

A tutorial on the model-dependent analysis of neutron and X-ray reflectometry*

Andrew R. McCluskey,^{1,2,†} Andrew R. J. Nelson,³ and Tim Snow^{1,4}
 (Open Reflectometry Standards Organisation)

¹*Diamond Light Source, Rutherford Appleton Laboratory,
 Harwell Science and Innovation Campus, Didcot, OX11 0DE, UK*

²*Department of Chemistry, University of Bath, Claverton Down, Bath BA2 7AY, UK*

³*Bragg Institute, Australian Nuclear Science and Technology Organisation,
 Locked Bag 2001, Kirrawee DC, NSW 2232, Australia*

⁴*School of Chemistry, University of Bristol, Bristol, BS8 1TS, UK*

(Dated: July 6, 2020)

Neutron reflectometry analysis is an inherently ill-posed, which is to say that there are many possible solutions which agree equally well with the measured data. This leads to the use of model-dependent analysis to interpret the experimental data, where information that is “known” about the system is integrated into the analysis. This tutorial aims to introduce the mathematics that underlies the use of model-dependent analysis in neutron reflectometry. I hope that those that are well experienced in reflectometry find it as useful as those who are just about to experience their first reflectometry experiment.

I. INTRODUCTION

The mathematical concepts that underpin reflectometry are not typically affected by whether the probing radiation is the neutron or a beam of X-rays. Therefore, I will discuss these agnostically and only reference a given radiation source when particularly relevant.

The collected data (once normalised) is described in terms of reflected intensity, $R(q)$, which depends on the wavevector, q , and is measured as,

$$R(q) = \frac{\text{specular reflected intensity at } q}{\text{incident radiation intensity}}, \quad (1)$$

where the denominator in Equation 1 is the total flux of X-rays or neutrons on the sample. It is this $R(q)$ that we want to calculate from our model and then compare with our data.

The Born approximation [1], which assumes a single scattering event for each incident radiation particle, allows the reflected intensity of be calculated as follows [2],

$$R(q) \approx \frac{16\pi^2}{q^4} \left| \int_{-\infty}^{+\infty} \rho'(z) \exp(-izq) dz \right|^2, \quad (2)$$

where, $\rho'(z)$ is the first derivative of the scattering length density profile. The scattering length density profile is our model, a functional description of how the scattering length density varies with depth normal to the interface, z .

The approach outlined above is referred to as the kinematic approach, and it has a significant problem that can be demonstrated by considering the simple case of a bare

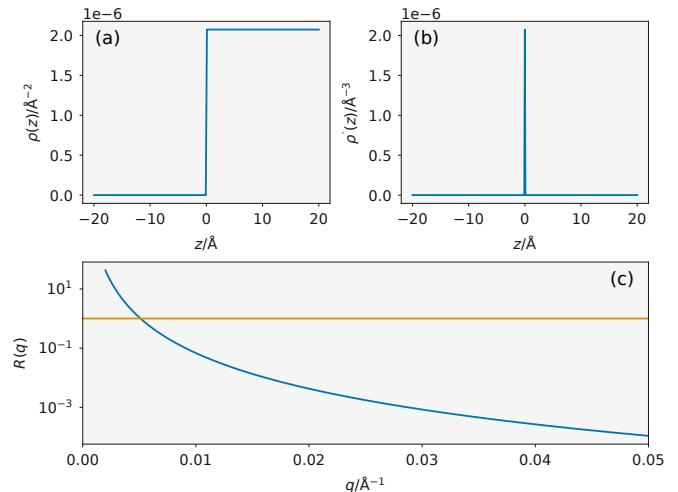


FIG. 1. A graphical representation of each aspect of the kinematic approach; (a) the Heaviside function describing the scattering length density profile for a bare silicon substrate, (b) the δ -function arising from the first derivative of the function in (a), and (c) the resulting reflectometry profile, where the orange line at $R(q) = 1$ identifies the breakdown of between experiment and theory.

