Some general instructions for analytic profiles.

June 26, 2012

These are some general instructions for using analytic profile code with Motofit. The reason for using analytic profiles is that you can encode specific knowledge about your system into the fitting process. For example, you may want to use a parabolic volume fraction profile for a grafted polymer layer, in which case you may want to use parameters such as grafting density and a power law exponent as fitting parameters.

Another example is the analysis of adsorbed lipid layers. You may want to parameterise the model in terms of area per molecule, and thickness of tail and head group regions. Doing so reduces the number of parameters normally used in a 'slab' fit, and also encodes the knowledge that there must be head/tail group equivalence.

Such analytic profiles normally reduce the number of free parameters, which greatly strengthens the fitting process. In addition the parameterisation usually provides relevant physical parameters that normally would have to be back calculated - such as area per molecule in the example given above, or the exponent in the parabolic volume fraction profile.

The main caveat is that, in general, you have to have a specific fitting function for each model structure. In otherwords, small changes to the model you wish to use normally entails a small amount of code changes. However, don't be put off, it's not difficult.

1 If you need help

If you need help email me and we can collaborate (I usually ask for co-authorship for a significant piece of work). andyfaff@gmail.com or anz@ansto.gov.au.

2 General Design of fitting function

We will discuss this in terms of the bilayer example (5). The LIPID function is referred to as the FIT FUNCTION. The LIPIDTOREF function is referred to as the HELPER FUNCTION. Lets consider the LIPID function first. We need to supply various arrays, or *waves*, to this function. The first, COEFS, contains the parameters for the fit. For this specific case inspection of the comments in the LIPIDTOREF function says that this array needs to be 20 rows long. The

exact meaning of each of the parameters in COEFS is outlined in the comments at the start of the LIPIDTOREF function. The second and third waves are the reflectivity and Q values for the points that have to be calculated.

The LIPID function is called by the fitting engine. LIPID calls the LIPIDTOREF function with the coefficients. The purpose of the LIPIDTOREF function is to convert the input parameters into a wave that can be used by the reflectometry kernel to calculate reflectivity. In this example the wave that the LIPIDTOREF function creates is called W_FORREFLECTIVITY (4). This wave is then used by the MOTOFIT function to calculate the reflectivity. You do not have to worry about the RR and QQ waves, the fitting engine provides those.

Simple examples might just use several layers, the LIPID example given here has 5. However, a parabolic function may have several hundred layers. These layers represent the SLD of the parabolic function at a given distance from the surface.

3 Using the analytic function in practice

1. start Motofit (*motofit->fit reflectivity data*)
2. In the reflectivity panel select the correct dq/q resolution smearing for the datasets you want to fit.
3. Load the data in using the *load data* button.
4. Bring up the genetic optimisation curvefitting panel. You can do this by pressing Cmd-1 (OSX) or Ctrl-1 (windows). Or select the *Analysis->Genetic Curvefitting* menu option.
5. On the *function and data* tab select the FIT FUNCTION for the analytic profile, e.g. LIPID.
6. On the same tab select the Y and X data for the data you want to fit. Loaded reflectivity data is normally kept in the ROOT:DATA:<DATASETNAME> datafolder.
7. On the *data options* tab select the error wave for the data you want to fit (if required).
8. On the *coefficients* tab select the coefficients wave you want to use to fit the data. For the LIPID example the wave needs to be 20 rows long. If no wave exists you can create one by selecting the *new wave* option, you will have to supply the name of the coefficient wave (I suggest something beginning with COEF_) and the number of parameters (20 for the LIPID example).
9. Enter initial guesses for each of the parameters. The meaning of the parameters in the LIPID example can be read from the code comments in the LIPIDTOREF function.

10. Enter lower and upper limits for each parameter.

11. You then need to select the parameters you want to hold during the fit. Often the analytic profile requires that you hold some. In the LIPID example you need to hold parameters 9, 10, 11, 12.

12. When you are ready press DO IT.

13. The fit proceeds and eventually finished (hopefully with a good fit).

14. The fitted coefficients are contained in the Igor data folder you are currently in. It can be a good idea to move them to the data folder that the data is contained in.

