Installation

The following packages are needed to run TroFit and are located in the package setups folder.

Python 3.2
Numpy 1.8.1
Scipy 0.13.3
Matplotlib 1.2.1
Lmfit 0.7 (other versions do not work)

When these packages are installed, create a *.trofit project file (see section below) and then start the program as described there.
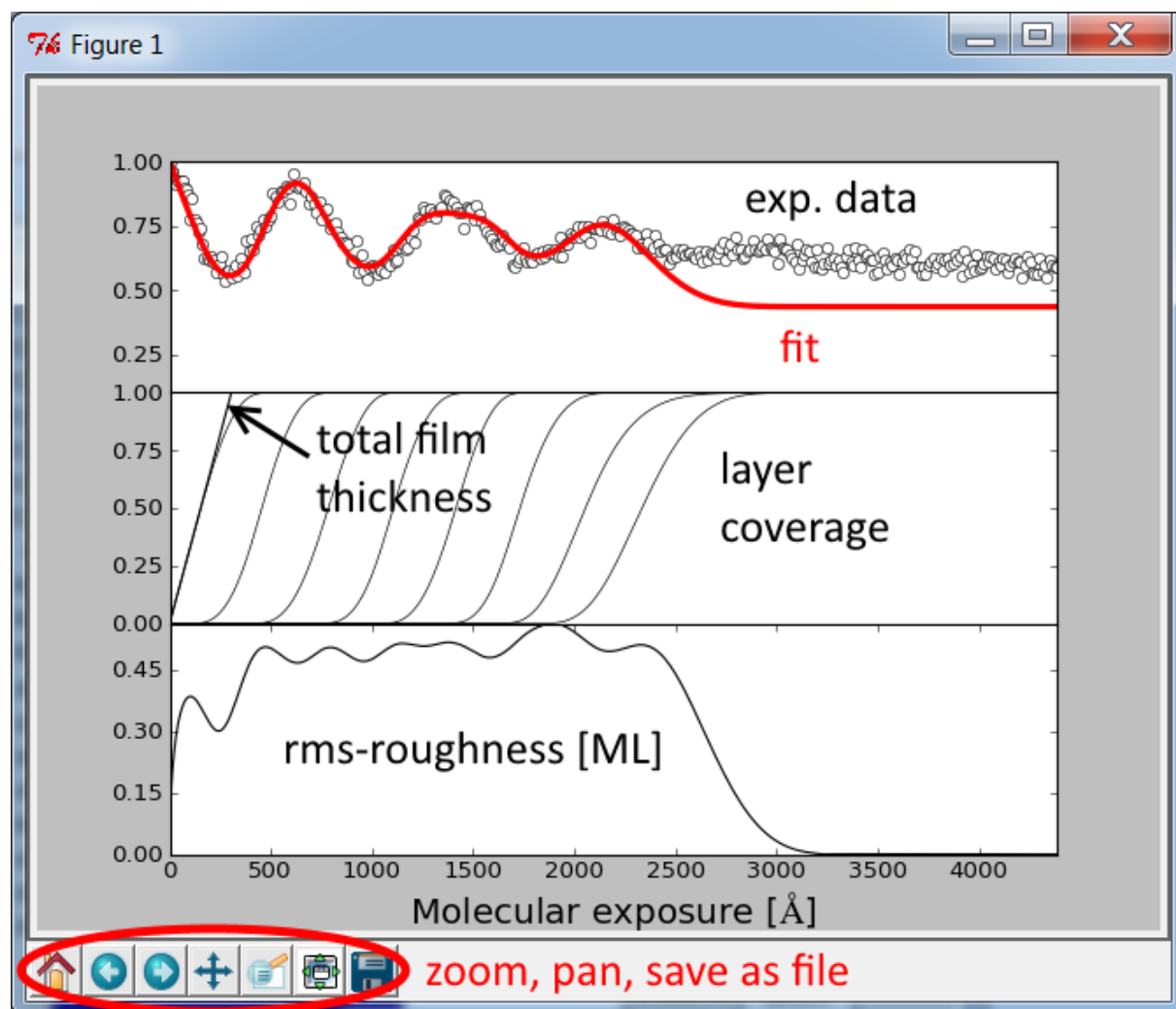
User interface

