GO-RXR: User Guide

Version: 2023-0.3

Lucas Korol

September 1, 2023

GO-RXR

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 multi-layered 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. 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.

The user manual will explain how to navigate the GUI of GO-RXR. The user manual will also explain the script feature and the file formats used by GO-RXR.

2 Project Workspace

The software can be run in Python or from the stand-alone executable file. The window in Figure 2.1 will appear after running the software.

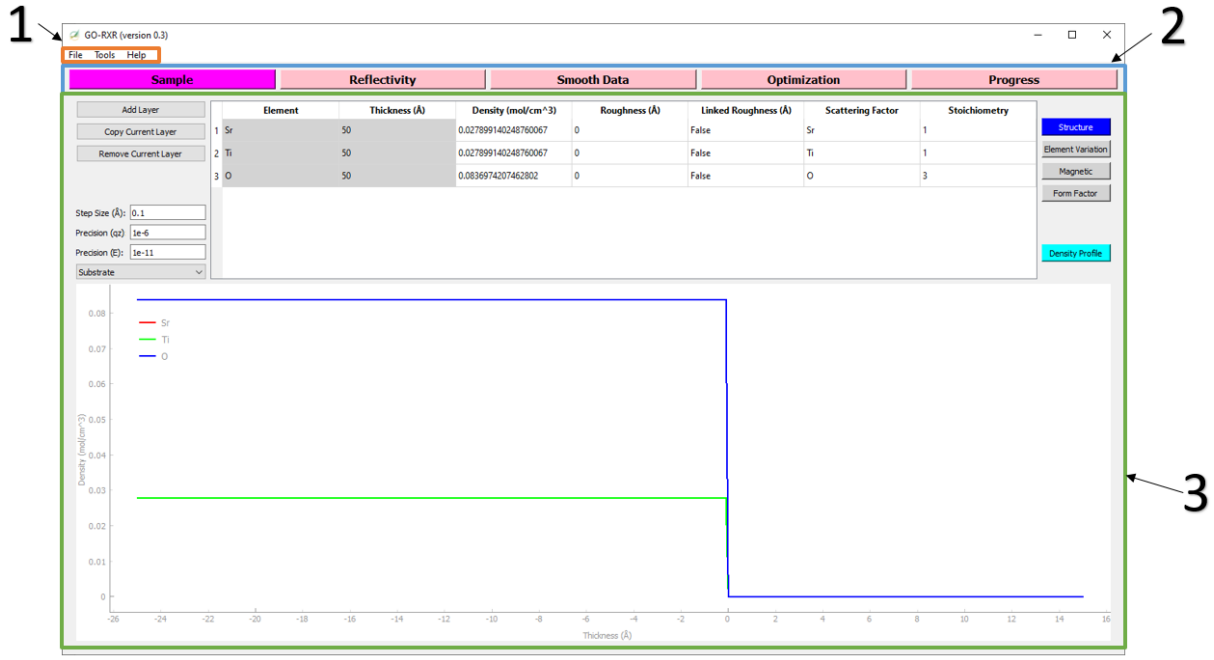

Figure 2.1: Start Screen

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. The options include transformations and adding grids. In particular, there is an option to take the derivative of the data. The derivative of the data is useful when checking the quality of the noise removal in the data smoothing feature. The plotting options are available in all plotting areas.

3 Toolbar

3.1 File

The Hierarchical Data Format version 5 (HDF5) is the file type chosen to save the workspace. An HDF5 file requires the '.h5' file extension. Not all files used in GO-RXR are HDF5 files (e.g.form factor files) and the files that do not belong to the HDF5 file type are mentioned in this user manual. The file tab is used to load and save data from 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

The **New Workspace** option opens up a new workspace. The new workspace is loaded in with default parameters, including a predefined substrate layer with the chemical formula of $SrTiO_3$. The default substrate layer can be removed by pressing the **Remove Current Layer** button. After removing the predefined substrate, a new substrate can be added by selecting the **Add Layer** button as shown in Figure 2.1. Once a new workspace is initialized, the previously loaded in experimental dataset is cleared from the GUI.

The **Load Workspace** option is used to load a GO-RXR workspace with the '.h5' file extension from a desired directory. The software scans the current working directory for form factor files. Any form factors found in the directory are included in the form factor database and used in the reflectivity calculations. Any form factors added to the project directory after the workspace file has been loaded will not be included into the form factor database. The workspace must be saved and reloaded to incorporate the new form factor files.

The **Load Sample** option loads in the model information. Loading in the model information is ideal when working on multiple samples with similar configurations, but possess slightly different characteristics. Loading in the model of the sample allows for an easy comparison between the various models versions of the models.

The **Save Workspace** option 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 **Save Sample** option saves only the model of the sample to the current project file. The **Save Simulations** option saves the simulations for the current model to the current project file. Once this option is selected a pop-up will appear. Click the **Start Saving** button and the software will begin calculating the simulated data. The progress bar indicates the progress of the calculations. The simulations will only be saved once all the calculations are finished. The automatic termination of the pop-up window signals the simulations are saved to the project file. If the pop-up is excited before the automatic termination of the window, then the simulations will not be saved.

The **Import Dataset** option 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 Section 6 of the user manual.

The **Load ReMagX** option allows the user to import the data from a ReMagX file with the '.all' extension. Only the experimental data is loaded from the ReMagX file into the project workspace. The model of the sample defined in ReMagX is not loaded in the project workspace because the model definitions in ReMagX differ from the model definition in GO-RXR.

The **Save Summary** option allows the user to save the workspace information to a text file. The file gives a summary of the model and the fitting parameters. The **Exit** option is used to exit the program. The program can also be exited by pressing the 'x' icon at the top right corner of the GUI.

3.2 Tools

The **tools** tab contains tools that can be used throughout the data analysis. The tools added are **Script** and **Form Factor**. The **Script** option is used to define internal function that can be used to constrain the parameters of the model during the data fitting. The **Form Factor** option opens a window that is used to view the form factors used in the reflectivity calculations.

Script

The script option allows for more flexibility in the data fitting. Figure 3.1 will appear after selecting the **Script** option.

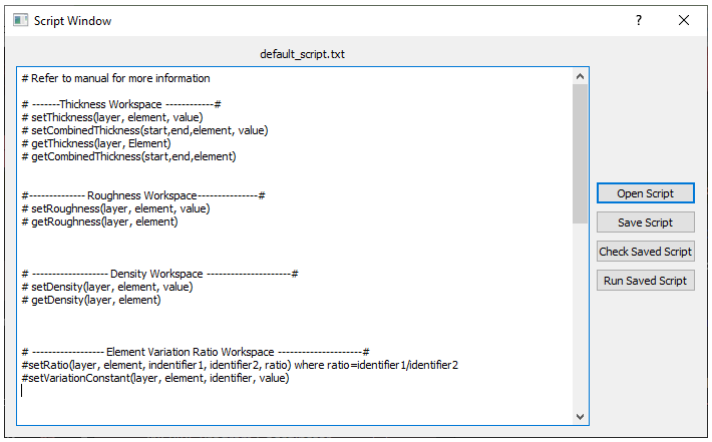

Figure 3.1: Script Window

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, 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. Custom scripts can also be used by loading the script in by selecting the **Open Script** button. From this point forward, anytime you press **Save Script** the most recently loaded in script is used. After exiting the project workspace, the software will always refer back to the default script.

The script features has a limited number of internal functions. Simple arithmetic and loops cannot be compiled in the script. In future versions, simple arithmetic and loops 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 **Check Saved Script** will compile the script and report any errors. If there are any errors in the script, an error message will report the line and error type of the first error detected. The error checker can only report a single error at a time. The **Run Saved Script** button runs the most recently saved script and applies all defined internal functions to the sample model. When the script is being used in a data fit, the internal functions of the script will override any fitting parameter. Any line of code can be commented out by using the # character.

As previously mentioned, the script can only use predefined internal functions. The internal script functions 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 in a Set function. Note that the value placeholder can only be used as an input for the Set function. The detailed descriptions of each function will indicate the parameters that can use the placeholder variables. Again, future updates will consider making the script more flexible. The Set function is defined as

```
setFunction(parameters)
```

where *parameters* are the input parameters of the Set function. It should also be made clear that a Get function cannot be input as a parameter to an internal function

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

where the *getDensity* function retrieves the density of manganese in the first layer in the model. The correct way to define a Get function using this same example is

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

where *rho* is the placeholder assigned to the density of manganese in the first layer of the model.

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 reflectivity calculations. The window shown in Figure 3.2 will appear after selecting the **Form Factor** option.

