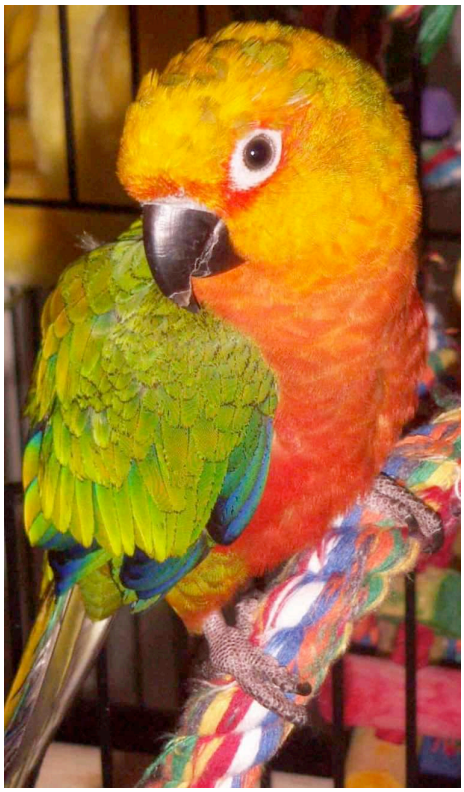
Motofit

February 19, 2009



http://motofit.sourceforge.net
Andrew_Nelson@users.sourceforge.net

Abstract

Motofit uses the Abeles Formalism to fit multiple contrast neutron and X-ray reflectometry data.

Contents

1 *Motofit*

Motofit aids the fitting of specular X-ray and neutron reflectivity data. It works within the analysis package IGOR PRO (Wavemetrics, OR). It was created in version 5 of IGOR.

The specular reflectivity is calculated using the *Abeles*[?] formulation (giving identical results to *Parratts*[?] recursion formula for stratified thin films), as a function of the perpendicular momentum transfer, Q_Z.

$$Q_z = \frac{4\pi}{\lambda} \sin\theta \tag{1}$$

Specular reflectivity (R) is defined as the ratio of reflected intensity over incident intensity, where the angle of reflection (θ) is equal to the angle of incidence.

Motofit was created with the aim of allowing a scientist to easily fit her/his reflectivity data. In particular, an easy way to analyse multiple contrast NR and XR data. This is achieved through a simple GUI, and genetic optimisation for fitting (Genetic Optimisation allows the use of initial guesses that are far from a final solution, which would otherwise trouble normal least squares fitting packages). Other reflectivity packages are obviously out there, but sometimes they don't allow you to do what you want.

A further advantage with *Motofit*/IGOR is that the graphics and output is publication quality, you just put the graph straight into your paper. You can control all aspects of the graphs you create. With most other analysis packages you have to export data into another plotting package, which take time. If you have to re-analyse data you have to do the export again. With *Motofit*, you just re-analyse, and the graphs are changed automatically. You can also save the whole experiment you are working on.

Motofit is also open source, which means you can change the behaviour of the program to whatever you want, if you don't like it. IGOR is a proven data-analysis program, which means you can rely on the numbers coming out

2 Acknowledging the use of *Motofit*

If you use *Motofit* to analyse data that you then use in a publication, then please cite *Motofit* in the references. This is so I can justify the time I spent writing the program to my bosses.

Inclusion in the reference should be done by referring to the *Journal of Applied Crystallography* paper[?].

A. Nelson. 'Co-refinement of multiple contrast neutron / X-ray reflectivity data using MOTOFIT.' *Journal of Applied Crystallography,* **39**, 273-276, 2006.

3 Fitting reflection from a stratified medium

The reflectivity is calculated using the *Abeles* matrix method for stratified interface[?].

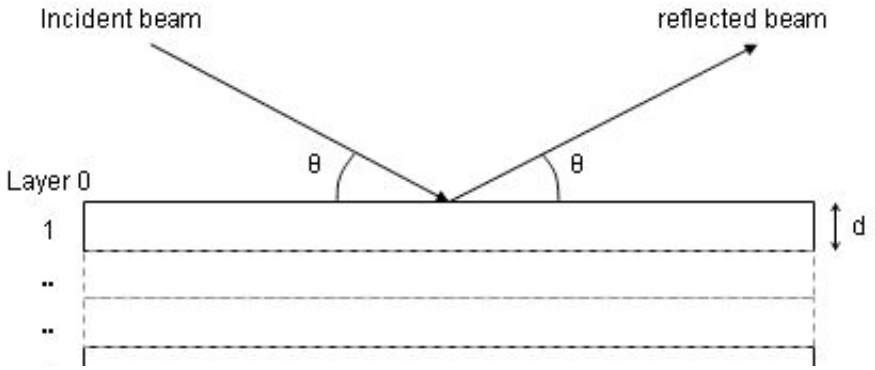


Figure 1: Reflection from a stratified medium

The measured reflectivity depends on the variation in the scattering length density (SLD) profile, $(\rho(z))$ perpendicular to the interface. Although the scattering length density profile is normally a continuously varying function, the interfacial structure can often be well approximated by a slab model in which layers of thickness (d_n), scattering length density (ρ_n) and roughness $(\sigma_{n,n+1})$ are sandwiched between the super- and sub-phases. One then uses a refinement procedure to minimise the differences between the theoretical and measured reflectivity curves, by changing the parameters that describe each layer.

In this description the interface is split into n layers. Since the incident neutron beam is refracted by each of the layers the wavevector, k, in layer n, is given by:

$$k_n = \sqrt{k_0^2 - 4\pi(\rho_n - \rho_0)} \tag{2}$$

The Fresnel reflection coefficient between layer n and $n+1$ is then given by:

$$r_{n,n+1} = \frac{k_n - k_{n+1}}{k_n + k_{n+1}} \tag{3}$$

$$r_{n,n+1} = \frac{k_n - k_{n+1}}{k_n + k_{n+1}} \exp\left(-2k_n k_{n+1}\sigma_{n,n+1}^2\right) \tag{4}$$

Since the interface between each layer is unlikely to be perfectly smooth the roughness/diffuseness of each interface modifies the Fresnel coefficient 4 and is accounted for by an error function, as described by Nevot and Croce[**?**].

A phase factor, β is introduced, which accounts for the thickness of each layer.

$$\beta_n = k_n d_n \tag{5}$$

A characteristic matrix,cn is then calculated for each layer.

$$c_n = \begin{bmatrix} \exp(\beta_n) & r_n \exp(\beta_n) \\ r_n \exp(-\beta_n) & \exp(-\beta_n) \end{bmatrix} \tag{6}$$

The resultant matrix is defined as the product of these characteristic matrices, from which the reflectivity is calculated.

$$M = \prod_0^n c_n \tag{7}$$

$$R = \left|\frac{M_{00}}{M_{10}}\right|^2 \tag{8}$$

4 What files do you need?

1. Motofit_all_at_once.ipf
2. Motofit_multilayers.ipf
3. Motofit_Fresnel.ipf
4. Motofit_Globalfit 2.ipf
5. Motofit_SetupGlobals.ipf
6. Motofit_Xrayreduction.ipf
7. Motofit_Batch.ipf
8. GeneticOptimisation.ipf
9. Abeles.xop (optional)

You will need to open them all for the program to work. I suggest creating a *Motofit* folder in *\Igor Pro Folder\User Procedures*. Then place a shortcut to the Motofit folder in the IGOR procedures folder. This will ensure that *Motofit* starts up whenever you open IGOR.

However, make sure that you remove previous versions of *Motofit* (keep the scattering length density database if you have changed it).

Please note that you can also download the optional Abeles XOP file. This file speeds up the reflectivity calculations by a factor of 5. You don't need this to run *Motofit*, but it is highly recommended, as you SAVE A LOT OF TIME. This XOP can be downloaded from a link on the *Motofit* website http://motofit.sourceforge.net `http://motofit.sourceforge.net`.

5 Getting Started

Open all the required procedure files (which should happen automatically if you install using the instructions above). Once the procedures have compiled then a *Motofit* menu should appear on the top menu. Use Fit Reflectivity Data to start fitting your data.

The *Motofit* menu has several sub-options:

1. Genetic Optimisation
2. Fit Reflectivity Data
3. Load Experimental Data
4. Scattering Length Density database
5. Create local Chi2map for requested parameter
6. Change Q range of theoretical data.
7. Reduce Xray data (reduces data from ANSTOs X-ray reflectometer)
8. Reduce Neutron data (reduces data from ANSTOs Neutron reflectometer).
9. Fit batch data (this fits multiple datasets, from kinetic data for example).
10. Moto_globalfit

5.1 Genetic Optimisation

This option allows the user to fit data to any fit function, using Genetic Optimisation with least squares. This allows the user to fit any other type of 1D data, e.g. Small-Angle Scattering measurements. Genetic Optimisation is a powerful way of fitting data, which allows the use of poor initial parameter guesses, but still finds the best fit solution. In comparison, normal least squares methods, such as Levenberg-Marquardt, are often quicker, but probably won't find the best fit if given poor initial guesses. With Genetic Optimisation you set up limits for parameters, and the optimisation finds the best fit from within the limits you set.

For fitting reflectometry data you will mostly use Genetic Optimisation from within the Reflectivity panel, but you can also run it from within the genetic optimisation panel created by this menu option.

It is extremely simple to set up your own user function to perform curvefitting, the following function fits a gaussian:

Function gaussian(w,x): Fitfunc
Wave w
Variable x
*return w[0]+w[1]*exp(-(x-w[2])^2/w[3]^2)*
End

See the IGOR manual for more details on creating your own fitfunction.

The genetic optimisation fitting procedure can be called from the command line, or from other user functions. If you wish to use this capability, please refer to the manual distributed with the standalone GeneticOptimisation package (on the Sourceforge website).

5.2 Fit Reflectivity Data

This option creates a user interface for you to fit your data.

When you click on this option a dialogue is brought up. This dialogue enables you to select some of the parameters required to produce the theoretical curve.