Fig1: Display of the data (white circles) and the fit (red) in the top panel. The middle panel shows the layer coverages from empty (0) to completely filled (1). Bottom panel shows the calculated rms-roughness in either units of monolayers or Å.
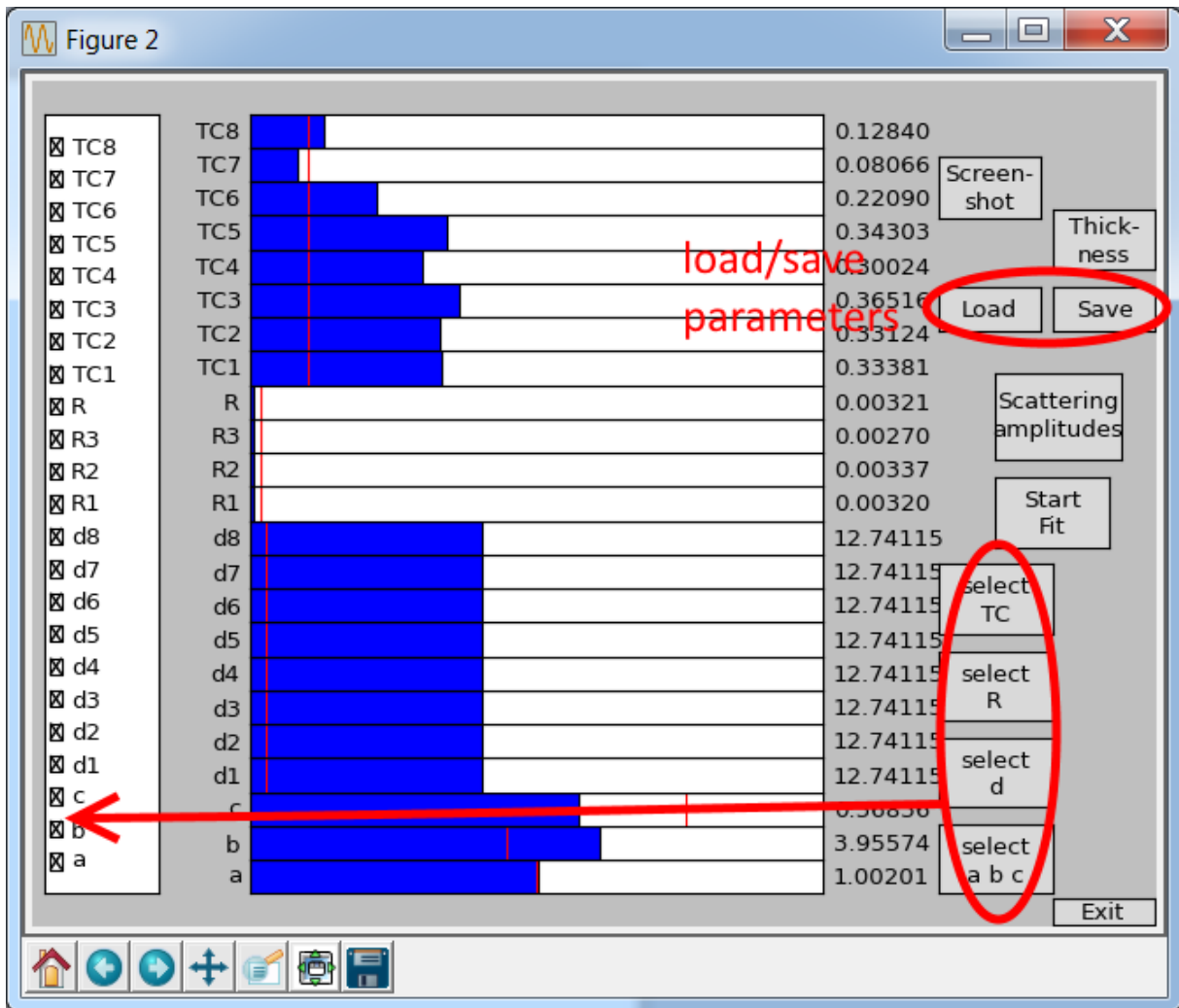
Fig2: Left checkboxes determine whether a parameter is used for the fit (checked) or left unchanged (unchecked). Middle blue sliders can be dragged with the mouse and the fit gets updated.