silicon surface [2], which can be modelled with a Heaviside function, shown in Figure 1(a),

$$\rho(z) = \begin{cases} 0 & \text{where } z < 0, \\ \rho_{\text{Si}} & \text{otherwise,} \end{cases} \quad (3)$$

where, $\rho_{\text{Si}} = 2.074 \times 10^{-6} \text{ Å}^{-2}$, the nuclear scattering length density for pure silicon. The first derivative of a Heaviside function is a scaled δ -function, shown in Figure 1(b),

$$\rho'(z) = \rho_{\text{Si}}\delta(z). \quad (4)$$

Then, using Equation 2, we can calculate the reflected

* Modified from a document prepared for the ISIS Neutron Training Course in March 2020, which was unfortunately cancelled due to the COVID-19 pandemic

† andrew.mccluskey@diamond.ac.uk; a.r.mccluskey@bath.ac.uk

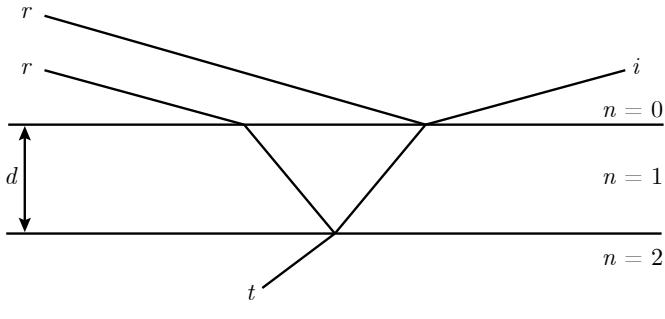


FIG. 2. A schematic diagram showing the reflected (r) and transmitted (t) waves when an incident (i) wave enters an interface of thickness d .

intensity with respect to q ,

$$\begin{aligned} R(q) &\approx \frac{16\pi^2}{q^4} \left| \rho_{\text{Si}} \int_{-\infty}^{+\infty} \delta(z) \exp(-izq) dz \right|^2 \\ &\approx \frac{16\pi^2}{q^4} \left| \rho_{\text{Si}} \exp(0) \right|^2 \\ &\approx \frac{16\pi^2 \rho_{\text{Si}}^2}{q^4}. \end{aligned} \quad (5)$$

The result of Equation 5 is shown in Figure 1(c), where it is clear that the agreement with an experimental profile would be poor as $q \rightarrow 0$ [3], where the reflected intensity is greater than 1. This violates the physical condition imposed by Equation 1, with more neutrons being reflected than were incident in the first place. The breakdown of this kinematic approach is due to the assumption of the Born approximation that each incident radiation particle is only scattered once. The reflectometry geometry, where a large path length arises from the shallow incidence angle means that multiple scattering events are likely, rendering the kinematic approach invalid.

II. RECURSIVE METHODS

The breakdown of the kinematic approach leads to the application of the Abéles [4] or Parratt [5], recursive model for the reflection of light at a given number of stratified interfaces, also referred to as the dynamical approach. This is an alternative methods for calculating the reflected intensity from a model, however, now the model is a series of layers of different scattering length density. The probing radiation can be either reflected or refracted at the interface between each of the layers. Figure 2 shows the refraction and reflections from a simple two layer system, where layer 0 is a vacuum above the sample. It is clear, by analogy to Bragg's law, how the two waves, labelled r , could interfere constructively or destructively depending on the thickness of layer 1, d .

The generalisation of this approach to any number of layers is possible and enables the reflected intensity to

be calculated at each value of q that was measured. The incident neutrons will be refracted by each of the layers, giving wavevectors for each layer, k_n ,

$$k_n = \sqrt{k_0 + 4\pi(\rho_n - \rho_0)}, \quad (6)$$

where, $k_0 = 0.5q$. The Fresnel equation coefficient between layers n and $n+1$, $r_{n,n+1}$, can then be found along with the phase factor, β_n , for the layer n ,

$$r_{n,n+1} = \frac{k_n - k_{n+1}}{k_n + k_{n+1}}, \quad (7)$$

$$\beta_n = k_n d_n. \quad (8)$$

From the Fresnel equation coefficient and phase factor, a characteristic matrix for each layer, M_n , can be constructed,

$$M_n = \begin{bmatrix} \exp \beta_n & r_{n,n+1} \exp -\beta_n \\ r_{n,n+1} \exp \beta_n & \exp -\beta_n \end{bmatrix}, \quad (9)$$

and the resultant matrix for a given q -vector, $B(q)$ is found from the product sum of the matrices from each layer,