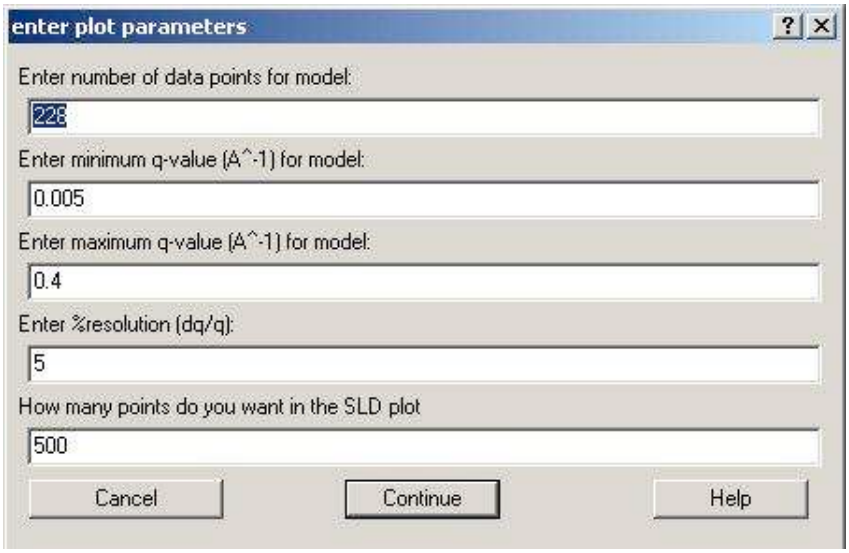


Figure 2: *Motofit* Initialisation

Most of the selections are obvious. You can control the Q range for the plot, as well as the number of points in that plot. dq/q relates to the resolution function of your instrument (note use dq/q=0 if you want quicker calculations), if you are unsure set this equal to zero.

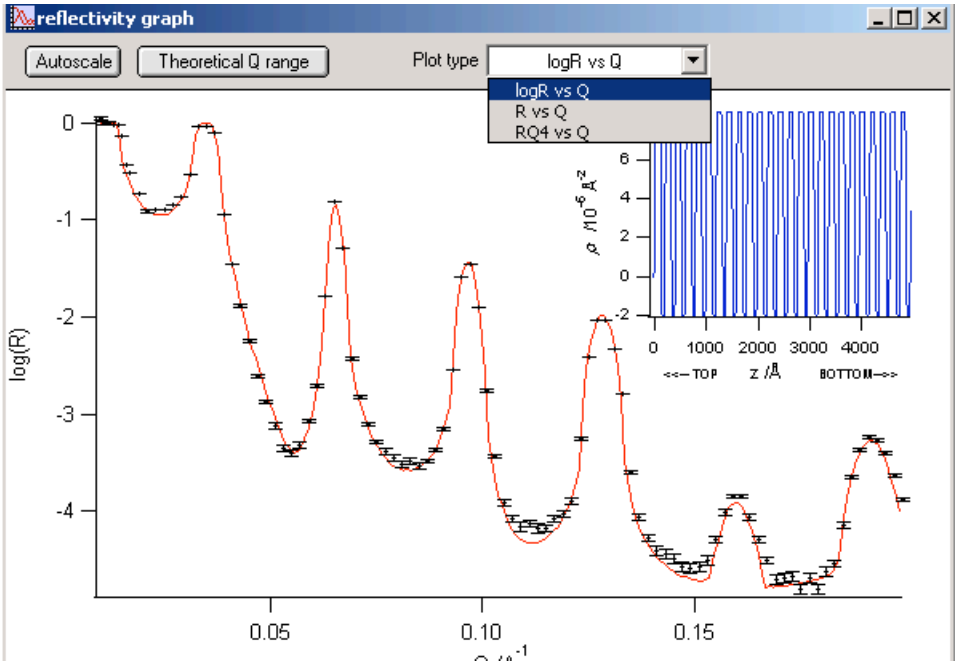


Figure 3: The reflectivity graph, showing a Ti/Ni multilayer

There are 500 points in the SLD profile graph that is created for your layered system. When you press continue the Reflectivity Fit Panel and Reflectivity Graph are produced.

The theoretical Q range can be altered at a later stage using the menu item *"Change Q range of theoretical data"*,or from within the reflectivity graph.

There are several sections to the panel:

5.2.1 Model parameters

The model parameters reside in the parameter wave *coef_ Cref*. This wave is updated when you change the values in the reflectivity panel.

The first box allows you to change the parameters required for the base model.

- NUMBER OF LAYERS : how many layers there are in your model. Layer 1 is closest to the superstrate, layer n is closest to the subphase (refer to Figure 3). You can have as many layers as you like. To increase the number of layers, change this value.
- SCALE : The scale factor for the reflectivity measurement (should be 1, or close to 1). All the values in the theoretical reflectivity curve are multiplied by this number.
- SLDTOP : Scattering length density of the Superphase (the incident medium for the radiation). If you are doing an air-solid measurement then this will 0. If you are doing a solid-liquid measurement, incident through the solid, then this will be the SLD of the solid.
- SLDBASE : Scattering length density of the subphase.
- BKG : This adds in a Q independent linear background to the fit.
- SIGMA_BASE : The roughness at the top of the subphase.

The second box allows you to change the parameters for each of the layers.

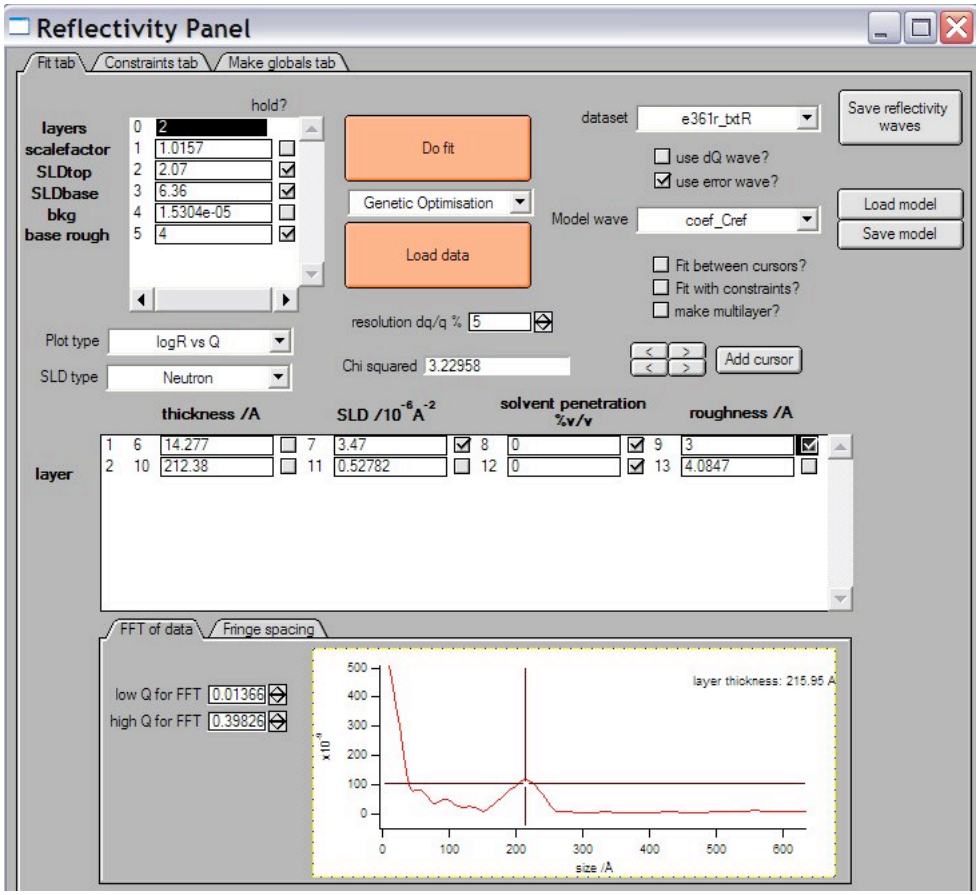


Figure 4: The reflectivity fit panel, setup for a two layer system.

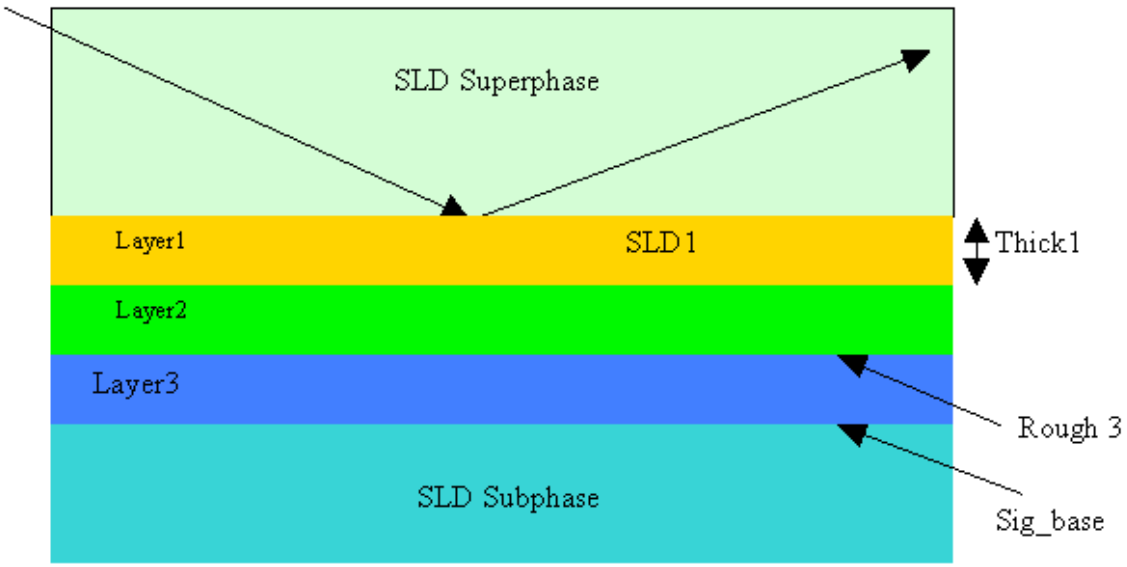


Figure 5: The *Motofit* model

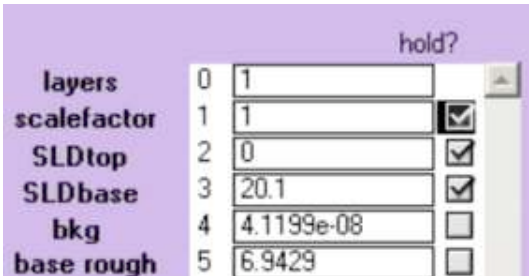


Figure 6: Parameters required for the basemodel. The numbers 0 to 5 represent the parameter number (which will be needed to set constraints)

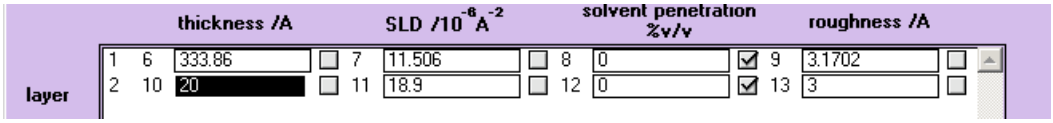


Figure 7: Parameters for each of the layers. The number at the far left hand side of the white box indicates the layer number. The number to the left of each parameter is the parameter number (which is needed to set constraints).

