

Session 1

Demonstrations of model building in some commonly-used analysis packages

This session was not aimed at resolving a particular issue, but rather to give people a sense of what is out there for reflectometry analysis.

Demos:

1. Andrew Nelson demonstrated refnx, as well as jupyter notebooks for running some of the python-based package calculations in an interactive session
2. Artur Glavic demonstrated the GenX package, complete with a loader for .ort (ORSO-draft-formatted) reflectivity data.
3. Joachim Wuttke showed the calculation in BornAgain, along with some non-specular calculations that BornAgain can perform.
4. Arwel Hughes showed the construction of a model in RasCAL, and showed the backend for the newer version that is under active development.
5. Brian Maranville demonstrated Refl1D, as well as a web-based simple model builder.
6. Mrinal Bera demonstrated XModFit, with multi-modal fitting available including XRR.

Discussion: One suggestion was to provide a communications channel where users could ask questions of the package maintainers on a more regular basis - a slack channel was floated as an idea, and the fact that the gitter.io chat had essentially gone quiet was mentioned.

Someone noted that there are already many structural similarities between the scripts that are written for defining models in the various programs.

People suggested that these videos are a good start for a tutorial video for users, for each of the packages.

Session 2

Part I: Discussion of multi-modal fitting

Andrew Caruana presented "Fitting laboratory XRR data – to use with NR"

The talk provided an overview of some of the common roadblocks encountered when trying to fit laboratory X-ray reflectometry data, including footprint corrections and beam profile, as well as presenting alternative goodness-of-fit criteria that can be used to manually alter the weighting given to different parts of the measurement (low-Q vs high-Q)

In the discussion, it was pointed out that some of this can be helped in newer XRR instruments with area or line detectors, where some additional corrections (rebinning) are possible after the measurement is complete.

The goodness of fit criteria were discussed at length, and it was pointed out that while using χ^2 is needed for some statistical interpretations of the results of fitting, that practical considerations often lead to people using a different weighting function that allows fitting of the higher-Q data as well as the lower-Q data.

Someone suggested that using fluorescent paper would allow direct imaging of the beam at the sample position, and someone else mentioned that knife-edge optics with scanning allow mapping of the beam profile, as well.

It was suggested by more than one person that a lower-intensity or less-focused beam was often preferred if the beam shape and resolution was better understood as a result.

In discussions of adding ellipsometry to the available multi-modal fits, it was pointed out that ellipsometry measurements typically do not come with computed statistical error bars at all, and so it is very difficult to mix them with other measurements in a combined fit.

Part 2: Demonstration of EasyReflectivity

Andrew McCluskey demonstrated a cross-platform GUI for setting up a model and fitting reflectivity, which will be able to interact with a number of different available backends for doing the calculations and the optimizing. Currently `refnx` and `BornAgain` are available as calculation backends, and `bumps` is available as an optimizer.

Part 3: Presentation of a draft declarative model language

Brian Maranville presented a draft set of definitions of building blocks for a declarative definition of a reflectivity model

During the discussion, people expressed a concern that this approach would be too limited and not extensible.

Some pointed out that a readable version of a model not tied to a particular modeling package might have value to users wanting to include their models in a publication.

Someone pointed out that having an intermediate “exchange” language like that would make it easier to write cross-package helpers like EasyReflectivity that was presented earlier in the session