# Advanced Neutron Reflectometry Data Analysis for Soft Matter Systems

## Introduction

During the following practical you will fit layer models to experimental reflectivity data sets and use these fits to gain quantitative information about the molecular structure of material at a buried interface. To do this you will use the RasCal software which calculates slab layer models using Abeles matrix formalism, which is conceptually very similar to Parratt's recursive formalism which is also used in reflectometry data analysis.

In these approaches each layer is described by its thickness, roughness and scattering length density (SLD, ρ). The thickness of each layer is used to determine the phase of the reflected waves from each "slab" layer interface (top and bottom). Kiessig fringes in the reflectivity data are due to interference between these reflected waves.

More information on the mathematics of Parratt's recursive formalism is given at the end of this document.

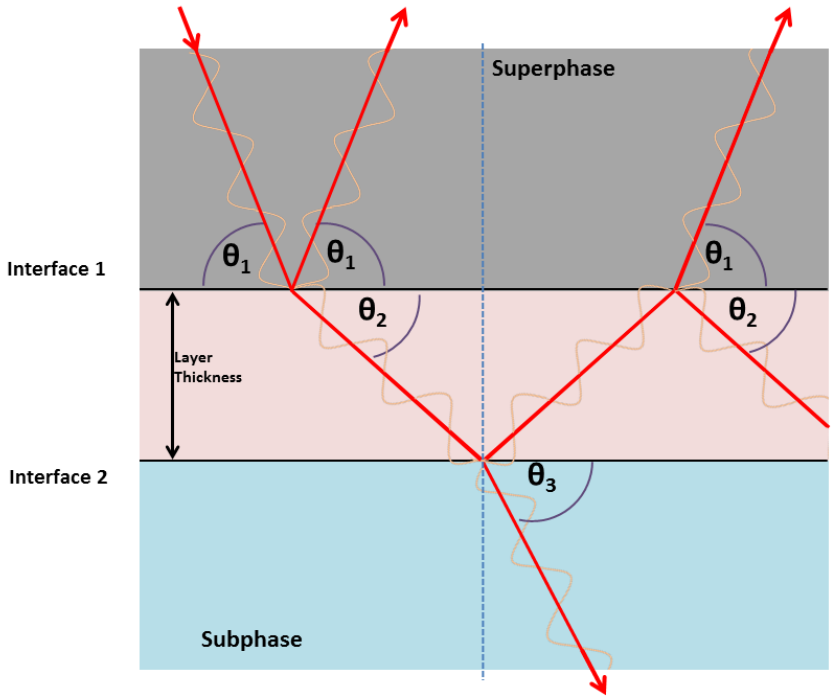



**Figure 1, The "slab" layer model is used to calculate model reflectivity data by both the Abele's matrix formalism and Parratt's recursive formalism. The difference in scattering length density (ρ) at an interface and the angle/wavelength of the neutron beam give the magnitude of reflection while thickness of the layer and its effect on the phase difference between the top and bottom reflections gives rise to Kiessig Fringes.**

In the following practical you will start by examining the structure of a silicon / $D_2O$ interface using the Rascals model building GUI followed by fitting the same data using a "custom" i.e. ***scripted modelling capabilities***. Then, the structure of a bilayer of 1,2-dipalmitoylphosphatidylcholine (DMPC) will be examined using multiple solution isotopic contrasts firstly with a simple "volume" fraction model followed by analysis using a more advanced "area per molecule" model. The data you use for this will be the reflectivity data obtained from the ISIS neutron training course solid/liquid flow cells practical on the CRISP reflectometer. Finally, time permitiing, we will take a look at a complex layer structure with a composite fringe pattern from a multi-layered membrane complex.

**Practical 1 – *The Silicon/Water Interface***

The silicon water interface is a common surface used in NR studies. For these experiments the reflection occurs within a silicon substrate.

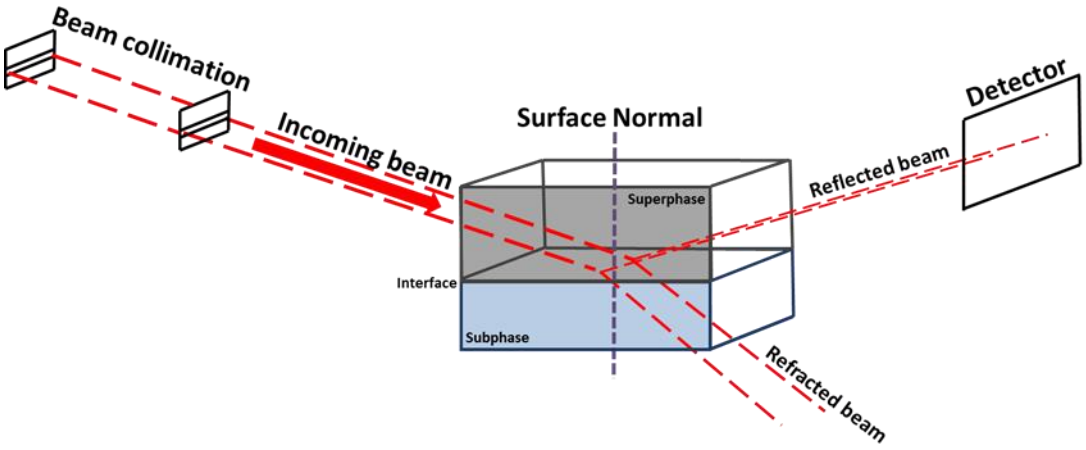



**Figure 2, Neutron Reflectometry in a solid liquid flow cell. Note: the neutron beam is reflected inside of the substrate.**

You will begin the practical by fitting a bare $SiO_2$ coated silicon/$D_2O$ interface using Rascals GUI option.

***Moving the Practical to the IDAaaS Desktop***

If you haven't already. Move the school folder onto your IDAaaS desktop so you can save your progress in the practical. To do this you must use the following steps:

**Applications>System>File manager**

In file manager navigate to:

**/mnt/ceph/auxiliary/reflectometry/**

Copy the **NTC2021** folder, and paste it at:

**/home/username/Desktop/**

**HINT:** Username is your **individual username**, mine is lc10795

Alternatively hit the **home** button (top right) and navigate to desktop from there.

***Starting the Software.***

In your IDAaaS session:

**Applications>Software>RasCal**

The **RasCal** software will load.