Button functionality

Screenshot — Save screenshots of the current graphs (.png (1000 DPI), .png (125 DPI), .svg (Vectorgraphics), .pdf (Vectorgraphics))

Thickness — Toggle displaying the total film thickness

Load — Load a new set of parameter values from a file

Save — Save the current parameter values (_parameter.dat), the fit (_fit.dat), the total film thickness and individual layer coverages (_coverages.dat) and the calculated film roughness (_roughness.dat). **FILES WILL BE OVERWRITTEN WITHOUT WARNING**

~~Scattering amplitudes — Shows an animation of the scattering amplitudes with evolving time~~

Start Fit — Start the fitting procedure (there is no stop until the fit is finished)

Select TC, R, d, abc — Select just a subset of the parameters for fitting

Exit — Exit the program

Fig3: Some useful output is printed in the console, while using the program.

Parameter explanations

a	A_{sub}	scattering amplitude of the substrate
b	φ_{sub}	phase difference substrate/film
c	F_{mol}	molecular form factor of the ad-layers
d		thickness of one single ad-layer
R		growth rate of the ad-layers
TC	θ_{cr}	critical coverages of the ad-layers

Reflectivity models

There are three models for the calculation of the reflectivity implemented. Custom models can be added as desired in the **TroFit_bib.py** script file. When implementing new reflectivity models, a function in **TroFit_bib.py** needs to be implemented/adjusted as well as the corresponding parameterset in **parameters.py**, which is also where the limits of all parameters for the fit are stored. The reflectivity is calculated according to this equation:

$$I_{reflected}(t) = \left| a \cdot e^{ib} + c \cdot \sum_{n} \theta_n(t) \cdot e^{i \cdot n \cdot q \cdot d} \right|^2$$

"reflectivity"	c and d are identical for all ad-layers
"reflectivity2"	1st ad-layer has different molecular form factor c1 and thickness d1 than the upper layers (c and d)
"reflectivity3"	there are exactly 8 ad-layers, with identical c and different d (d1, d2, … , d8)

The currently implemented Trofimov model has 4 different growth rates (the first 3 ad-layers are distinct and ad-layers 4+ have all the same rate).

Project files

Project files (*.trofit) are simple text files containing the following six lines:

Test	Projectname, used to name the saved parameters etc…
test_data.dat	Filename of datafile, a xy text file(relative path to the project file or absolute path)
500	Max number of datapoints to read from **test_data.dat**
0.284779	Value of the q-point in $Å^{-1}$
4 4 10	Number of parameters (1st = total number of a,b,c,d parameters, 2nd = 4, 3rd = total number of theta criticals Tc1Tc2Tc3…)
reflectivity	Which reflectivity model to use (defined in **TroFit_bib.py**)

An example project file is located in the folder testdata. The working directory, where all parameter files, fit files, screenshots etc… are saved, is the directory where the project file is.

When a project file is created it can be opened with TroFit by drag'n'drop onto **TroFit.py** or by registering **TroFit.bat** as a standard program for *.trofit files and then double-clicking the project file.

Troubleshooting

If the program closes unexpectedly please check that the content of the project file (*.trofit) is valid and that the parameter file (*_parameter.dat) has the correct number of parameters inside.

Additional tools

In the folder tools there are some useful scripts for preparing your data:

data_import_helper.py can import a *.xy data file or a scan from a *.dat specs file and adjust the spacings of the x-values in such a way, that a constant rate throughout the whole film growth is achieved. For this purpose a file of the QCM monitor has to be supplied. Open the file in a text editor and then adjust the first lines in the "Start user input" section as needed, then run the script.

footprint.py shows the raw and footprint corrected version of a reflectivity scan (drag and drop *.xy data file(s) onto the script file).

svg2emf.py Converts a *.svg file to *.emf using Inkscape, which needs to be installed. For usage just drag and drop the *.svg file(s) onto the script file.