- THICK : thickness of the each layer (in Angstrom).
- SLD : scattering length density of the first layer.
- SOLV : how much of the subphase (solvent) penetrates into layer 1. This is expressed as a percentage.
- ROUGH : the roughness at the top of the first layer (in Angstrom).

All the scattering length densities are expressed as $\rho/10^{-6}$Å^{-2}. Roughnesses are implemented in terms of an error function, according to Nevot+Croce[**?**]. In this situation the roughness is the standard deviation of the erf function (Reflfit from NIST uses FWHM for this value, which is approx. 2.35 times larger).

When you right click on a SLD cell (please see 7.2), a prompt appears that asks if you would like to set the SLD value from the scattering length database. . When the user selects a chemical using this prompt then the SLD is automatically inserted into the relevant SLD parameter. This is useful for quick initialisation of models. You can choose if an X-ray or neutron value is imported by changing the SLD type.

The checkboxes in these tables are used to define which parameters are held during the fit (held=checked).

The model can be saved by using the *Save model* button. If you want to use a saved parameter file as a starting point then you can use *Load model*, then select the parameter file in the *model wave* popup.

5.2.2 Load Data

This button loads experimental data. The program asks if you want to append the data to the theoretical reflectivity curve. When you load data the program removes NaN's and sorts the data as increasing Q_Z. You can load either 2, 3 or 4 column data:

Two column data is Q,R

Three column data is Q, R, dR

Four column data is Q, R, dR, dQ.

Q, and its associated uncertainty dQ (FWHM, not standard deviation) are measured in reciprocal Angstroms, $Å^{-1}$. Only Q and R are required for a fit. The uncertainty (standard deviation) in reflectivity, dR is required if you wish to weight the data according to counting statistics. dQ is required if you wish to use resolution smearing on a point by point basis.

5.2.3 Dataset / Model Wave

These popup menus select the experimental datasets that you wish to fit. The default coefficient (parameter) wave is *coef_ Cref.* However, each time you fit a different set of data another coefficient wave is produced (*coef_ Rwave*), corresponding to the fit coefficients. You can use these coefficients as a starting point for a new fit. (However, if you are fitting the same dataset then beware, the starting coefficients get overwritten).

You can save and load coefficient waves using the *save and load model* buttons. This saves a lot of effort.

The coefficient wave is saved as an IGOR text file (**.itx*). This means that you can double click on the file and it loads straight into IGOR, and initialises the reflectivity panel. The saved coefficient wave remembers the exact state of how the data was fitted (including which parameters were held).

5.2.4 Use weight wave

If you check this box the fitting process uses the statistical errors on each of the datapoints to weight the fit. See the FAQ section for different weighting scenarios. Please note that your dataset should contain

5.2.5 Plot type

You can change the plot type by simply using this pop up menu. The data fitting operates on the sample plot-type. E.g. if you select RQ4 vs Q, you fit the data as RQ4 vs Q. The plots update as you change the selection in this box.

5.2.6 Checkboxes

If you want to hold a parameter while fitting then tick the checkbox. If left unchecked then the parameter will vary in the fitting process.

5.2.7 Resolution

There are two different ways of handling the resolution smearing. The first method assumes that the resolution of the instrument is a constant percentage, e.g. *dq/q=5%*, where dq is defined in 9. This is often the case if the instrument measures at constant footprint. This value resides in the wave *resolution[0].*

$$\left(\frac{\Delta Q}{Q}\right)^2 = \left(\frac{\Delta\theta}{\theta}\right)^2 + \left(\frac{\Delta\lambda}{\lambda}\right)^2 \tag{9}$$

In 9 ΔQ is the uncertainty in Q, $\Delta\theta$ is the uncertainty (FWHM) in the incident angle and $\Delta\lambda$ is the uncertainty (FWHM) in the wavelength of the incident radiation. Please note that Motofit assumes that ΔQ is the FWHM of the gaussian, not the standard deviation.

If the data is from a nice instrument that outputs the resolution function as an extra (4th) column (containing dQ values) in the data file, then you can use this to resolution smear the data point by point. This is handy if the resolution varies across the Q range. This is often the case if a single slit setting is used.

The first method is automatically used (as it's faster). To use point by point resolution smearing the select the *use dQ wave* checkbox. Note that the second option is only used when you are fitting, it's not used while you are setting the model up.

In both cases smearing is done by gaussian convolution. If you use *constant dq/q* then the convolution is handled using IGOR's convolve operation and is quite fast. If you select the *use dq wave* option then the convolution is performed by the following calculation:

```
y=f(w,z)
y+=0.1354*f(w,z+2*(dq/2.35482))
y+=0.1354*f(w,z-2*(dq/2.35482))
y+=0.24935*f(w,z-1.666667*(dq/2.35482))
y+=0.24935*f(w,z+1.666667*(dq/2.35482))
y+=0.4111*f(w,z-1.333333*(dq/2.35482))
y+=0.4111*f(w,z+1.333333*(dq/2.35482))
y+=0.60653*f(w,z-1*(dq/2.35482))
y+=0.60653*f(w,z+1*(dq/2.35482))
y+=0.80074*f(w,z-0.6666667*(dq/2.35482))
y+=0.80074*f(w,z+0.6666667*(dq/2.35482))
y+=0.94596*f(w,z-0.3333333*(dq/2.35482))
y+=0.94596*f(w,z+0.3333333*(dq/2.35482))
y *= 0.137023
```

5.2.8 Fit between cursors

If you want to limit the region of the dataset that the fit function uses then tick this box. Note that you should have both cursors (A and B) on the *dataset* you are fitting.

5.2.9 Use constraints

If you want to use constraints then you should tick this checkbox. Constraints are numbered with respect to their parameter. The parameter numbers are listed in the parameter selection boxes. For example, the roughness of the first layer is referred to as K9, whilst the thickness of the first layer is K6. The constraints should be entered in the constraints tab, using *add or remove constraint*. You should not leave any blank boxes, otherwise nothing will happen when you try to do the fit. You can either enter simple constraints, e.g. K6>5 or complicated constraints (K10>5*K2). You can also make equality constraints (special to *Motofit*), e.g. K5=K10. For further details on valid constraints please refer to the Wavemetrics manual.

One often used example is when all the roughness are constrained. If you have a two layer system then you will have three roughness parameters, K5, K9 and K13. If you want to make these equal during fitting then you have to enter two constraints.

- K9=K5
- K13=K5

NOTE: If you have ticked the use constraints box, then you must have at least one constraint, or nothing will appear to happen. In a similar manner, nothing will happen if you enter a constraint then don't allow it to vary when you do the fit.

When you have a multilayer system, then the coefficients of the reflectivity panel take precedence, then the coefficients of the multilayer panel follow on. (if you have two layers in the "base" model then there will be parameters up to K13. The first parameter in the multilayer panel is then referred to as K14). The parameter number for the a multilayer system is simply number of params in base model + multilayer parameter number.

CONSTRAINTS DON'T APPLY DURING GENETIC OPTIMISATION!

5.2.10 Make multilayer

This sets up the multilayer panel for analysing multilayer repeat systems. The multilayer created by the multilayer panel is inserted in the base model (the layer system contained in the reflectivity panel), at a position determined by the user. The multilayer is defined in terms of the number of repeats and how many individual layers are in each of those repeats. For a clearer picture try messing around with the test dataset e179r, using the test coefficients provided.

This option will be really useful for people studying samples such as cadmium arachidate.

Note that in this case you are fitting the data with parameters contained in *coef_ MultiCref*, not *coef_ Cref.* For more guidance examine the e179r dataset that is provided as an example.

5.2.11 Do fit

Fits the experimental dataset with the theoretical model. You have a choice of using Genetic Optimisation or Levenberg-Marquardt as the fitting algorithm. These methods minimise the Chi2 value. The fitfunc that IGOR uses is called *Motofit.*

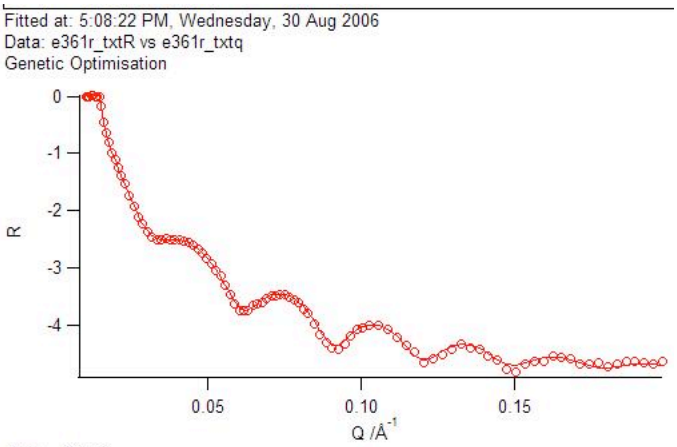
Output

1. Fit curves for the data you analysed. An SLD profile is also appended to the SLDplot
2. a coefficient wave called *coef_ Rwave* (which can be printed/saved as Igor text files)
3. a wave, *W_ Sigma*, containing the uncertainty in each of the fit parameters
4. The normalised correlation matrix is contained in a wave called *M_ Covariance*
5. An IGOR notebook called 'Reflectivity Fitting Report' is produced. This notebook contains a record of all the reflectometry fits that you perform in an experiment (containing information on fit parameters, fit errors, a graph of the fitted data, the Chi2 value, time, date, and the method of fit). If you have the full version of IGOR this notebook can be saved in RTF format (readable by Word, etc) or HTML format, etc.

Genetic Optimisation This method is very efficient at finding global minima in the Chi2 function, even if the starting guesses are far from the final solution. The algorithm used in the program is a differential evolution technique from Wormington et al[**?**] and Storn and Price[**?**]. This method should be the first point of call if confronted with a difficult reflectivity pattern. I suggest using this as the default fitting procedure.

The program asks for information to set up the genetic optimisation.

- *Number of generations.* This is effectively the number of iterations you want to do. The more generations you use, the longer it takes, but you should expect a closer answer. I suggest trying 100 generations to start with.
- *Mutation constant.* The mutation constant is used to create genetic diversity. A larger number means more genetic diversity, but may mean that the solution takes longer to find. This value should lie between 0 and 1. Start out with 0.7 and see how you go.
- *Population size multiplier.* The genetic algorithm works by starting off with a large population of initial randomised solutions (that lie in between the limits you set), which are then refined. The size of this population is set by *popsizemultiplier*the number of free parameters.* If you have a large parameter space to examine, then it may help to set this value larger, but 10 has worked in the past.



