# GO-RXR: User Guide

Version: 2023-0.3

Lucas Korol

July 24, 2023

G O - R X R

## Introduction

Global Optimization of Resonant X-ray Reflectivity (GO-RXR) is a data analysis package used to uncover the structural, electronic, and magnetic properties of multilayered thin films by simulating and fitting the x-ray reflection to experimental data. GO-RXR is capable of simulating non-magnetic and magnetic reflectivity spectra of varying energies and polarization.

# Contents

# 1 Overview

GO-RXR offers many different features all with the common goal of facilitating the data analysis process and reducing the overall time spent analyzing a material. These features include:

- graphical user interface,
- material model defined with a compound profile,
- adaptive layer segmentation,
- internal database of form factors,
- energy shift and scaling of form factors,
- allows for the selection of form factors in the project directory,
- magnetism capabilities,
- compatibility with ReMagX,
- script functionality,
- customizable cost function in data fitting,
- data smoothing functionality for total variation,
- progress workspace to view fit progress.

# 2 Project Workspace

Upon running the software a window similar to figure 2.1 will appear.

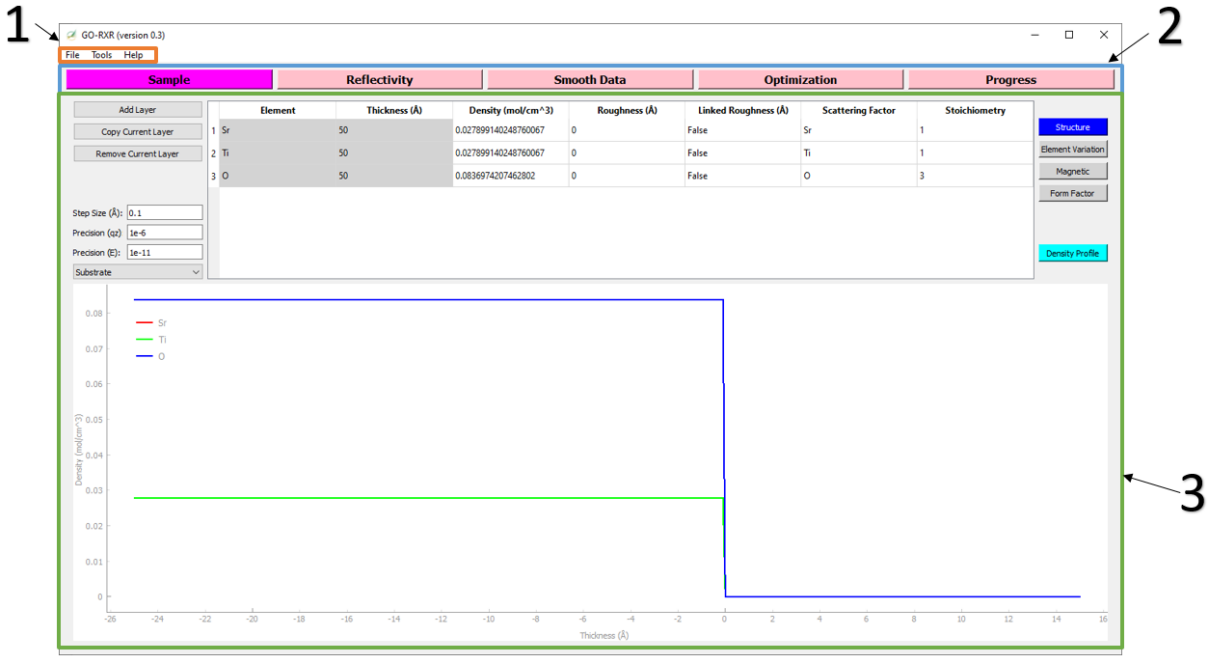


Figure 2.1: Start Screen

There are three main sections in the window:

1. **Toolbar** - enclosed by orange square
2. **Workspace Navigator** - enclosed by blue square
3. **Workspace Area** - enclosed by green square

It is important to note that in the plotting workspace, there are multiple different options the user can select by right-clicking on the plot. This includes transformations and adding grids. In particular, there is an option to take the derivative of the data in the plotting space which is useful when checking the quality of the data smoothing.

# 3 Toolbar

## 3.1 File

The file type chosen to save the workspace is The Hierarchical Data Format version 5 or otherwise known as HDF5. This requires a '.h5' file extension for the project workspace file. Not all files need to be an HDF5 file (e.g. form factor files) and will be indicated in the user manual.

The file tab is used to load and save data for the current workspace. The file tab contains the options:

- New Workspace
- Load Workspace
- Load Sample
- Save Workspace
- Save Workspace As
- Save Sample
- Save Simulation
- Import Dataset
- Load ReMagX
- Save Summary
- Exit

### New Workspace

The **New Workspace** option starts a completely new workspace in the directory with the selected file name. The workspace will be loaded in with the default parameters including a pre-defined substrate layer with the chemical formula of $SrTiO_3$. The default substrate layer can easily be removed by pressing the **Remove Current Layer** button and then adding your desired substrate by selecting the **Add Layer** button as shown in figure **??**. It should also be noted that when creating a new workspace the previously loaded experimental dataset will be cleared from the GUI.

### Load Workspace

The **Load Workspace** option is used to load a GO-RXR workspace with the '.h5' file extension from a desired directory. Upon loading in the workspace the software will scan the workspace directory for magnetic and non-magnetic form factor files that are being used in the selected workspace. The GUI will not be able to access any form factor files that are included after the workspaces were already loaded, hence, if a new form factor file is included the workspace must be reloaded. If unknown form factors are requested then the software will not be able to simulate the experimental data.

### Load Sample

**Load Sample** allows the user to load in only the sample model. This is ideal when working on multiple samples with similar configurations with slightly different characteristics. This allows the user to load in an optimized material model for similar configurations for different data sets, which can serve as an excellent starting point for the data analysis, ultimately, reducing the time spent analyzing the data.

### Save Workspace and Save As Workspace

The **Save Workspace** saves the project workspace to the current file. The **Save As Workspace** allows the user to save the current workspace to a new directory with a new file name. Both of these options save all the data except for the simulated data. The reason for this is it takes a long period to calculate the simulations for all the experimental data. Considering the data analysis process is an iterative process and the user will save the project multiple times during a data analysis session, it was decided to neglect to save the simulations each time and to provide the option to save the simulations once a desired configuration is achieved.

### Save Sample

The **Save Sample** option allows the user to save only the sample information to the current project data file.

### Save Simulations

The **Save Simulations** option will save the simulations for the current sample model to the current project data file. Once this option is selected pop-up will appear. Click the **Start Saving** button and the software will begin calculating the simulated experimental data. There is a progress bar that will demonstrate the progress of the calculations. Note that the simulations will only be saved once the calculations are finished and the automatic exit of the pop-up will signal the calculations have been saved to the project file.

### Import Dataset

The **Import Dataset** is used to load the experimental data from a workspace or data file with the HDF5 file format. To create a data file with an HDF5 file format refer to the **File Format** section of the user manual.

### Load ReMagX

The **Load ReMagX** option allows the user to import the data from a ReMagX file with the '.all' extension. It is important to note that only the experimental data will be loaded into the workspace as the definition of the material model differs significantly.

### Save Summary and Exit

The **Save Summary** allows the user to save the workspace information to a text file. This will provide a summary of the sample configuration and the fitting parameters. Finally, the **Exit** option can be used to exit the program.

## 3.2 Tools

### Script

The script option allows for more flexibility in the data fitting. Upon selecting the script option a new window as shown in figure 3.1 will appear.

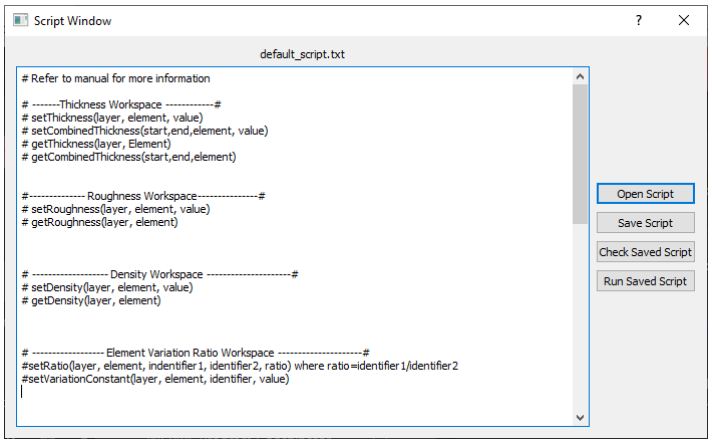


Figure 3.1: Script Window

The default setting is to use a predefined script that is available to all projects. Anytime a change is made to this script and saved by selecting **Save Script** button, the saved changes will be made available to any other project. Any changes saved to the default script will be lost when the program is terminated. If you would like to be able to save a custom script you can include a text file into your project directory (or any directory) and then use the **Open Script** button. You can then search and select the desired script. From this point onward, anytime you press **Save Script** the changes will be saved to the opened script file.
It is important to mention that the script has a limited set of internal functions and simple arithmetic and loops cannot be applied. In future versions, simple arithmetics and a loop functionality will be included. To make sure the script has the proper notation and has no obvious errors the **Check Saved Script** button can be clicked. The software will then load in the script and check to see if any errors have been made. If an error has been made a pop-up will emerge stating there is an error in the script. The current version GO-RXR does not provide an advanced debugging option but can provide the line and possible type of error. The **Run Saved Script** button will run the most recently saved script and apply all internal functions to the sample model in the sample workspace. When the script is being used in a data fit the script will override any fitting parameter. For example, if the thickness of a specific layer and element is chosen to vary but in the script the same thickness was set to be constant, in the end, the thickness will remain constant.

As previously mentioned, the script can only use predefined internal functions that are separated into 'get' and 'set' functions. The 'get' functions are used to retrieve a value and can only be used as follows

```
a = getFunction(parameters)
```

where 'a' acts as a value placeholder that can later be input into a 'set' function. Note that the value placeholder can only be used as an input for the 'set' functions. The detailed

descriptions of each function will indicate which parameters can use the placeholder variables. Again, future updates will consider making the script more flexible. The set function is defined as follows:

```
setFunction(parameters)
```

It should also be made clear that a 'get' function cannot be input as a parameter as shown below.

```
setDensity(0,Sr,getDensity(1,Mn))
```

The correct way to define a 'get' function is shown below.

```
rho = getDensity(1,Mn)
setDensity(0, Sr, rho)
```

**Get Function Definition**

**getThickness(layer, element)**

Purpose: Get the thickness of an element in a specified layer

Parameters:

**layer –** index of the layer and must be entered as an integer (substrate starts at 0)

**element –** element symbol, if 'all' is entered then this assumes the thickness for all the elements in the layer are the same and the thickness for the first element is used.

**getCombinedThickness(start, end, element)**

Purpose: Get the total density from layer 'start' to layer 'end' for the specified element

Parameters:

**start –** index of the start layer and must be entered as an integer (substrate starts at 0)

**end –** index of the end layer and must be entered as an integer

**element –** element symbol, if 'all' is entered then this assumes the thickness for all the elements in the layer are the same and the thickness for the first element is used.

**getRoughness(layer, element)**

Purpose: Get the roughness of an element in a specified layer

Parameters:

**layer –** index of the layer and must be entered as an integer (substrate starts at 0)

**element –** element symbol, if 'all' is entered then this assumes the roughness for all the elements in the layer are the same and the roughness for the first element is used.