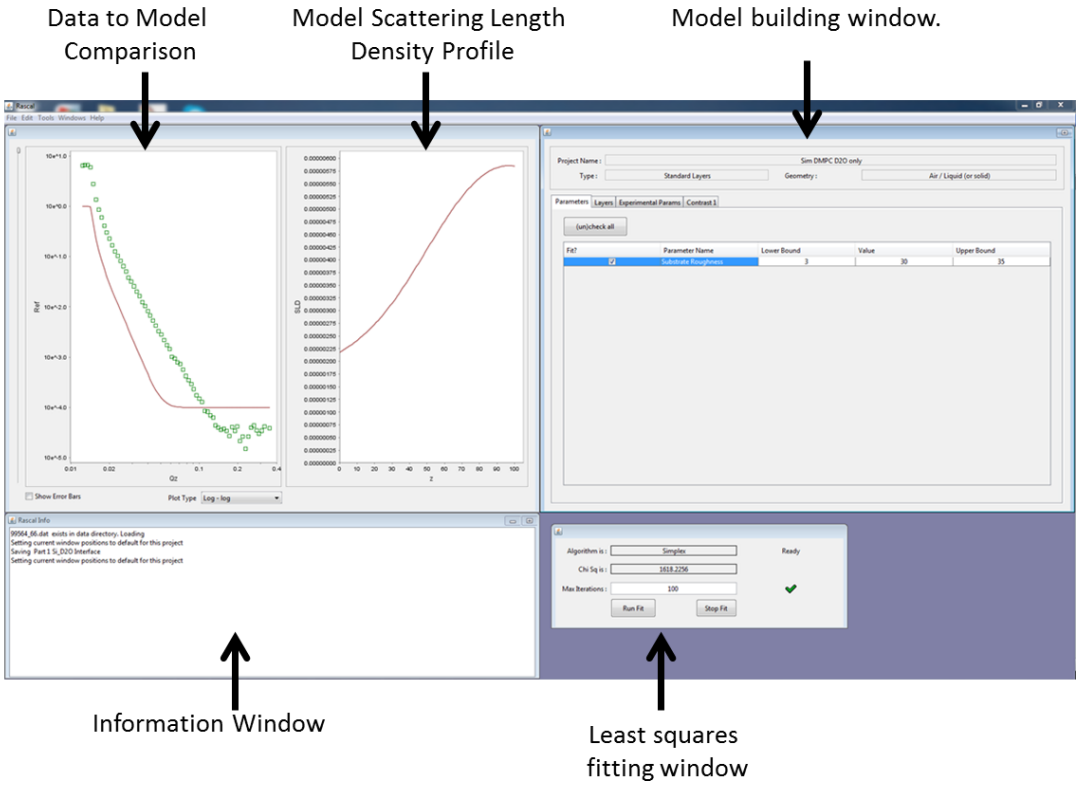
In **RasCal**: **Windows>Tile**

***Setting up the model to fit***

Once the Rascal software has been started you will need to load the project to begin working. Find your way to the following folder:

**File>load>/home/username/Desktop/NTC2021/Practical7_Membranes/**

Go to the folder named "**Part 1 Si_D2O Interface**" and double click on the icon called "**Part 1 Si_D2O Interface.mat**". Then **Windows**>**Tile**. You should now see the window shown below appear. :



The 1st thing you will notice is that the real and model reflectivity data is not on the same scale, i.e. the scale factor is incorrect. There are similarities and differences between the model data (**red**) and the experimental reflectivity data (**green**). Specifically, the critical edge of the reflection is in the same position in both and while the general decay of the model data intensity against momentum transfer ($Q_z$) and the background do not match the experimental data.

Therefore we begin fitting by setting the correct experimental parameters.

1. On the model building window click on the tab labelled **experimental parameters**.

2. Next correctly scale the data using the **scale factor** (do this so the model and data critical edges meet) and then set the **background** for the sample (the flat region in the high $Q_z$ regime). If you want you can fit these parameters by ticking the fit box against each parameter and clicking on "**run fit**".

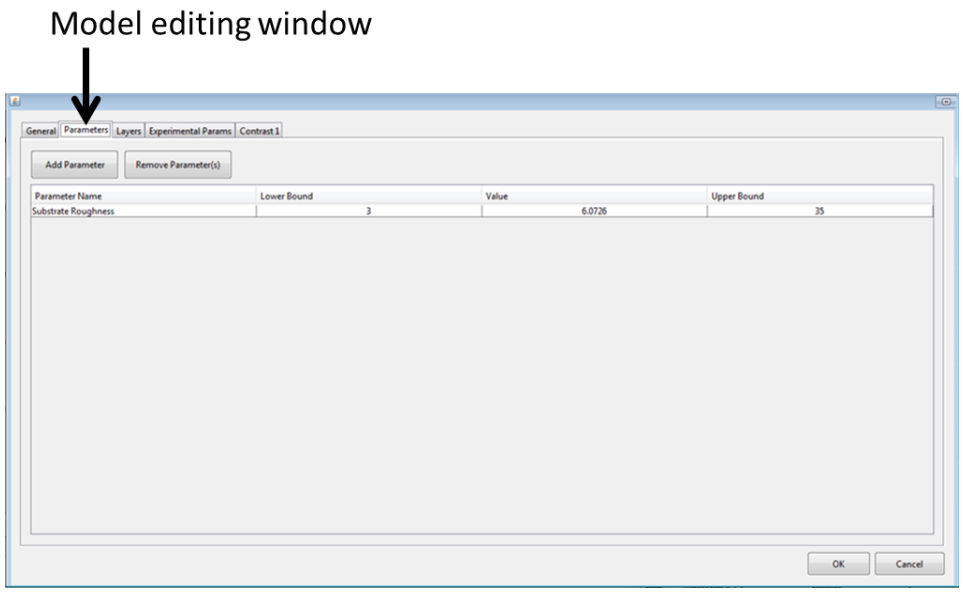
**TIP**: if you do this, make sure you set the number of iterations in the **least squares fitting window** to be more than 100, if you type **inf** it will run until the fit is complete.

3. Once these are set fit the substrate roughness (the roughness between the bulk interfaces) to gain an approximate fit of the data. You will notice you get a good fit to the experimental reflectivity data producing a reflectivity profile with a defined step function.

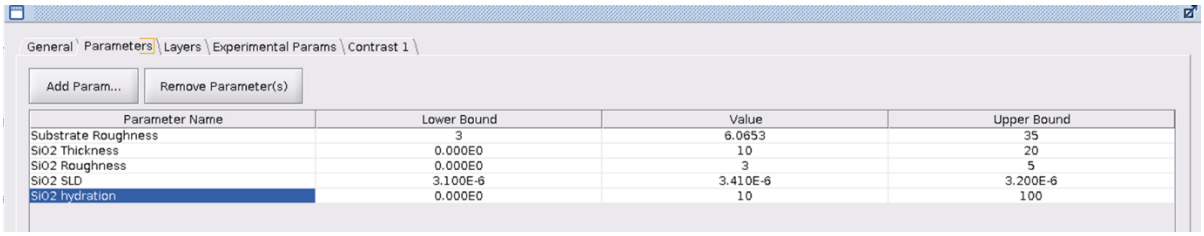
4. However, the roughness will be artificially high as the model does not accurately portray the interfacial structure. The silicon substrate will have a thin (~10 Å) silicon dioxide layer on the surface. We will now add this layer by editing the model.

**Adding an interfacial layer.**

1. Click **Edit** > **Edit Project** and you will now be able to edit the model window:



2 Click on the parameters tab and add the following four parameters with the following bounds:




| Parameter Name | Lower Bound | Value | Upper Bound |
|---|---|---|---|
| Substrate Roughness | 3 | 6.0653 | 35 |
| SiO2 Thickness | 0.000E0 | 10 | 20 |
| SiO2 Roughness | 0.000E0 | 3 | 5 |
| SiO2 SLD | 3.100E-6 | 3.410E-6 | 3.200E-6 |
| SiO2 hydration | 0.000E0 | 10 | 100 |

Place the fitted value between the lower and upper bounds for each parameter except the $SiO_2$ SLD where you should use the value given above.

When you run the fit do not fit the SiO2 SLD value (un-tick the box for this).

