Australian Government

ANSTO

refnx - reproducible neutron + X-ray reflectometry analysis

Andrew Nelson

In a nutshell

- Python based / Open-source / Fast / Well tested
- Co-refinement of Neutron, X-ray (and ellipsometry) data
- Scriptable/Jupyter notebooks - reproducible research
- Standalone GUI
- Bayesian analysis framework – posterior distribution for parameters, model selection, include prior knowledge in modelling system {nested sampling, MCMC}
- Different fitting algorithms – differential evolution, Levenberg-Marquardt, etc
- Batch Fitting
- Parallelisable on cluster
- `MixedReflectModel` – model patchy surfaces
- `Component` - defines subset of SLD profile, uses physically relevant parameters
- `Interface` – different types of roughness between `Component`
- Various types of resolution smearing
- Simulate NR datasets

ANSTO

Sources of Information

- Source code - https://github.com/refnx/refnx
  - Issues, **Contribute your changes**
- Documentation - https://refnx.readthedocs.io/en/latest/
  - Getting Started, FAQs, Examples, API reference, etc
- Video tutorials - https://www.youtube.com/channel/UCvhOxwZsdFMGqSzasE0ZSOw
- User contributed models/`Component` - https://github.com/refnx/refnx-models

- **ASK**

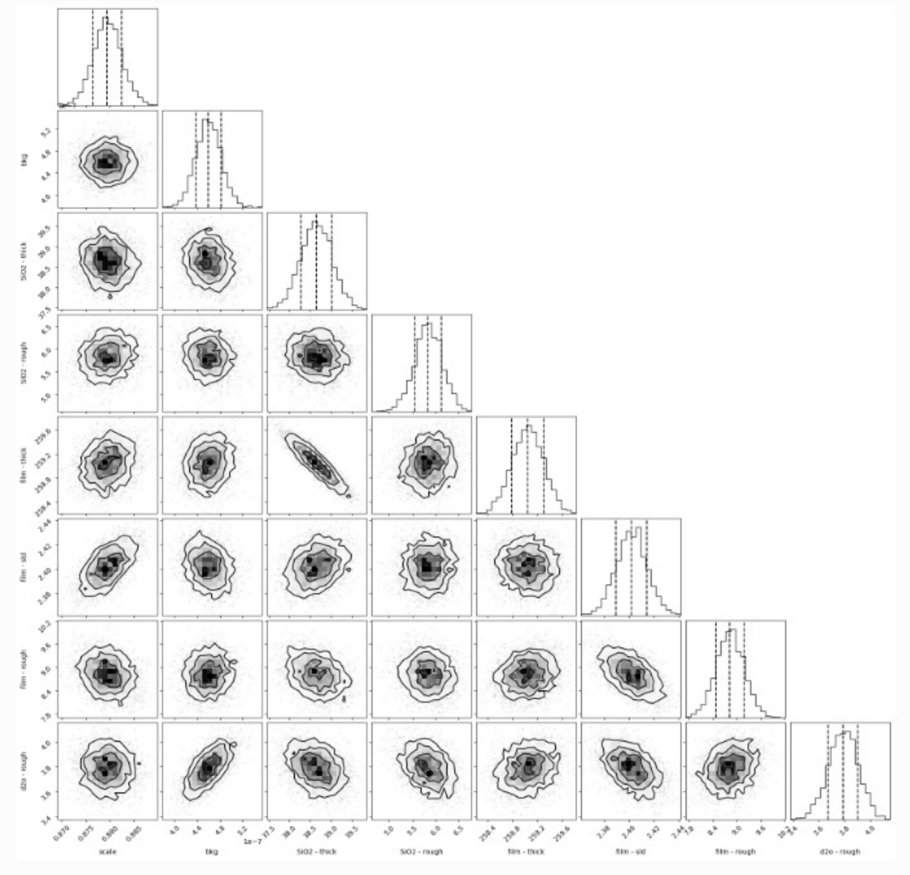
New features/highlights

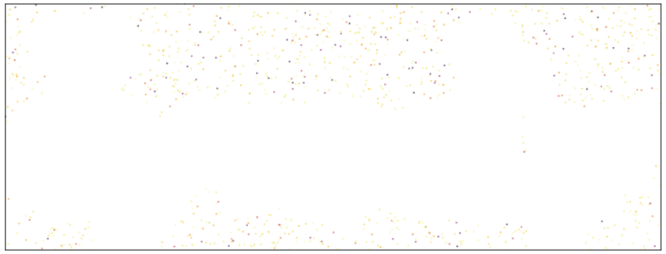
- Energy dispersive Scatterers (ellipsometry/RSoXRR?)
- Components
  - FreeformVFP
  - MaxEnt
  - LipidLeaflet
- Detailed resolution smearing kernel
- Simulate datasets from ToF reflectometers
- MixedReflectModel (incoherent averaging)
- Read ORSO files