**getDensity(layer, element)**

Purpose: Get the density of an element in a specified layer

Parameters:

**layer –** index of the layer and must be entered as an integer (substrate starts at 0)

**element –** element symbol, if 'all' is entered then this assumes the density for all the elements in the layer are the same and the density for the first element is used.

**getMagDensity(layer, element, variation)**

Purpose: Get the magnetic density of an element in a specified layer

Parameters:

**layer –** index of the layer and must be entered as an integer (substrate starts at 0)

**element –** element symbol

**variation –** element variation identifier, if the element does not have any variation then input the element symbol again

**getEshift(ffname)**

Purpose: Get theenergy shift of a form factor

Parameters:

**ffName –** name of the form factor

**getMagEshift(ffmName)**

Purpose: Get the energy shift of a magnetic form factor

Parameters:

**ffmName –** name of the form factor

**Set Function Definition**

**setThickness(layer, element, d)**

Purpose: Set the thickness of element in specified layer

Parameters:

**layer –** index of the layer and must be entered as an integer (substrate starts at 0)

**element –** element symbol, if 'all' is entered then this assumes the thickness for all the elements in the layer are the same and the thickness for the first element is used.

**d –** thickness value entered as a float or placeholder value

**setCombinedThickness(start, end, element, d)**

Purpose: Set the total density from layer 'start' to layer 'end' for the specified element

Parameters:

**start –** index of the start layer and must be entered as an integer (substrate starts at 0)

**end –** index of the end layer and must be entered as an integer

**element –** element symbol, if 'all' is entered then this assumes the thickness for all the elements in the layer are the same and the thickness for the first element is used.

**d –** thickness value entered as a float or placeholder value

**setRoughness(layer, element, sigma)**

Purpose: Set the roughness of element in specified layer

Parameters:

**layer –** index of the layer and must be entered as an integer (substrate starts at 0)

**element –** element symbol, if 'all' is entered then this assumes the roughness for all the elements in the layer are the same and the roughness for the first element is used.

**sigma –** roughness value entered as a float or placeholder value

**setDensity(layer, element, rho)**

Purpose: Set the density of element in specified layer

Parameters:

**layer –** index of the layer and must be entered as an integer (substrate starts at 0)

**element –** element symbol, if 'all' is entered then this assumes the density for all the elements in the layer are the same and the density for the first element is used.

**rho –** density value entered as a float or placeholder value

**setRatio(layer, element, id1, id2, ratio)**

Purpose: Sets the ratio between id1 and id2 to be constant (e.g. ratio = id1/id2). Note that this function can only be used for 3 element variations

Parameters:

**layer –** index of the layer and must be entered as an integer (substrate starts at 0)

**element –** element symbol, the parameter 'all' cannot be used as an input in this case

**id1 –** Element variation identifier

**id2 –** Element variation identifier

**ratio –** Ratio value defined as ratio=id1/id2

**setVariationConstant(layer, element, id, value)**

Purpose: Sets the ratio of a specific element variation to be constant. This function makes sure that all other element variation ratios are changed such that the summation of all element variation ratios is equal to one.

Parameters:

**layer –** index of the layer and must be entered as an integer (substrate starts at 0)

**element –** element symbol, the parameter 'all' cannot be used as an input in this case

**id –** Element variation identifier

**value –** Constant ratio value

**setMagDensity(layer, element, variation, rho)**

Purpose: Set the magnetic density of element in specified layer

Parameters:

**layer –** index of the layer and must be entered as an integer (substrate starts at 0)

**element –** element symbol

**variation –** element variation identifier, if element is not an element variation than input the element symbol again

**rho –** density value entered as a float or placeholder value

**setEshift(ffName, dE)**

Purpose: Set the density of element in specified layer

Parameters:

**ffName –** form factor name

**dE –** energy shift entered as a float or placeholder value

**setMagEshift(ffmName, dE)**

Purpose: Set the density of element in specified layer

Parameters:

**ffmName –** magnetic form factor name

**dE –** energy shift entered as a float or placeholder value

## Form Factor

The form factor tool is used to view the form factors used in the workspace. Upon selecting the form factor tool the window in figure 3.2 will appear.

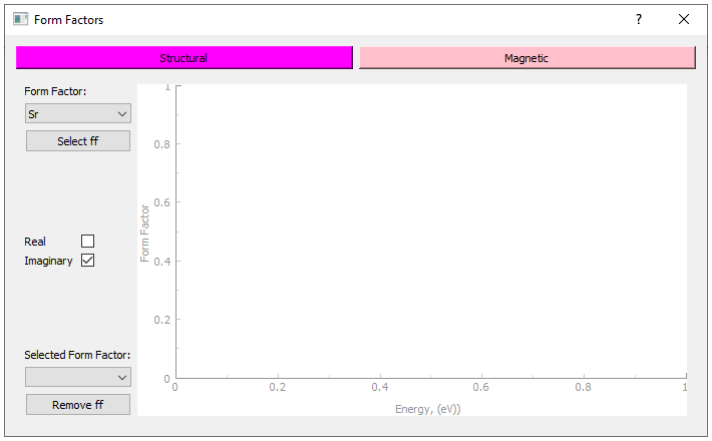


Figure 3.2: Form Factor Start Screen

The buttons labeled **Structural** and **Magnetic** are used to toggle between the non-magnetic and magnetic form factors, respectively. On the top left corner of the window, there is an option box that contains all the form factors that are found in the project workspace. It is important to note that the form factor will not appear when clicking the desired form factor from the option box, but the button label **Select ff** must be clicked to view the form factor. Once **Select ff** is clicked then the real and imaginary components of the form factor as a function of energy will be plotted in the plotting space. Multiple form factors can be selected and viewed at the same time using the **Select ff**. This is very useful when dealing with experimental datasets that are difficult to reproduce cia calculation as the form factors provide valuable information on the elements that have the largest impact on reflectivity for the energy in question. An example of what the window will look like is shown in figure 3.3.

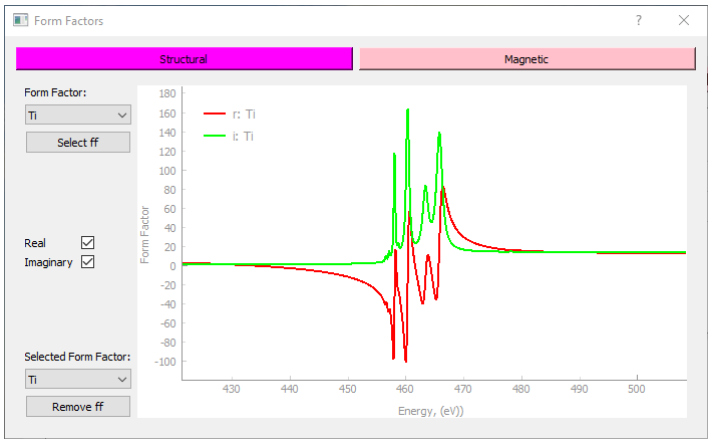


Figure 3.3: Form Factor Start Screen

Once a form factor has been selected it will appear in the option box located in the bottom left corner of the window. This option box can be used to remove a form factor from the viewing space by selecting the desired form factor and then pressing the **Remove ff** button.

The two checkboxes labeled **Real** and **Imaginary** allow the user to select whether the user wants to view the real and/or imaginary component of the form factor.

### Help

The help tab contains the options **License** and **About**. The **License** option will provide information about the licensing of GO-RXR. The **About** option will open up the user manual.

# 4 Workspace Navigator

The workspace navigator contains five buttons that are used to navigate through the different workspaces available. The five options are **Sample**, **Reflectivity**, **Smooth Data**, **Optimization**, and **Progress**. The **Sample** button will navigate to the sample workspace and is used to construct the model of the sample. The **Reflectivity** button will navigate to the reflectivity workspace and is used to view the experimental data and simulations. This workspace also allows the user to alter the background shift, scaling factor, boundaries, and weights of the different scans. The **Smooth Data** button will navigate to the data smoothing workspace and provides the user the ability to smooth the data. The **Optimization** button will navigate to the optimization workspace which is used to select the data fitting algorithm, and parameter boundaries, and start a data fit process. Finally, the **Progress** button will navigate to the progress workspace and is used to view the progress of the currently running data fitting process. A more detailed explanation of each workspace will be provided in the workspace section of the user manual.

# 5 Workspace

## 5.1 Model Definition

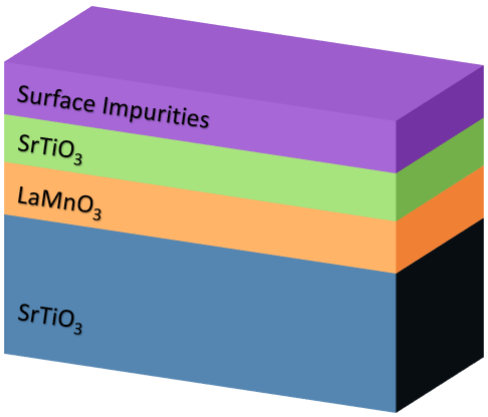


Figure 5.1: Model Definition

In the development of the software, the idea was to construct the model of the sample using a layer-by-layer approach where each layer is defined as compounds with known stoichiometry. A visual representation of this type of model is shown in figure 5.1. This type of model was chosen in hopes of simplifying the model and making it easier to maintain a physically possible material model. GO-RXR was also developed considering that elements can have a linked roughness at the interface of each layer. This requires that all layers have the same number of elements defined in their chemical formula. If there are multiple layers with a different number of elements the issue can be resolved by using 'dummy' variables. This is done by filling the 'missing' elements in the chemical formula with a pre-defined variable where their atomic mass and form factors are given to be zero. The possible dummy variables that can be used include the symbols A, D, E, G, J, L, M, Q, R, T, X, Z. An example will be provided in the **Adding and Removing Layers** section of the user manual.

## 5.2 Structure Workspace

The structure workspace can be selected by pressing the **Structure** button. When this button is pressed a table will appear with the headers element, thickness, density, roughness, linked roughness, scattering factor, and stoichiometry, along with their units, as shown in figure 5.2.

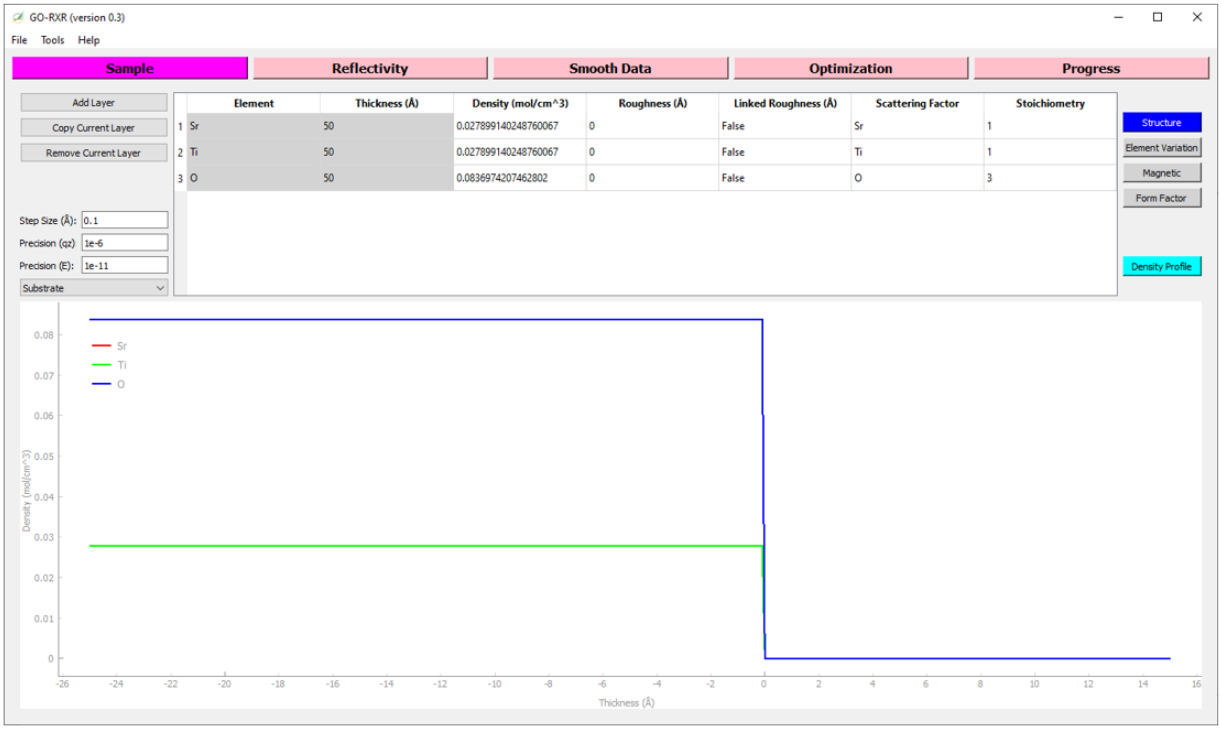


Figure 5.2: Structure Workspace

Below the element header are the symbols of the elements found in the layer. The user can navigate the different layers by selecting the layer option box labeled **Substrate** in figure 5.2. The element symbol cannot be changed in the structure workspace once it has been defined. Note that all grey columns demonstrate parameters that cannot be changed.