15. If you need to create an SLD profile corresponding to the analytic model use the following steps from the IGOR command line:

```
//make the SLD wave, give it your own name
make/n=1000/o/d SLD_profile
//calculate the SLD curve
//the W_forreflectivity wave from the last iteration of the fit needs to
//be in the same data folder as you are currently in
Moto_SLDplot(w_foreflectivity, SLD_profile)
//alternatively you can calculate it again
lipidtoref(coefs_for_fit)
Moto_SLDplot(w_foreflectivity, SLD_profile)
//Display the SLD plot
display SLD_profile
```

4 parameterisation of W_FORREFLECTIVITY

W_FORREFLECTIVITY should be 4N+6 rows long, where N is the number of layers in the model of the interface. The meaning of the 4N+6 parameters is as follows:

w[0] = N
w[1] = scale factor
w[2] = SLD of fronting medium
w[3] = SLD of backing medium
w[4] = linear background
w[5] = roughness between the backing medium and layer N
w[6] = thickness of layer 1
w[7] = SLD of layer 1
w[8] = solvent penetration, expressed as a percentage, in layer 1
w[9] = roughness between layer 1 and fronting medium
.....
w[4(N - 1) + 6] = thickness of layer N
w[4(N - 1) + 7] = SLD of layer N

w[4(N - 1) + 8] = solvent penetration, expressed as a percentage, in layer N
w[4(N - 1) + 9] = roughness between layer N and N-1

All SLD's are used in units of $10^{-6}A^{-2}$ (the NSLD of Si would be entered as 2.07). All thicknesses are in Angstrom. All roughnesses are described by an error function with that s.d.. Solvent penetration is expressed in terms of percentage solvent in layer N. The overall SLD in layer N is a volume fraction weighted sum of the SLD of layer N and the amount of backing medium (solvent) in layer N.

5 Bilayer example

```
#pragma rtGlobals=1 // Use modern global access method.
Function lipid(coefs,rr,qq):fitfunc

    wave coefs,qq,rr
    lipidtoRef(coefs)
    Wave W_forReflectivity
    motofit(W_forreflectivity,RR,qq)
    RR=log(RR)

End
Function lipidtoRef(coefs)

    Wave coefs
    //coefs[0]=scalefactor
    //coefs[1]=SLDsuperphase (silicon?)
    //coefs[2]=SLDsolvent (d2o mix?)
    //coefs[3]=background
    //coefs[5]=oxide thickness
    //coefs[6]=SLDoxide
    //coefs[7]=roughness of si/sio2
    //coefs[8]=roughness of inner lipid head/sio2
    //coefs[9]=Vh, headvolume     (cubic A)                      FIX
    //coefs[10]= bh, sum of scattering lengths of headgroup     FIX
    //coefs[11]= Vt, molecular volume of tailgroup (cubic A)    FIX
    //coefs[12]= bt, sum of scattering lengths of tailgroup     FIX
    //coefs[13]=Area per molecule (inner)
    //coefs[14]=thickness of inner leaflet head
    //coefs[15]=thickness of inner leaflet chain
    //coefs[16]=Area per molecule (outer)
    //coefs[17]=thickness of outer leaflet chain
    //coefs[18]=thickness of outer leaflet head
    //coefs[19]=roughness of lipid layers
    make/o/d/n=(5*4+6) W_forReflectivity
    W_forReflectivity[0] = 5
    W_forreflectivity[1] = coefs[0]
    W_forreflectivity[2] = coefs[1]
```

```
    W_forreflectivity[3] = coefs[2]
    W_forreflectivity[4] = coefs[3]
    W_forreflectivity[5] = coefs[19]
    //SiO2 layer
    W_forreflectivity[6] = coefs[5]
    W_forreflectivity[7] = coefs[6]
    W_forreflectivity[8] = 0
    W_forreflectivity[9] = coefs[7]
    //inner head
    W_forreflectivity[10] = coefs[14]
    W_forreflectivity[11] = coefs[10] / coefs[9]
    W_forreflectivity[12] = 100 - 100 * (coefs[9] / (coefs[13] * coefs[14]))
    W_forreflectivity[13] = coefs[8]
    //inner chain
    W_forreflectivity[14] = coefs[15]
    W_forreflectivity[15] = coefs[12] / coefs[11]
    W_forreflectivity[16] = 100 - 100 * (coefs[11] / (coefs[13] * coefs[15]))
    W_forreflectivity[17] = coefs[19]
    //outer tail
    W_forreflectivity[18] = coefs[17]
    W_forreflectivity[19] = coefs[12] / coefs[11]
    W_forreflectivity[20] = 100 - 100 * (coefs[11] / (coefs[16] * coefs[17]))
    W_forreflectivity[21] = coefs[19]
    //outer head
    W_forreflectivity[22] = coefs[18]
    W_forreflectivity[23] = coefs[10] / coefs[9]
    W_forreflectivity[24] = 100 - 100 * (coefs[9] / (coefs[16] * coefs[18]))
    W_forreflectivity[25] = coefs[19]
End
```