Chi2 = 3.2296
2 +/- 0
1.0157 +/- 0.0052743
2.07 +/- 0
6.36 +/- 0
1.5304e-05 +/- 5.133e-07
4 +/- 0
layer 1: 14.277 +/- 0.78871 3.47 +/- 0 0 +/- 0 3 +/- 0
layer 2: 212.38 +/- 0.48378 0.52782 +/- 0.029414 0 +/- 0 4.0847 +/- 1.0756

Figure 8: A report is produced for each fit, so you can compare different approaches.

- *Recombination constant.* Should lie between 0 and 1. This controls how much of the original guess is used in the evolved guess. If you use a low number then the original guess will not evolve that much. If you use a large number then the guess will evolve a lot, making it more likely to find a solution, but taking longer in the process. Try starting out with 0.5.
- *Fractional tolerance to terminate fit.* If the fractional decrease in the best chi2 value is less than this value then the fit will terminate.

The user is then presented with a table showing the initial guesses, the limits to be placed on those parameters during the fit. The optimisation is unable to search for solutions outside those bounds. When you are satisfied with the setup press *continue* and the optimisation will start. If the limits are not set correctly (e.g. your upper limit is less than your lower limit) you will be asked to set them up again.

While the optimisation is proceeding the updated Chi2 value will be displayed in a graph(9). This graph is used to show the colourised fit parameters for each of the members of the genetic population. As the fit proceeds towards convergence then the colour for a particular parameter, over all the population, should converge.

On finishing the optimisation (or aborting it part way through) a coefficient wave and fitwaves will be produced, a pointwise estimate of the parameter uncertainties is contained in the wave *W_Sigma*.

NOTE: CONSTRAINTS DO NOT APPLY DURING GENETIC OPTIMISATION.

Levenberg-Marquardt The Levenberg-Marquardt (LM) method is a much faster and efficient method for fitting than genetic optimisation. However, it only has a good chance of working if your initial guess is close to the final fit, as it has problems escaping from local minima in the Chi2 function. It is often best to use genetic optimisation to find the best solution, then employ LM to speedily take you the rest of the way. LM has the advantage of being able to employ inter-parameter constraints, whereas genetic optimisation is only able to set boundary limits on parameters.

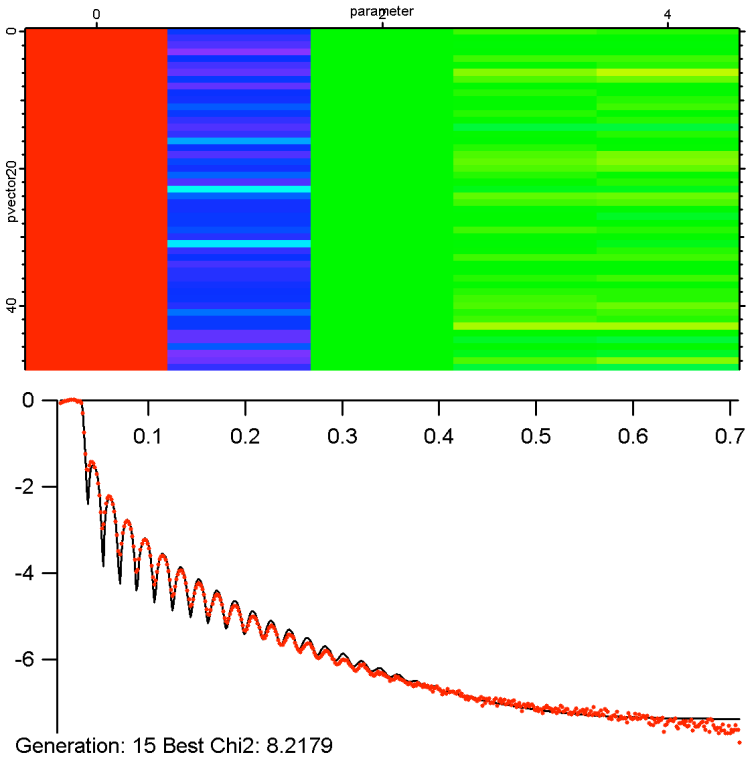


Figure 9: Following genetic optimisation on some X-ray reflectometry data. The fit is converging. Note the Chi2 value in the bottom corner.

Genetic - LM This option performs a genetic optimisation followed by a Levenberg-Marquardt fit, the best of both worlds.

NOTE:

1. Remember to check the final answer, to make sure the model is something that is physically realistic.
2. Negative numbers may appear as fit results, for the thickness and roughness. DON'T PANIC. Internally in *Motofit* the absolute value of these numbers are taken, i.e. abs(thickness).
3. *Coef_ Cref* is always used as the coefficient wave for the fit. If you specify the use of a different coefficient wave using the popup, then the action of selecting the popup makes *Coef_ Cref* equal to *Coef_ otherwave.*

5.2.12 FFT of data / Fringe spacing

The Fourier transform of the selected dataset (from the dataset popup) is displayed in the bottom of the reflectivity panel. If you have oscillations in your reflectivity curve you will get a peak in the fourier transform that corresponds to your layer thickness. Use the cursor in the Fourier transform plot to get values for each of the peak positions. This option is excellent for setting up fits.

If you have many layers present then you will get multiple peaks present in the fourier transform. You can change the region over which the FFT is applied. The low Q region should be above the critical edge of the data, and the high Q end should stop before the background is reached.

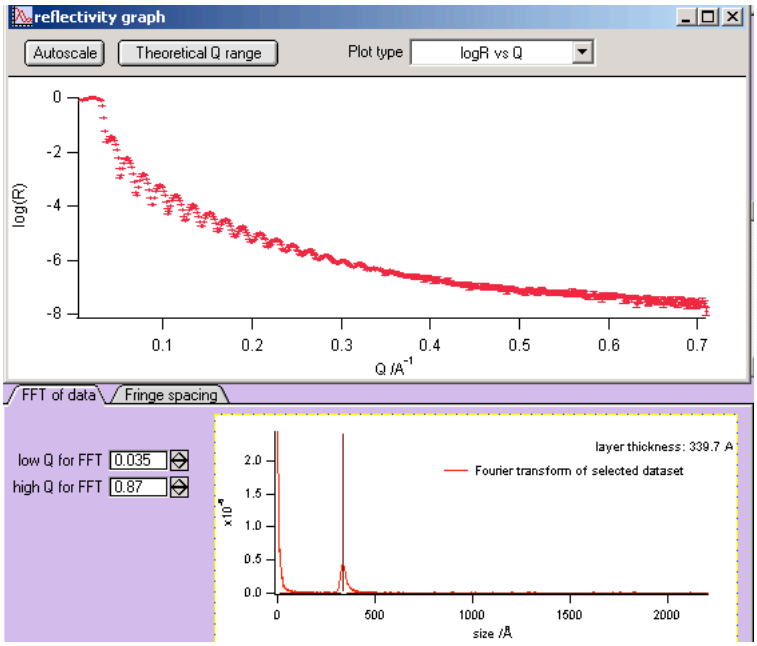


Figure 10: Fourier transform of some X-ray data. This data fitted to a 338A layer

(for the techno-nerd: the reflectivity curve is multiplied by Q4, then Fourier transformed with an FFT. The data is zero padded before the fourier transform, and a cosine window is applied during the transform).

If you have multiple Kiessig fringes then you can estimate layer thicknesses by assessing the FRINGE SPACING.

Put the cursors on the reflectivity graph by pressing *Cursors*. This button appends the cursors to the trace in the *Dataset* popup. Move the cursors until you have selected an integer number of fringes, change the number of fringes, the layer thickness is then displayed.

5.2.13 χ^2 value and other statistics

As the model is updated the χ^2value is also updated, which tells you if your fit is improving. The χ^2 value is normalised to the number of points.

$$\chi^2 = \sum_{n=1}^{L} \frac{1}{L-P} \left(\frac{y_{n,obs} - y_{n,calc}}{y_{n,error}} \right)^2 \tag{10}$$

After fitting data the normalised correlation matrix (*M_ Covariance*) is produced in the root datafolder, which indicates interdependence of fit parameters. To see this correlation matrix type *Edit M_ Covariance* in the command line.

5.3 Reflectivity Graph

This plot shows the theoretical reflectivity curve (3) for the layer model specified in the Reflectivity Fit Panel. Whenever you change a parameter in the panel then the graph will update. The main part of the graph is the theoretical reflectivity profile, the other part the SLD profile corresponding to your model.

If you wish to change the theoretical Q range of the plot then select the *Change Q range of theoretical data*. You can also change the plot type.

The SLD profile is calculated using 11. Where N is the total number of layers, z is the distance from the top interface and *erf* is the error function.

$$\rho_z = \sum_{i=0}^{N} \frac{\rho_i - \rho_{i+1}}{2} \left(1 + \mathrm{erf}\left(\frac{z - z_i}{\sqrt{2}\sigma_i}\right)\right) \tag{11}$$

If you have the full version of IGOR you can export the graph in whatever way you would like. This manual was created with the EPS export.

5.4 Load Reflectivity data

When you bring up this option from the top menu you load in 2, 3 or 4 column reflectivity data. The data format is tab delimited, with the data columns as:

Q, R, dR, dQ

If you have a graph already open (e.g. you have already opened the reflectivity panel), and this graph is on top, then it asks if you want to append it to that graph, which is quite useful. In other words, you see your experimental data and the theoretical data on the same plot. If there is no graph already open, then it creates a new graph. You can then open (and append) further reflectivity datasets to this graph.

5.5 Fitting Multiple Contrasts / Global Fitting

Fitting multiple (neutron + X-ray) contrasts is a cinch in *Motofit*. You use the global fit package. Simply follow the following steps.

1. Load all the experimental data (preferably in the same graph).
2. Fit the individual datasets/contrasts separately to start off with, with a best guess model. This should give you a good starting point. You should use the same number of layers for each contrast. If you can't do this for each contrast, then the model is probably incorrect.
3. This procedure should give you coefficient waves, *coef_ Rwave*, for each of your datasets.
4. Select the *Make globals* tab. Enter the correct number of layers, and the correct number of contrasts. Now select which parameters you want to be "global", i.e. the same for all the datasets.
5. Select the coefficient waves for each of the datasets. Note: the program uses the names of the coefficient waves to set up the fit, so they must tally with the dataset. For example, the coefficient wave is *coef_ e361r_ datR*; so the reflectivity wave *e361r_ datR* and a Q wave *e361r_ datQ* MUST exist.
6. Now press *Make Global Coefficient* wave. At this step you should have the global fitting dialogue there.
7. Select the parameters you want to hold. Don't forget to hold the number of layers, this should be invariant!
8. If you want to weight the datasets, or make constraints, this is the time to do it.
9. Press "*Fit*". Choose either Levenberg Marquardt or Genetic Optimisation.