The thickness header is in units of angstrom, the typical units used when studying thin films. GO-RXR does allow for different thicknesses to be set for each element in the layer, but this is not recommended as it can be confusing and difficult to keep track of how the difference in thickness propagates throughout the rest of the sample model. For example, let's define an element with the arbitrary symbol 'A'. In the sample, it is believed that element 'A' may terminate before the rest of the elements in the layer. Instead of altering the thickness of element 'A' in the current layer to be smaller than the rest of the elements it is suggested to add another layer. After adding another layer the termination can be modeled by settting the density of element 'A' to zero and altering the combined thickness of the added layer. Both methods have been used in the modeling of layer terminations, but adding an additional layer is much more useful. It should also be noted that the element and thickness column are grey. This signals that the value cannot be changed (element header) or the change will not have an impact on the density profile (thickness header). It is important to clarify that the thickness column will only be grey for the substrate layer. This is because in the model the substrate thickness does not matter and is always assumed to be

infinite relative to the incoming x-rays.

The density header is the density of the element in units of $mol/cm^3$. The density of each element can be set to any desired value independently of the pre-defined stoichiometry. The stoichiometry will only be used in the data fitting of the density if signaled. This will be discussed further in the **Optimization Workspace** section.

The roughness header is the roughness of the element and describes how the density of the material transition at the interface between two layers. The roughness is modeled using the Nevot-Croce roughness approximation and uses the error function shown in equation 1.22.

$$erf(z,\sigma) = (\sigma\sqrt{2\pi}) \int_{-\infty}^{z} e^{-\zeta^2/2\sigma^2} d\zeta \tag{1}$$

where sigma is the values input in the roughness column. In general, the roughness column refers to the rate at which the elements density decreases at the interface.

The linked roughness determines whether or not the increase in the density of the elements is directly related to the decrease of a different element in the previous layer that is in the same positional argument of the chemical formula. For example, for perovskite oxides with the chemical formula $ABO_3$ all symbols found in the *A*-position are linked. Similarly, all elements found in the *B*-position are linked. If the linked roughness is set to *False* or is left empty then it is assumed that the roughness is linked. If a numerical value is entered into the linked roughness column then the rate the density increases for the element in the current layer will be different than that of the rate of decrease for the element in the previous layer.

The scattering factor column (which is the synonymous with form factor) indicates the name of the scattering factor that will be used in the reflectivity calculation. A database of scattering factors does exist for all elements and can be selected by inputting the symbol of the element. Upon loading the workspace GO-RXR will search the directory for form factor files. Any form factor file name that matches one of the defined form factors in the model will be replaced by the form factor file found in the directory. It is suggested to use unique names and to avoid using any element symbols as a name, such that the form factors found in the database and the directory can easily be toggled. The format of such a file will be discussed further in the **Form Factor File** section of the user manual. Note that if an element has multiple element variations and has been defined as such in the model (e.g. different oxidation states) then a list of form factors will appear in the scattering factor field for that element. At this point, the form factors cannot be altered in the structure workspace and the scattering factors must be changed in the element variation workspace. It should also be noted that when a scattering factor for an element is changed in a layer, then the scattering factor will automatically be changed for all layers where the element's symbol is found.

The stoichiometry column determines the stoichiometric relation between the different elements in the layer. This is used in a select few cases but is mainly used when fitting the density profile or when adding a new layer. For example, when fitting the density there is an option to perform an element fit or a compound fit. The element fit will ensure that the density of the selected element is fit independently of the stoichiometry of the layer. However, in the case of the compound fit the density will be changed for all elements in

the layer while maintaining the stoichiometry.

## 5.3 Element Variation Workspace

The element variation workspace is used to add the different element variations that may exist for an element (e.g. oxidation states) and is shown in figure 5.3. The default is to allow for two possible element variations, but the add button can be used to include an arbitrary number of variations. Note that whenever an element variation is added, the element variation will appear in every layer where that element exists. The option box on the top left corner of the workspace contains all the different elements present in the currently selected layer. The delete button will only remove the added element variations to the minimum number of rows of two. The element variations remaining in the rows can be removed from the layer (and all layers) by manually removing all values in the row.

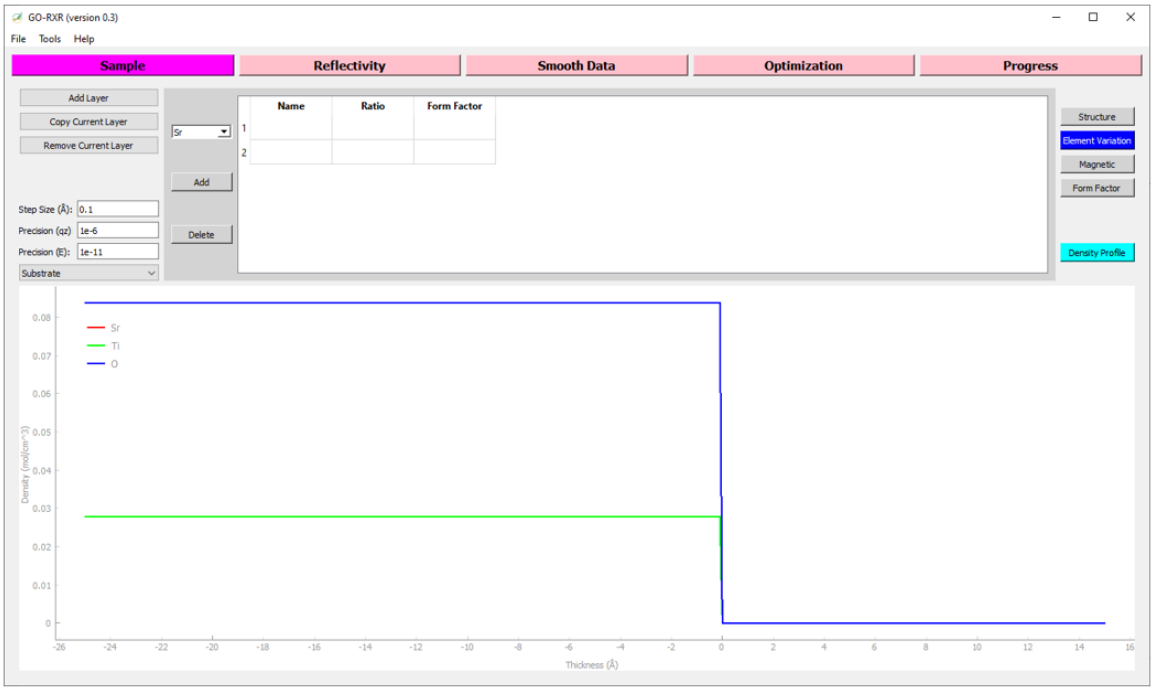


Figure 5.3: Element Variation Workspace

When adding an element variation three parameters need to be considered, which include name, ratio, and form factor. The name can be any string and can include the '+' or '-' symbols. The only requirement is that each name must be unique and cannot be the same as any other already defined element found in the structure workspace. The ratio must be a value between 0 and 1, where the sum of all the ratios are equal to 1. The form factor works the same way as the form factors in the sample workspace. When an element variation is initialized the form factor in the sample workspace will appear as a list containing all the form factors used for each element variation. This helps to clarify which elements have an element variation associated with them, and what they might be.

## 5.4 Magnetic Workspace

The magnetic workspace is used to create the magnetic density profile and is shown in figure 5.4. In the magnetic workspace a row will be given for each element or element

variation in the current layer. Only the elements or element variations that have a magnetic component need their rows filled out. The column with the header magnetic density has the units of $mol/cm^3$. The magnetic density is not necessarily the best term to use and can be related to the number of bohr magnetons per element but what is more important is to be able to quanitfy the magnetic contribution as a function of the material's depth. The form factor column is where the name of the magnetic form factor is included. Currently, only form factors for Nickle (Ni) and Cobalt (Co) have been included into the database.


Figure 5.4: Magnetic Workspace

In the top right corner, there is an option to include the magnetization directions. There are only three options provided in the options box which include the x-direction, y-direction, and z-direction (default). A future version of GO-RXR will include an arbitrary magnetization direction defined by $\phi$ and $\gamma$. Currently, when the magnetization direction is changed in one layer this will change the magnetization direction to be the same in the rest of the layers. In future versions, it is being planned to provide the ability to give unique magnetization directions for each layer and allowing to fit the direction to the experimental data.

Note that the thickness and roughness of the magnetic density profile are controlled by the structure workspace. The idea behind this is that it was assumed that the magnetic profile would have a similar profile as the non-magnetic component. This also helps make sure that the magnetic component does not exist in regions where its non-magnetic counterpart does not exist.