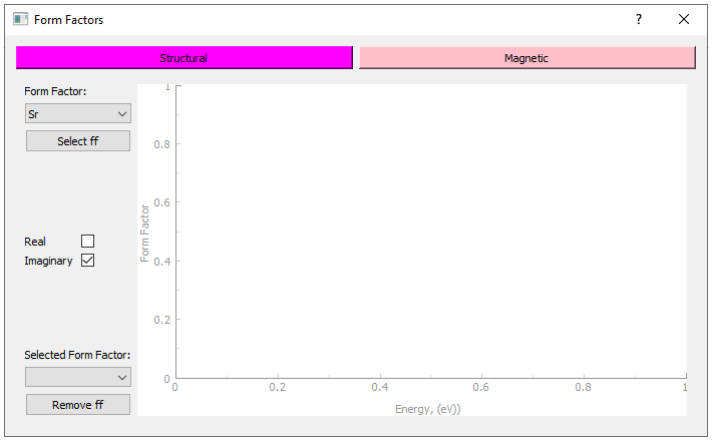

Figure 3.2: Form Factor Start Screen

The buttons labeled **Structural** and **Magnetic** are used to toggle between the non-magnetic and magnetic form factors, respectively. At the top left corner of the window there is an option box that contains all the form factors in the project directory. The form factors are not plotted after clicking the form factor in the option box, but the button label **Select ff** must be clicked to view the selected form factor. Once **Select ff** is clicked, the real and imaginary components of the form factor as a function of energy will be plotted in the plotting area. Multiple form factors can be selected and viewed at the same time by selecting multiple form factors using the **Select ff** button. This is very useful tool when dealing with experimental datasets that are difficult to recreate in the reflectivity calculation. The form factors provide valuable information regarding the sensitivity of each element at a select energy. Figure 3.3 illustrates the plotting of the real and imaginary component of the titanium atomic form factor.

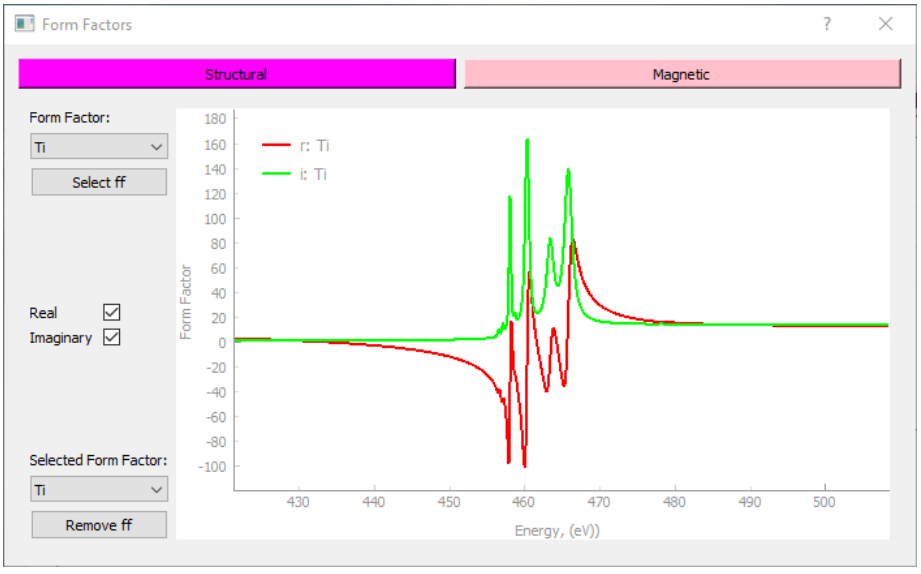

Figure 3.3: Real and imaginary component of the titanium atomic form factor.

Once a form factor has been selected it will appear in the **Selected form factor** option box located at the bottom left corner of the window. The **Selected form factor** 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 toggle between viewing the real and imaginary components of the atomic form factor.

Help

The help tab contains the options **License** and **About**. The **License** option gives the appropriate information about the licensing of GO-RXR. The **About** option opens up the user manual.

4 Workspace Navigator

The workspace navigator contains five buttons that are used to navigate the various workspaces in GO-RXR. The five buttons are **Sample**, **Reflectivity**, **Smooth Data**, **Optimization**, and **Progress**. The **Sample** button navigates to the sample workspace and is used to construct the model of the sample. The **Reflectivity** button navigates to the reflectivity workspace and 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 various scans. The **Smooth Data** button navigates to the data smoothing workspace and provides the user the ability to smooth the data. The **Optimization** button navigates to the optimization workspace and used to select the data fitting algorithm, parameter boundaries, and to start a data fit process. The **Progress** button navigates to the progress workspace that is used to view the progress of the currently running data fitting process. A more detailed explanation of each workspace will be provided in Section 5 of the user manual.

5 Workspace

5.1 Model Definition

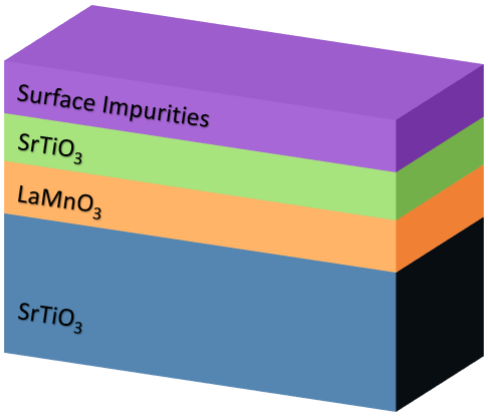

Figure 5.1: Model Definition

In the development of the software, the idea was to construct the model of the material using a layer-by-layer approach, where each layer is defined as compounds with a known stoichiometry. This type of model is called the slab model and is visually illustrated in Figure 5.1. GO-RXR was also developed considering that elements can have a linked roughness at the interface of each layer. The linked roughness 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. Dummy variables are used by filling the missing elements in the chemical formula with a predefined variable, where their atomic mass and form factors are zero. The possible dummy variables are A, D, E, G, J, L, M, Q, R, T, X, Z. An example will be provided in Section 5.6 of the user manual.

5.2 Structure Workspace

The structure workspace can be selected by pressing the **Sample** button at the top of the window. A table will appear after pressing the **Sample** button with the headers including element, thickness, density, roughness, linked roughness, scattering factor, and stoichiometry. Figure 5.2 displays an example of the table for the predefined substrate.

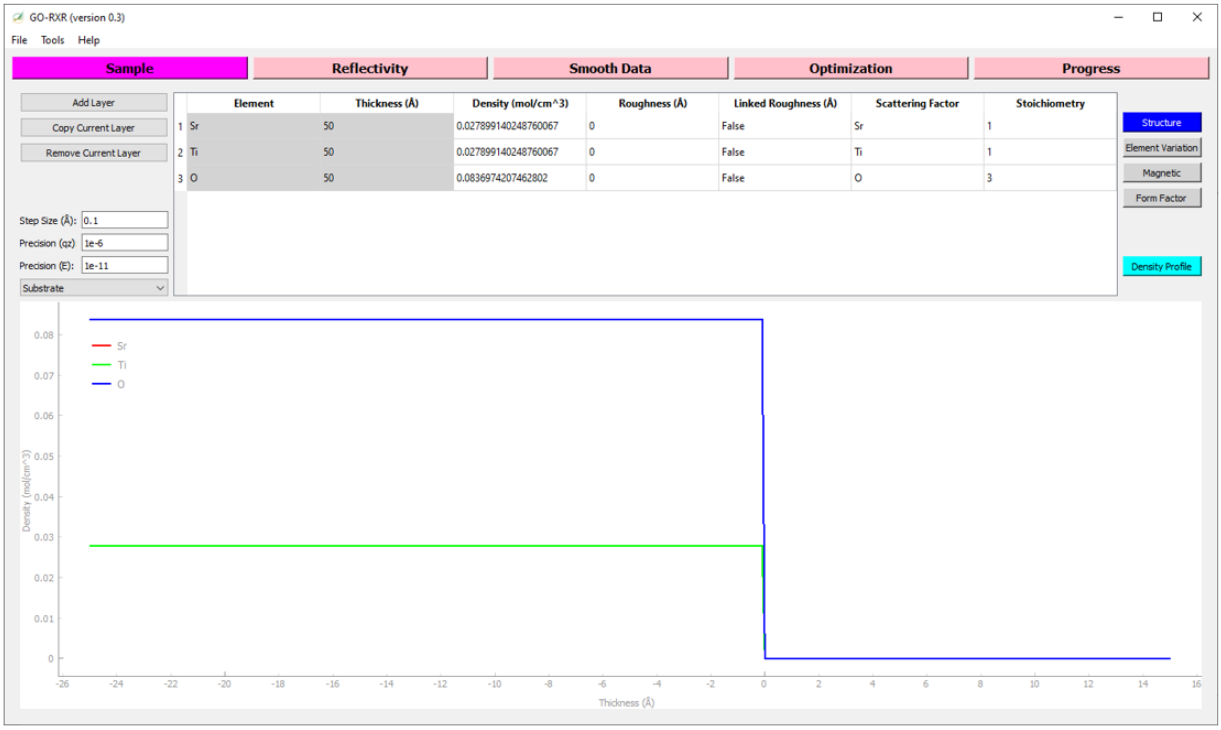

Figure 5.2: Structure Workspace

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 that is labeled **Substrate** in Figure 5.2. The layer option box is used to navigate between the different layers in the model. The element symbol cannot be changed in the structure workspace after it has been defined. To remove an element, all layers that contain the element must be removed. All the grey columns denote parameters that cannot be changed or varied in the data fitting.

The thickness header is in units of angstroms (Å), 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. It should also be noted that the element and thickness column are grey for the substrate. It is important to clarify that the thickness column will only be grey for the substrate layer.

The density header refers to the density of the element in units of $\mathrm{mol/cm^3}$. The density of each element can be set to any desired value independently of the predefined stoichiometry. The stoichiometry will only be used in the data fitting. This will be discussed further in Section 5.10 of the user manual.

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

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

where σ correspond to the width of the error function. The values input into the roughness column are the σ values.