$$B(q) = \prod_{n=0}^{n_{\max}} M_n. \quad (10)$$

The characteristic matrix for each layer describes the reflected and refracted radiation intensity, so the product sum is able to describe the total reflected intensity. In particular, this is given by the following elements of the resultant matrix,

$$R(q) = \frac{B_{1,2}(q)}{B_{1,1}(q)}. \quad (11)$$

This algorithm models the layers as perfectly flat layers, which may not be strictly true. This has resulted in the use of correction terms to be added to Equation 7 to account for this roughness. The most common of these is Nérot and Croce Gaussian broadening [6], in which the Fresnel equation coefficient is evaluated as,

$$r_{n,n+1} = \frac{k_n - k_{n+1}}{k_n + k_{n+1}} \exp(-2k_n k_{n+1} \sigma_{n,n+1}^2), \quad (12)$$

where, $\sigma_{n,n+1}$ is the interfacial roughness between the layers n and $n+1$. Applying this dynamical approach to a perfectly flat silicon surface is shown in Figure 3. There is a clear difference between the kinematic and dynamical approaches as $q \rightarrow 0$, with the dynamical approach adhering to the physical constraint that causes the breakdown of the kinematic approach.

Figure 4 shows the effect that increasing roughness can have on the scattering length density of some system, in this case D₂O on a silicon substrate. Notice that as the roughness gets to values similar to half the layer

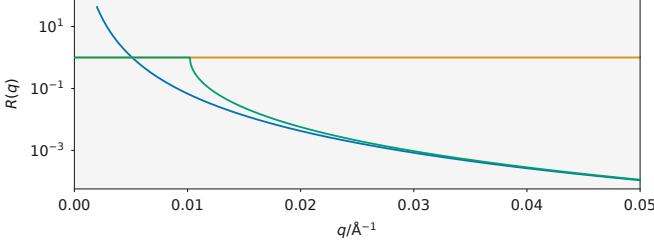


FIG. 3. A comparison of the kinematic (blue) and dynamical (green) approaches to determine the reflected intensity from the material with the scattering length density profile given in Figure 1(a).

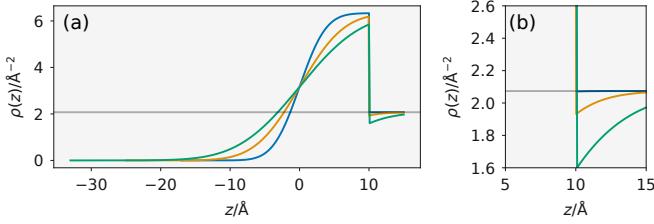


FIG. 4. The effect of a roughness greater than a given layer's thickness on the scattering length density profile. The layer thickness is 10 Å. The scattering length densities are those of D₂O on a atomically flat silicon. The roughness of the water layer is varied from 3 Å (blue), through 5 Å (orange), to 7 Å (green). The full scattering length density profile is given in (a), while (b) represents a zoomed in representation.

thickness, artifacts appear in the scattering length density profile. Below, where the roughness is too large, the scattering length density is less than the substrate scattering length density. This is unphysical, and indicates an important factor that we should try and account for by not allowing large roughnesses with respect to the layer thickness, a common rule of thumb is to use less than one quarter of the layer thickness.

III. GLOBAL OPTIMISATION

The recursive method described above gives an accurate method to obtain a model reflected intensity. However, this is just the first step in the analysis of a neutron reflectometry dataset. Now we are interested in optimising our model such that the reflected intensity from it matches our experimental data as best as possible. This is the problem of parameter optimisation, which is a broad area of mathematics and computer science that we will not dwell on here. However, we will introduce the basics of optimisation and discuss the most common global optimisation method used in reflectometry.