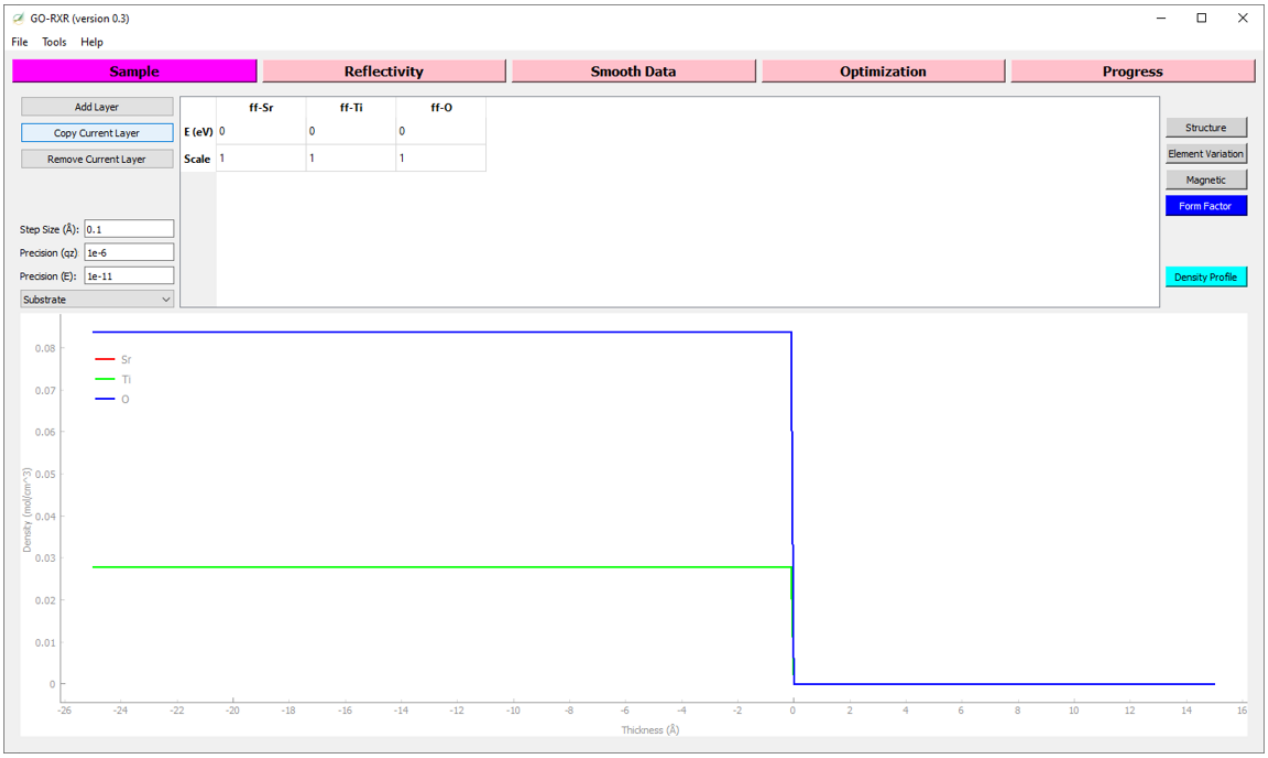


Figure 5.5: Form Factor Workspace

## 5.5 Form Factors Workspace

The form factor workspace is used to add energy shifts or to scale the form factors. The form factor workspace is shown in figure 5.5. The name of the form factors is found in the headers where the 'ff-' prefix denotes a non-magnetic form factor and 'ffm-' denotes a magnetic form factor. There are two rows where the *E* row is the energy shift and *Scale* is the scaling factor. The energy shift can be entered as positive or negative numbers where the positive numbers will shift the form factor to higher energies and a negative value will shift to lower energies. It is important to add energy shifts when the resonance peaks for the simulations and experimental data do not align. The scaling factor is typically only used for the magnetic form factors so that the magnetic density can have a similar scale as the non-magnetic density. However, in some cases, we are unsure whether the scaling of the form factors in resonance is correct. The scaling of the non-structural form factors has been found useful when estimating the amount the scaling in the resonance regions needs to be changed.

## 5.6 Adding and Removing Layers

Upon clicking the add layer option a window similar to 5.6 will appear. The added layer requires that the layer information is input in compound mode.

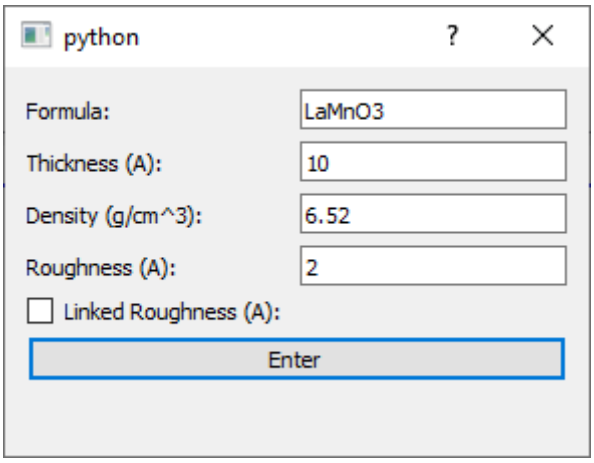


Figure 5.6: Adding Layer Window

The formula text box requires the user to enter the chemical formula. In the case where a dummy variable is required, the user enters the chemical formula as normal but inputs the dummy variable in the location where there is no linked roughness (e.g. LaMnDO3 where D is the dummy variable). The thickness field is the thickness of all the elements within the layer. The value must be input as a number. The density is input in units of $g/cm^3$ for the layer in compound form and is converted to $mol/cm^3$ for each element in the structure workspace depending on the stoichiometry. There is also a database of atomic masses for each element in the periodic table where the pre-defined dummy variables (A, D, E, G, J, L, M, Q, R, T, X, Z) have an atomic mass of zero. Perovskite materials with known densities will automatically be input into the density field. The roughness field is the roughness of all the elements in the layer. The value must be entered as a number. The linked roughness field can be toggled on or off. If the check box is selected the linked roughness will appear and the user can input a linked roughness for all elements. If the linked roughness check box is not selected the linked roughness of each element will be set to False.
Once all the parameters are set then **Enter** can be clicked. This will create a new layer in the structure workspace, which will be entered above the currently selected layer. For example, if there are two current layers called substrate and top film (substrate, top layer) and the currently selected layer is the substrate, the new layer will then be input between the substrate and top layer (substrate, new layer, top layer). If an element has already been defined to have a magnetic structure or element variation GO-RXR will automatically include the appropriate parameters to the layer.

The **Copy Current Layer** button will take the currently selected layer and create a new layer on top of the currently selected layer with the same parameters. The **Remove Current Layer** button will remove the currently selected layer. If an element in the layer that was just removed no longer exists within the model, than all prior information (e.g. form factor, element variation, magnetic) will be removed as well.

**Surface Layer**

The surface layer is required to have the same number of elements as all the layers in the model. However, in most cases, the surface impurities are made up of oxygen and carbon. How can this requirement be fulfilled if the layer in the model has more than two elements? There are a few ways this can be done:

1. dummy variables
2. multiple carbon entries

In the dummy variables case, the user can input carbon and oxygen in the desirable locations in the chemical formula and fill the remaining positions with the available dummy variables. For example, let's say we have a chemical formula with 4 elements. A possible way of defining the surface layer chemical formula is *CADO* where A and D are the dummy variables and C and O are carbon and oxygen, respectively. In the multiple carbon case, the user can put the oxygen in the desirable location and fill the remaining positions in the chemical formula with carbon. For example, let's say we have a chemical formula with 4 elements again. A possible way of defining the surface layer's chemical formula is *CCCO* where C and O are carbon and oxygen, respectively. It should be noted that in the structure workspace, each carbon will be given a number to help differentiate between the different carbons. In the case of *CCCO* in the name column the symbols *C1*, *C2*, *C3*, *O* will appear. Now it is also possible to input the surface layer with multiple oxygens, but if oxygen exists in previous layers this is not recommended. In both cases, the user can include stoichiometry to the surface layer chemical formulas (e.g. C2ADO3 or CCC2O3).

In the software, a value of 15 angstroms is on top of the last defined layer. This is done to account for the fact that an added roughness can cause the layer to extend past the defined thickness of the layer. There are, however, cases in which the surface is too rough and extends past the 15 angstroms resulting in the cutting off of the density profile. This can impact the simulations and must be addressed. The first way to address this issue is by adding another surface layer on top where the density of all the elements are zero. The thickness of the layer can then be altered such that the density profile is no longer being cut off. The other method is by increasing the thickness of an element in the surface layer that already has a given zero density.

## 5.7 Fitting Parameters

In the sample workspace the parameters that can be fit include:

1. Thickness
2. Density
3. Roughness
4. Linked Roughness
5. Energy Shift
6. Element Variation Ratio

7. Magnetic Density

To fit a parameter the user must select the desired cell and then right-click. For the structure parameters (thickness, density, roughness, and linked roughness) there are two kinds of fits: compound fit and element fit. A compound fit means that all parameters in the selected column will be tied together where the parameter boundaries are set by the element in the first row. For example, in the thickness case when a compound fit is being performed the thickness for all elements will be changed by the same amount. The thickness value will only be the same for all elements if all the values start with the same value. For example, let's say the current layer has the elements *A* (first row), *B*, and *C* with the thicknesses set to 10, 15, and 7 respectively. As a result of the data fit the thickness of element *A* changes by 5 angstroms making its new thickness 15 angstroms. This will then add angstroms to the thickness of the other remaining elements resulting in the thickness of element *B* and *C* changing to 20 and 12 angstroms, respectively. A similar process will occur for the roughness parameter. If a compound fit is selected for density then the density for all the elements will change according to the stoichiometry. For the linked roughness a compound fit can only be selected if all the linked roughness fields are numerical values and not 'False' or empty. The element fit option will only fit the parameter of the selected cell and will treat all parameters independently. An element fit will color the selected cell with a green color while the compound fit will color the entire row purple. This can be shown in figure 5.7. In the case where you have previously selected compound fits and element fits for your parameters and want them to switch, it is not required to select the unfit option and then select the desired fit. One can simply select the desired parameter and fitting mode and the software will automatically replace the old fitting mode with the new mode.

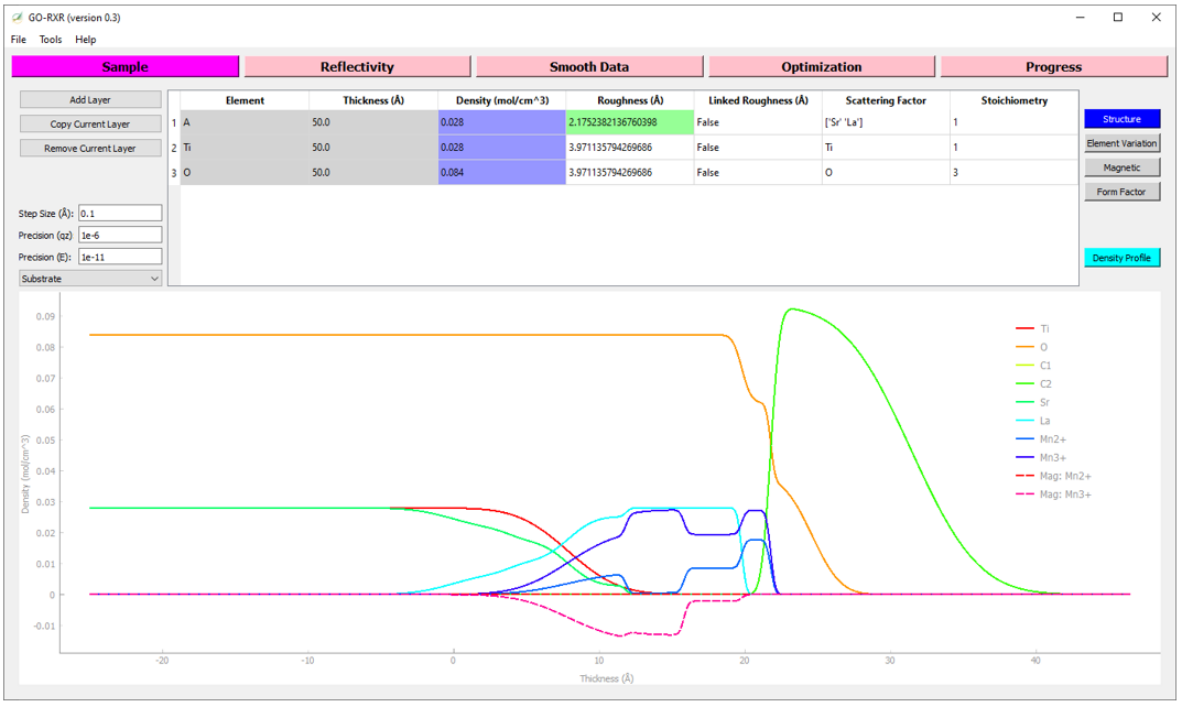


Figure 5.7: Fitting Parameters

For the element variation ratio, only one ratio can be selected for each layer. In the case of two-element variations when the selected ratio is being changed in the data fit, the non-selected ratio will automatically change such that the condition that all ratios are equal to one is fulfilled. If more than two element variations exist the default is to change the remaining element variation ratios such that their sum equals one but their respective ratios to each other remain constant. If there are three element variations the user can use the script to deviate from the default setting. Refer to the script section for more information. The energy shift can be selected by right-clicking on any of the form factors in the structure, element variation and magnetic workspaces. The user can also select the energy shift by navigating to the form factor workspace and right-clicking on the energy shift cell for the desired form factor. In some cases, it is desired to tie the energy shifts of multiple form factors (especially with oxidation states). The best way to do this is to select one of the form factors to fit, and then in the script use *getEshift* to get the changed energy shift and then use *setEshift* for the form factors that you would like to have the same energy shift. The magnetic density can be selected in the magnetic workspace. The software also allows the user to highlight multiple cells to include a compound fit, element fit, or unfit, and apply the selection to the selected cells.