The linked roughness is used to link the roughness of elements together at an interface. If the linked roughness is set to *False* or 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 corresponds to the atomic form factors. The symbol or name provided to the scattering factor column indicates the atomic form factor to use in the reflectivity calculations. A database of scattering factors is provided for all the elements in the periodic table. 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. The format of a form factor file will be discussed further in Section 6.4 of the user manual. If an element is defined as an element variation, then a list of form factors will appear. In this case, the form factors can only be changed in the **Element Variation** workspace.

The stoichiometry column determines the stoichiometric relation between the different elements in the layer. The stoichiometry 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 of an element (e.g.oxidation states). The element variation table is illustrated 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 that the element has previously been defined. The option box on the top left corner of the workspace contains all the different elements present in the currently selected layer. The element that an element variation is supposed to be applied to needs to be selected in this option box.

The name, ratio, and form factor need to be considered when adding an element variation. The name can be any string and 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. The sum of all the ratios in the ratio column must equal to one. The form factor works the same way as the form factors in the sample workspace.

5.4 Magnetic Workspace

The magnetic workspace is used to create the magnetic density profile. The magnetic workspace is displayed in Figure 5.4. In the magnetic workspace, a row will be given for

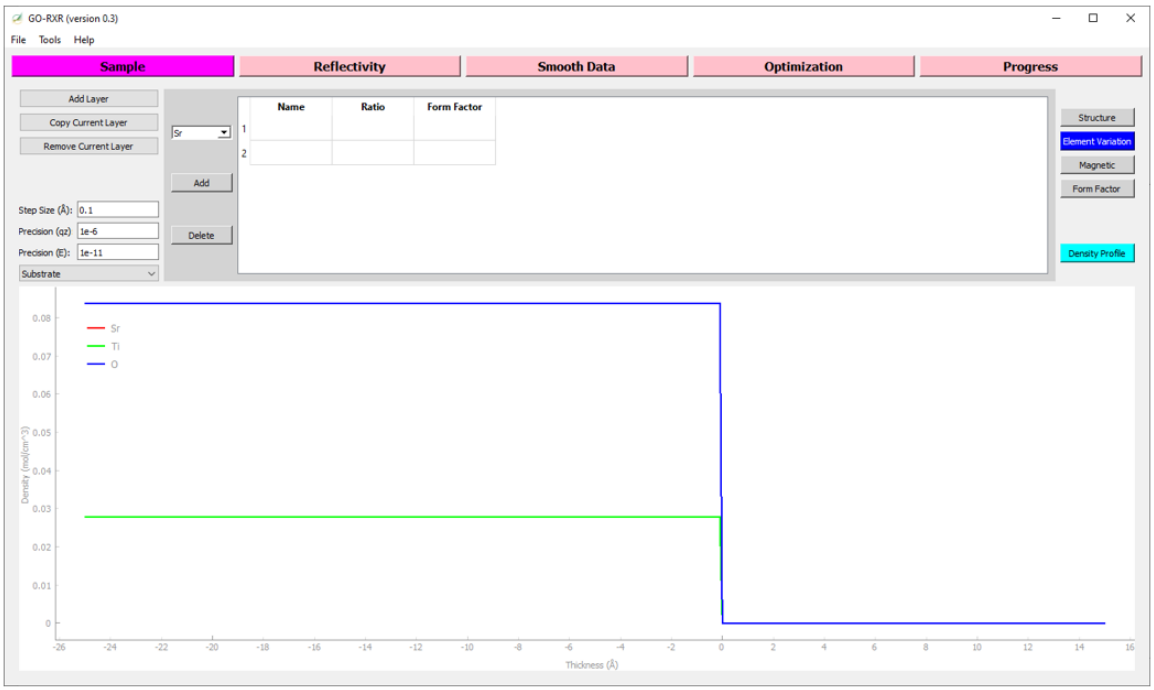

Figure 5.3: Element Variation Workspace

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 is expressed in terms of the units $\mathrm{mol/cm^3}$. 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.

At the top right corner there is an option to include the magnetization directions. There are only three options provided and include the x-direction, y-direction, and z-direction (default). Future versions of GO-RXR will include an arbitrary magnetization direction defined by ϕ and γ. Currently, when the magnetization direction is changed in one layer this will change the magnetization direction in all the layers. In future versions, it is planned 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 the magnetic structure is assumed to be related to the crystal structure of the material.

5.5 Form Factors Workspace

The form factor workspace is used to add energy shifts or to scale the atomic 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. The *E* row is the energy shift and *Scale* is the scaling factor. The energy shift can be entered as a positive or negative number, where a positive value will shift the form factor to higher energies and a negative value will shift to lower energies. The scaling factor is typically used for the magnetic form factors to ensure that the element and magnetic densities are of comparable scales. Therefore, the scaling factor is mainly used for visualization purposed of the structure of the material.

Figure 5.4: Magnetic Workspace

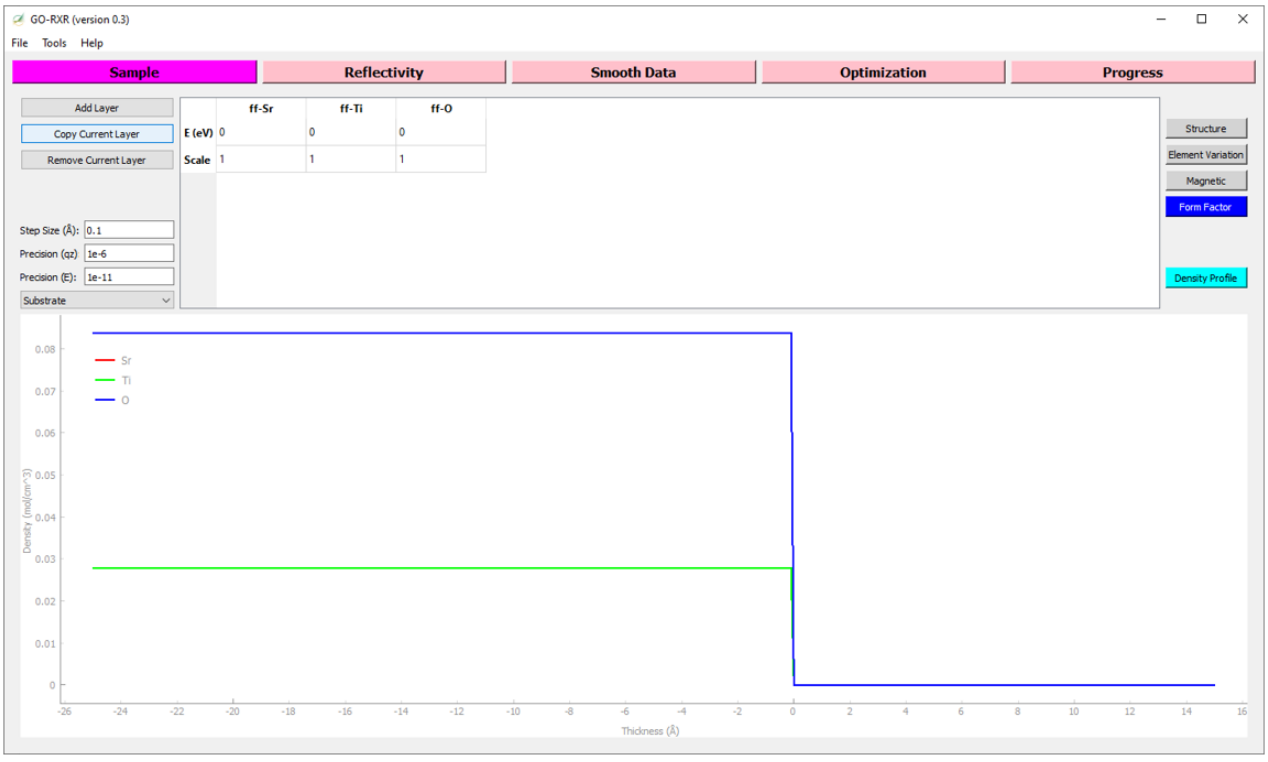

Figure 5.5: Form Factor Workspace

5.6 Adding and Removing Layers

The window in Figure 5.6 will appear after clicking the **add layer** button. The window required that the layer information is entered as a compound.

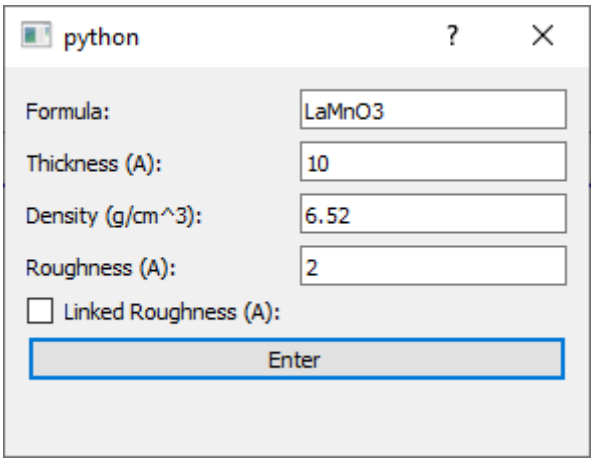

Figure 5.6: Adding Layer Window

The **Formula** text box is where the chemical formula of the layer is entered. In the case where a dummy variable is required, the user enters the chemical formula as normal but adds 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 where the thickness of the layer is entered in units of angstroms. The **Density** is the density of the compound as is entered in units of g/cm^3. 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 **Linked Roughness** field can be toggled on or off. If the **Linked Roughness** is toggled on then the user can input a value for the linked roughness. If the **Linked Roughness** is toggled off check box then the linked roughness is set to False. The **Enter** button should only be clicked once all the parameters are properly set. Clicking **Enter** will create a new layer in the structure workspace. The newly created layer is entered above the currently selected layer.