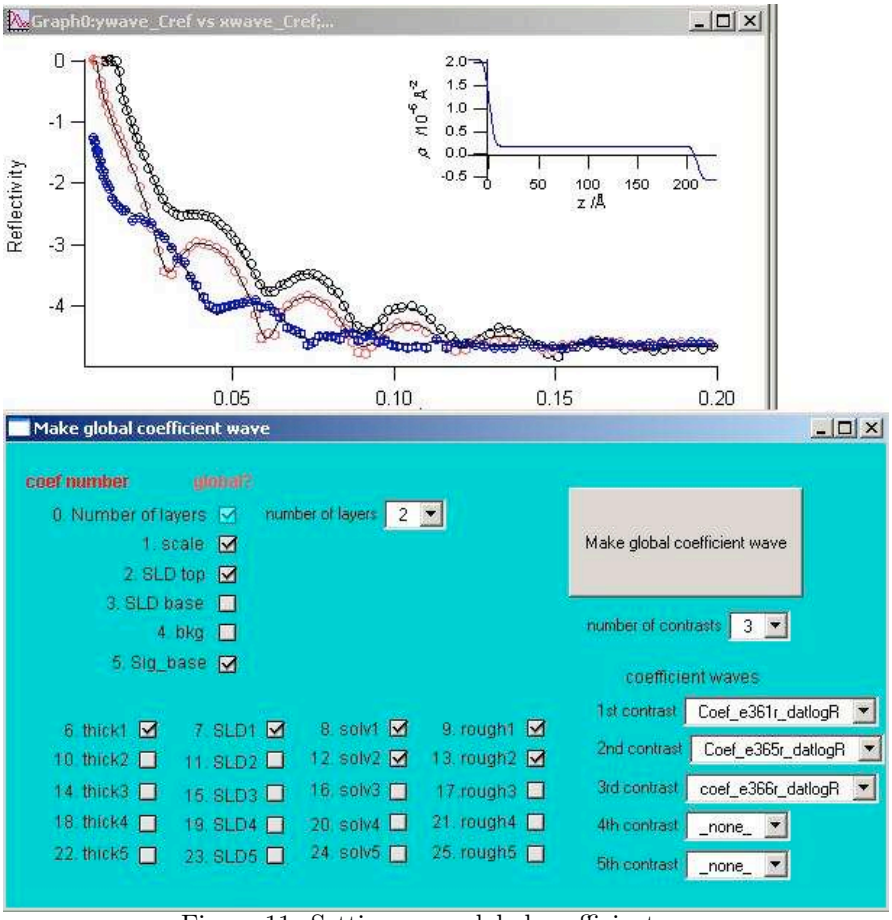


Figure 11: Setting up a global coefficient wave

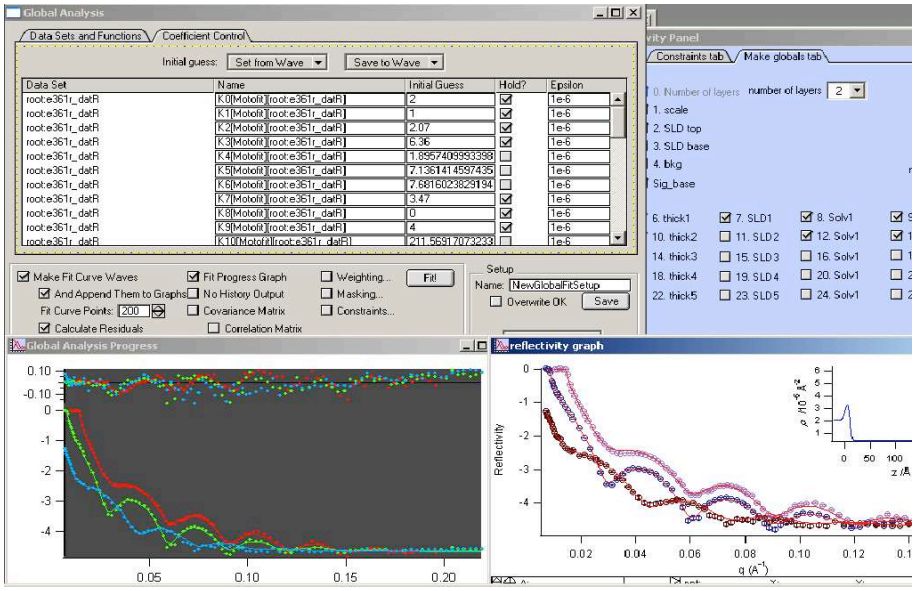


Figure 12: The global fit dialogue

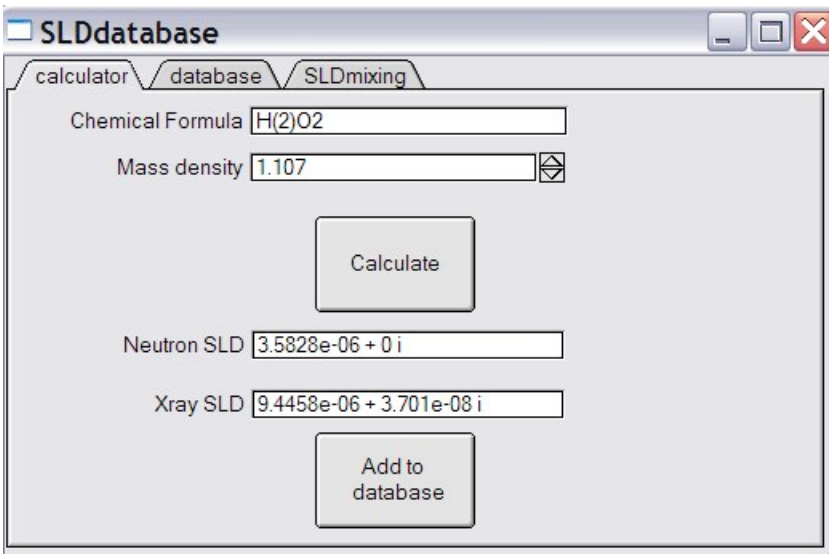


Figure 13: The scattering length density database panel

The global fit appends fit curves to the top graph. Unfortunately at this stage the SLD profiles don't update.

1. Note that you don't have to use the panel as described, you can enter all the parameters by hand. It just makes setting up the global wave a LOT QUICKER!!!
2. The global fitting only uses the constant dQ/Q option at the moment (i.e. It doesn't use the relevant dQ wave, if it exists).
3. You can fit X-ray and Neutron data at the same time.
4. Global fit is extremely powerful, and you can make more complicated linkages than described here. For example one can link parameter x of dataset y, with parameter a of dataset b. This is particularly suited when you would like to fit a air-solid dataset at the same time as a solid-liquid dataset.
5. The constraints you entered for fitting a single dataset in the reflectivity panel no longer apply for global fitting, you have to enter them in the global fit panel. Note that any entered constraints only apply to the Levenberg Marquardt fit, they don't apply to Genetic Optimisation!

5.6 Scattering Length Density Database

As Motofit v3.04 a scattering length density calculator is integrated into the program, along with the SLD database. These features can be accessed from the reflectivity menu, or from the SLDdatabase panel created when Motofit is initialised.

Figure 13 demonstrates a scattering length density calculation. One enters the chemical formula (including isotope information) and density and the Neutron and X-ray scattering lengths (8047eV) are calculated for you. The typing of the chemical formula is strict and case sensitive. One can use fractional atomic compositions

Formulas that work:

- H(2)2O - heavy water (this is also the same as D2O)
- D2O

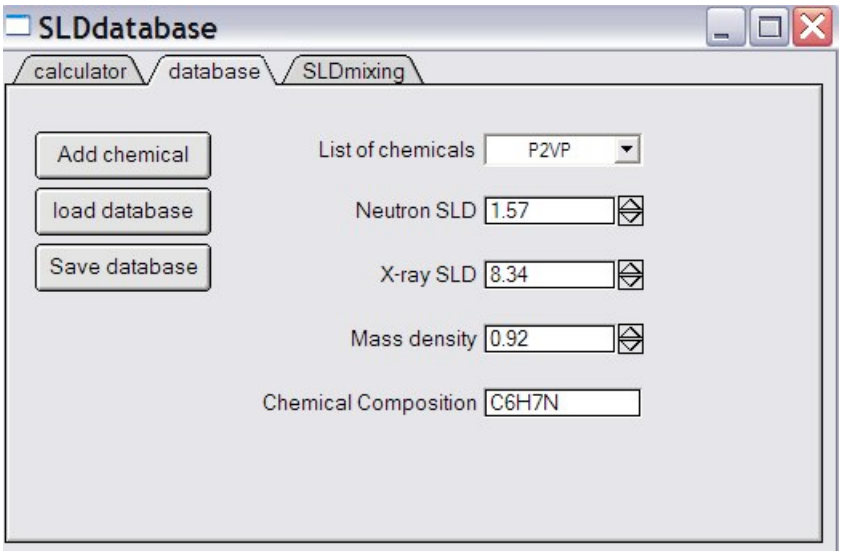


Figure 14: The SLD database holds a list of common chemicals

- Al2O3
- AlO
- Al1O1
- Al2.023O3.056 (fraction compositions allowed)

Formulas that don't work

- h(2)2o (Elements are case sensitive)
- al2O3 (Elements are case sensitive)
- H()2O (there is no isotope information)
- H(100)2O (there is no isotope of Hydrogen with weight of 100).

The SLD calculation is performed using the method described in Section **??**. The calculator uses the neutron lengths tabulated at the NIST `http://www.ncnr.nist.gov/resources/n-lengths/` website. The X-ray scattering lengths were obtained from the scattering factors on the Lawrence Livermore `http://www-phys.llnl.gov/Research/scattering/asf.html` website.

You can then enter the calculated value into the SLD database, on the next tab.

If you can't remember all the SLD values that you use then you can add them to the database. You can add as many new values as you like. (You can edit the database using Excel as well). The SLD database is automatically loaded when the reflectivity macro's are initialised, so should be present in the *Motofit* folder. Values from the SLD database can be automatically inserted into a structural model by right clicking in the SLD cells in the reflectivity panel (section 7.2). The format for new lines of the SLDdatabase is:

Chemname <tab> chemformula <tab> SLDneut <tab> SLDXray <tab> massdensity.

The last tab of the SLDdatabase panel enables the user to calculate SLD's from the ideal mixing of two components.

5.7 The curve fitting dialogue

Although the panel can handle as many layers as you want sometimes you may want to use the curvefitting dialogue. Heres what you have to do.