## 5.8 Reflectivity Workspace

The reflectivity workspace is used to view, select, and customize the reflectivity scans. Upon selecting the reflectivity workspace a window as shown in figure 5.8 will appear.

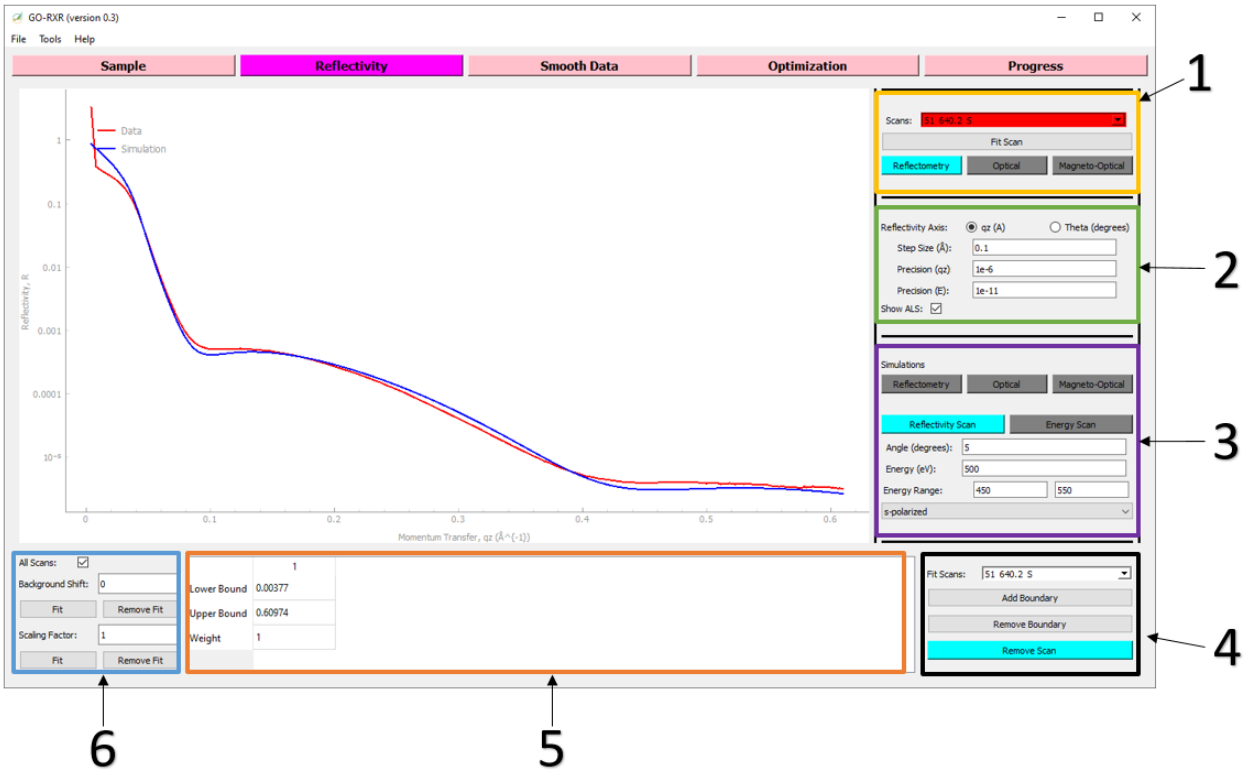


Figure 5.8: Reflectivity Workspace

The workspace can be separated into six separate regions:

1. **Data Scans** - enclosed by yellow square
2. **Axis Toggle** - enclosed by green square
3. **Simulations** - enclosed by purple square
4. **Fit Scans** - enclosed by black square
5. **Boundary and Weights** - enclosed by orange square
6. **Background and Scaling** - enclosed by blue square

### Data Scans

The data scans section is used to view and select scans that you would like to fit. The name of the scans will appear in the option box. If the option box is empty, then that means that a data file has not been loading yet. There is a unique naming convention for the data scans which will be elaborated in the **File Format** section. When a data scan has been selected in the option box the simulated and experimental data (along with the simulation) will be plotted in the plotting area. If you would like to use the selected experimental data in the data fitting then the **Fit Scan** button can be clicked. The scan name will then appear in the options box in the **Fit Scans** region located at the bottom right corner of the window. There are also three buttons in this region labeled **Reflectometry**, **Optical**, and **Magneto-Optical**. If the **Reflectometry** button is clicked then the reflectivity scan will be displayed in the plotting region. The **Optical** button will display the optical profile as a function of depth for the sample model. The **Magneto-Optical** button will display the magneto-optical profile as a function of depth for the sample model.

The axis toggle section is mainly used to toggle between momentum transfer and grazing angle on the x-axis for the reflectivity scans (no changes for the energy scans). This momentum transfer option can be selected by clicking the **qz** option and the grazing angle can be selected by clicking the **Theta** option.

The step size field provides the user the ability to change the thickness step size for the density profile (this field is also found on the sample workspace). The precision fields are used in the adaptive layer segmentation implementation. The precision field labeled with $qz$ signifies the precision value used for the reflectivity scans, while the precision field labeled $E$ refers to the precision value for the energy scans. Unique fields are provided for the different types of scans to provide more flexibility to the user. There is also a checkbox labeled **Show ALS** which is used to signal if the adaptive layer segmentation is to be visualized in the plotting space for the optical and magneto-optical plotting options.

The simulation section is used to view the model simulations for a defined angle, energy, energy range, and polarization. This section can be used regardless if a data set has been loaded into the workspace. There are three buttons at the top of this section which include **Reflectometry**, **Optical**, and **Magneto-Optical**. These button work in the same manner as in the data scans section. The **Reflectivity Scan** and **Energy Scan** buttons are used to toggle between which type of simulation you would like to view in the plotting area. The **Reflectivity Scan** option will plot the reflectivity with a varying momentum transfer or

grazing angle, which depends on the selection made in the axis toggle section, for a constant energy. The **Energy Scan** option will plot the reflectivity with respect to varying energies at a constant grazing angle. The **Angle** field is the grazing angle and is the constant grazing angle used in the energy scan. The values for the grazing angle must be found between 0 and 90 degrees, but not inclusive. The **Energy** field is the constant energy for the reflectivity scan. The values for the energy can be any value larger than 0, not inclusive. The Energy Range field provides the interval for the energy scan. The leftmost entry is the lower limit while the rightmost entry is the upper limit of the energy scan. The upper limit must always be larger than the lower limit. The option box found at the bottom of the simulation section contains the different possible polarizations. This includes s-polarized, p-polarized, right circular, and left circular light. It also allows for the display of linear and circular asymmetry.

The fit scans section is used to view and change the boundaries and weights of a selected scan. The fit scans option box allows the user to determine which scan to alter and view in the plotting area. The **Add Boundary** button adds an additional column to the boundary and weights section for that specific scan. The **Remove Boundary** button will remove a column from the boundary and weights section for the selected scan. The **Remove Scan** button will remove the selected scan from the scans to fit.

The boundary and weights section is used to change the boundaries and add weights to those boundaries. For example, if multiple regions in the reflectivity scan are very noisy the user can create boundaries and exclude those regions in the data fit. In figure 5.8 the reflectivity scan demonstrates a fairly noisy portion for the interval of 0.5 to 0.8 $A^{-1}$. If we wanted to neglect this region in the data fit we would need to input the lower bound as 0.00471 and the upper bound as 0.5 in the first column. Now, let's assume we do not want to neglect this region, but we would like for this region to have less of an impact on the data fit. In this case, we can add another boundary by clicking the **Add Boundary** button. Now there will be two columns visible. In the first column, we will put the lower boundary to be 0.00471 and the upper boundary to be 0.5 with a weight of 1. The second column will have a lower boundary of 0.5 and an upper boundary of 0.8 with a weight of 0.5. The result of this is that cost of the boundary from 0.00471 to 0.5 will be multiplied by 1, while the cost of the boundary from 0.5 to 0.8 will be multiplied by 0.5. In some cases, there are specific characteristics of a scan that are important, for example, a large dip in the fringes. What I like to do is create a boundary that encapsulates that region and provides a large weight to it. I found that by doing this results in a simulation where the highly weighted region has good agreement. There may also be the case where you want to fit several scans, but there are a few that are more important than the rest. What I like to do is provide larger weights, around values of 2-4, for the more important scans. It is important to note that each scan can have a different set of weight boundaries allowing for biased cost functions.

Finally, the background and scaling section can be used to apply a background shift or a scaling factor to the scans. The check box labeled **All Scans** determines if the user would like to apply the background shift or scaling factor to all the scans that have been selected for a data fit, or the currently selected scan. If the box is checked then all the scans in the fit scans option box will have the scaling factor and background shift applied to them. An unchecked box will only apply the background shift and scaling factor to the selected scan. The **background shift** field shifts the scan up or down and typically ranges between values

of $-5 \times 10^{-8}$ and $5 \times 10^{-8}$. You will have to be careful with the background shift, especially with a negative shift, as this may cause some of the values to be negative. Considering that the data fitting is done most often using a logarithmic transformation then a negative background shift may result in undefined values. The **Scaling Factor** field is used to scale the reflectivity value of the scans. Essentially, the value input into the field will be used to multiply the reflectivity values. Typically, the scaling factor is found between 0.8 and 1.2. The **Fit** button is used to fit the background shift or scaling factor. If a fit will occur for a selected scan then the **Background Shift** or **Scaling Factor** field will turn red. If the background shift or scaling factor will not be fit then their respective fields will have a white color. The scaling factor and background shift fits can be removed by simply clicking the **Remove Fit** option.

## 5.9 Smooth Data Workspace

The smoothing data section is used to remove as much noise as possible for the selected scans to improve the total variation calculation. The total variation is calculated by essentially comparing the derivatives of the simulated and experimental data, which makes it quite obvious why the removal of noise is required for this kind of analysis. This is done using spline interpolation or a Fourier filter.