Appropriate range for fitting for the SiO2 layer are:

**SiO2 Thickness:** Lower = 0 Å, Upper = 20 Å

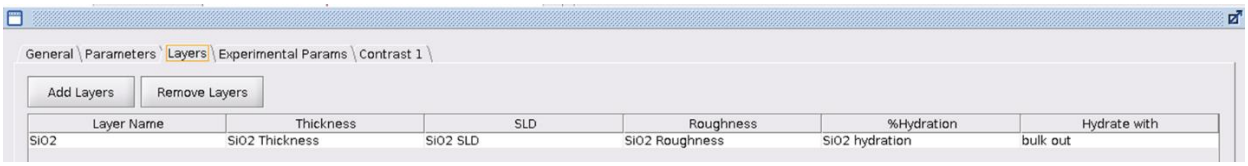
**SiO2 Roughness:** Lower = 0 Å, Upper = 5

**SiO2 SLD:** Set to 3.41e-6 $Å^{-2}$ and **do not fit** (untick)

**SiO2 Hydration:** Lower = 0%, Upper = 30%

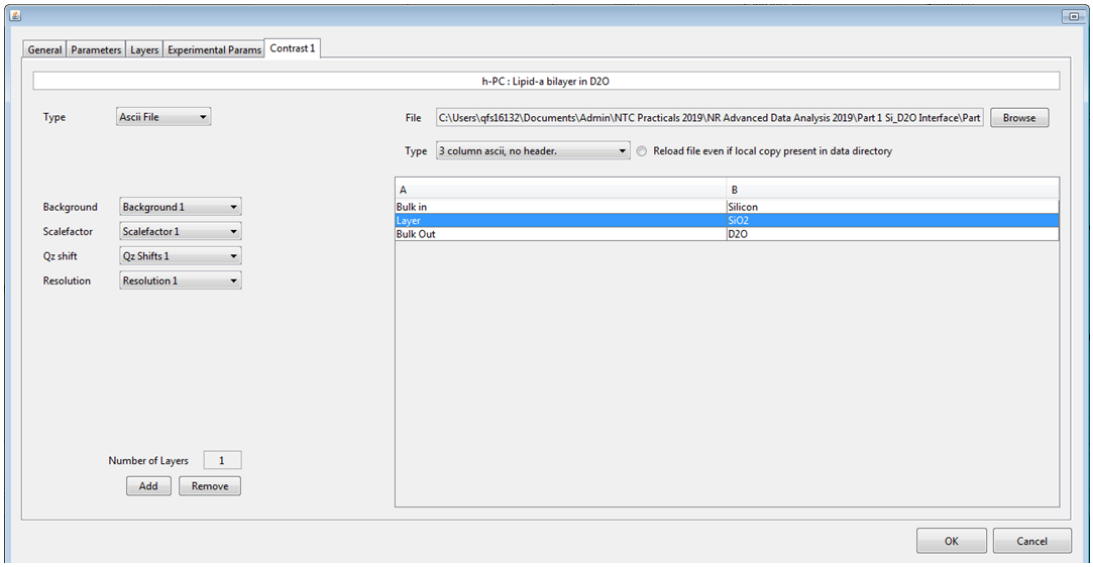
**HINT!**: The value must be set between these bounds before you start the fit.

3. Next in the "**Layers**" tab add a layer and then populating the layer with each parameter in the correct place and naming the layer appropriately:




| Layer Name | Thickness | SLD | Roughness | %Hydration | Hydrate with |
|---|---|---|---|---|---|
| SiO2 | SiO2 Thickness | SiO2 SLD | SiO2 Roughness | SiO2 hydration | bulk out |

4. In the "**Contrast 1**" tab add the layer between the bulk phases by clicking on **bulk out** row and pressing the add button, then clicking on the new layer ($2^{nd}$ column) and selecting the $SiO_2$ layer from the drop down menu.




Now click **OK** and the model should be updated with your new layer. Now rerun the fit.

5. You should now see that you have a new layer between the bulk phases which is rather ambiguous i.e. poorly described by the experimental data.

Think about why this is ambiguous and what might help to better resolve this layer? Write you answer thoughts in the box below:

**6.** Place the fit values in the table overleaf:

| | |
|---|---|
| **Substrate roughness/Å** | |
| **SiO2 thickness/Å** | |
| **SiO2 roughness/Å** | |
| **SiO2 Hydration/%** | |
| **Background 1** | |
| **Scale factor 1** | |

**Now save the project using a name of your choice on the IDAaaS desktop by:**

**File>save as new project>home icon**  **>/home/Desktop/name**

# Custom Model Fitting

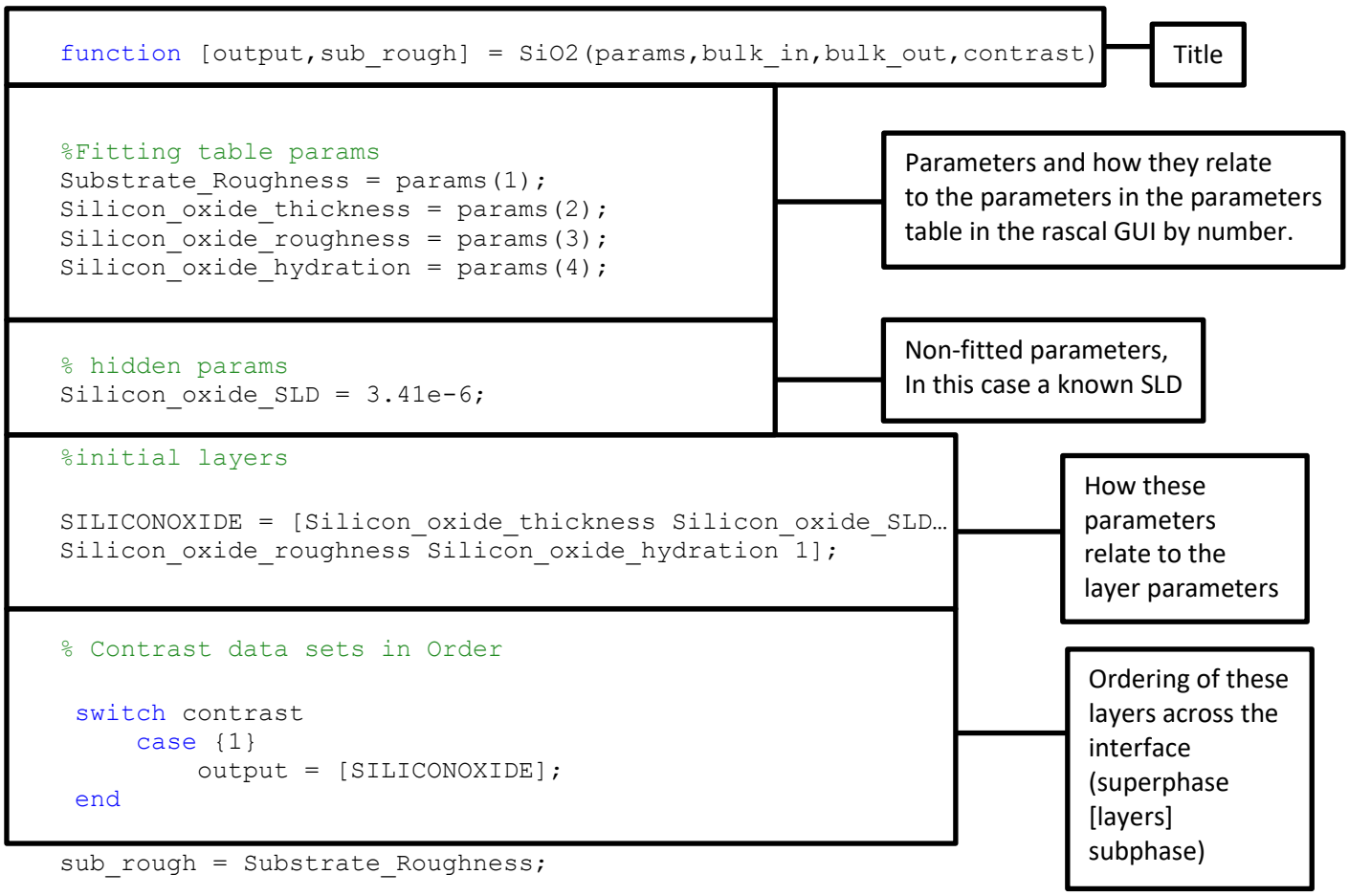
Using RasCals GUI is an easy way to begin fits but the real flexibility in the software comes from using the custom model approach where the fit is defined using a script. We will start by fitting a predefined Custom model of the Silicon/$D_2O$ interface we have already fitted.

1. Edit the project and delete the parameter for the **SiO2 SLD** from the parameters table.

2. On the **General** tab change the **Project type** to **Custom Layers (NOT custom XY)** and click on **Browse**. Select the file called **SiO2.m** Next Click **Edit** in the General Tab. You will now be able to edit this script.