MCMC sampling – emcee/ptemcee/dynesty/pymc

PyMC

emcee
the mcmc hammer

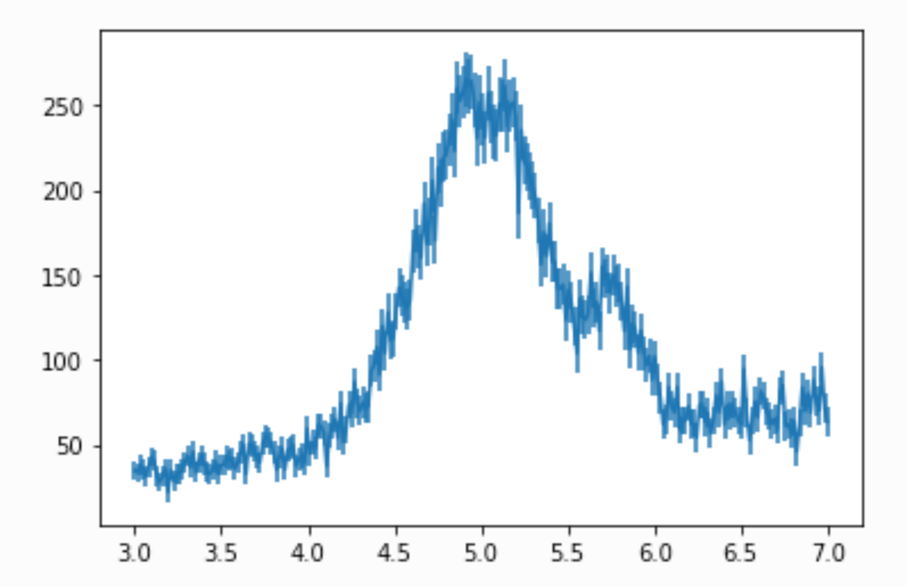


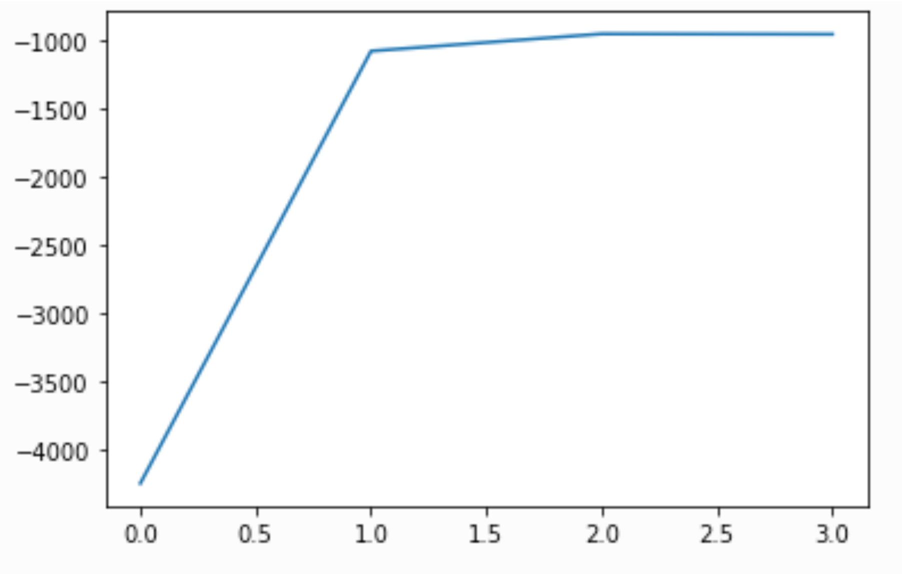


$$p(\theta \mid D, I) = \frac{p(\theta \mid I)\, p(D \mid \theta, I)}{p(D \mid I)}$$