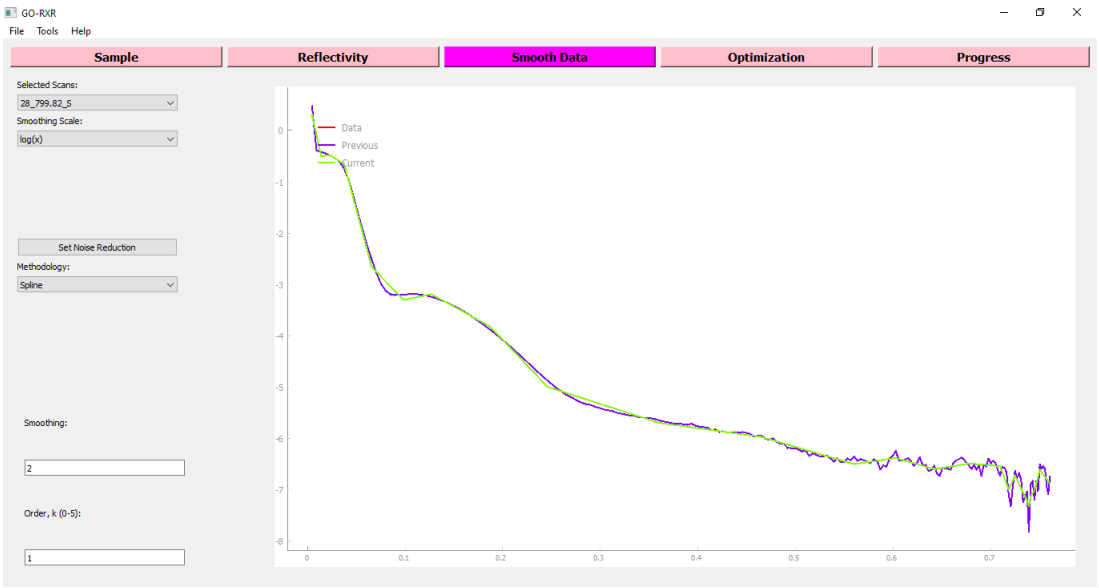


Figure 5.9: Spline Interpolation

Upon navigating to the smooth data workspace a window similar to figure 5.9 will appear. On the top left corner of the window, there are two option boxes. The option box labeled **Selected Scan** contains all the scans selected for data fitting in the reflectivity workspace. The **Selected Scans** option box is used to select which scan to work on for the data smoothing. The **Smoothing Scale** option is used to transform the reflectivity (R). The possible transformations include $R$, $log_{10}(R)$, $ln(R)$, and $R(qz) * qz^4$. Note that $R(qz) * qz^4$ takes the reflectivity data at each momentum transfer value and multiplies the reflectivity by the momentum transfer to the power of 4. On the left side directly in the middle, there is a button **Set Noise Reduction** and the option box labeled **Methodology**. The button **Set Noise Reduction** will save the current smoothed data attempt for the data fit. It should be noted

that every time the smooth data workspace is exited that the previously fit parameters are not saved, but the smoothed data is. This means that smoothed data will only be saved locally and not to a GO-RXR data file. The **Methodology** option box contains the different smoothing methodologies to use. The only two available so far are spline interpolation and Fourier filters. In future versions, it is being considered to include a neural network method.

When the spline option is selected the bottom left corner of the window will appear similar to that shown in figure 5.9. The **Smoothing** field is simply a smoothing factor. A smaller value for the smoothing factor results in less smoothing, while a larger value will result in more smoothing. The smoothing must be both positive and real value. The **Order** field is the parameter that dictates the degree of the smoothing spline. The value of the order must be a positive and real integer found between 1 and 5, inclusive. The default value is a cubic spline with an order value of 3.

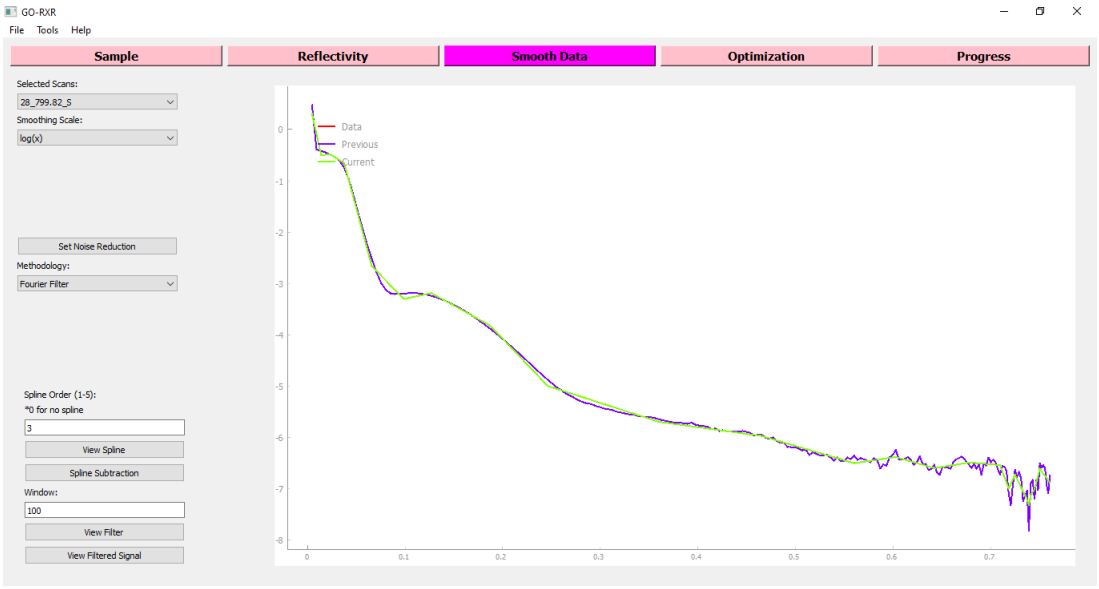


Figure 5.10: Spline Interpolation

Once the **Fourier Filter** method is selected the bottom left corner of the smooth data window will resemble that of figure 5.10. The **Spline Order** field is used to select the order of the best-fit spline. The spline order must be an integer found between 1 and 5, but a value of 0 will result in no spline being used in the filter. Once a spline order has been fit the data and the best-fit spline can be viewed in the plotting area by clicking the **View Spline** button. This will result in a plotting area similar to that found in figure 5.11.

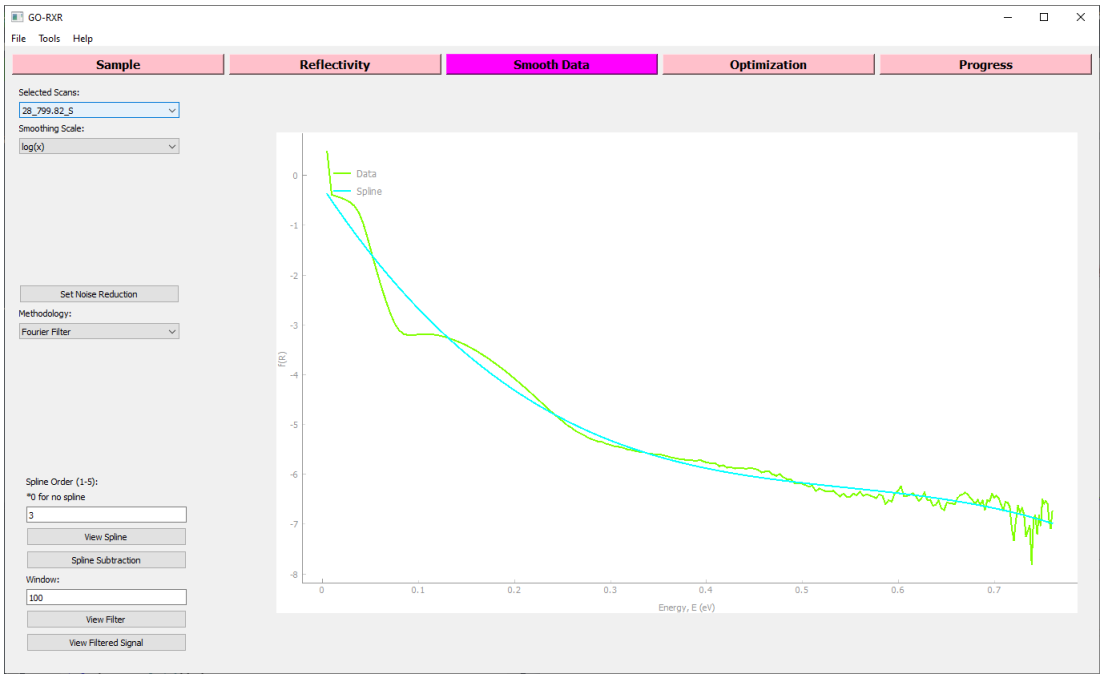


Figure 5.11: Spline Comparison

To view how the spline subtraction encapsulates the oscillations in the reflectivity scan the user can then click the **Spline Subtraction**, which is shown in figure 5.12. The spline subtraction is exactly what it sounds like, which takes the original data and subtracts the best-fit spline from it.

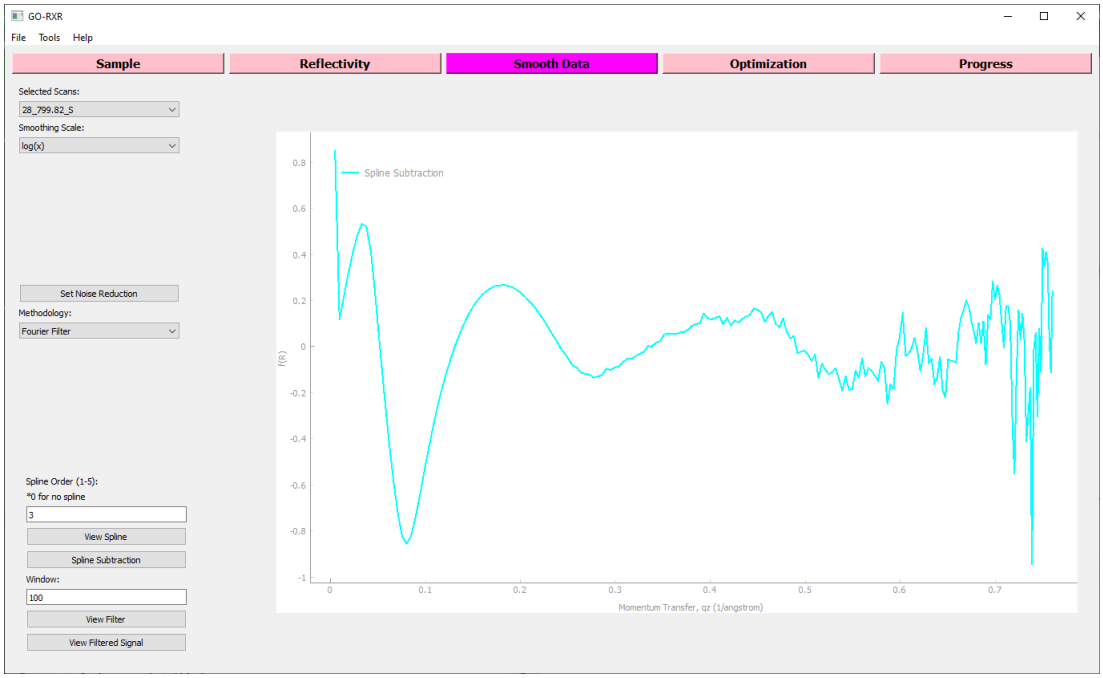


Figure 5.12: Spline Subtraction

Once satisfied with the spline subtraction the user can then select the size of window to use for the filter. To view the window with respect to the Fourier transform of the spline subtraction the **View Filter** can be selected. An example of a filter is shown in figure 5.13.