When we measure a reflectometry profile, we measure the reflected intensity, and some uncertainty in that measurement, at discrete points in the wavevector, $R(q) \pm \delta R(q)$. Using the recursive method discussed

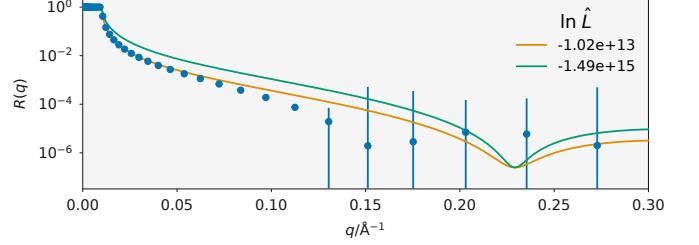


FIG. 5. The maximum likelihood fit, the blue data points indicate some experimental data, while the yellow line shows the maximum likelihood model fit to this dataset and the green line show another model (which doesn't maximise the likelihood).

above, we can calculate a model reflected intensity at these same q values, $R_m(q)$. We then aim to reduce the difference between the measured and modelled reflected intensity through the optimisation (maximisation) of the likelihood, L ,

$$L = \exp \left\{ -\frac{1}{2} \sum_{q=q_{\min}}^{q_{\max}} \left[\frac{R(q) - R_m(q)}{\delta R(q)} \right]^2 + \ln[2\pi\delta R(q)] \right\}. \quad (13)$$

This parameter described the probability that the model m accurately describes the observed reflectometry data. The aim in reflectometry modelling is to obtain a model reflectometry that has the maximum likelihood possible for the given experimental data. Figure 5 shows the maximum likelihood model for some experimental data, in addition to another model which does not agree well with the data and therefore has a lower likelihood as a result.

The global optimisation of a reflectometry model is a particularly difficult problem, this is due to the ill-posed nature of this data, this is where there are many reasonable solutions to a particular reflectometry profile. However, a particular global optimisation method has shown substantial utility in the fitting of reflectometry data [7], differential evolution [8]. This has lead to the inclusion of this method in many common reflectometry analysis packages [9].

Differential evolution is an iterative, genetic algorithm, designed to mimic the evolution processes observed in biology [10]. The method consists of two vectors, the parent population \mathbf{p} , and the offspring population, \mathbf{o} . These vectors are of shape $(i \times j)$, where i is the number of parameters in the model and j is the number of candidate solutions being considered. The offspring population is generated as a result of some trial method, here we will discuss only a simple classical trial method.

A classical trial method consists of two stages, mutation and recombination. The mutation stage involves the creation of a mutant population, \mathbf{m} . The magnitude of the mutation is dependent on the first of our hyperpa-

rameters, the mutation constant, k_m ,

$$\mathbf{m}_{i,j} = b_i + k_m(\mathbf{p}_{i,R1} - \mathbf{p}_{i,R2}), \quad (14)$$

where b is the candidate solution with the greatest likelihood, and $\mathbf{p}_{i,R1}$ and $\mathbf{p}_{i,R2}$ are randomly chosen members of the parent population. The mutation constant hyperparameter controls the size of the search space, with a large k_m corresponding to a wider search.

The recombination step creates the offspring population vector by taking a sample from either the parent or mutant population with some frequency, which depends on our second hyperparameter, the recombination constant, k_r ,

$$\mathbf{o}_{i,j} = \begin{cases} \mathbf{m}_{i,j} & \text{where } X < k_r, \\ \mathbf{p}_{i,j} & \text{otherwise,} \end{cases} \quad (15)$$

where, X is a random number selected from a uniform distribution between 0 and 1. The recombination constant hyperparameter controls the mutation frequency in the offspring population.

The final stage is to compare the offspring and parent population vectors, in the selection stage, to create the parent population for the next iteration. Here, the likelihood is used to compare between subunits from the offspring or parent populations,

$$\mathbf{p}_{*,j} \leftarrow \begin{cases} \mathbf{o}_{*,j} & \text{where } L_{\mathbf{o}_{*,j}} > L_{\mathbf{p}_{*,j}}, \\ \mathbf{p}_{*,j} & \text{otherwise,} \end{cases} \quad (16)$$

where, the $*, j$ subscript notation indicates all objects from the population, j . Following the use of the differential evolution algorithm, typically a more common (gradient-based) approach is used to ensure that the likelihood has been maximised within the space identified by the differential evolution.

The differential evolution algorithm can be seen in action applied to the negative two-dimensional Ackley function [11], in Figure 6. The Ackley function is a common function used in the assessment of global optimisation functions. This is due to there being a large number of local minima and only a single global minimum to this function. Here, we want to maximise the value, so the negative Ackley function is used. This function can maximise the value of the negative Ackley function. While this does not offer a clear example of this algorithm's application to reflectometry analysis, the popularity of this method in the fitting of reflectometry profiles cannot be denied [9, 12].