https://refnx.readthedocs.io/en/latest/emcee_pymc_dynesty.html

Dynesty - model selection

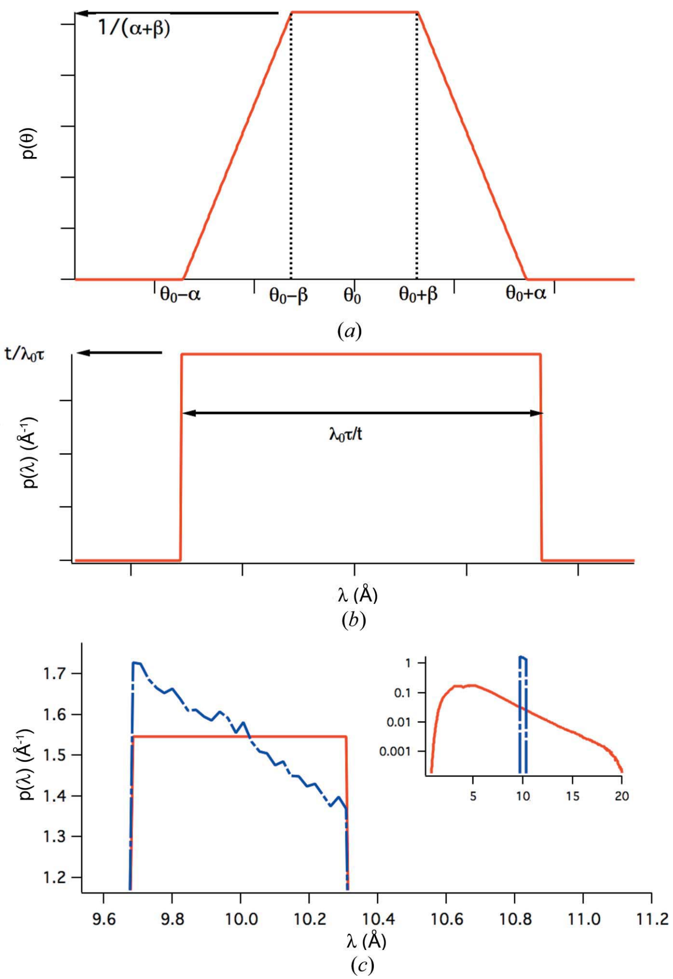




$$p(\theta \mid D, I) = \frac{p(\theta \mid I)\, p(D \mid \theta, I)}{p(D \mid I)}$$

https://refnx.readthedocs.io/en/latest/model_selection.html

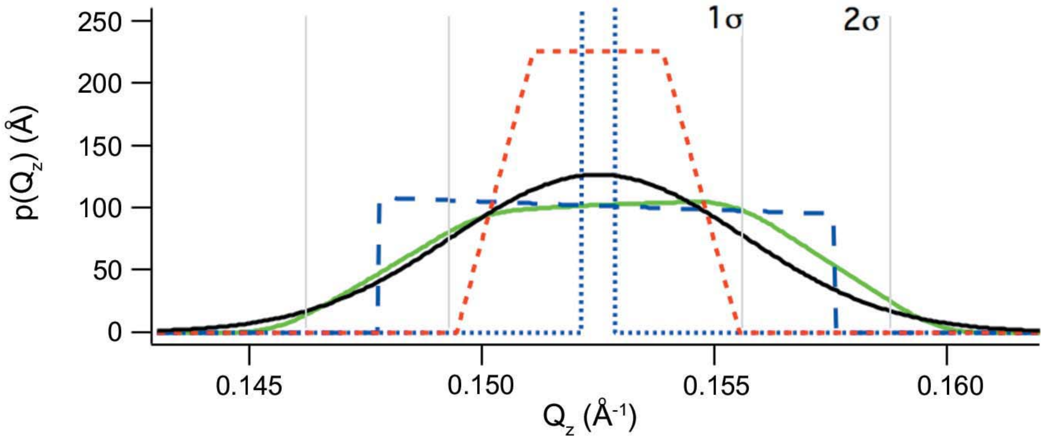
Detailed resolution kernel



(*a*)

(*b*)

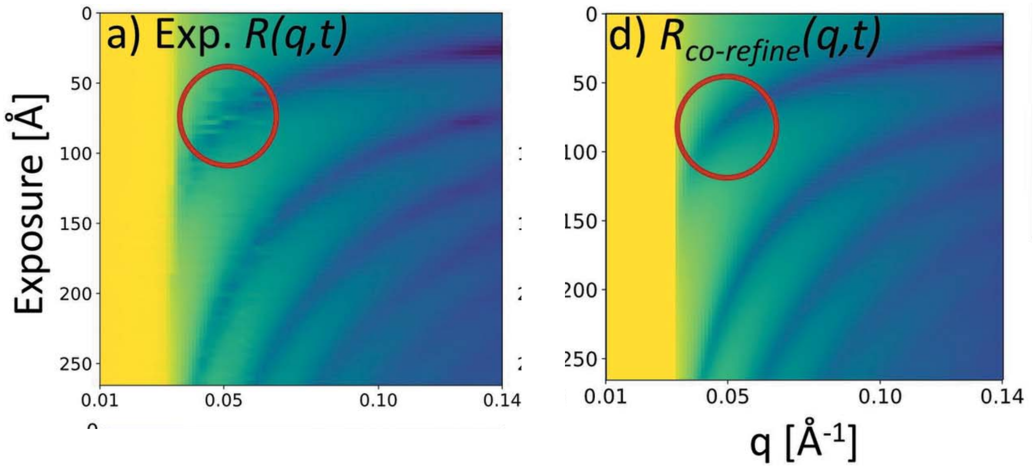
(*c*)