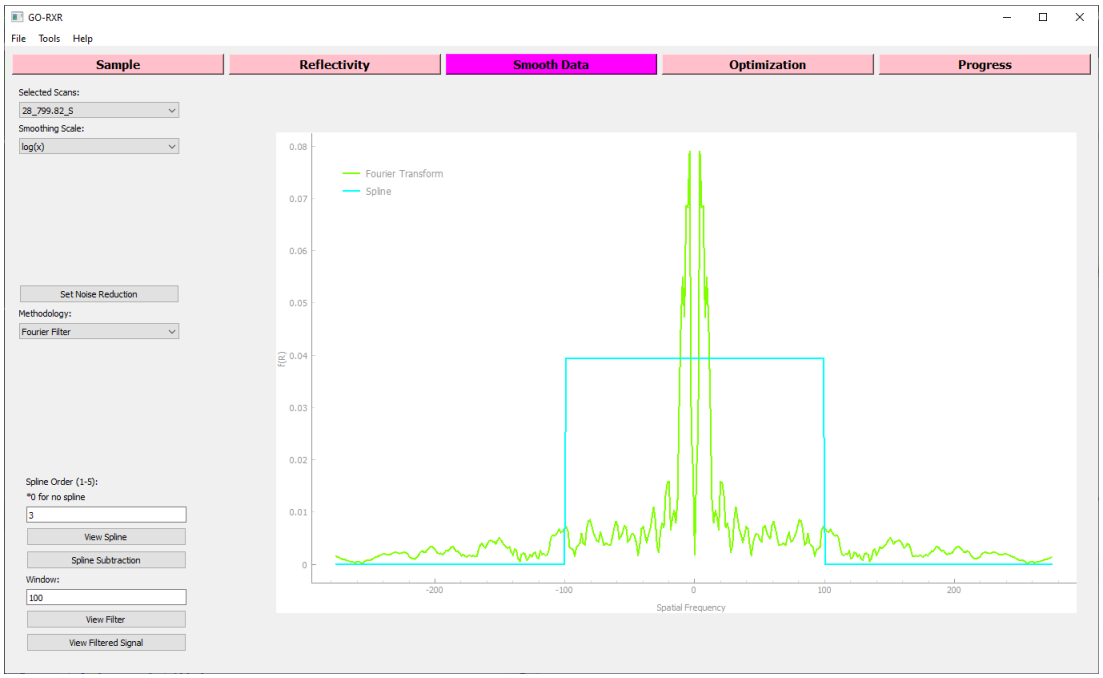


Figure 5.13: Figure 5.13: Filter Window

Once satisfied with the window the user can then view the filtered signal by clicking the **View Filtered Signal** button. It is recommended to use the Fourier Filter method on really noisy signals and the spline interpolation options for less noisy data.

## 5.10 Optimization Workspace

The optimization workspace is used to set up the optimization parameters. Upon navigating to the optimization workspace a window similar to that in figure 5.14 will appear.

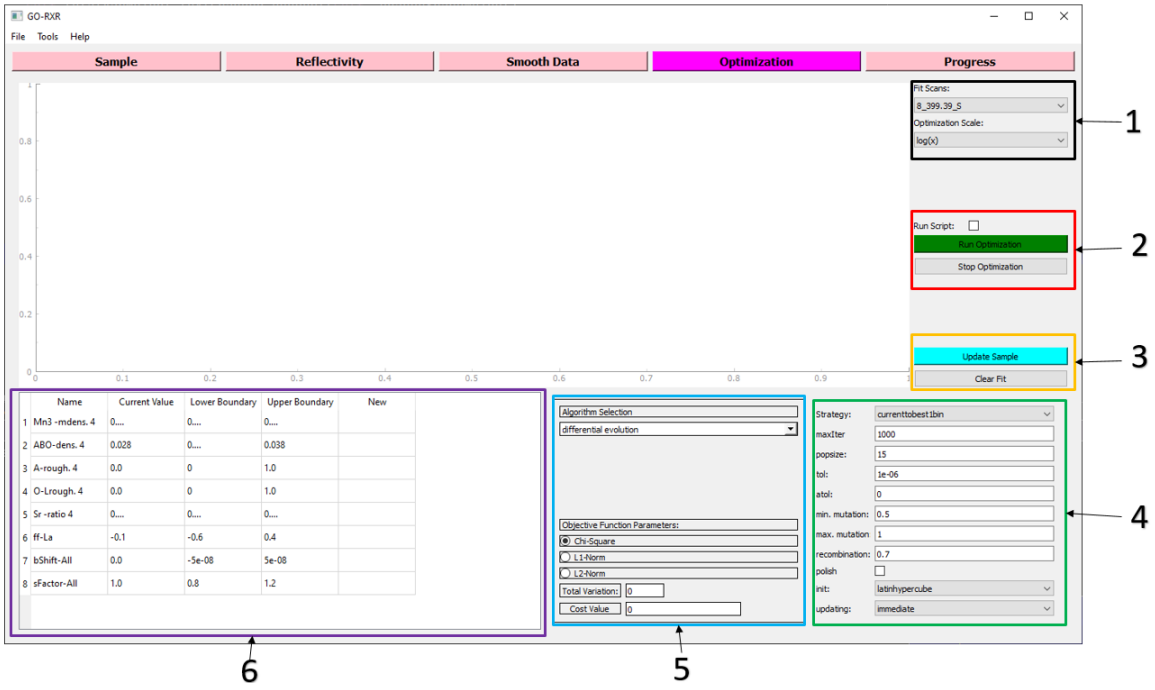


Figure 5.14: Optimization Workspace

There are 6 main sections in the optimization workspace:

1. **Scans** - enclosed by the black square
2. **Start Optimization** - enclosed by the red square
3. **Update Fit** - enclosed by the orange square
4. **Algorithm Parameters** - enclosed by the green square
5. **Optimization Parameters** - enclosed by the blue square
6. **Parameter Boundaries** - enclosed by the purple square

The **Scans** section has two main option boxes and is labeled **Fit Scans** and **Optimization Scale**. The **Fit Scans** determines which scan will be viewed in the plotting area. In the plotting area, both the simulations for the model before and after the data fit will be included in the plotting space. The **Optimization Scale** determines which transformation will be applied in the cost function. Similar to the smooth data workspace, the possible options are $R$, $log_{10}(R)$, $ln(R)$, and $R(qz) * qz^4$.It is required that the asymmetry scans use the $R$ transformation as they can have negative values. It is suggested that the $log_{10}(R)$ and $ln(R)$ are used for the reflectivity scans and $R$ is used for the energy scans.

The **Start Optimization** section is used to start and stop the data fit. The **Run Script**

check box determines whether to use (checked box) or not (unchecked box) the most recently saved script in the data fit. It should be noted that if there exists an error in the script the data fit will run without using the script. The **Run Optimization** button is used to start a data fit. When a data fit is currently not being run the **Run Optimization** button will be green in color. Once a data fit is in progress the button will turn red. This button can be clicked during a data fit and will not start a new data fit while the current one is running. The user must wait until the current data fit process fit has stopped before starting a new process. The **Stop Optimization** button allows the user to stop the current data fit process with the most recent optimized parameters. Note that pressing the **Stop Optimization** will not immediately terminate the data fit process, but can take some time. This is because the software must wait for an iteration of the data fit algorithm to finish before the process can be terminated manually. Unlike the global optimizers, the local optimizers must complete and cannot be manually terminated by pressing **Stop Optimization**.

The **Update Fit** section contains the **Update Sample** and **Clear Fit** buttons. The **Update Sample** button is clicked when a data fit has yielded desirable results. This will apply the optimized parameters resulting from the data fit to the sample workspace. This will not automatically save the optimized parameters in the GO-RXR data file, so this must be done manually each time. The **Clear Fit** button will remove all the fitting parameters. This makes it a lot easier when you would like to start a new data fit with completely new parameters without having to unfit all the parameters in the sample workspace and reflectivity workspace. Note that this will not remove the scans from the fit and they will need to be removed in the reflectivity workspace.

The **Optimization Workspace** is used to select which data fitting algorithm to use and to customize the cost function. The **Algorithm Selection** option box contains all the different possible algorithms that can be used. It is recommended to use the differential evolution algorithm for most cases. The only other algorithm I would recommend is the least squares when fitting the energy shift. The **Objective Function Parameters** contains three checkboxes: chi-square, L1-norm, and L2-norm. These check boxes determine the regularization that will be applied to the experimental and simulated data difference. The **Total Variation** field determines the weighting that will be applied to the total variation in the cost function. A value of 0 should be used if the total variation is included in the cost function. From much testing, it is recommended to use a value between 500-1500 for the weight. The cost function value (neglecting the total variation) for the selected scan and regularization chosen will be displayed upon pressing the **Cost Value** button. The cost function value will include any weighting applied to the scan.

The **Algorithm Parameters** section allows the user to customize the different algorithms of the selected data fitting algorithm. I would recommend using the default parameters. The definition of the different algorithm parameters can be found in the SciPy documentation [2].

The **Parameter Boundaries** section is used to set the lower and upper boundaries of the values that the fitting parameters can be changed to in the data fit. This section also displays the current parameter value, and when a data fit has finished, the parameter value from the data fit. It should be noted that only the lower boundary and upper boundary columns can be edited by the user. In the name columns, it should be noticed that there is a new naming

convention. Any name that begins with a *ff-* (non-magnetic) or *ffm-* (magnetic) signals that the energy shift of the form factor symbol that follows is to be fit. Names that begin with *bShift* or *sFactor* signal that the background shift or scaling factor of a scan is to be fit. If *all* follows either bShift or sFactor then that means that the background shift and scaling factor will be applied to all the scans that are being fit. Otherwise, the name of the scan will appear. The other parameters are the sample model parameters and follow the generic naming convention of *symbol-parameter. layer*. If in the *symbol* position the symbol there are two possible options. If a single element/variation symbol appears, then that means that the fit is to be done in element mode. If the symbol of all the elements in the layer appears in the *symbol* position then this means that the fit is to be done in compound mode. The *layer* position displays the layer indices that the parameter is from (note that the indices start at 0 for the substrate). The *parameter* position contains the key or short form of the parameter that is being fit. The different parameter possibilities are summarized in Table 1.

Table 1: Parameter Keys

| **Key** | **Parameter** |
|---|---|
| th | Thickness |
| rough | Roughness |
| Lrough | Linked Roughness |
| dens | Density |
| mdens | Magnetic Density |
| ratio | Element Variation Ratio |

## 5.11 Progress Workspace

The progress workspace is used to display the progress of the current running data fit. Upon navigating to the progress workspace a window similar to figure 5.11.


Figure 5.15: Progress Workspace

The **Total Cost** button will display the cost function value (y-axis) with respect to the number of iterations (x-axis) that have occurred. Note that this will update each iteration. The **Norm** will display the regularization (chi-square, L1-norm, L2-norm) portion of the total cost function with respect to the number of iterations. The **Variation** button will display the total variation portion of the cost function with respect to the number of iterations. It should be noted that not only does it display the cost function for the sum of all the scans, but the cost function for each individual scan. This provides valuable information about which scan is being impacted the most in the data fit. The **Parameters** button will display the evolution of the fitting parameters with respect to the number of iterations. The **Density Profile** button will display the density profile of the current iteration in the data fit. The **Selected Scans** option box contains the experimental data that is being fit. When a scan is selected from this option box the simulation of the current fitting iteration will be displayed along with experimental data. The **All Scans** option box contains all the experimental data and allows the user to view the simulation of the current fitting iteration for any experimental data. This has been a very useful tool to use to determine when to stop a data fit. It has also been very useful in determining whether the data fit is converging how we would like it to. This does make the data analysis process more involving, but it does provide valuable insight into what needs to be done to improve the fits. A possible feature that could be included in the progress workspace is the ability to view a sensitivity analysis of the data fit. This would be an interesting feature that would help determine which parameters impact which scans and to what degree.