The script is a sequential description of individual parameters, how these parameters relate to the interfacial layers and how those layers are organised between the bulk phases.

The simple script for the silicon water interface is given below with an explanation of its structure:




3. Click **Save** on the script go back into Rascal and Click **OK** on the **Edit Project window** to compile the script.

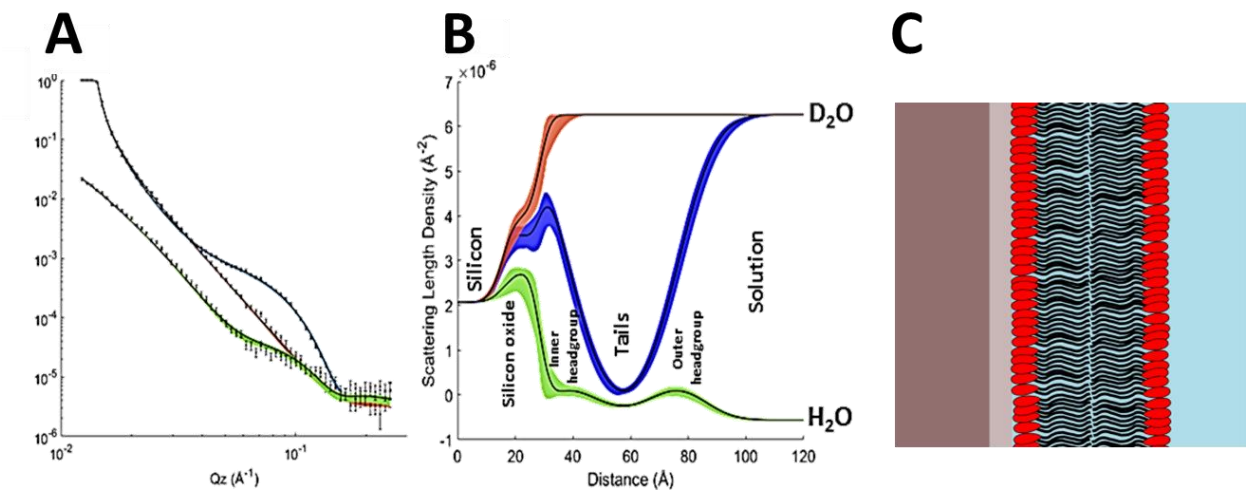
4. Refit the data. You will note that the fit should be fairly identical to what we have found previously.

## Save the Project at this point on the IDAaaS desktop

## Fitting a Custom Model of the Bilayer by Volume Fractions

We will now create a custom model of the interfacial structure coated with a 1,2-dimyristoyl-sn-glycero-3-phosphocholine (DMPC) bilayer and fit the experimental.

Below is an example of neutron reflectometry data from two solution contrasts ($D_2O$; blue, and $H_2O$; green, **A**), associated scattering length density (SLD) profiles (**B**) and a schematic of the interfacial structure these profiles represent (**C**) for a DMPC bilayer at the silicon/water interface:




The simultaneous fitting of these same two solution isotopic contrast data sets ($H_2O$ and $D_2O$) will be used to resolve the interfacial lipid bilayer structure here, which will be constrained against the data for the $Si/D_2O$ Interface only.

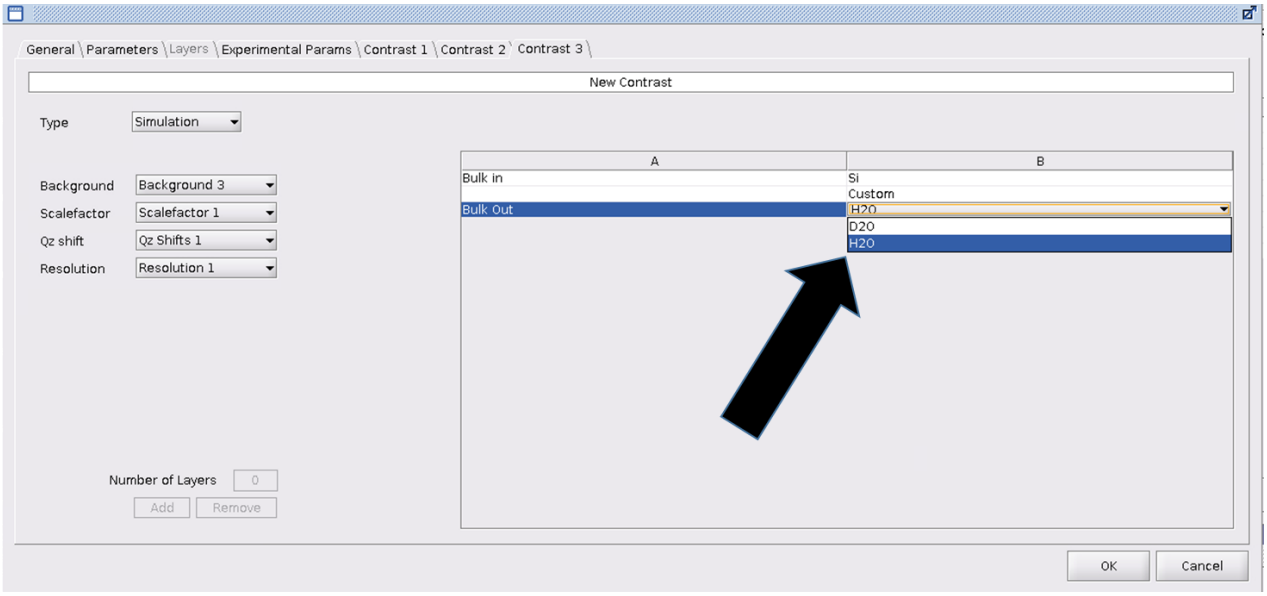
1. Let's start by setting up the contrasts and parameters. Add two extra contrasts to the project by **Edit > Edit project > General > Number of contrasts > Add > Add.**

Two new tabs will have appeared called **Contrast 2** and **Contrast 3**. These are for our $D_2O$ and $H_2O$ bilayer data sets respectively.

2. Add new backgrounds for the new contrasts using the relevant **Add** tabs in **Experimental Parameters**. Set the ranges on these new parameters correctly.

3. Additionally, add an extra **SLD bulk 2 (beam out)** for **$H_2O$**, the value for this is **-0.56e-6**, set the lower and upper bounds close to this (~-0.6e-6 and -0.4e-6).

4. In **Contrast 2** and **Contrast 3** Change to the correct **Backgrounds**, **Scale factors** (use scalefactor1) and in the case of Contrast 3 (the $H_2O$ contrast) the **Bulk out**. See Below:




5. Now let's decide on our known and unknown parameters.

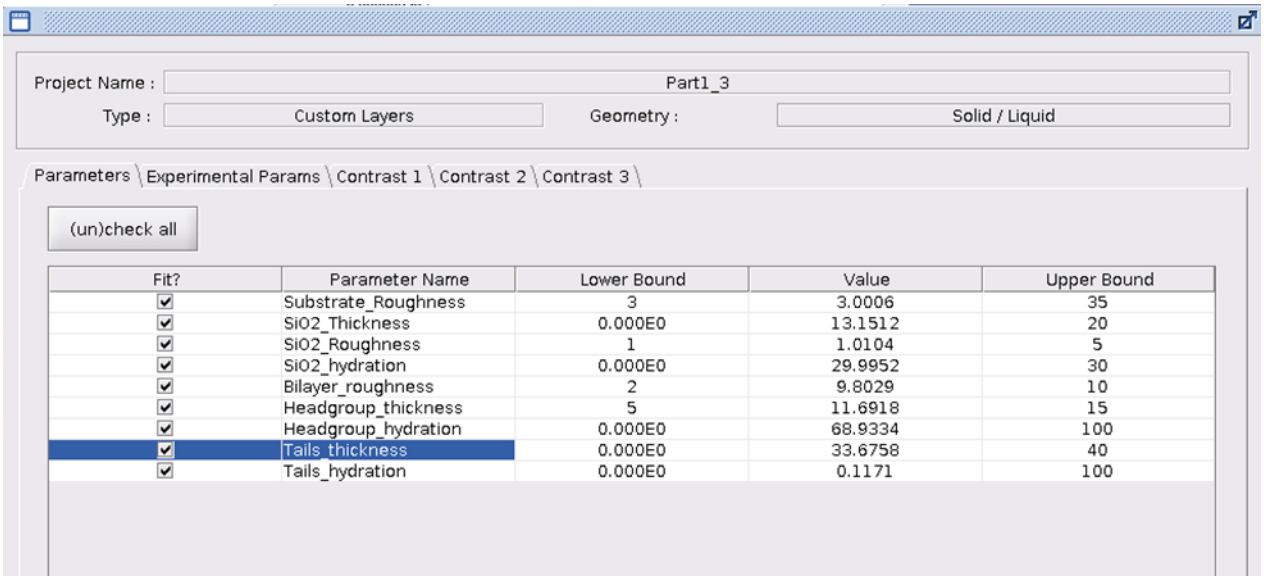
We know the scattering length densities of the DMPC headgroups and tails:

$$\text{Tails} = \mathbf{-0.37 \times 10^{-6}\ \text{Å}^{-2}}$$

$$\text{Head groups} = \mathbf{1.98 \times 10^{-6}\ \text{Å}^{-2}}$$

But this only takes into account a situation where the layer is composed of a single component only. The DMPC bilayer, like the $SiO_2$ layer is immersed in solution and therefore hydrating water is present in the head groups and potentially in the tails if the bilayer has defects, therefore we should fit two hydration parameters (heads and tails).

Additionally we should fit the **tails thickness**, **headgroup thickness**, and a single **bilayer roughness** parameter. Add these to the **Parameters** tab using the bounds shown below.



6. Now we need to edit the custom fit script and build our bilayer structure. Go to the **General tab** and click **Edit**. Now edit the script to add the new parameters and the known parameters. Define layers for the headgroups and tails of the bilayer and create two new contrast cases both with silicon oxide layers but additionally with a bilayer layers across the interface as well (**Headgroup; Tails; Headgroup**). Use the structure of the SiO2 custom model as your guide.

You will be given **30 minutes to complete this task**!

**Top-Tip**: You can save **SiO2.m** as a different file name in the project folder to avoid confusion over script names but make sure you **browse** and **select** the new **custom model** script name in the **general tab** before hitting **OK.**

**HINT!: How to construct your bilayer parameters.**

**To complete this assignment you need to describe a structure across the interface for contrasts 2 and 3 which is composed of the following layers:**

**[SILICONOXIDE; HEADGROUPS; TAILS; HEADGROUPS];**

**Refer to page 10 to see what this should look like. Below is an example of this done with one <u>Headgroup and SiO2 layers only as a guide:</u>**

```
%Fitting table params
Substrate_Roughness = params(1);
Silicon_oxide_thickness = params(2);
Silicon_oxide_roughness = params(3);
Silicon_oxide_hydration = params(4);

Bilayer_roughness = params(5);
Headgroup_thickness = params(6);
Headgroup_hydration = params(7);
Tails_thickness = params(8);
Tails_hydration = params (9);

% hidden params
Silicon_oxide_SLD = 3.41e-6;
Headgroup_SLD = 1.98e-6;

SILICONOXIDE = [Silicon_oxide_thickness Silicon_oxide_SLD…
Silicon_oxide_roughness Silicon_oxide_hydration 1];

HEADGROUPS = [Headgroup_thickness Headgroup_SLD Bilayer_roughness…
Headgroup_hydration 1];

switch contrast
     case {1}
         output = [SILICONOXIDE];
     case {2}
         output = [SILICONOXIDE; HEADGROUPS];
     case {3}
         output = [SILICONOXIDE; HEADGROUPS];
```

```
end
```

**BEWARE!: Parameter names in the RasCal custom model cannot have spaces between words while in the RasCal GUI parameter list they can.**

7. Compile your new script by saving the script and clicking **OK** on the **Edit project** window. It will not compile if there are errors.

**Now save the project on the desktop on IDAaaS using a name of your choice.**

8. Add experimental data from the DMPC bilayer to **Contrasts 2** and **3**.

This data is found in: **/home/username/Desktop/NTC2021/Practical7_Membranes/Part 1 Si_D2O Interface/dataFiles/**

To add this data:

- In the correct contrast tab change data type (top left) from **simulation** to **Ascii File.**
- Hit **Browse>dataFiles>**Select the appropriate data set (**DMPC_D2O.dat** or **DMPC_H2O.dat**).
- Once the correct data sets are in the correct tabs hit **OK** to compile.

**Now fit the data.**

**Discuss on slack/zoom chat the structure you have resolved for the bilayer (thickness and coverage) to check consistency. We will discuss the results as a group.**

**Once finished your fits should look like those shown in panel A of the diagram on page 10.**

## Save your project as a new project on the IDAaaS desktop.

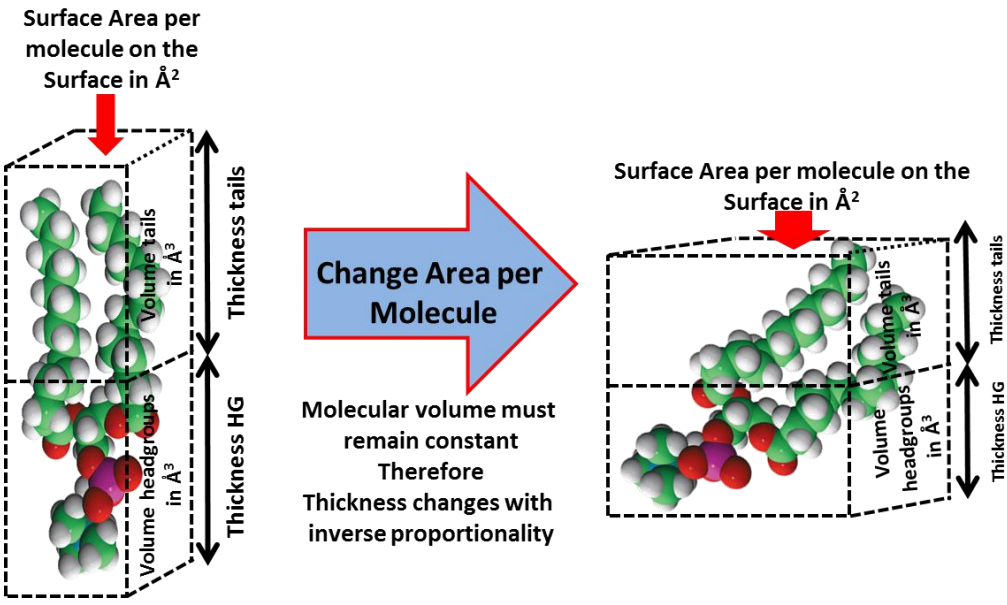
## Fitting a Custom Model of the Bilayer by Area per Molecule

In the previous section different layers arising from the same molecular species are fitted in an unconstrained manner. This can cause some ambiguity in the analysis as there can be differing total molar amounts of head and tail components in the resolved structural data.