The **Copy Current Layer** button takes currently selected layer and creates 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) is removed.

Surface Layer

The surface layer is required to have the same number of elements defined as all the other layers in the model. However, in most cases, the surface impurities are made up of oxygen and carbon. The surface layer can be defined two ways by using dummy variables or multiple carbon (or oxygen) 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, each layer in the material has four elements defined. A possible way to define the surface layer is with the chemical formula of 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. A possible way of defining the chemical formula of the surface layer is CCCO. 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, the symbols of C1, C2, C3, O will appear in the name column.

5.7 Fitting Parameters

The parameters that can be varied in the data fitting include the thickness, density, roughness, linked roughness, energy shift, element variation ratio and 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 types 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. If a compound fit is selected for the density, then the density for all the elements will change according to their 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 only fits the parameter of the selected cell and will treat all parameters independently. An element fit will color the selected cell with a green, whereas a compound fit will color the entire column purple. An example of an element and compound fit is shown in Figure 5.7.

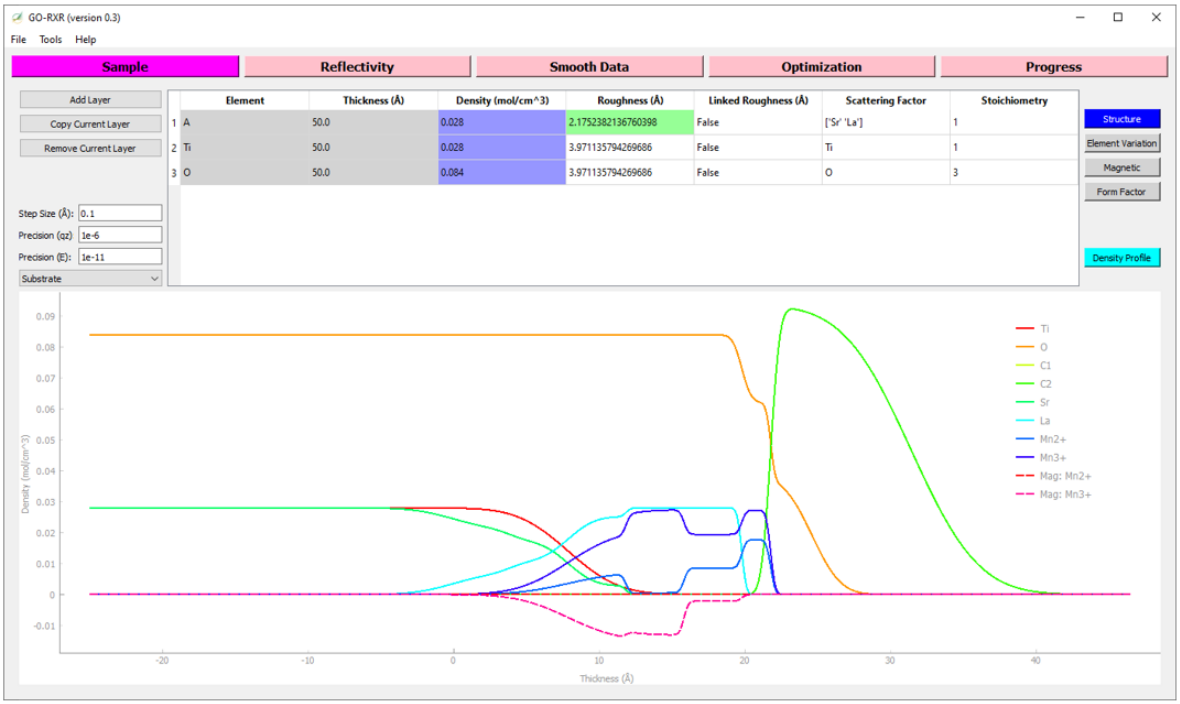

Figure 5.7: Fitting Parameters

For the element variation ratio, only one ratio can be selected to fit in each layer. All the other element ratios will be changed based on predefined function, or an internal function defined in the script. The energy shift can be fitted 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. In some cases, it is important to tie the energy shift of multiple form factors. The energy shifts are most commonly tied together when dealing with the oxidation states of an element. The energy shifts of several form factors can be linked by using the internal functions *getEshift* and *setEshift*. The magnetic density can be selected in the magnetic workspace. The software also allows the user to highlight multiple cells and apply a compound fit, element fit, or unfit to all the highlighted 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.

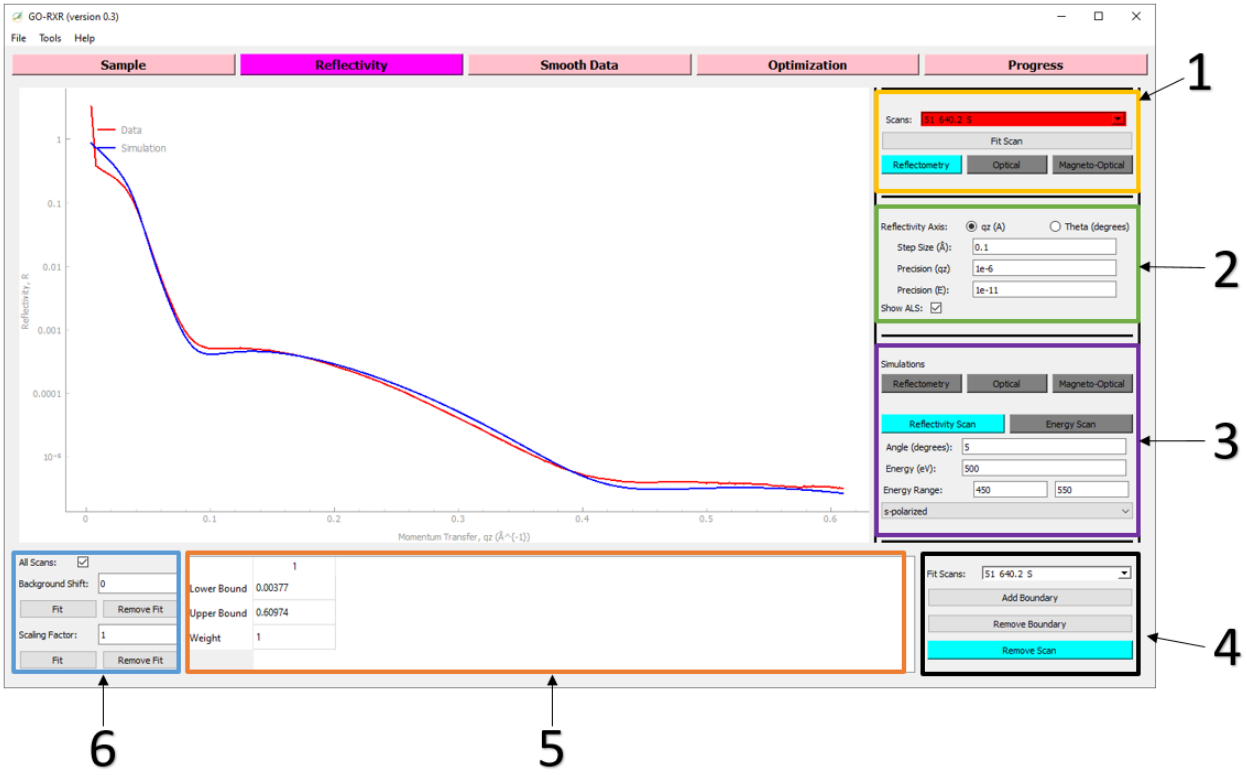

Figure 5.8: Reflectivity Workspace

The reflectivity workspace can be separated into six 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. The option box in the data scans section contain the names of the experimental data in the dataset. The option box will remain empty if a data file has not been loaded into the program. Selecting a scan from the option box will result in the selected scan being plotted in the plotting area. The **Fit Scan** button is pressed to select an experimental scan to use in the data fitting. The scan name will then appear in the **Fit Scans** options box in the located at the bottom right corner of the window. Multiple scans can be selected for a data fit. The three buttons labeled **Reflectometry**, **Optical**, and **Magneto-Optical** in the data scans section are used to view the reflectivity, optical profile, and magneto-optical profile of the model. Each of these options use the energy defined in the name of the selected scan to view.

The axis toggle section is used to toggle between momentum transfer and grazing angle for the theta/two-theta reflectivity scans. The axis toggle section is only implemented on the theta/two-theta reflectivity scans and ignored when plotting the energy scans. The momentum transfer option is selected by clicking the **qz** option and the grazing angle is selected by clicking the **Theta** option.

The step size field provides the user the ability to change the thickness step size in the density profile (this field is also found in the sample workspace). The precision fields are used in the adaptive layer segmentation implementation. The precision field labeled with qz denotes the precision value used in the theta/two-theta reflectivity calculations, whereas the precision field labeled E refers to the precision value for the energy scan calculations. The checkbox labeled **Show ALS** is used to plot the slices taken in the adaptive layer segmentation. The slices are plotted with the optical and magneto-optical profiles.