# 6 File Format

## 6.1 Data File

The easiest way to load a data file is by loading it in a ReMagX file with the '.all' extension. However, if you are unfamiliar with ReMagX then it would be easier to create a data file using HDF5. To do this I would suggest using the *h5py* library found in python [1]. The architecture of the HDF5 file is shown in figure 6.1.

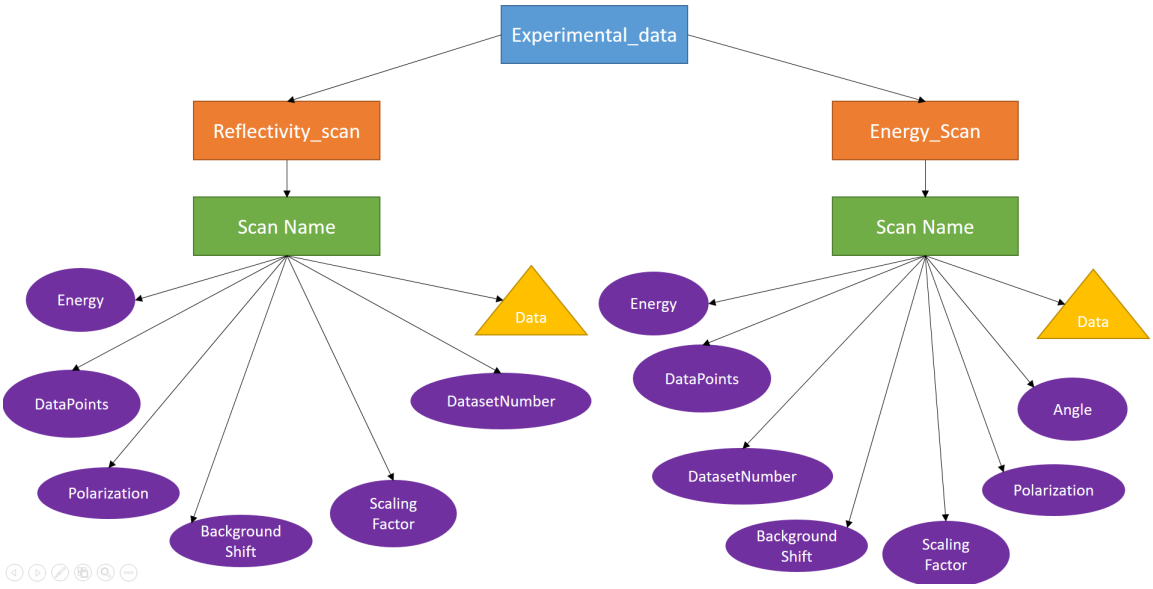


Figure 6.1: Data File Flowchart

The squares in the flowchart demonstrate a group. The blue square is the main group, while the orange square is a subgroup of the blue square, and the green squares are subgroups of the orange squares. It should be noted that the group names must be spelled the same way as they appear in the flowchart. The purple ovals are the attributes of the group and the yellow triangle is where the data is stored. Note that in the flow chart, there is only one green square but for more than one data scan a new group will need to be added for each scan with a unique name. The naming convention will be discussed in more detail in the **Naming Convention** section. Below you will see some sample code of how to create such a structure using h5py:

```
# reflectivity_scans and energy_scans are assumed to be dictionaries with all
# the appropriate information
reflectivity_scans = ... # reflectivity scans
energy_scans = ... #  # energy scans

f = h5py.File(fname, 'a')  # creates a h5py file capable of writing data

h = 4.135667696e-15  # Plank's constant eV*s
c = 2.99792458e8  # speed of light m/s
```

```
grp1 = f.create_group("Experimental_data")  # experimental data group

grpR = grp1.create_group("Reflectivity_Scan")  # reflectivity scan group
grpE = grp1.create_group("Energy_Scan")  # energy_scan group

# Assumes data_scans is a dictionary that has a similar structure to the data
#structure

for idx, name in enumerate(reflectivity_scans.keys()):
    scan = reflectivity_scans[name]  # retrieve scan
    qz = scan['Data']['qz'] # momentum transfer (list or numpy array)
    R = scan['Data']['R'] # reflectivity (list or numpy array)
    Theta = np.arcsin(qz / energy / (0.001013546247)) * 180 / np.pi
    dat = np.array([qz, Theta, R])  # data

    polarization = scan['Polarization']  # polarization
    energy = scan['Energy'] # energy in units of eV
    datasetpoints = len(qz)  # number of data points
    scanNumber = scan['Scan Number]

    new_name = str(ScanNumber) + "_" + str(np.round(energy,2)) + "_" + polarization

    m = np.shape(dat)  # shape of the data
    dset = grpR.create_dataset(name, m, data=dat, maxshape=(3, None), chunks=True)
    dset.attrs['Energy'] = float(energy)  # adding attribute to data
    dset.attrs['Background Shift'] = float(0)
    dset.attrs['Scaling Factor'] = float(1)
    dset.attrs['Polarization'] = str(polarization)
    dset.attrs['DataPoints'] = int(datasetpoints)
    dset.attrs['DatasetNumber'] = int(scanNumber)

for idx, name in enumerate(energy_scans.keys()):
    scan = energy_scans[name]  # retrieve scan
    qz = scan['Data']['qz'] # momentum transfer (list or numpy array)
    R = scan['Data']['R'] # reflectivity (list or numpy array)
    E = scan['Data']['E]  # energy (list or numpy array)
    Theta = np.arcsin(qz / energy / (0.001013546247)) * 180 / np.pi
    Angle = scan['Angle']  # grazing angle (this value is used in simulations)
    dat = np.array([qz, Theta, R, E])  # data

    polarization = scan['Polarization']  # polarization
    energy = scan['Energy'] # lowest energy
    datasetpoints = len(qz)  # number of data points
    scanNumber = scan['Scan Number]

    new_name = str(ScanNumber) + "_E" + str(np.round(energy,2)) + "_Th" +
               str(np.round(angle,2)) + "_" + polarization
```

```
    m = np.shape(dat)  # shape of the data
    dset = grpR.create_dataset(name, m, data=dat, maxshape=(4, None), chunks=True)
    dset.attrs['Energy'] = float(energy)  # adding attribute to data
    dset.attrs['Angle'] = float(Angle)
    dset.attrs['Background Shift'] = float(0)
    dset.attrs['Scaling Factor'] = float(1)
    dset.attrs['Polarization'] = str(polarization)
    dset.attrs['DataPoints'] = int(datasetpoints)
    dset.attrs['DatasetNumber'] = int(scanNumber)

f.close() # close the file
```

Note that by replacing the blue square with the name *Simulated_data* a simulation group can be created instead of the experimental data group. If there is any further problems consult the *data_structure.py* python file found at GitHub repository `https://github.com/lucaskorol21/GO-RXR`.

## 6.2 Attributes

The attributes of the project workspace file are:

**Energy -** This is the energy of the incoming x-ray energy and should be entered as a float type.

**Angle -** This is the grazing angle of the incoming x-rays and should be entered as a float type.

**Background Shift -** This is the background shift and should be as a float type with value zero

**Scaling Factor -** This is the scaling shift and should be a float type with value 1

**Polarization -** For s-polarized light the polarization should be entered as 'S'. For p-polarized light the polarization is to be entered as 'P'. For linear asymmetry the polarization is to be entered as 'AL'. For right circular polarization the polarization is entered as 'RC'. For left circular polarization the polarization is entered as 'LC". For circular asymmetry the polarization is entered as 'AC'.

**DataPoints** - This is the number of data points and is entered as an interger type.

**DatasetNumber -** This is the scan number and is entered as an integer type.

It should be noted that the linear asymmetry is calculated as:

$$R = (R_S - R_P)/(R_S + R_P) \tag{2}$$

where the subscript $S$ and $P$ denote s-polarized and p-polarized light, respectively.

Similarly, the circular asymmetry is calculated as:

$$R = (R_L - R_R)/(R_L + R_R) \tag{3}$$

where the subscript $S$ and $P$ denote left circular and right circular polarizations, respectively.

## 6.3 Naming Convention

### Reflectivity Scan

For reflectivity scans the naming convention is *scanNumber_Energy_Polarization*. Let's assume we have a scan with the input photon energy of 500 eV, a $\sigma$-polarization, and is the 3rd scan. Then the name of the scan will be:

```
name = 3_500.00_S
```

However, there is a slight difference between a linear and circular asymmetry scan. Let's assume we use the left and right circular scans with the scan numbers 8 and 9, respectively, at 640.2eV. Then the name of this particular scan would be:

```
name = 8-9_640.2_AC
```

### Energy Scans

For reflectivity scans the naming convention is *scanNumber_Energy_Theta_Polarization*. Let's assume we have an energy scan between 450-480eV, with an $\sigma$-polarization, grazing angle of 5.0 degrees, and is the $15^{th}$ scan. Then the name of the scan will be:

```
name = 15_E450.00_Th5.0_S
```

Note that the energy has the prefix *E* and the energy has the prefix *Th* and is required. However, there is a slight difference between a linear and circular asymmetry scan. Let's assume we are using the same energy and grazing angle, but we are calculating the circular asymmetry for the left circular scan with scan number 18 and the right circular scan with scan number 19. Then the resulting name will be

```
name = 18-19_E450.00_Th5.0_AC
```

The naming convention is extremely important where the information included in the name is used by the software.

## 6.4 Form Factor File

The form factor must be found with the file extension '.ff' or '.ffm'. The file extension of '.ff' tell the software that the file is a non-magnetic form factor while the extension '.ffm' tells the software it is a magnetic form factor. Both of these files are saved as a text file. In the text file, the first column is the energy. The second column is the real component of the form factor. The third column is the imaginary component of the form factor. Note that each column must be separated by a space. An example of a form factor file is shown in figure 6.2.

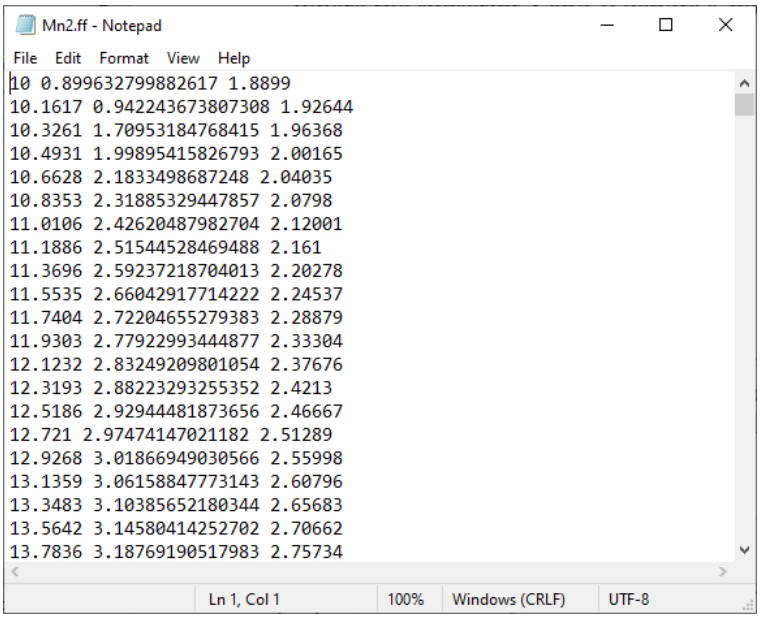


Figure 6.2: Figure 6.2: Form Factor File

# References

[1] Andrew Collette. *HDF5 for Python*. 2014. URL: https://docs.h5py.org/en/stable/ (visited on 04/12/2023).

[2] SciPY. *Optimization and root finding*. 2023. URL: https://docs.scipy.org/doc/scipy/reference/optimize.html (visited on 04/12/2023).