IV. UNCERTAINTY QUANTIFICATION

Reflectometry measurements offer an average description of the out of place structure of a material. This means that it is pragmatic to describe the uncertainties in the values of our model parameters in some fashion. In

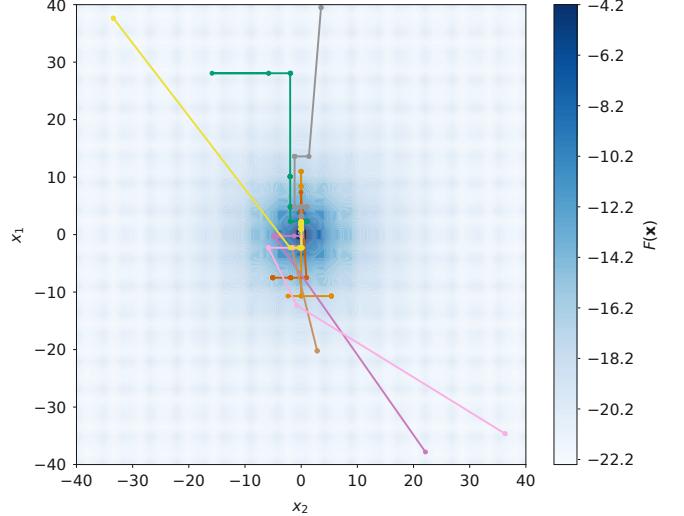


FIG. 6. An example of a differential evolution algorithm applied to the negative of an Ackley function. In this implementation $k_m = 0.5$ and $k_r = 0.5$. Each line represents a different candidate solution. The optimisation was stopped after 100 iterations had run.

this section, we will introduce two potential methods to determine the uncertainty on a set of model parameters.

The first is computationally straightforward and is substantially more common. This is where the parameter uncertainties are found from the Hessian matrix of the optimisation space, where the uncertainties are the square-root of the matrix diagonal. The previous sentence introduced a lot of terminology that may not be familiar, however we shall work through it slowly. The above optimisation method, differential evolution, is what is referred to as a stochastic method, where steps taken arise from a randomly determined process. However, another common type of optimisation methods are known as gradient methods, where the gradient of the optimisation space at a given point is calculated and this is used to determine the next position. Consider maximising the likelihood for a reflectometry model fit, you would like to make steps that increase the likelihood to reach the maximum, so calculating the gradient will be able to tell you in which direction to travel.

The Hessian matrix, H describes the curvature of optimisation criteria, the likelihood, with respect to each of the model parameters, and it is frequently used in the gradient-based optimisation methods. This means it is an $i \times i$ matrix, where i is again the number of model parameters, θ , for a two parameter model, this would look like this,

$$H = \begin{bmatrix} \frac{\partial^2 L}{\partial \theta_1^2} & \frac{\partial^2 L}{\partial \theta_1 \partial \theta_2} \\ \frac{\partial^2 L}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 L}{\partial \theta_2^2} \end{bmatrix}. \quad (17)$$

If the parameter distribution is Gaussian in shape then

the second derivative at the Gaussian maximum will give the variance of that distribution. Therefore, when these gradient approaches are used, the Hessian matrix is calculated and the diagonal of that matrix gives the uncertainty in each of the parameters. The off-diagonal elements contain information about the covariance between different parameters, however this is rarely leveraged and the covariances are taken to be negligible. Additionally, we note that often the Hessian is not calculated in an optimisation so the Jacobian matrix (the first derivative analogue of the Hessian) is used to estimate Hessian matrix.

The determination of parameter uncertainties from the Hessian matrix is dependent on two major assumptions about the parameter space:

1. the parameters are normally distributed with a single maximum,
2. the covariance between the parameters is negligible.