The simulation section is used to calculate the reflectivity from the defined model. The reflectivity can be calculated for a constant angle with a defined energy range or a constant energy for a defined grazing angle range. The simulation section also allows for the selection of the polarization of the light. Data does not need to be loaded in order to simulate the reflectivity, but a model does need to be defined in the sample workspace. The **Reflectometry**, **Optical**, and **Magneto-Optical** buttons will display the simulated reflectivity, optical profile, and magneto-optical profile, respectively, in the plotting area. The **Reflectivity Scan** button will display the theta/two-theta reflectivity scans with the defined constant energy and grazing angle range. The **Energy Scan** option will display the energy scans with the defined constant grazing angle and energy range. The **Angle** field is where the constant grazing angle is entered. The angle entered is used in the energy scan calculations and must be found between 0 and 90 degrees. The **Energy** field is the constant energy used in the theta/two-theta reflectivity calculations. The values for the energy can

be any value larger than 0, not inclusive. The Energy Range field denotes the energy interval of the energy scan to be displayed in the plotting area. The leftmost entry is the lower limit and 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. The polarization section also gives an option to display the linear and circular asymmetry.

The fit scans section is used to view and change the boundaries and weights of a selected scan. The option box in the fit scans section contains all the scans that have been selected to include in the data fitting. Selecting a scan in this option box will display the experimental data (along with the simulation) in the plotting area. The table in the boundaries and weight section will also change to values that correspond to the selected scan. The table is used to alter the boundaries of the selected scan and to apply the appropriate weights to the boundaries. The **Add Boundary** button adds an additional column to the boundary and weights section for the selected scan. The **Remove Boundary** button removes 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 theta/two-theta reflectivity scan are noisy, the user can create boundaries to exclude those regions in the data fitting process. In Figure 5.8, the theta/two-theta reflectivity scan has noisy data in the interval from 0.5 A^{-1} to 0.8 A^{-1}. To remove the noisy region from the data fitting process, the lower bound in the first column needs to be input as 0.00471 A^{-1} and the upper bound as 0.5 A^{-1}. The boundaries and weights section can also be used to provide a bias to certain intervals in the data. Instead of completely removing the noisy interval from the data scan, a weight can be included into the boundaries section that will limit the bias of the noisy interval. In this case, another boundary can be included in the table by clicking the **Add Boundary** button, creating two columns in the table. In the first column, the lower boundary can be set as 0.00471 A^{-1} and the upper boundary can be set with the value 0.5 A^{-1}. The weight in this boundary can be set to one to ensure no bias is applied to this region. The second column is used to define the noisy interval by setting the lower boundary to 0.5 A^{-1} and setting the upper boundary as 0.8 A^{-1}. To minimize the bias the noisy boundary has on the data fitting, this region can be giving a weight of 0.5. The applied boundaries and weights multiplies the cost function by 1 in the 0.00471 A^{-1} to 0.5 A^{-1} interval, whereas the cost function will be multiplied by 0.5 in the 0.5 A^{-1} to 0.8 A^{-1} interval. Multiplying the cost function by a reduced weight decreases the influence the noisy interval has on the overall cost.

The background and scaling section is used to apply a background shift or a scaling factor to the experimental data. Checking the **All Scans** check box signals that any changes made to the background shift and scaling factor field will be applied to all the data in the dataset. Unchecking the **All Scans** check box signals that any change made to the background shift or scaling factor will only be applied to the selected data. The **background shift** field shifts the scan up or down by adding a constant value to the reflectivity. The typical range of the background shift is between values of -5×10^{-8} and 5×10^{-8}. It is important to be aware that a negative background shift can result in negative reflectivity values. This is extremely important to consider when applying the logarithmic transformation in the data fitting because logarithms are undefined for negative values. The **Scaling Factor** field scales the experimental reflectivity by multiplying the reflectivity with the entered value. The typical values entered for the scaling factor are found between 0.8 and

1.2. The **Fit** button is used to fit the background shift or scaling factor. Experimental data with background shift or scaling factor selected to be fit will have the background shift and scaling factor field turn red. The scaling factor and background shift fits can be removed by clicking the **Remove Fit** button.

5.9 Smooth Data Workspace

The smoothing data section is used to remove the noise in the experimental data for the total variation penalty implementation. The total variation penalty is calculated by taking the absolute difference between the total variation of the simulated and experimental data. The total variation is used to calculate the one-dimensional arc-length to quantify the shape of the data. The presence of noise in the experimental data adds additional variation such that the shape of the experimental data is not properly quantified. Therefore, the noise needs to be remove to capture the trend of the experimental data. Spline interpolation and Fourier filtering are the methods available to remove the noise from the experimental data.

Upon navigating to the smooth data workspace the window in 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. The scan to work on is selected in the **Selected Scans** option box. The **Smoothing Scale** option box is used to transform the reflectivity and includes the transformations of R, $\log_{10}(\mathrm{R})$, $\ln(\mathrm{R})$, and $\mathrm{R}(q_z) * q_z^4$. In the middle-left region of the window, there is a button labeled **Set Noise Reduction** and an option box labeled **Methodology**. The **Set Noise Reduction** button will internally save the applied noise removal, but the parameters used for the noise removal are not internally saved. Therefore, every time the program is closed the noise removal will need to be reapplied to the data. The **Methodology** option box contains the different smoothing methods available in GO-RXR. The only two methods available are spline interpolation and Fourier filtering. A neural network approach is currently being developed to include into the software.

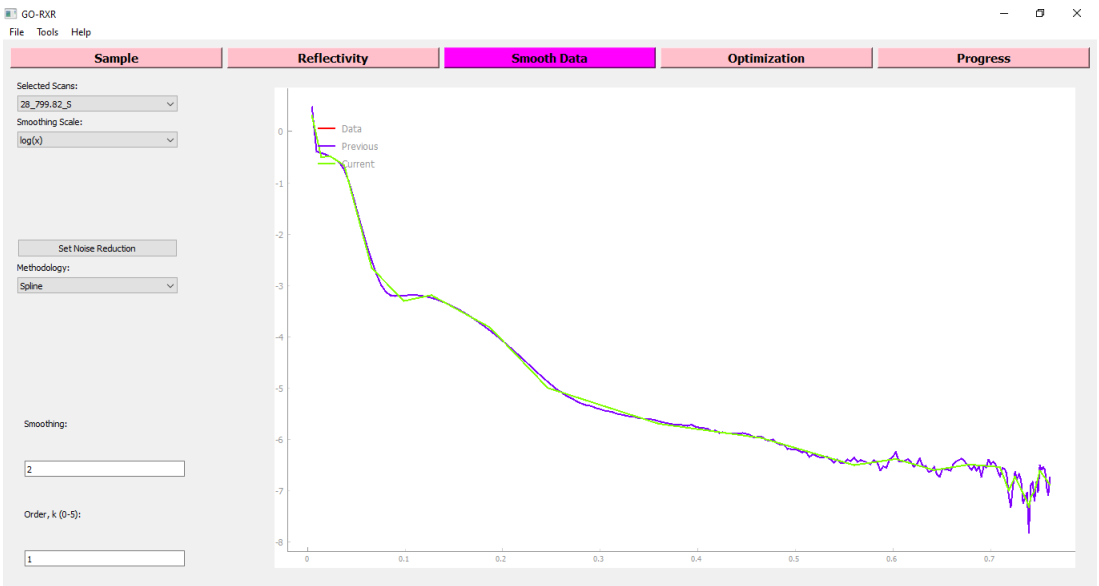

Figure 5.9: Spline Interpolation

The spline method used the spline interpolation to remove the noise. Figure 5.9 displays

the parameters that will appear at the bottom-left of the window when the spline method is selected. The **Smoothing** field is a parameter that dictates the amount of smoothing that is applied to the spline interpolation. A small smoothing parameter will smooth the data less, whereas a large value will result in more smoothing. The smoothing parameter must be a real positive value. The **Order** field dictates the degree of the smoothing spline. The value of the order must be a positive and real integer between 1 and 5, inclusive. The default value is a cubic spline with an order value of 3.