A better approach is to couple these parameters together using a known shared parameter. In most cases this will be area of the surface occupied by the molecule. As each lipid head and tail, being a part of the same molecule, will occupy the same 2-dimentional area of the sample surface. This area will be related to the thickness of the sample by:

$$Occupied\ Surface\ Area\ = \frac{Component\ Volume}{Layer\ Thickness}$$

The volume of the tail and head groups are linked to the sum of the atomic volumes which make up the molecule and therefore will not change (within a given phase). The thickness is inversely correlated to the area per molecule and therefore only a single parameter for both the head and tail groups needs to be fitted, that is the area per molecule at the surface. Below is a diagrammatic representation of this concept:




In the example we will use the molecular volumes of the head and tail components of DMPC to construct a model which fits the bilayer at the silicon water interface by four parameters only. That is, the **area per molecule**, the **total bilayer hydration** (i.e. presence of defects

across the surface) and the **headgroup hydration** (i.e. water bound to the hydrophilic headgroups) and **bilayer roughness**.

For this section of the practical you will edit a **custom area per molecule model**, completing parts of the code which are missing.

***Starting the project***

In Rascal load the project: **File>Load>Practical7_Membranes >Part 2 Custom Area Per Mol>Part 2 Custom Area Per Mol.mat.**

You will see three sets of data, Contrasts 1, 2 and 3, and Model data sets which describe the $SiO_2$ layer at the interface only (i.e. no bilayer structure is present).

The relationship between layer thickness, component volume and area per molecule is:

$$Thickness = \frac{Volume}{Area\ Per\ Moelcule}$$

And component volume, scattering length and scattering length density is:

$$SLD\ (\rho) = \frac{\sum b}{Volume}$$

The volume and the scattering lengths (∑b) of each of the interfacial components are known. By fitting the experimental reflectivity data we can determine the interfacial lipid area per molecule and layer thicknesses.

The SLD of the lipid headgroups and tails can be calculated using priori information, using this, the solution is present in each of the headgroup and tail regions of the bilayer can be determined.

There is a differential in solution content within the headgroups and tails. As the tails will only have water content through defects while the headgroups have hydration waters as well as water present due to defects. The model built during this part of the exercise will discriminate between these hydration types.

Edit the model and then edit the custom model. **Edit>Edit project>General>Edit**.

You will find the following incomplete code to describe the interfacial layer structure:

```
function [output,sub_rough] =
Area_per_molecule(params,bulk_in,bulk_out,contrast)

%%Fitting Table parameters
Substrate_Roughness = params(1);
SiO2_thickness = params(2);
SiO2_roughness = params(3);
SiO2_hydration = params(4);

HG_bound_waters = params (5);
%DMPC_AreaPerMolecule = params (6);
%Bilayer_roughness = params (7);
%Bilayer_hydration = params (8);


%%Known SLDs
SiO2_SLD = 3.41e-6;

%%Known Volumes in Angstrom cubed
DMPC_HG_Volume = 320.9;
DMPC_Tails_Volume = 783.3;
Water_Volume = 30.4;

%%Known Scattering Length in Angstrom
DMPC_HG_SL = 6.41e-4;
DMPC_Tails_SL = -3.08e-4;
H2O_SL = -1.64e-5;
D2O_SL = 2e-4;

 %% Relate HG bound waters to SL and Volume

Headgroup_water_D2O_SL = HG_bound_waters * D2O_SL;
Headgroup_water_H2O_SL = HG_bound_waters * H2O_SL;
Headgroup_water_Volume = HG_bound_waters * Water_Volume;

%% Add that to the HG Volumes and SLs in both contrast cases

%Volume_HG = ? + ?; %%Clue it's the same as before but with the water added

%DMPC_HG_SL_D2O =  ? + ?; %%Clue it's the same as before but with the
correct contrast water added
%DMPC_HG_SL_H2O =  ? + ?;

%%Calculate the SLD of the HG in both contrast cases

%SLD_HG_D2O = ?/?; %%SLD = sum b / v
%SLD_HG_H2O = ?/?;

%% Calculate the thickness from the HG volume over the lipid Area per
%% molecule

%HG_thickness = Volume_HG/DMPC_AreaPerMolecule; %% Thickness = v/APM

%%Calculate the SLD of the tails
```

```
%SLD_Tails = DMPC_Tails_SL/DMPC_Tails_Volume;

%%Calculate the thickness of the tails

%Tails_thickness = DMPC_Tails_Volume/DMPC_AreaPerMolecule;

%% Now construct your layers adding in a single roughness parameter and a
%% parameter for defects (hydration)

SiO2 = [SiO2_thickness SiO2_SLD SiO2_roughness SiO2_hydration 1];

%HEADGROUP_D2O = [? ? ? ? 1];

%HEADGROUP_H2O = [? ? ? ? 1];

%TAILS = [? ? ? ? 1];

 switch contrast
    case {1}
        output = [SiO2];
    case {2}
        output = [SiO2];
    case {3}
        output = [SiO2];
end

sub_rough = Substrate_Roughness;
```

In this code you will notice there are single commented lines (**%**) and double commented (**%%**) lines. The **double commented** lines are **notes and hints** while the **single commented** lines are the **lines of code you must complete** by the replacing the questions marks (?) with the correct information.

The parameters you wish to fit are commented out in the parameters section of the code. These are:

```
HG_bounds_waters = params (5);
DMPC_AreaPerMolecule = params (6);
Bilayer_roughness = params (7);
Bilayer_hydration = params (8);
```

Suitable parameter ranges for these are:

**HG_bound_waters**: lower = 0 waters, upper = 10 waters;

**DMPC_AreaPerMolecule**: lower = 0 Å$^2$, Upper = 100 Å$^2$.

**Bilayer_roughness**: lower = 2 Å, upper = 10 Å.

**Bilayer_hydration**: lower = 0%, upper = 100%.

**HINT: Add these parameters to the list in RasCal**

The code below the fitted parameters details the relationship between these parameters and the parameters which define the structure i.e. the layer parameters SLD, thickness, roughness and hydration.

Once you have uncommented and placed the correct info in all the lines of code you must add the bilayer structure to cases **{2}** and **{3}**.

**HINT! Input the following layer structure in contrast (Case) 2 and 3 once you have determined the relationship between area per molecule, thickness and SLD and defined your layers:**

```
Case {2}
        output = [SiO2; HEADFGROUP_D2O; TAIL; TAIL; HEADGROUP_D2O];
Case {3}
        output = [SiO2; HEADFGROUP_H2O; TAIL; TAIL; HEADGROUP_H2O];
```

**Note: Here we define the tails as two layers rather than a single layer.**

You will be given **40 minutes** to complete this practical. After which time we will discuss the resolved structural information as a group.

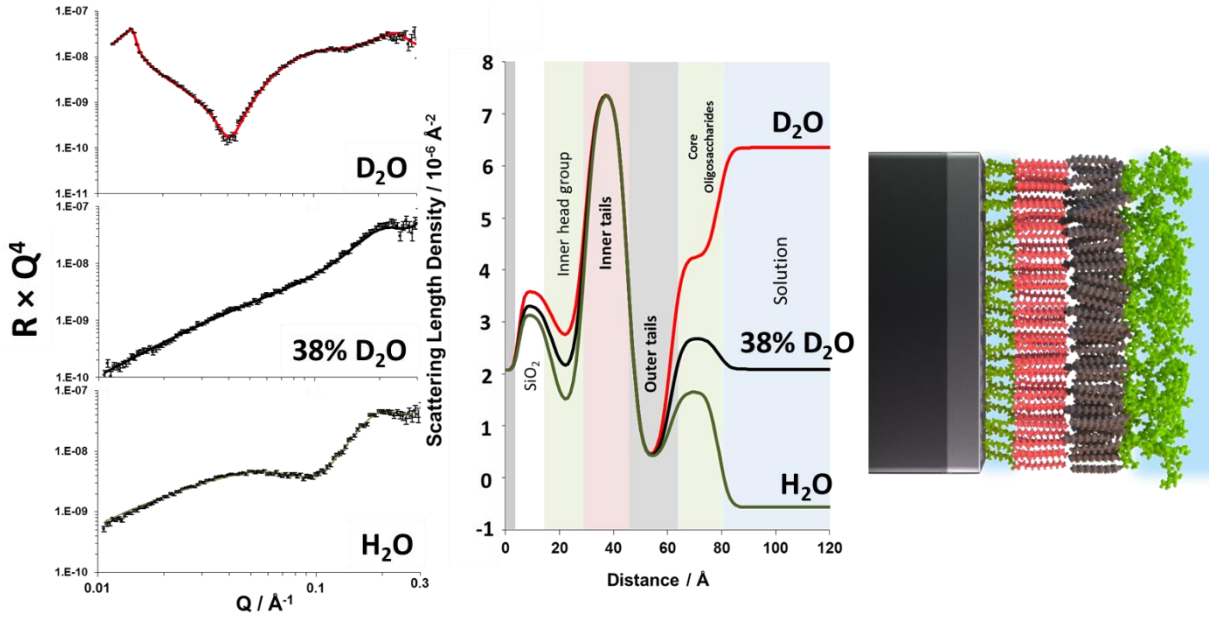
## Save your project as a new project on the IDAaaS desktop.