However, this may not always be the case [13], therefore it is necessary to use methods to completely describe the parameter probability distribution. One such method for this is Markov-chain Monte Carlo (MCMC), which samples the posterior probability distribution for each of our parameters to obtain an analytical description [14]. Typically, MCMC is used on already optimised solutions to a particular problem, in reflectometry analysis it is usually applied after the differential evolution has optimised the structure. In addition to being able to quantify the inverse uncertainties (the name given to the uncertainties in the model parameters) MCMC also offers a more complete understanding of the correlations between different parameters [15], which is particularly important in the ill-posed reflectometry analysis.

Once an optimised set of model parameters, θ , are obtained which maximise the likelihood, L , some random perturbation is applied,

$$\Theta = \theta + aR, \quad (18)$$

where R is some normally distributed number centered on 0 with a standard distribution of 1 and a is the step size. A new L is found for Θ and the probability that this transition is will occur is found,

$$p = \exp \left[\frac{L(\Theta) - L(\theta)}{2} \right]. \quad (19)$$

This probability is then compared with a uniformly distributed random number from 0 to 1, n ,

$$\theta \leftarrow \begin{cases} \Theta & \text{where } n < p, \\ \theta & \text{otherwise.} \end{cases} \quad (20)$$

This process is repeated until some desired number of samples has been obtained. It should be noted that it is important to allow the Markov chains to have some

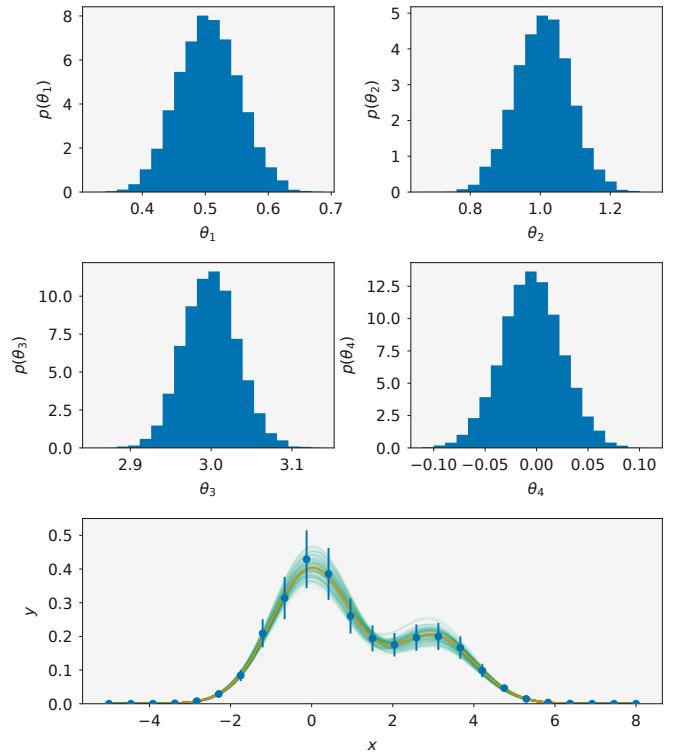


FIG. 7. An example of the result of MCMC uncertainty sampling on data synthesized from two overlapping Gaussian functions. θ_1 and θ_2 are the integrals of the Gaussian functions while θ_3 and θ_4 are the positions. The data is shown with blue circles, the optimised model with an orange line, and samples from the posterior distributions with green lines.

“burn-in” period, which is not included in the final samples. This allows the MCMC algorithm to settle into the search-space.

Figure 7 shows the result of an MCMC sampling for a pair of overlapping Gaussian functions, performed using the `emcee` Python package [16]. The posterior probability distributions that are available to each of the four parameters are shown along with the data, the optimised fit and a subset of models within the posterior distributions. All of the samples from the posterior distributions fit within the uncertainty error bars on the data. This shows the ability for MCMC to sample the range of the distribution that is allowed by the experimental uncertainty.

We note here that MCMC may also be used to facilitate the use of Bayesian inference in the analysis of reflectometry data, where other prior knowledge about the experimental system is able to influence the result of the analysis. In this tutorial, we won’t say any more on this subject, but if you are interested, this is discussed in the work of Nelson and Prescott [12] and McCluskey and co-workers [17].