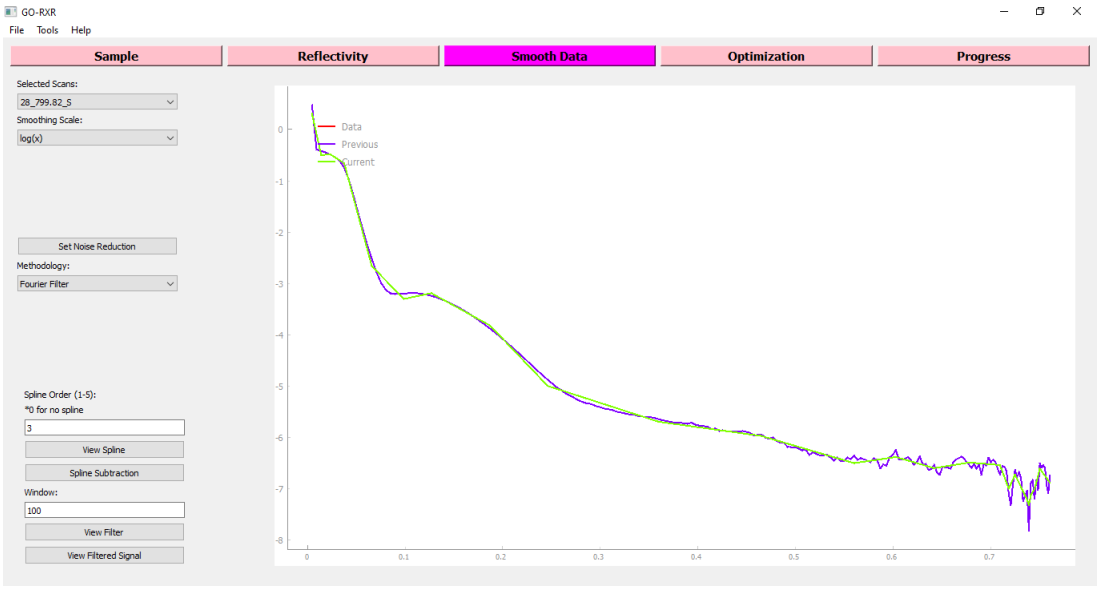

Figure 5.10: Spline Interpolation

Figure 5.10 illustrates the parameter field in the bottom-left corner of the window once the **Fourier Filter** method is selected. 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. The best-fit spline can be viewed in the plotting area by clicking the **View Spline** button. Figure 5.11 illustrates the best fit spline as displayed in the plotting area.

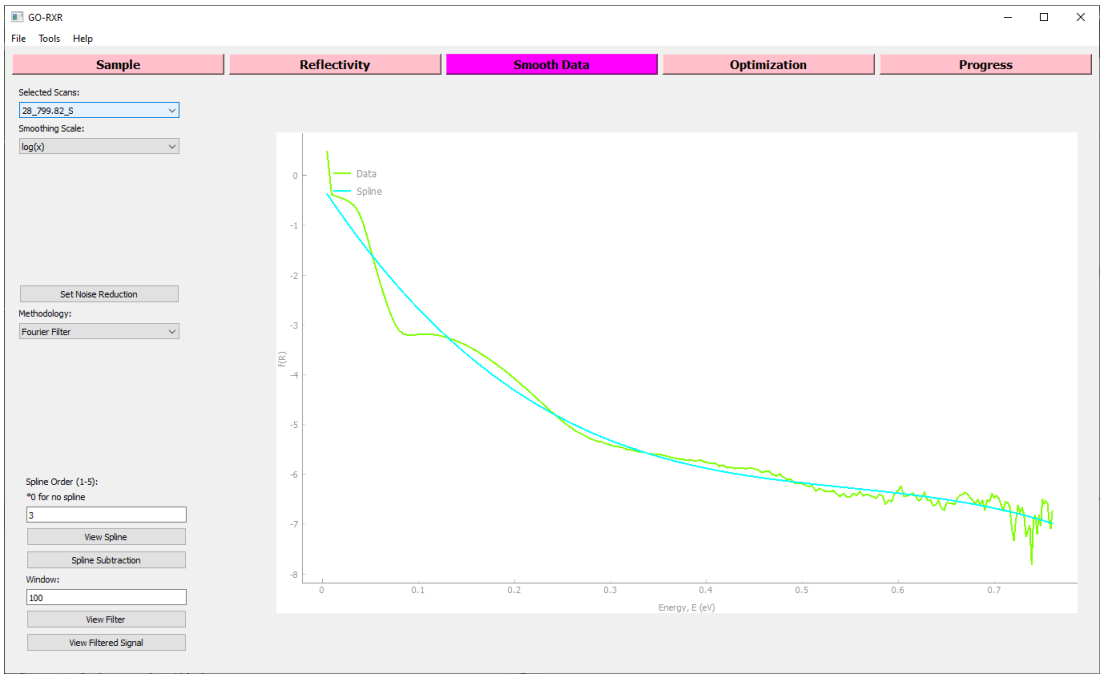

Figure 5.11: Spline Comparison

The best-fit spline is then subtracted from the reflectivity data and is known as a spline subtraction. The spline subtraction is required in the Fourier transform to introduce periodicity into the experimental data. The spline subtraction can be viewed by clicking the **Spline Subtraction** button. An example of the spline subtraction is displayed in Figure 5.12.

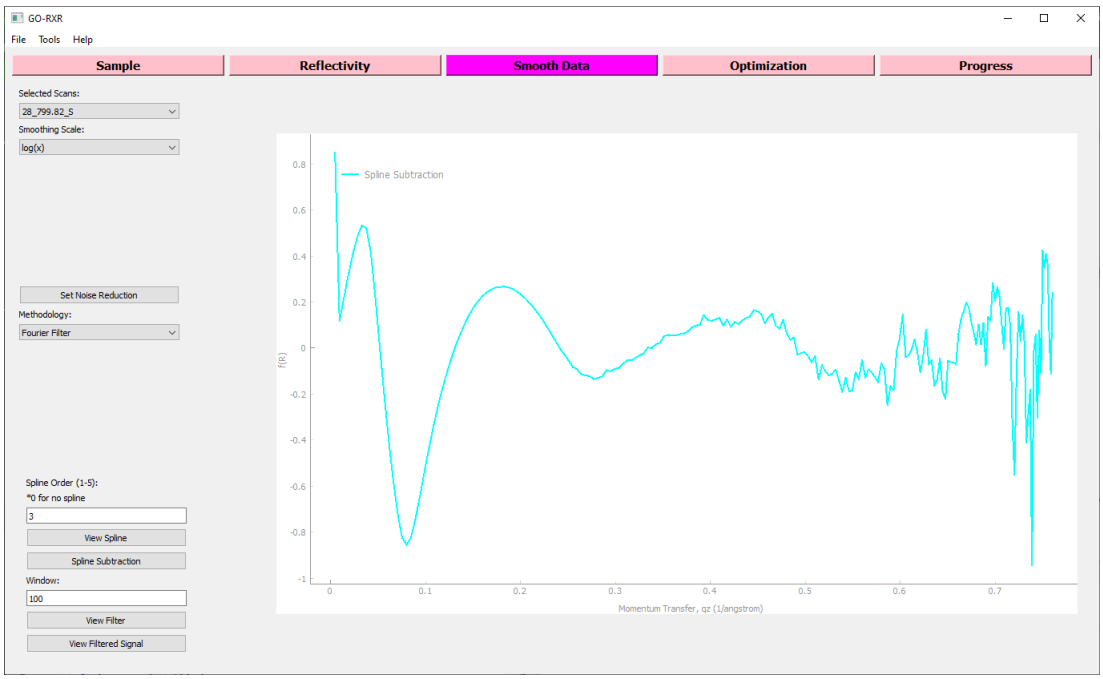

Figure 5.12: Spline Subtraction

After applying the spline subtraction the Fourier filter is designed. The Fourier filter is a boxcar function, where the width of the boxcar is controlled by the window parameter.

The designed Fourier filter with respect to the Fourier transformed data can be compared by clicking the **View Filter** button. The designed filter is superimposed on top of the Fourier transformed data, illustrating the frequencies that will be removed from the data. An example of a filter is shown in Figure 5.13. Once finalizing the design of the Fourier filter, the filter can be applied to the experimental data. The filtered data can be viewed by clicking the **View Filtered Signal** button.

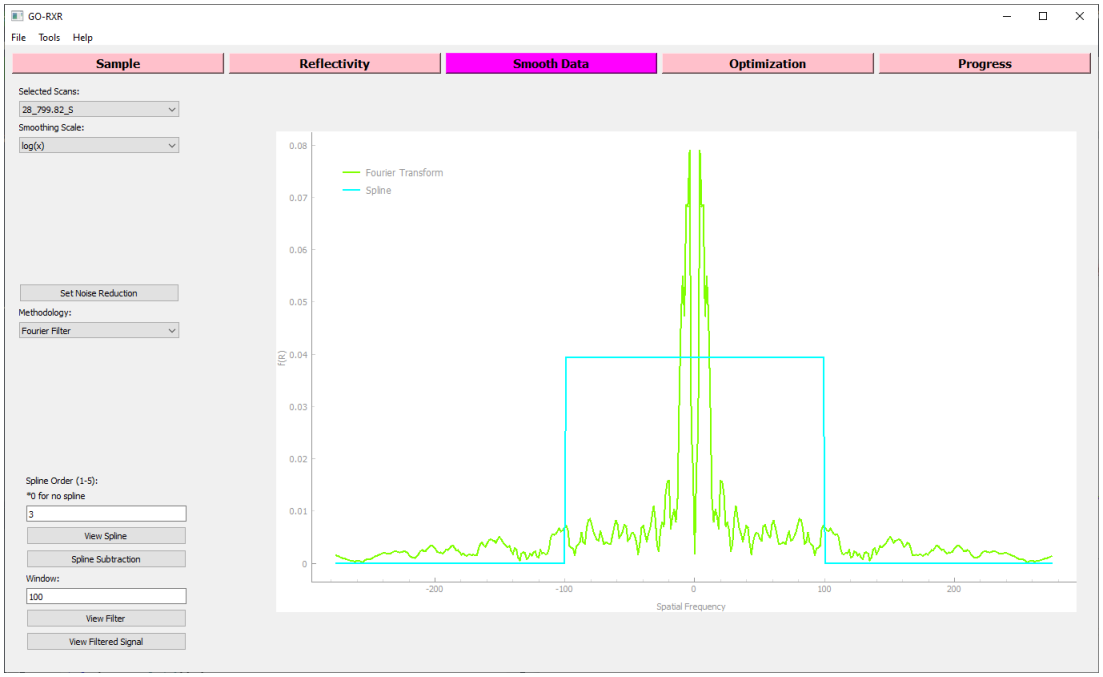

Figure 5.13: Figure 5.13: Filter Window

The spline interpolation is the preferred noise removal method for moderately noisy data because the Fourier filter introduces oscillations into the experimental data that make the noise removal ineffective for the total variation implementation. The oscillations are present due to the Fourier filter treating the experimental data as a Fourier series. However, the Fourier filter is preferred for extremely noisy data as it tends to capture the general trend of the data better than the spline interpolation, such that the oscillations introduced by the Fourier filter are negligible compared to the quality of the noise removal using the spline interpolation method.