1. Bring the parameter table, *coef_ Cref*, to the front.
2. Enter the required number of layers in *coef_ Cref*[0]. (Wave numbering starts from 0).
3. You will need *4n+6* parameters in total, so type in: *Redimension/n=(4n+6) coef_ Cref* in the command line. where *4n+6* is a number. Alternatively you can enter the numbers directly into the table if you want to. The extra parameters will have the following index numbers: *thick(n)=4n+2, SLD(n)= 4n+3, solv(n)= 4n+4, rough(n)= 4n+5.* i.e. the SLD of the 7th layer should be in coef_Cref[31].
4. You have to have the correct number of parameters for the number of layers entered.
5. Start the curve fitting dialogue, Analysis Curve Fitting.
6. In the Function tab select *Motofit* as the fitfunc and select the correct y and x waves.
7. If you want to use the cursor positions to select a particular fit range then press cursors on the data options tab.
8. In the coefficients tab select the *coef_ Cref* as the coefficient wave. Select the hold checkbox if you want to hold that parameter. You HAVE to hold the first parameter.
9. If you have more parameters than are required for the number of layers then hold the extra parameters.
10. If you want to enter constaints use the constraints wave.
11. Press Do-it. The curve will hopefully fit without a problem

Note that all the reflectivity panel does is perform the same procedure as outlined in the 11 steps above.

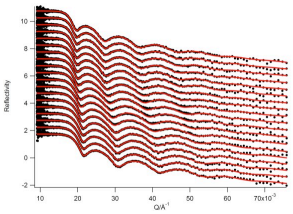
IGOR can perform a lot more advanced functions than those outlined above. You should be able to find more options to suit you, if you want them. (You can get covariance matrices, residual waves, etc).

5.8 Batch fitting datasets

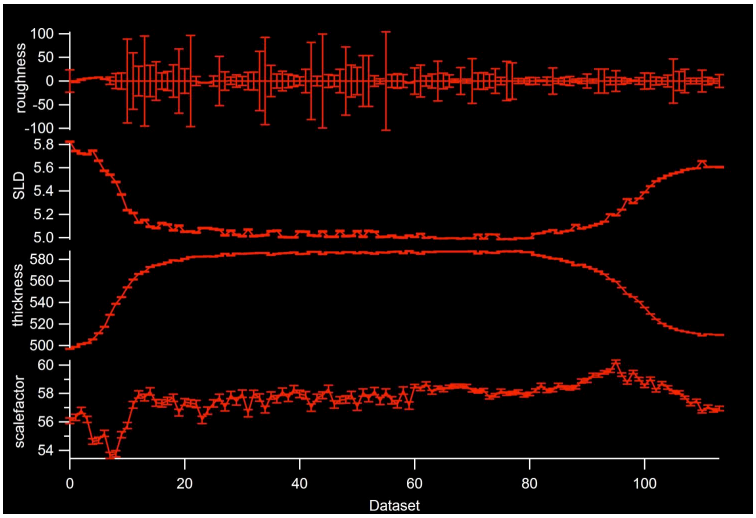
This function is essential for people with many kinetic datasets (e.g. >5). When you select *Load batch data* (from the *Reflectivity* menu) you are prompted for a folder in which your data resides. The program then loads in all the files in that directory, with the extension you provide. Please take care, if you specify loading all ".txt" files, then move any non-reflectivity text files from that directory beforehand. The program only loads 3 column reflectivity data (Q, R, dR). You can also specify an offset for the plotting, this offsets all the datasets in the batchdata graph so that you can see each of them clearly. The data is placed in the *MOTOFIT_ batchfits* datafolder.

Now use the reflectivity panel to set up the parameters for the first dataset (you should try fitting that dataset before loading the batch data). You can fit with Genetic Optimisation or Levenberg Marquardt (or both), if you have a lot of datasets Genetic Optimisation will take a long time. If you use Levenberg Marquardt you can use constraints, etc and the fit will be a lot faster.

Now select *fit batch data* from the menu *Motofit->Batchdata*. This fits the first dataset (as normal), using the parameters in the reflectivity panel as a starting point. It then uses the fit parameters from that first dataset as a starting point for the second dataset. The program then cycles through all the datasets fitting them all. In testing around 100 files



(a) These datasets were fitted in <10 seconds



(b) The output from the entire batch fit process

Figure 15: Batchfitting is easy in Motofit

were fitted in 5 minutes. On installing the AbelesXOP this time decreased to 20seconds (with Levenberg Marquardt)!!! All the fit curves, coefficients and errors are collected as the batch fit proceeds. Comprehensive graphs detailing the fits and trends in parameters are available at the end, ready to export to your Nature paper.

5.9 Create local chi2map for requested parameter

This option is available from the *Motofit* menu. You can use this to explore how Chi2 varies for selected parameters, around their current values.

If you choose the 1D version you are prompted to enter a parameter number (available from the reflectivity panel), and a range to explore (expressed as a percentage of the value). A graph is then displayed of Chi2 versus parameter value.

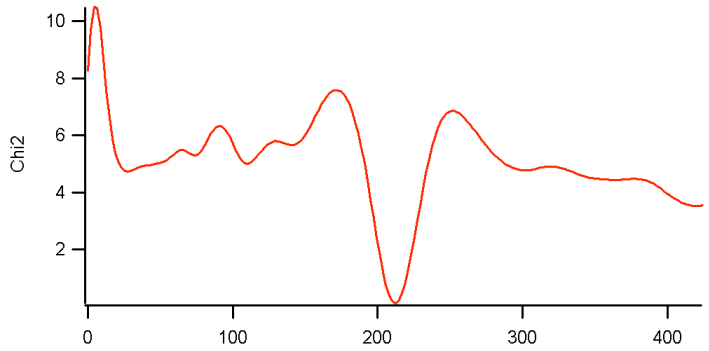


Figure 16: Layer thickness Chi2 map for a sample with well defined Kiessig Fringes (212A)

The 2D version allows you to examine a Chi2 surface, as a function of two parameters. For example you can look at how Chi2 varies as a function of layer thickness and layer scattering length density.

5.10 Fitting user defined functions.

If you have a specific profile to fit, then this will probably replace *Motofit*. In this situation you will have to use the curve-fitting dialogue, as the panel will be useless. You will create a specific parameter wave for your model, which will be sent to a user function. This function will then create a specific *coef_Cref* wave which can be read by the kernel that calculates the reflectivity, *Abelesreflectivity*. This probably sounds harder than it really is. If you want help then email me and I'll try to help.

5.11 Test datasets

The distribution package contains several test datasets (can you fit them?). Also included are the coefficient waves for each dataset. The resolution for these experiments is dQ/Q = 4%

Dataset e179 is a multilayer sample. This sample consists of a two layer stack of Ti/Ni, which has 25 repeats. The multilayer is appended to layer 0.

e361r, e365r and e366r are solid-liquid datasets of silicon vs water. On top of the silicon substrate is a native SiO_2 layer, on top of that is a polymer layer; which makes it a two layer system. The only difference between the different datasets is the scattering length density of the water phase. Of the 14 parameters (0 –> 13 inclusive) the only parameters that aren't global are: *SLDbase, bkg, SLD* (layer 2). Therefore these datasets are ideal for the global fit routines.

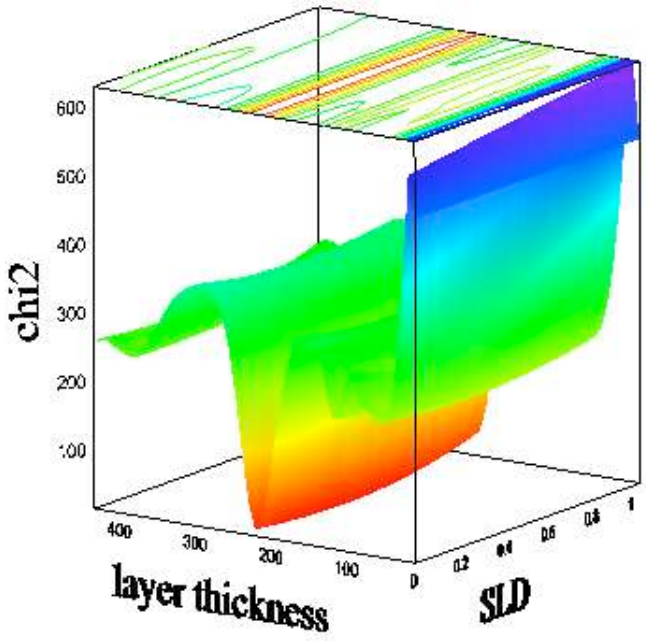


Figure 17: A 2D chi2map of layer thickness and SLD

Sitest.txt is a solid air experiment of a silicon wafer vs air. The Si wafer has a native oxide layer (SiO_2) of around 15A thickness on top.

6 Useful functions that not many people know about

To access these functions you need to type commands in the "command line". To pull this window up type Ctrl-J.

6.1 getrid()

This helper function gets rid of all graph windows, all tables, and kills all loaded waves. Instead of this you could type Ctrl-N, which starts a new experiment.

6.2 moto_holdall()

Holds all the parameters in the reflectivity panel. Useful when you've used globalfit and want to update the SLD profiles. The only way to do that is by performing a "spoof fit", i.e. performing a fit on an individual dataset, with all the parameters held.

6.3 moto_restoremodel()

If you perform a fit and you don't like what happens to the parameters, then you can use this command to "revert" to the initial guesses.

6.4 moto_backupmodel()

This makes a backup of the current guesses. e.g. You can change several parameters and if you don't like the new theoretical curve you can revert the the backed up guesses using moto_restoremodel(). The backup is overridden when you perform a fit.

6.5 moto_offsetq(qwave,thetaoffset,lambda)

If your sample is misaligned, then you can correct your q values (contained in the wave qwave), assuming you know the angular offset (thetaoffset) and the wavelength of the radiation (lambda).

7 Frequently asked questions

7.1 How do I calculate a scattering length density?

The easiest way is to use the inbuilt SLD calculator.

But you should also be able to calculate them for yourself. The formula for calculating a scattering length density is given in 12. In this equation V_m is the molecular volume of the material and b_{c_i} is the bound coherent scattering length of the i th atom of a molecule with n atoms.

$$\rho = \frac{\sum_{i=1}^{n} b_{c_i}}{V_m} \tag{12}$$

You can also write this as 13, where N_a is Avogadro's number, ρ_{mass} is the mass density of the material and M_R is its relative molecular mass.

$$\rho = \frac{N_a \rho_{mass}}{M_R} \times \sum_{i=1}^{n} b_{c_i} \tag{13}$$

You can find a table of neutron scattering lengths at:

- `http://www.ncnr.nist.gov/resources/n-lengths/`

For X-rays you have to calculate the scattering length for each atom before doing the calculation, 14.

$$b_i = \frac{e^2}{4\pi\epsilon_0 mc^2} f_{1i} \tag{14}$$

where e is the charge on a single electron, ϵ_0 is the permittivity of free space, m is the mass of an electron and c is the speed of light. f_{1i} is the scattering factor for an atom of element i and may be found from:

- `http://www-phys.llnl.gov/Research/scattering/asf.html`

To use this website you will need to know the wavelength of the X-ray radiation that you are using (1.54Å=8047eV).