Nelson and Dewhurst, *J. Appl. Cryst*, **47**, 2014

Complex constraints

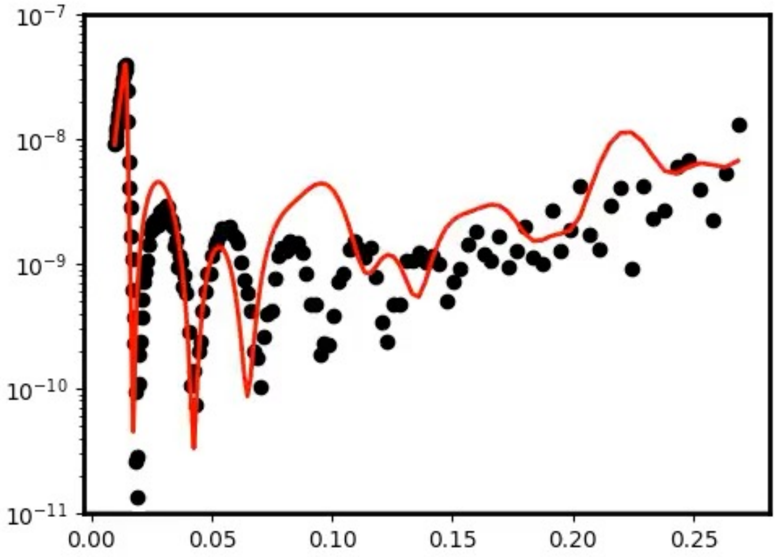
- Priors can be any statistical distribution
- Functional relationships $x_0 = \sin(x_1)$ or $x_0 = f(x_1)$
- Inequality constraints $lb < f(x_{0\dots n}) < ub$