5.10 Optimization Workspace

The optimization workspace is used to set the parameters for the data fitting. Figure 5.14 displays the window that appears after navigating to the optimization workspace.

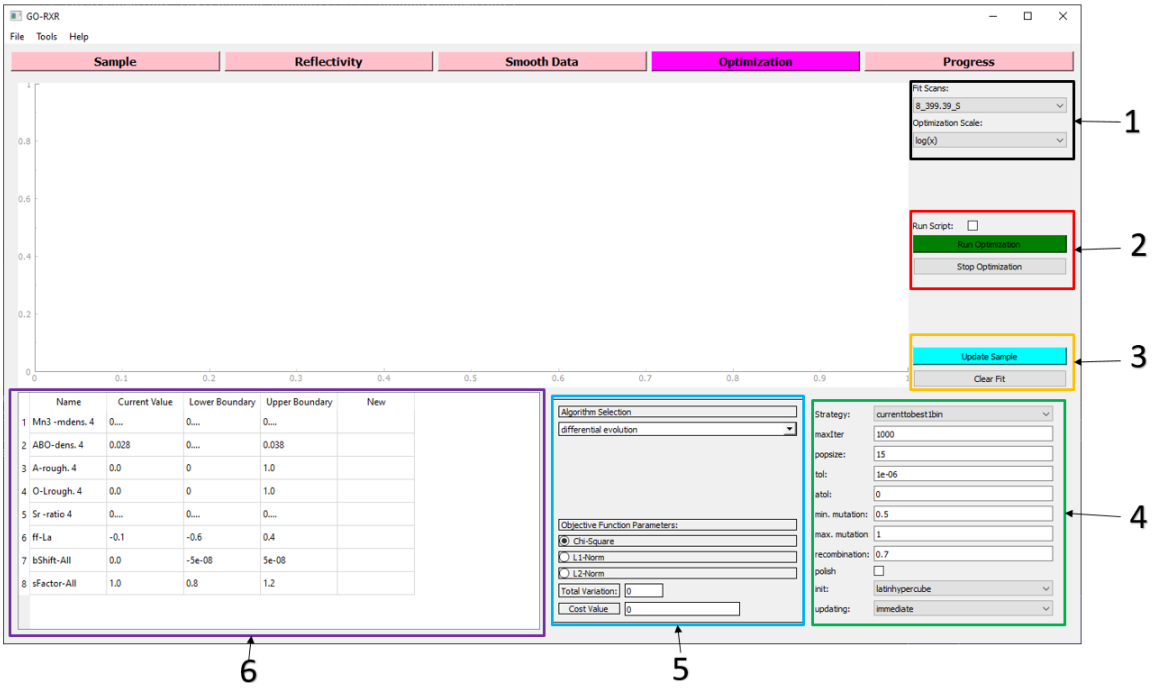

Figure 5.14: Optimization Workspace

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 contains two option boxes and are labeled **Fit Scans** and **Optimization Scale**. The scan to view in the plotting area is selected in the **Fit Scans** option box. Before starting a data fit, the experimental data and the simulated data are plotted. The simulated data is calculated from the model defined in the sample workspace. After a data fit has terminated, the experimental data, the simulation of the initial model, and the simulation of the fit model are displayed in the plotting area. The **Optimization Scale** option box determines the transformation that is applied to the reflectivity in the optimization. The transformations available are the same as in the smooth data workspace and include R, $\log_{10}(\mathrm{R})$, $\ln(\mathrm{R})$, and $\mathrm{R}(\mathrm{qz}) * \mathrm{qz}^4$. The software requires that the theta/two-theta asymmetry curves and asymmetry spectrum use the R transformation because of the possibility of negative values. The $\log_{10}(\mathrm{R})$ and $\ln(\mathrm{R})$ transformations are recommended for the theta/two-theta

reflectivity scans and the R transformation is recommended for the energy scans.

The **Start Optimization** section is used to start and stop the data fit. The **Run Script** check box in this section signals whether to use the script in the optimization. A checked box indicates that the script will be used in the data fit, whereas an unchecked box indicates the script will not be used. The script check box state cannot be applied to the data fit while a data fit is running. Only the state of the script checkbox right before a fit is started is used. The script will be ignored in the data fit if there is an error detected in the script. The **Run Optimization** button is used to start a data fit. A green **Run Optimization** button denotes that a data fit can be started. During a data fit, the **Run Optimization** button turns red to indicate that a data fit is currently in progress. Clicking the **Run Optimization** button while a data fit is running will not start a new data fit. Therefore, multiple data fits cannot be run simultaneously in parallel. The **Run Optimization** turns green once the data fitting process is finished. The **Stop Optimization** button allows the user to stop the current data fit before the algorithm terminates on its own. The **Stop Optimization** button does not immediately stop the current data fit, and will terminate once the most recent iteration of the algorithm terminates. The software will update the fitting parameters with the results from the last iteration of the global optimization algorithm.

The update fit section contains the **Update Sample** and **Clear Fit** buttons. The **Update Sample** button is used to apply the results from the data fit to the model defined in the sample workspace. Clicking **Update Sample** will not automatically save the updated model to the project file, and will need to be manually saved. The **Clear Fit** button clears the fitting parameters, including the results of the most recent data fits. Clicking **Clear Fit** will not clear the selected scans and must be removed in the reflectivity workspace.

The optimization parameters section is used to select the global or local optimization algorithm and to customize the objective function. The **Algorithm Selection** option box contains the global and local optimization algorithms available. The recommended global optimization algorithm is the differential evolution algorithm. The **Objective Function Parameters** displays three check boxes: chi-square, L1-norm, and L2-norm. The selected check box will be used as the cost function in the data fitting. Recently, the arctan cost function has been included into the optimization parameters section. The **Total Variation** field determines the weight that will be applied to the total variation comparison function. A value of zero excludes the total variation penalty from being applied to the objective function. The recommended value for the weight of the penalty term is between 100 and 1000. These values have been found to ensure that the total variation penalty is smaller than the cost function. The value of the cost function for the selected scan can be displayed by clicking the **Cost Value** button. The cost value is displayed in the field next to this button.

The algorithm parameters section allows the user to set the parameters of the global and local optimizations algorithms. It is recommended to use the default algorithm parameters. The role of the various algorithm parameters are defined in the SciPy documentation for the optimize object [2]. The default parameter values for the differential algorithms are shown in Table 1.

Table 1: Differential Evolution Parameters	
Parameters	**Values**
strategy	currenttobest1bin
maxIter	50
popsize	15
tol	10^{-6}
atol	0
min. mutation	0.5
max. mutation	1
recombination	0.7
init	latinhypercube
updating	immediate

The parameter boundaries section is used to set the lower and upper boundaries of the fitting parameters. This section displays the current parameter value, the lower boundary, the upper boundary, and the results from the data fit. Only the lower boundary and upper boundary columns can be edited by the user. The first column indicates the name of the parameter that is being fit. The naming convention for the energy shift of an atomic form factor is '*ff(m)-XX*', where '*ff*' indicates a non-magnetic atomic form factor and '*ffm-*' indicates a magnetic atomic form factor. The name of the atomic form factor is found in the '*XX*' location. The naming convention for the background shift and scaling factor are ŕextitparameter-scan', where the options for '*parameter*' are '*bShift*' to denote a background shift and '*sFactor*' to denote a scaling factor. The '*scan*' field indicates the scan that the background shift or scaling factor is applied to, where a '*scan*' field set with '*all*' indicates that the background shift or scaling factor is applied to all the data scans selected for the data fitting. The model parameters contain the naming convention of '*symbol-parameter. layer*'. The '*symbol*' field can indicate if the parameter is being fit in element or compound mode. A single element/identifier in the '*symbol*' field indicates that the parameter is being fit in element mode, whereas a '*symbol*' field that contains the compound representation of the layer is being fit in compound mode. The '*layer*' field indicates the layer where the parameter is found. A '*layer*' field that is given a value of '0' denoted the substrate layer. The '*parameter*' position contains the short form of the parameter that is being fit. The short form of the parameters are displayed in Table 2.