This is an exact way of calculating b_i, a much quicker approximation is to use 15, where Z is the atomic number and $r_e$is the Compton radius ($r_e = 2.8179 \times 10^{-5}$Å)

$$b_i = Zr_e \tag{15}$$

To find out how to enter the scattering length densities are entered into the program please see Section 5.2.1.

7.2 Why does right clicking in the SLD cells in the reflectivity panel not always work.

Right clicking in the SLD cells of the reflectivity panel has the ability to bring up a dialogue, from which you can use the SLD database to enter values into your model. This procedure always works, but you need to hold down the right mouse button until the dialogue appears, as it may take some time to show. The only time the dialogue will not appear if you right click, is when you are already editing that SLD cell. In that case select a different cell, then right click in the SLD cell of interest.

7.3 Why can't I install the program properly?

Did you follow the installation instructions carefully? If you think you have:

- make sure that you have loaded all the required files.
- update to the latest version of IGOR.
- if all these steps fail then please email:
  `mailto:Andrew_Nelson@users.sourceforge.net`

7.4 Do I need to install Abeles.xop?

Abeles.xop is a pre-compiled function written in C that calculates reflectivity. Although it is not necessary to install this file, I strongly recommend it, as it makes the program run around 5-6 times faster. This means that if you have lots of layers, or prefer to fit with Genetic Optimisation, or want to fit batch data, then the fit is blazing fast, and you don't need to go for a cup of tea in the meantime. There is a link on the *Motofit* website (http://motofit.sourceforge.net `http://motofit.sourceforge.net`) directed towards the XOP.

7.5 Installation issues for the XOP?

Please refer to installation section (4). Note that the XOP is platform dependent, so please don't install the Windows version on a MAC, and vice versa.

7.6 What is a wave in IGOR?

A wave is a data object, which contains a column of data, similar to Excel. When you load a reflectivity dataset with Q, R and dR, you are loading 3 waves. One wave contains the Q values, and there is a wave for the Reflectivity values and a wave for the error in reflectivity values. Obviously Q,R and dR should have the same number of points. Each of the waves has a different name. If you opened the same dataset in Excel you would see 3 columns containing the Q,R and dR values. IGOR is a program that performs mathematical operations on these waves.

7.7 What is a coefficient wave?

A coefficient wave is a wave containing the parameters for the fit. They can be saved as IGOR text files (*.itx), which have some extra advantages to normal text files, try double clicking on one of the coefficient files.

7.8 What should I call my coefficient waves when I save them to disk?

You can save them as anything, but call it something you will remember.

7.9 Why are the layer thicknesses and roughnesses sometimes negative after the fit?

Motofit uses the modulus of the thickness and roughness values, so there is no need to worry.

7.10 The model doesn't seem to be sensitive to roughness?

Sometimes you may fit some data and some of the roughness parameters appear to go haywire. This is because in a lot of situations the modelled data isn't sensitive to roughness and when you start to fit the roughnesses oscillate dramatically. This often occurs in situations where there aren't many features in the reflectivity profile and where the data doesn't extend to large Q values. To combat this change the troublesome roughnesses to a realistic value, then fix them during the fit. Once you get close to a good solution then this problem usually gets better. Be sure to look at the correlation matrix to determine any problems.

7.11 Can I fit X-ray data and Neutron data at the same time?

Using global fit you can fit as many neutron contrasts as you like. You can also fit X-ray and neutron data together. However, difficulties may arise because X-ray datasets normally have much better statistics than neutron data. Therefore when you try to fit neutrons and X-rays together the data is sometimes preferentially weighted towards the X-ray set.

7.12 What format should my dataset be in?

The dataset should have either 2,3 or 4 columns. If there is a two column dataset then the columns should consist of Q and R values. For a three column dataset you should have Q, R and dR (where dR is error in reflectivity). For a four column dataset you should have Q, R, dR and dQ (where dQ is uncertainty in Q). The reflectivity values should be a value between 0 and 1. i.e. you cant load in Q, logR, dR or Q, RQ^4, dR data. You should get rid of errant datapoints which have $R<0$ (which may happen if you subtract a background). All columns should be the same length. For best results the file should be saved without a header (i.e. the data starts in the first line), and in the tab delimited format. This file manipulation is easy in Excel.

7.13 Why doesn't the global fit produce SLD profiles for each dataset?

The global fit procedures don't produce SLD profiles for each dataset and there is no intention at the moment to implement this functionality. To get around this and get SLD profiles for each dataset it is suggested that you fit each dataset individually, but hold all the parameters.

7.14 What is a global parameter in IGOR?

If you have several datasets on a particular system, and the data can be described by the same structure, then some of the fit parameters may be the same for all the different datasets. Such parameters are referred to as global because they are common to all the datasets.

7.15 When should I use Levenberg-Marquardt or genetic optimisation?

I would use Genetic Optimisation to perform the fitting (for both single datasets and multiple contrast fitting), then follow up with Levenberg Marquardt to calculate errors, etc. This is because Genetic Optimisation is very good at finding global minima from poor initial guesses, but Levenberg Marquardt is much faster.

7.16 Can I fit normalised Fresnel reflectivity curves?

Yes! Thats what the *Fresnel.ipf* file is for. However, if you want to fit curves like this you will have to use the curve fitting dialogue. Follow these steps: It will help if you are acquainted with the curve fitting dialogue, since all the curve fitting will go on in there.

- Load the experimental datasets that you measured, using the "*Load data*" button.
- Make sure that the plot type is "*R vs Q*" for all the steps (important).
- subtract the background from your dataset, you can find out the name of the dataset in IGOR by using the databrowser (Data –> Databrowser) OR by using the *dataset* popup menu in the reflectivity panel. The R data will be something like *dataset_ extR*.
- You subtract the background by using this command in the command window (ctrl-J): *dataset_ extR=dataset_ extR-backgroundvalue*
- You then need to normalise the dataset by the Fresnel reflectivity. (still keep the plot type as "R vs Q").
- First put the correct parameters in the Reflectivity panel (i.e. number of layers=0, scale=1, SLDtop=SLDtop, SLDbase=SLDbase, bkg=0, sigma_base=0).
- Now enter the following command in the command line:
  Calcfresnel(coef_ Cref,dataset_ extR,dataset_ extQ).
  This will take the Rwave (*dataset_ extR*) and the Q wave (*dataset_ extQ*), calculate the Fresnel reflectivity at each of the points in your experimental dataset, then divide *dataset_ extR* by the Fresnel reflectivity. At this stage you should have the normalised curve.
- TO display the normalised curve type:
  display dataset_ extR vs dataset_ extR
  in the command line. You will see a line plot, but you can change this by clicking on various aspects of the graph.
- Now put the parameters you expect for the system in the Reflectivity panel, i.e. the required number of layers, thicknesses, solvent penetration, roughnesses, etc. This updates the wave "*coef_ Cref*"

**YOU MUST KEEP THE SCALE, SLDBASE, SLDTOP, SIGMA_BASE THE SAME AS FOR THE FRESNEL CURVE (AND THE BKG=0), BUT OBVIOUSLY USE THE NUMBER OF LAYERS THAT IS RIGHT FOR YOUR SYSTEM.

- You are now are at a stage to do the fit. Go to the Igor menu and select: *Analysis–>Curve Fitting*. Use "*Fresnelreflectivity*" as the fitfunction. Use the normalised reflectivity curve (*dataset_ extR*) as the ydata, and *Dataset_ extq* as the xdata. Go to the coefficients tab and use "*coef_ Cref*" as the coefficient wave.

**YOU MUST HAVE UPDATED THE LAYER PARAMETERS TO REFLECT YOUR SYSTEM. ** THE PLOTYPE MUST BE "R VS Q" ** THE FIT WILL NOT BE WEIGHTED!

- Now select the parameters to hold. YOU MUST HOLD PARAMETERS 0 TO 5 (NUMLAYERS, SCALE=1, SLDTOP, SLDBASE, BKG=0, SIGMA_BASE). You have to hold these because you have normalised your data by the fresnel reflectivity created by these parameters. Select any other parameters to hold, then do the fit (DO IT button).
- There will be a parameter wave produced as a result, (*coef_ dataset_ extR*) and the graph that you produced will have the fit curve on it.

- You can also global fit with this function, once you have normalised all your contrasts by their individual fresnel reflectivity.

CAVEATS: I've tried fitting X-ray data this way, and it works ok. However it doesn't work very well if the data is very different to the fresnel reflectivity (very likely at high Q). This may be true if you have X-ray data extending to high Q (and therefore low reflectivity). Therefore it's *VERY, VERY, VERY* sensitive to the roughness (*sigma_base*) of the fresnel reflectivity curve that you normalise by. In order to get this to fit I created a theoretical reflectivity out to Q=0.7, and experimented with different *sigma_base*. In addition YOU ALSO HAVE TO SUBTRACT THE RIGHT AMOUNT OF BACKGROUND. You might want to restrict the range you fit over using the "cursors".

If you don't choose the correct values then the reflectivity curve will have a large upturn at high Q, this is bad.

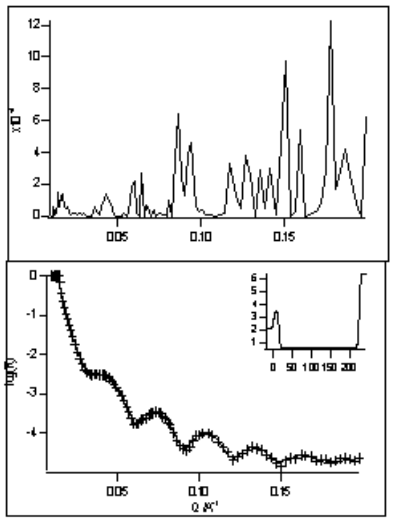
7.17 I want a normalised correlation matrix, how do I do it?