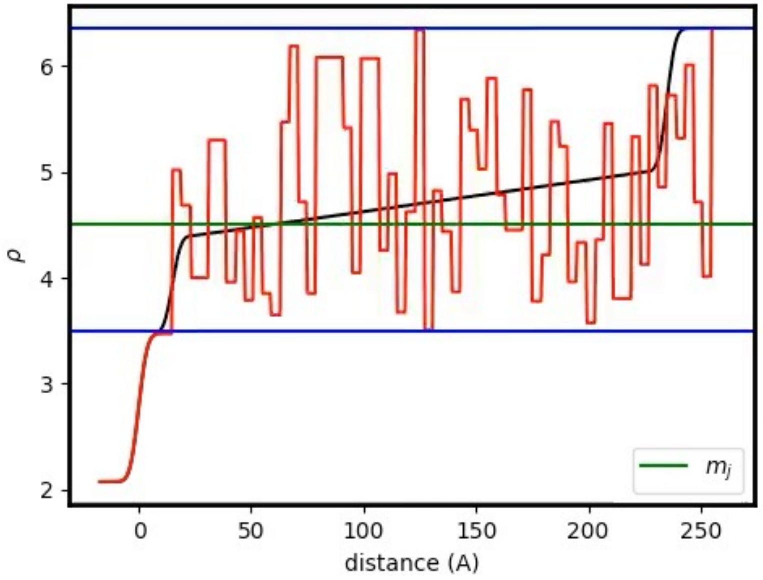


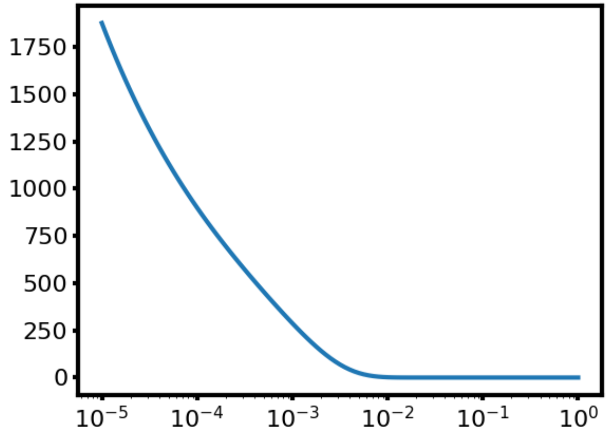
$$\theta_{n,\mathrm{cr}} = \begin{cases} a\,\dfrac{0.5\tanh[-0.5(t+b)]+0.5}{0.5\tanh(-0.5b)+0.5}, & \theta_{n,\mathrm{cr}} \geq c,\\ (c-f)\exp[-d(t-t_{\mathrm{c}})]+f, & \theta_{n,\mathrm{cr}} < c,\end{cases}$$

Marecek et al, *J. Appl. Cryst.* (2022). **55**, 1305–1313

Maximum entropy





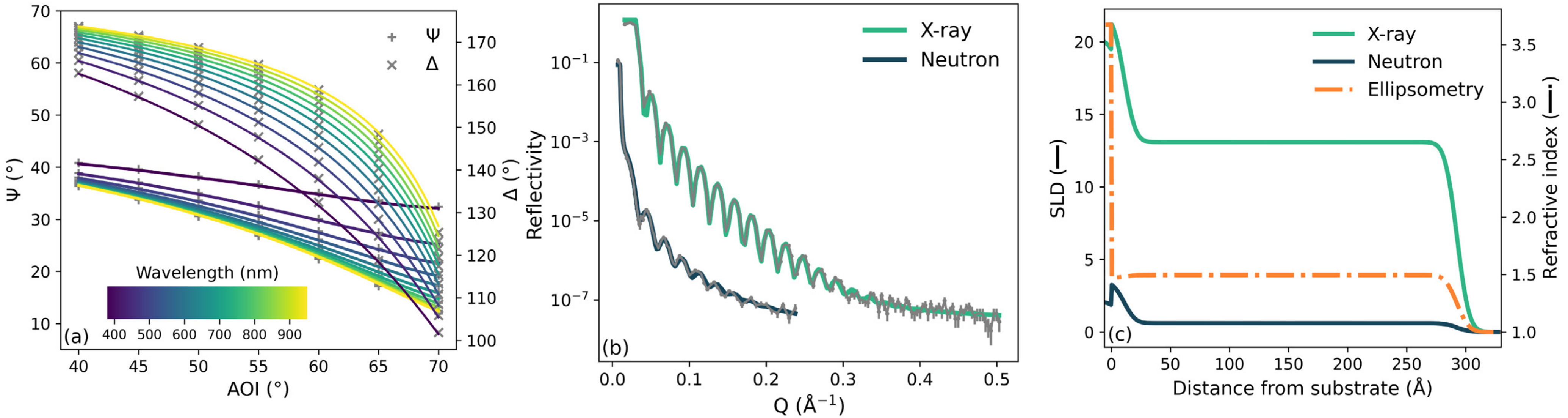


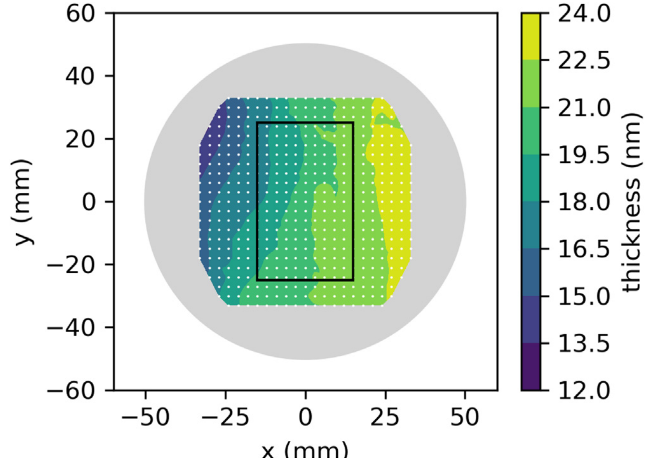
$$S = \sum_i \rho_j - m_j - \rho_j \log\left(\frac{\rho_j}{m_j}\right)$$

$$p = \frac{S}{\alpha}$$

Add to log-posterior

refellips





- Available from PyPI
- Different oscillator models
- Model creation identical to *refnx*
- Co-refine XRR/NR/Ellipsometry
- Surface mapping
- Batch fitting

Optimisations

- Calculation kernels
  - C/Python/Cython/numba/Parratt/JAX
  - JAX allows automatic differentiation
- Internal overhead
- HPC sampling speed

ANSTO