Table 2: 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. Figure 5.11 displays the window after navigating to the progress workspace.
The **Total Cost** button will display the addition of the cost function and the total variation penalty value (y-axis) with respect to the number of iterations (x-axis). The **Total Cost**

Figure 5.15: Progress Workspace

will update after each global optimization algorithm iteration. The **Norm** button displays the cost function (chi-square, L1-norm, L2-norm, arctan) without applying weights to the boundaries with respect to the number of iterations. The **Variation** button displays the value of the total variation penalty term with respect to the number of iterations. The **Parameters** button displays the evolution of the fitting parameters with respect to the number of iterations. The **Density Profile** button will display the density profile with the parameters values from the most recent iteration of global optimization algorithm. The **Selected Scans** option box indicates the data that is included in the data fitting process. A scan can be selected from **Selected Scans** option box, where the simulation and experimental data for the selected scan is displayed in the plotting area. The simulated scan is calculated from the parameters from the most recent iteration of the global optimization algorithm. The **All Scans** option box contains all the experimental data, where any scan can be selected and viewed in the plotting area. Both the experimental and simulated data is displayed in the plotting area.

6 File Format

6.1 Data File

The most common way to load the experimental data is from the ReMagX file with the '.all' file extension. However, if unfamiliar with ReMagX then it would be easiest to create a data file using HDF5. It is suggested to use *h5py* library found in python [1] to create the HDF5 data file. Figure 6.1 illustrates the architecture of the HDF5 data file.

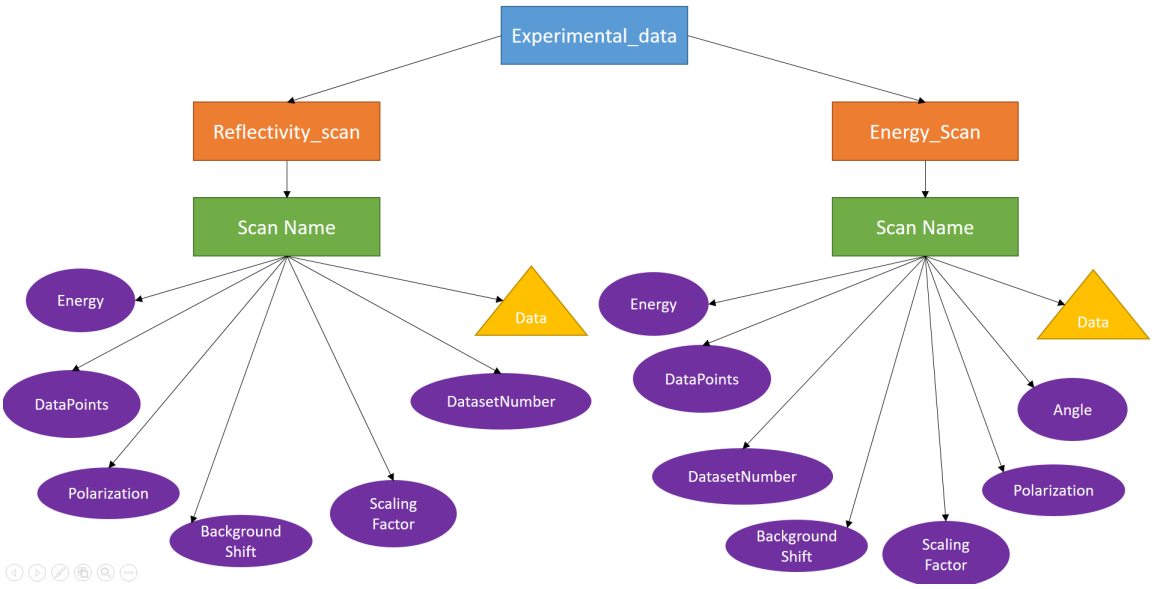

Figure 6.1: Data File Flowchart

The squares in the flowchart demonstrate a group. The blue square is the main group, whereas 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 triangles refer to the location the data is stored. The green square refers to a data scan, where a new subgroup will need to be added for each scan in the dataset. Each scan included must be given a unique name that follows the naming convention defined in Section 6.3 in the user manual. A sample of the code used to create the HDF5 data file is demonstrated below.

```
# reflectivity_scans and energy_scans are assumed to be dictionaries with
# all the appropriate information
# e.g. {'name1': {'Data',[...],'Energy': E,'Angle': theta,
#       'Polarization': pol}}

reflectivity_scans = ... # load reflectivity scans
energy_scans = ... #  # load 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

# theta/two-theta reflectivity group
grpR = grp1.create_group("Reflectivity_Scan")
# energy scan group
grpE = grp1.create_group("Energy_Scan")

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
    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
```

A simulation group can be created by setting the group name as *Simulated_data* instead of giving the name *Experimental_data* for the experimental data group. The functions used to create the HDF5 data files are found in the *data_structure.py* python file in the `https://github.com/lucaskorol21/GO-RXR` Github repository.

6.2 Attributes

The attributes define other import information regarding the experimental data. The attributes included in the HDF5 data files are

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

Angle - Grazingr angle of the incoming x-rays and should be entered as a float type.

Background Shift - Background shift and should be entered as a float type

Scaling Factor - Scaling shift and should be entered as a float type

Polarization - Polarization of the incoming x-rays. For s-polarized light the polarization is entered as 'S', p-polarized is entered as 'P', linear asymmetry is entered as 'AL', right circular polarization is entered as 'RC', left circular polarization is entered as 'LC", and circular asymmetry is entered as 'AC'.

DataPoints - Number of data points and is entered as an interger type.

DatasetNumber - Scan number and is entered as an integer type.

In GO-RXR, the linear asymmetry is calculated by

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

where the subscript S and P denote s-polarized and p-polarized light, respectively. Similarly, the circular asymmetry is calculated by

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

where the subscript L and R denote left circular and right circular polarizations, respectively.

6.3 Naming Convention

Reflectivity Scan

For theta/two-theta reflectivity scans the naming convention is '*scanNumber_Energy_Polarization*'. For example, a theta/two-theta reflectivity scan with the x-ray energy of 500 eV, a σ-polarization, and a scan number of 3 will have the name

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

where the energy is typically shown as a float. The linear and circular theta/two-theta reflectivity curves have a similar naming convention, but differ slightly. For example, a circular theta/two-theta reflectivity curve that uses the left and right circular theta/two-theta reflectivity scans with the scan numbers of 8 and 9, and a x-ray energy of 640.2 eV will have the name

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

where the '*scanNumber*' includes the scan number of both theta/two-theta reflectivity scans used in the asymmetry calculation.

Energy Scans

The naming convention of energy scans is '*scanNumber_Energy_Theta_Polarization*'. An energy scan between 450 eV and 480 eV, with an σ-polarization, grazing angle of 5.0 degrees, and a scan number of 15 will be identified as

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

where the prefix *E* denotes the lower energy limit of the energy scan and the prefix *Th* refers to the constant grazing angle. The asymmetry spectrum follow a slightly different naming convention. For example, a circular asymmetry spectrum that has minimum x-ray energy of 450 eV, a grazing angle of 5.0 degrees, and uses the left circular scan with scan number 18 and the right circular scan with scan number 19 has the name

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

where the '*scanNumber*' indicates the scans used in the asymmetry calculation. The naming convention used in GO-RXR is the same naming convention used in ReMagX. The naming convention is extremely important because the energy, grazing angle, and polarization are used to calculate the reflectivity from the model.

6.4 Form Factor File

The form factor files have the file extension '.ff' or '.ffm'. This is the convention used in ReMagX and used in GO-RXR to allow easy transition between both softwares. The file extension of '.ff' indicates to GO-RXR that the file is a non-magnetic form factor, whereas the extension '.ffm' indicates a magnetic form factor. The form factor files can be written in a text editor and store the energy and form factor pairs. The energy of the form factors are included as the first column in the file. The second column includes the real component of the form factor and the third column contain the imaginary component of the form factor. Each column in the form factor file must be separated by a space and cannot load in form factor files that separate the columns with a comma. Figure 6.2 illustrates an example of a form factor file.

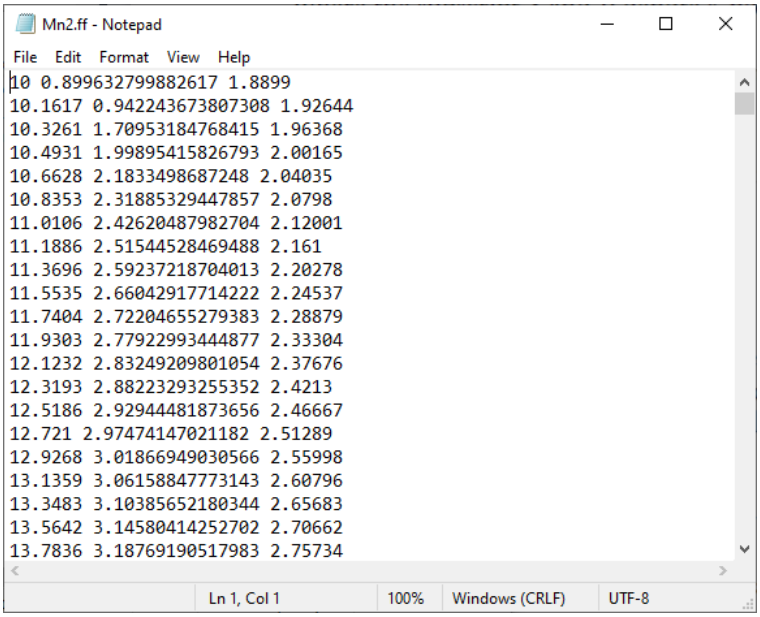

Figure 6.2: Figure 6.2: Form Factor File

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).