V. CONCLUSIONS

The aim of this document was to give an introduction to some of the mathematical concepts that underpin modern reflectometry model-dependent analysis. We have looked at how reflectometry is calculated from a layer model description of the scattering length density profile, and why the kinematic approach fails. Then we have discussed the importance of the differential evolution algorithm in reflectometry analysis and detailed the operation of this algorithm. Finally, Markov-chain Monte Carlo was introduced in the context of uncertainty quantification for model-dependent analysis, with particular importance for reflectivity discussed. While not exhaustive, I hope that this document will give you the confidence in understanding to look in more detail into how the analysis of reflectometry measurements are performed.

AUTHOR CONTRIBUTIONS

ARM wrote the first draft of this text, and ARJN and TS offered input on additional content to cover and clarifications to include.

proposed, implemented, and applied the Bayesian model comparison framework, with input from T.A., J.F.K.C. and T.S.; J.F.K.C. and T.S. sourced funding; A.R.M. wrote the manuscript, with input from all authors.

ACKNOWLEDGEMENTS

ARM is supported by the Ada Lovelace Centre a joint initiative between the Science and Technology Facilities Council (as part of UK Research and Innovation), Diamond Light Source, and the UK Atomic Energy Authority. The authors would like to thank Luke Clifton of ISIS Neutron and Muon Source for insightful input on the manuscript.

-
- [1] M. Born. *Z. Physik*, 38:803827, 1926. doi: 10.1007/BF01397184.
 - [2] D. S. Sivia. *Elementary Scattering Theory: For X-Ray and Neutron Users*. Oxford University Press. ISBN 978-0-19-922868-3.
 - [3] C. F. Majkrzak and N. F. Berk. *Phys. Rev. B*, 58(23): 15416, 1998. doi:10.1103/PhysRevB.58.15416.
 - [4] F. Abels. *Ann. Phys.*, 12(3):504–520, 1948. doi: 10.1051/anphys/194812030504.
 - [5] L. G. Parratt. *Phys. Rev.*, 95(2):359–369, 1954. doi: 10.1103/PhysRev.95.359.
 - [6] L. Névot and P. Croce. *Rev. Phys. Appl. (Paris)*, 15(3): 761–779, 1980. doi:10.1051/rphysap:01980001503076100.
 - [7] A. van der Lee, F. Salah, and B. Harzallah. *J. Appl. Crystallogr.*, 40(5):820–833, 2007. doi: 10.1107/S0021889807032207.
 - [8] M. Wormington, C. Panaccione, K. M. Matney, and D. K. Bowen. *Philos. Trans. R. Soc. London Ser. A*, 357(1761):2827–2848, 1999. doi:10.1098/rsta.1999.0469.
 - [9] M. Björck. *J. Appl. Crystallogr.*, 44(6):1198–1204, 2011. doi:10.1107/S0021889811041446.
 - [10] J. H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, 2 edition, 1992. ISBN 978-0-262-58111-0.
 - [11] D. H. Ackley. *A Connectionist Machine for Genetic Hill-climbing*. University of Michigan.
 - [12] A. R. J. Nelson and S. W. Prescott. *J. Appl. Crystallogr.*, 52(1):193–200, 2019. doi:10.1107/S1600576718017296.
 - [13] A. R. McCluskey, A. Sanchez-Fernandez, K. J. Edler, S. C. Parker, A. J. Jackson, R. A. Campbell, and T. Arnold. *Phys. Chem. Chem. Phys.*, 21(11):xx–xx, 2019. doi:10.1039/C9CP00203K.
 - [14] D. S. Sivia and J. Skilling. *Data Analysis: A Bayesian Tutorial*. Oxford University Press, 2 edition, 2006. ISBN 978-0-19-856832-2.
 - [15] W. Gilks, S. Richardson, and D. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC Interdisciplinary Statistics. CRC Press, 1995. ISBN 978-0-412-05551-5.
 - [16] D. Foreman-Mackey, W. Farr, M. Sinha, A. Archibald, D. Hogg, J. Sanders, J. Zuntz, P. Williams, A. Nelson, M. de Val-Borro, T. Erhardt, I. Pashchenko, and O. Pla. *J. Open Source Softw.*, 4(43):1864, 2019. doi: 10.21105/joss.01864.
 - [17] A. R. McCluskey, T. Arnold, J. F. K. Cooper, and T. Snow. *Mach. Learn.: Sci. Techno.*, xx(xx):xx–xx, 2020. doi:10.1088/2632-2153/ab94c4.