface fractal regime. Analysing these surface fractal dimensions is usually referred to as Porod analysis.

In Porod analysis, the specific surface area (SSA), measuring the surface area per ice volume, is related to the Porod constant (K) such that:

$$SSA = \frac{\lim_{Q\to\infty}(I(Q).Q^4)}{2\pi\Delta\rho^2} = \frac{K}{2\pi(\Delta\rho)^2}, \tag{1}$$

where $\Delta\rho$ is the difference in scattering length density between the pore and the solid[2]. This, along with the atomic number density, will need to be changed in the CLASSES.py sript at line 98 as it is currently set to the value for D_2O by default.

The constant K, can be found from the intercept of the quasi-plateau of an $I(Q).Q^4$ plot, as shown in Figure 2. To do this, $I(Q)$ is simply multiplied by Q^4 and the Python analysis script then selects the linear region by computing the rate of change of $I(Q) \cdot Q^4$ and defines that region as the area where the rate of

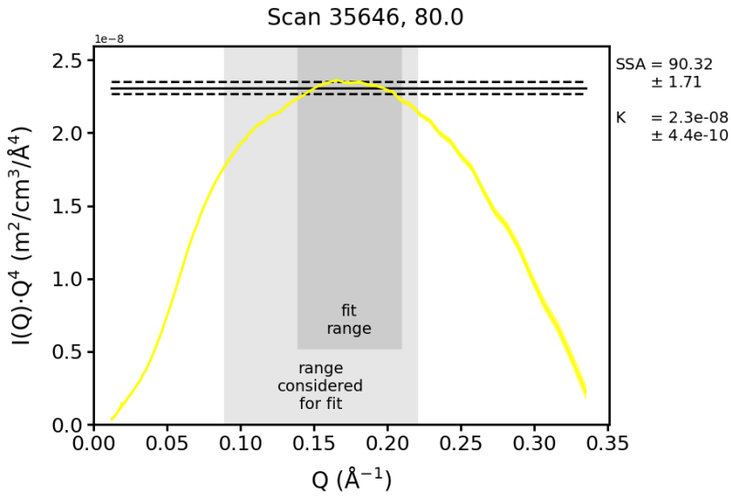

Figure 2: Example quasi-plateau in $I(Q).Q^4$ plot. Solid line indicates midpoint of linear region which is chosen as the Porod constant. From scan 35646 of a sample at 80 K.

change falls below a certain value. This cut-off point is dependent on the gradient of the linear region (lower point when there's a lower gradient). Setting this point is a matter of trial and error. Once the program has extracted a Porod constant, the plots should be visually examined to check that the right linear region has been chosen. If this is not the case, alter it by adding different 'q fit range min' and 'q range max' in the Porod section of the Input_Parameters_Python_Analysis.csv excel sheet. With this, you will need to set a number of points to fit over, where a value of 10 is typically acceptable. Once altered, re-run and continue this iterative process until the fit is satisfactory. With the Porod constant correctly extracted, Equation (1) is used to then calculate the SSA.

3 Guinier-Porod

For information about pore structure, the Guinier-Porod model is commonly used and can be fitted to the SANS curve to extract information about pore size, size and surface roughness:

$$I(Q) = \frac{G}{Q^s} \exp\left(\frac{-Q^2 Rg^2}{3-s}\right) \quad \text{for} \quad Q \leq Q_1, \tag{2}$$

$$I(Q) = \frac{D}{Q^d} \quad \text{for} \quad Q \geq Q_1, \tag{3}$$

$$Q_1 = \frac{1}{Rg}\sqrt{\frac{(d-s)(3-s)}{2}}, \tag{4}$$

where Q and D are scaling factors, Rg is the radius of gyration, s is related to pore shape ($s = 0$ indicates spheres, $s = 1$ indicates cylinders and $s = 2$ indicates platelets), and d is the Porod exponent related to surface roughness (d= 3 for very rough surfaces, $d = 4$ for very smooth surfaces and $d \geq 4$ for diffuse surfaces). An example of such fitting is shown in Fig.3. The python scripts have set default parameters for where to fit, which can all be changed either by using Input_Parameters_Python_Analysis.csv or by changing the default parameters in CLASSES.py.

The Input_Parameters_Python_Analysis.csv excel sheet gives you the option to give the scripts different starting parameters than the default set in CLASSES.py for the GP model fitting. You can do this by filling in the 'manual start' section for each parameter. This is useful if you can see that the fit was not quite right so you need to nudge it in the right direction. If simply just setting new start parameters does not work, then you can force the fit using the sections titled as 'manual fit' for each parameter. If forcing a fit, you will need to set a 'manual fit uncertainty', wherein 5 % is typically acceptable. If you do not want to do this, just leave all the options blank as it will use default numbers set out in CLASSES.py.

References

[1] Boothroyd, A. 2020, *Principles of Neutron Scattering from Condensed Matter*, 1 edn, Oxford University Press.

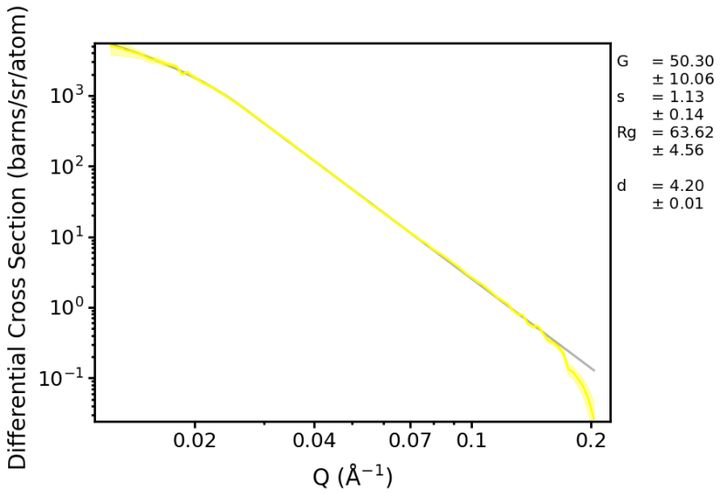

Figure 3: Fitting of GP model to a DCS vs Q plot. GP model in grey and data in yellow.

[2] Feigin, L. A. and Svergun, D. I. 1987, *Structure Analysis by Small-Angle X-Ray and Neutron Scattering*, Springer US, Boston, MA.
URL: *http://link.springer.com/10.1007/978-1-4757-6624-0*

[3] Hammouda, B. 2010, 'A new guinier-porod model', *Journal of Applied Crystallography* **43**, 716–719.

[4] Porod, G. 1952, 'Die röntgenkleinwinkelstreuung von dichtgepackten kolloiden systemen', *Kolloid-Zeitschrift* **125**, 51–57.
URL: *https://api.semanticscholar.org/CorpusID:197930362*

[5] Wei, Y. and Hore, M. J. A. 2021, 'Characterizing polymer structure with small-angle neutron scattering: A tutorial', *Journal of Applied Physics* **129**(17), 171101.
URL: *https://doi.org/10.1063/5.0045841*