The normalised correlation matrix is automatically produced after a fit. To get this displayed type the following in the command window (Ctrl-J):
EDIT M_COVARIANCE

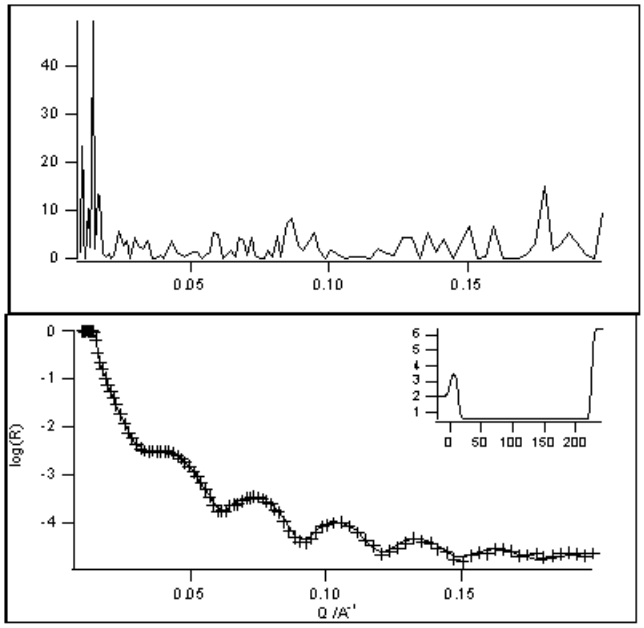
7.18 What type of weighting should I use?

Here we explore the χ^2 function for different types of weighting. I have plotted χ^2 as a function of Q for all the types I use and they are listed below. I have also investigated the situation where dR=1 (i.e. weighting has no effect). This should give a guide on how to weight data. Note that the investigation is on a single dataset only. In addition, the SLD of the top and subphase is fixed. This means that if I'm slightly off with those values then there might be spikes in χ^2 at low Q. I created all these graphs sequentially, from situation a). In other words I performed a) first, then used this as a starting point for all the other fits. This is a significant assumption when drawing conclusions, as this does not take into account the fact that each of the different ways of weighting may have different 'descent pathways' to the global minimum. Thus one should consider various starting points.

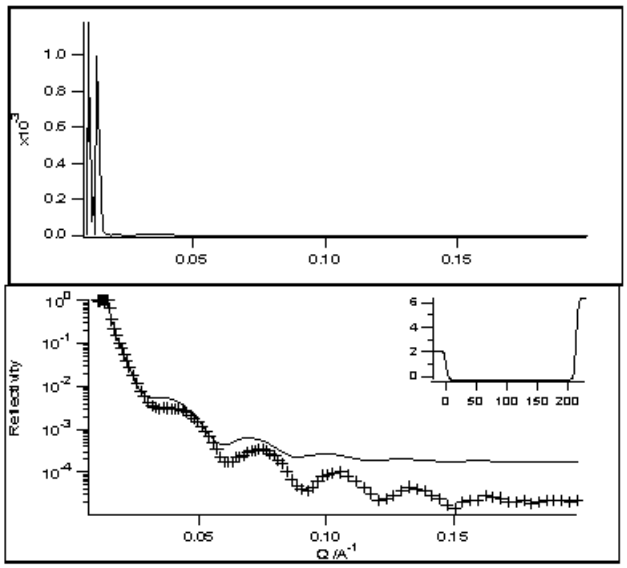
- fitting as log R vs Q, no weighting (dR=1). As you see this option has a good fit, and the χ^2 function starts to weight the data towards high Q (which in this case isn't bad).



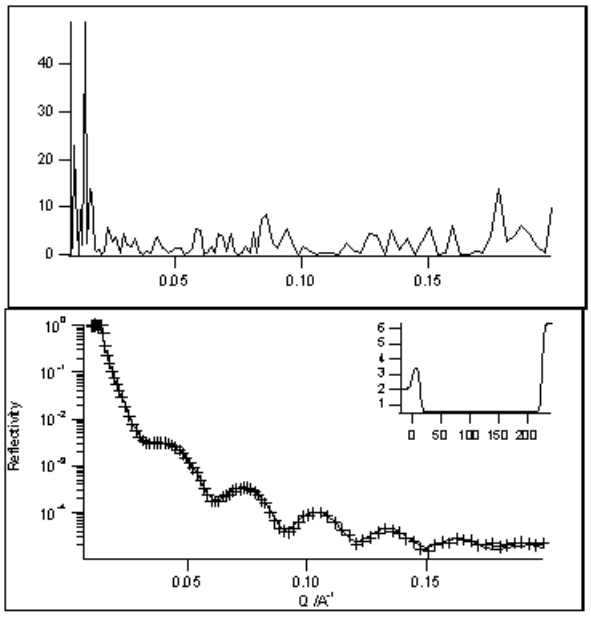
- Fitting logR vs Q, with errors. When you take this option you get rid of the bias towards the higher Q end, and the program will essentially fit uniformly over the Q range. However, you have to be careful with low Q data, as poor fitting around the critical edge (sometimes not unusual) will increase χ^2 significantly and may weight towards low Q, which is bad!



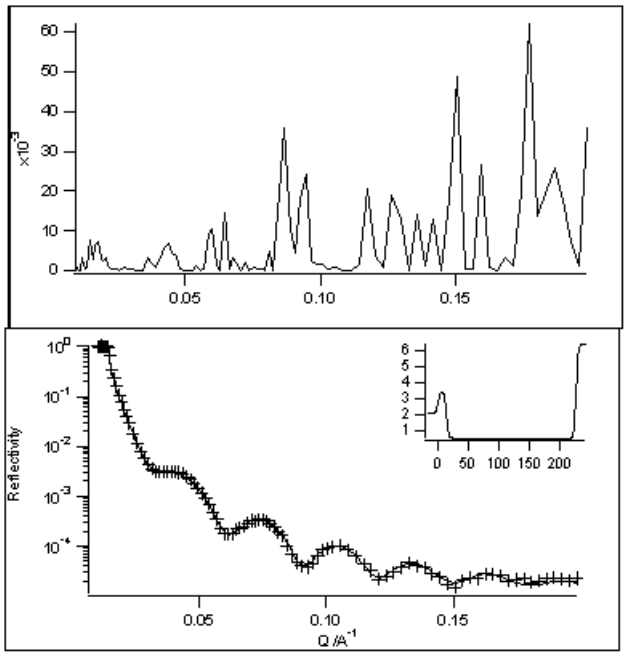
- Fit R vs Q, no weighting This one really doesn't work, never try this option. This is appalling.



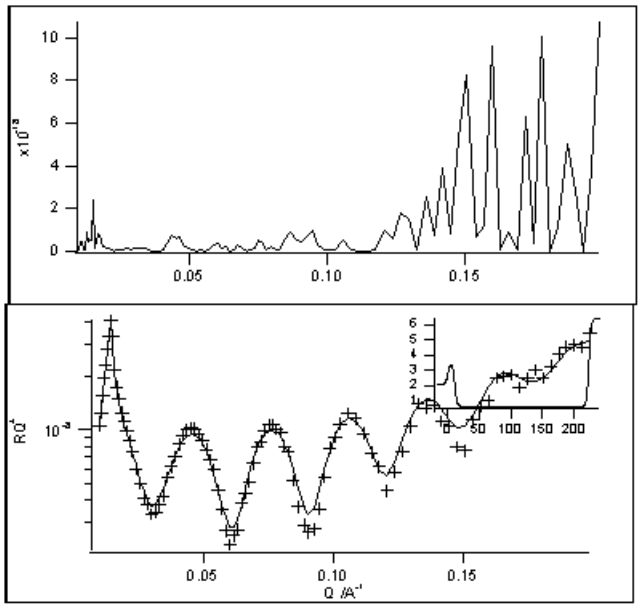
- Fitting R vs Q, with error weighting. Ok, but similar to b). In this situation you really need to make sure that the fit is good around the critical edge, or the fit is bad. The fit is often bad around the critical edge if you are not handling the resolution correctly, or if you don't allow the top or subphase SLD's to vary (which control the position of the critical edge).



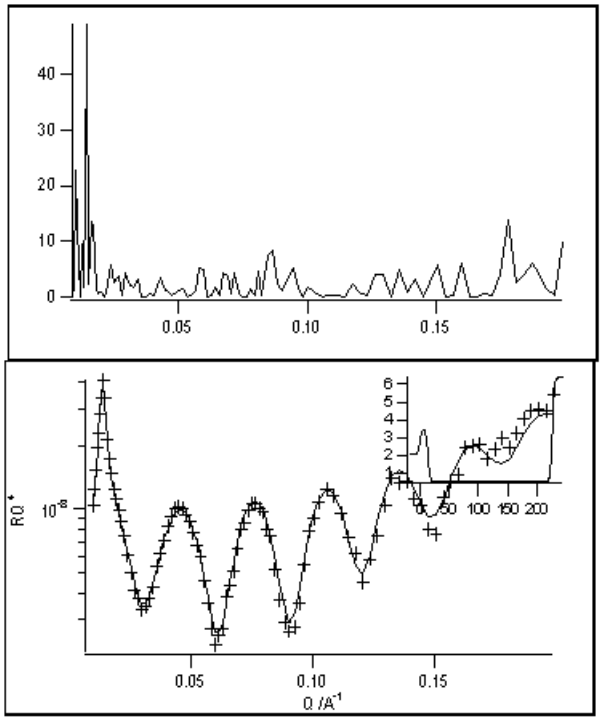
- Fitting R vs Q, weighting as dR=1/R. This is often used as a way of weighting, as it handles the entire Q range successfully. I like this option. Unfortunately you're not using errors.



- Fitting as RQ^4, no weighting. Seems to handle entire Q range ok, but there is some bias towards high Q.



- Fitting as RQ^4vs Q, with errors. This one is telling us that the fit is correctly weighting all the Q range, but will show some bias towards low Q, if the critical edge isn't handled correctly.



So what are the conclusions? These examples show that it's essential to make sure the critical edge is handled correctly if you want to fit with the error bars! This is because all the examples with error weighting have spikes in the χ^2 function at low Q. Of course, the critical edge region is the part of the reflectivity curve that has the least information about the sample. If I was fitting with error bars I would do these types of plots to make sure that

the crit edge region was being handled correctly, otherwise your fits may be skewed towards low Q. Fitting programs and reduction programs should make sure that the resolution is being used properly, in order to remove this problem.

The graphs show that fitting without error bars is quite robust if you fit as logR vs Q or RQ^4 vs Q (R vs Q doesn't work). This is fairly effective at weighting equally over the entire Q range, but there is a slight risk of bias towards higher Q.

Overall I would use all the options except c). The graphs show that there isn't really an advantage to fitting one way or another, so I suggest that it's a fallacy to say that fitting as RQ^4vs Q is better! All this option offers is the removal of the Fresnel influence, which makes it easier to see what is happening to all the layers. Of course this is a simple analysis, so take these conclusions with a pinch of salt.

8 Bugs

Send me details of any bugs and I'll fix them. I have benchmarked the program against other programs and I know that the reflectivity is calculated correctly. If you have any suggestions on how to improve things/would like to help develop things then please email me.

`mailto:Andrew_Nelson@users.sourceforge.net`

References