## Fitting a Complex Protein-Bound Complex Membrane Structure

The final part of this practical is a challenge of you newfound fitting and scripting skills and will test some basic trigonometry.

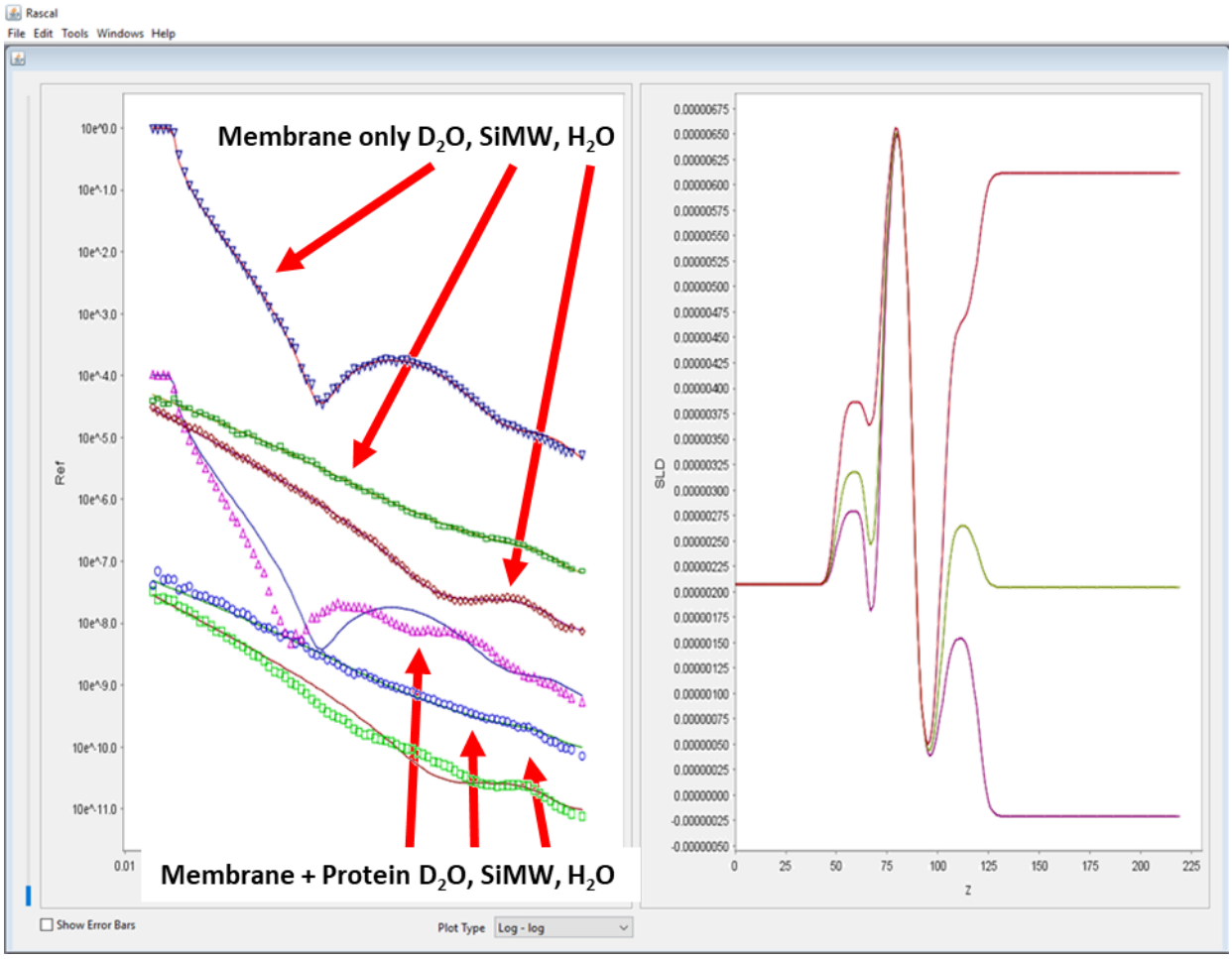
In the folder labelled **"Part 3 Protein-Bound Membrane Structure"** you will find a complete project with data which has previously been published in ***Langmuir 32 (14), 3485-3494***. In this example we have a complex membrane structure at the silicon/water interface which is representative of the Gram negative bacterial outer membrane.

This structure contains an asymmetric distribution of lipids with an inner leaflet composed tails deuterated phospholipid (in this case $d_{62}$-dipalmitoylphosphatidylcholine) and an outer leaflet of bacterial lipopolysaccharides which are hydrogenous (or 99.98% protium labelled). The figure below gives some details of this structure:




Above we have reflectometry data (shown in $R \times Q^4$ format, great for publications to show quality of fits) from the asymmetric membrane structure examined under three differing solution contrasts ($D_2O$, Si-MW (38% $D_2O$) and $H_2O$). From the analysis of the data we reveal the internal structure of the bilayer at the solid/liquid interface. The deuterated phospholipid component is located in the inner membrane leaflet close to the silicon interface and the hydrogenous lipopolysaccharide is located in the outer leaflet close to the bulk solution. Note that the profile for this consists of lipid tail and sugar head-group (core oligosaccharide) region.

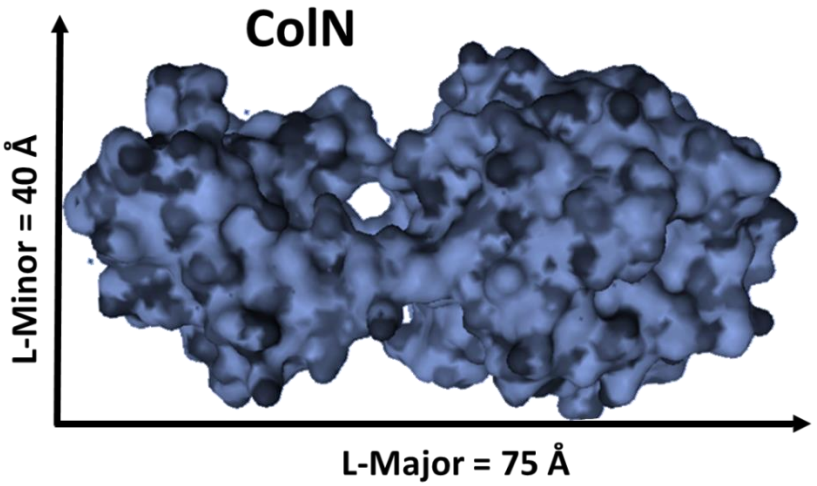
Upon opening the **Part 3 Protein-Bound Membrane Structure>Protein-Bound Membrane Structure.mat** the following data sets and fits will load in RasCal:



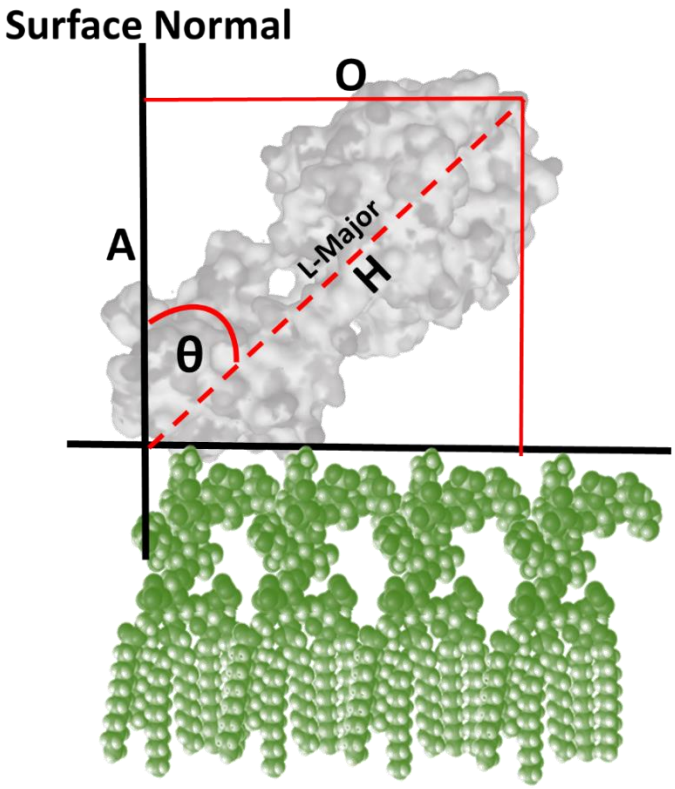

You will see six NR data sets which describe an asymmetric membrane structure across the silicon/water interface.

The top three data sets come from the membrane only and you can see are well fitted to the SLD profile shown on the left. The bottom three data sets are the same membrane after the binding of the antibacterial protein colicin-N (ColN) to the surface and, as can be seen, are not well fitted to a model which does not have the protein on the surface (line is model fit and error bars are data).

ColN is a cigar shaped protein:




This prolate shape means that **a;** with a single layer of protein on the surface of the membrane the thickness of that layer will be between **40** and **75 Å** and **b;** that from that thickness we can determine the angle of the protein relative to the surface normal:




Using RasCal we will fit the angle of ColN relative to the surface normal and convert this value in the RasCal custom file into a thickness which will be input into the Abeles matrix calculation in the software.

The area of code you need to pay the most attention to is the area where we convert protein surface angle to thickness (lines 26 to 41):

```
%%ColN angle to thickness conversion code! Remember your trigonometry
%%%%

ColN_minor_axis = 40; %COlN is a cigar shaped protein and this length is
the minor (thinnest axis)

ColN_major_axis = 75; %This is the longest axis.

ColN_Layer_angle = (ColN_Layer_angle * (pi/180)); %Here we convert the
angle to radians.

ColN_ratio = 0; %%%%%; %Here we you need to specify how we determine the
ratio between our two length scales.

ColN_Layer_Thickness = 0; %%%%%%%; %Here we then need to relate the ratio
to the thickness of the proteins minor and major axis.

%%%%% from here you line 99 you don't need to change anything
```

Note the parameters that describe what you want to fit for the protein are already in the parameter list but they have not been added to the structure yet which you need to do once the relationship between angle and protein layer thickness has been defined.

Simply add the ColN layer to contrasts 3-6 (the protein bound data sets) in the structure to look like this:

```
COLN = [ColN_Layer_Thickness ColN_SLD ColN_Layer_Roughness ColN_Layer_hydration
1];

switch contrast
    case {1}
        output = [SILICONOXIDE; INNERHG; INNERTAILS; OUTERTAILS; CORE];
    case {2}
        output = [SILICONOXIDE; INNERHG; INNERTAILS; OUTERTAILS; CORE];
    case {3}
        output = [SILICONOXIDE; INNERHG; INNERTAILS; OUTERTAILS; CORE];
    case {4}
        output = [SILICONOXIDE; INNERHG; INNERTAILS; OUTERTAILS; CORE; ColN];
    case {5}
        output = [SILICONOXIDE; INNERHG; INNERTAILS; OUTERTAILS; CORE; ColN];
    case {6}
        output = [SILICONOXIDE; INNERHG; INNERTAILS; OUTERTAILS; CORE; COlN];
end
```

Once you have added the ColN layer to the three data sets and successfully compiled the model, fit the angle of ColN relative to the surface normal and note it down.

**HINT: Just select the protein layer parameters to fit**

**SAVE PROJECT ON THE IDAAAS DESKTOP IF YOU WISH TO LOOK AT IT FURTHER**

## HINT!: Trigonometry:

SOH-CAH-TOA

**Sin = Opposite/Hypotenuse**

In Matlab:

y=sin(x);

x is angle in radians (angle in degrees × π/180)

**Cos = Adjacent/ Hypotenuse**

In Matlab:

y=cos(x);

**Tan = Opposite/Adjacent**

In Matlab:

y=tan(x);

## A Brief Introduction to Parratt's Recursive Algorithm:

**INPUT**:

*Q* - single value [$Å^{-1}$] (from an iteration loop)

**CONSTANTS**:

*N* - total number of layers (excluding subphase).

$\rho_n$ - scattering length density [$Å^{-2}$] of layer *n.*

$\sigma_n$ - roughness [Å] of layer *n.*

$d_n$ - layer thickness [Å] of layer n N - total number of layers (excluding subphase).

**INITIAL CONDITIONS**:

Neutron beam incomes via layer n = 0

Layer *N + 1* is the subphase with $d_{N+1} = \infty$ (e.g. $D_2O$, Si etc)

$R_{N+1} = 0$, which implies $r_{N+1} = r_{N,N+1}$

For *N* layers loop over n by decrementing from n = *N* +1 to *n* = 1

**EQUATIONS**

$$K_0 = \sqrt{\frac{Q^2}{4} + 4\pi\rho_0} \tag{1}$$

$$K_n = \sqrt{K_0^2 + 4\pi(\rho_{n-}\rho_0)} \tag{2}$$

For the last film layer when *n* = *N*:

$$r_{(N,N+1)} = \frac{K_N - K_{N+1}}{K_N + K_{N+1}} e^{-2i[\sigma_{N+1}]^2 K_N K_{N+1}} \tag{3}$$

$$r_N = \frac{r_{(N-1,N)} + r_{(N,N+1)} e^{2id_n k_N}}{1 + r_{(N-1,N)}\, r_{(N,N+1)} e^{2id_n k_N}} \tag{4}$$

For all other interfaces where n < N:

$$r_{(n-1,n)} = \frac{K_{n-1} - K_n}{K_{n-1} + K_n} e^{-2i[\sigma_n]^2 K_{n-1} K_n} \quad (5)$$

$$r_N = \frac{r_{(n-1,n)} + r_{(n+1)} e^{2id_n k_n}}{1 + r_{(n-1,n)}\, r_{(n+1)} e^{2id_n k_n}} \quad (6)$$

Where $k_0$ is the wavenumber of the incoming neutron beam; $k_n$ is the neutron beam wave-vector in layer $n$; $r_{N,N+1}$ is the reflectivity from the interface between the last layer $N$ and the subphase $N + 1$ and $r_{N-1,N}$ can be found using the equation (5). Note that for $n = N - 1$ the $r_{n+1}$ in equation (6) is equal to the value of $r_N$ from equation (4).

Finally once the process has been recursively calculated for each layer interface moving from bottom to top R(Q) is calculated from the reflectivity coefficient of the top interface ,r1, multiplied by its complex conjugate (7):

$$R(Q) = r_1 (r_1)^{\dagger} \quad (7)$$