MatSpecGUI 1.0, guide

J. Novák*
ID10B beamline, European Synchrotron Radiation Facility (ESRF), Grenoble, France

April 10, 2009

Contents

*E-mail: novak@esrf.fr

1 Introduction

MatSpecGUI is a Matlab based utility which allow for processing and imaging of 2D x-ray data acquired via SPEC software. In particular, the utility is aimed to handle x-ray data measured via 1-dimensional detectors (position sensitive detectors, PSDs). MatSpecGUI provide a graphical user interface for a convenient access to basic operations with data, including:

- extracting data from a SPEC file
- normalization of measured intensity for beam attenuators and a primary beam monitor
- transformation of angular coordinates to Q-space
- plotting data
- saving plots to image-files
- exporting data to ASCII files.

The first part of the guide gives an overview of usage of MatSpecGUI at an user level. Additionally, representation of SPEC data inside the package is described, which could be useful for users interested in further data manipulations using Matlab. MatSpecGUI is deliver with functions for transformation from angular space to Q-space, which are basically specific for the ID10B beam-line at the ESRF. However, the utility can be easily used for treatment of data from other x-ray instruments as well. For that purpose, the guide provides information on the structure of the transformation functions and their binding to MatSpecGUI. Particularly, if a set-up with the same organization of goniometer motors and orientation of PSD as for already implemented transformation functions is used and only difference is in the motor names, it is sufficient to change motor names provided to the transformation function from within the graphical user interface.

2 Installation and running MatSpecGUI

2.1 Installation

MatSpecGUI should run on any computer where Matlab is accessible. The utility is distributed in a zip-file MatSpecGUI.zip, which can be downloaded from http://www.physics.muni.cz/ novak/MatSpecGUI/MatSpecGUI.html. The zip-file should be extracted to an installation directory. It is advisable to add the installation directory to the Matlab lookup path via `File -> Set Path ...` in the Matlab menu.

The distribution zip-file contains two executable files gridq1_01 and gridq1_01.exe besides other files. The two files are a Linux and a MS Windows, respectively, compiled versions of a *C* program and serve for mapping unevenly spaced data to a regular grid (see Sec. 3.7). The functionality of the files should be checked after extraction of the package. Some users observed that after extracting the zip-file to a

UNIX-like system the program gridq1_01 does not have executable access right on. In that case, go to the installation directory change the access rights to the program by the command

```
UNIX> chmod u+x gridq1_01
```

If it is still not possible to execute the file, the last resort is to compile file gridq1_01.cc from the zip-file gridq1_01.zip by a c++ compiler. By the GNU C compiler it is done by

```
UNIX> g++ -s -o gridq1_01 gridq1_01.cc
```

2.2 Running MatSpecGUI

Start a Matlab session and be sure that MatSpecGUI is in the Matlab look-up path or change current directory to the Matlab installation directory. Start the MatSpecGUI graphical user interface by the command `MatSpecGUI` from within the Matlab session:

```
>> MatSpecGUI
```

3 Graphical user interface

This section describes work with MatSpecGUI at basic user level using graphical user interface (GUI).

3.1 Main menu

Once you started MatSpecGUI (see Sec. 2.2) you are facing the main menu (see Fig. 1) of the utility. Buttons of the menu give access to sub-menus dedicated to individual operations with measured data. In the following, GUI elements (e.g., buttons, editable text windows, and drop-down lists) of menus in figures are marked with numbers by which they are referred in the text. The order of operations as they are listed corresponds approximately to the order in which x-ray data need to be processed before one obtains a reciprocal space image of data from a SPEC file.

The operations to which main menu gives access are as follows:

(1) Loading one or more scans from a SPEC file (see Sec. 3.2).

(2) Loading MatSpecGUI data from a Matlab file (see Sec. 3.3).

(3) Normalizing intensity for absorbers and a monitor (see Sec. 3.4).

(4) Plotting x-ray intensity versus a motor position and channels of PSD as a color-scaled 2-dimensional plot (see Sec. 3.5).

(5) Transforming angular coordinates to Q-space (reciprocal space) coordinates (see Sec. 3.6).

(6) Mapping unevenly sampled data (i.e., intensity measurements) to a regular rectangular grid (see Sec. 3.7).

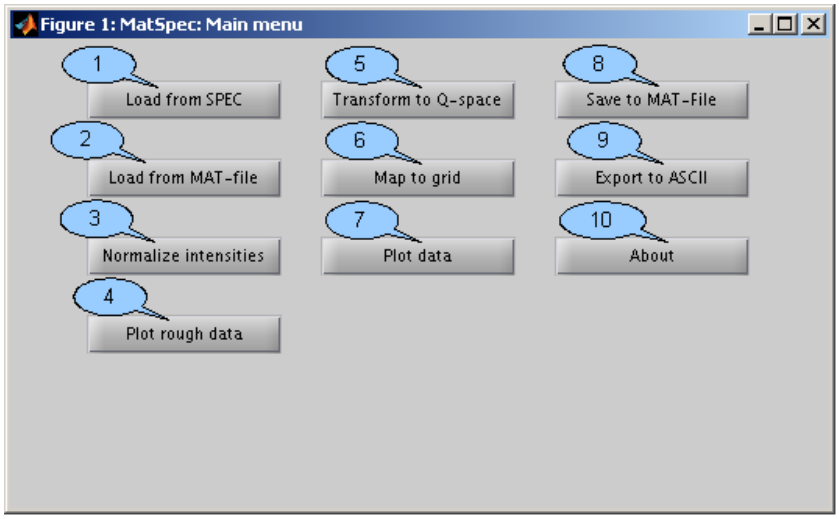

Figure 1: Main menu.

(7) Plotting data in Q-space as a color-scaled 2-dimensional plot. (see Sec. 3.8).

(8) Saving MatSpecGUI data to a Matlab file (see Sec. 3.9).

(9) Saving data to an ASCII file (see Sec. 3.10).

(10) Information about MatSpecGUI.

3.2 Loading data from SPEC file

The sub-menu shown in Fig. 2 serves for loading one or more scans from a SPEC file. Several scans can be merged together. The loaded data are stored in data-sets (see Sec. 5) referred by a name given by user. All further data processing and operations in other sub-menus are done over these data-sets. A data-set contains information on motor positions, intensities on detectors, and head(s) of scan(s) from the source SPEC file as variables. New variables are appended to data-set in some sub-menus by doing transformations on data (e.g., intensity of a detector is normalized and results into a new variable). The name of variable with data from a 1-dimensional detector is `PSD`. Multiple data-sets can be stored in MatSpecGUI. The required content of the elements of the sub-menu is as follows:

(1) A name, or path + name if needed, of a SPEC file from which data should be sourced.

(2) The button opens a open-menu for browsing directory structure to input SPEC file to the box (1).

(3) A scan number or a (Matlab) vector of numbers of scans to be sourced. The notation for one scan is simply a number of scan (i.e., an integer). To load more scans at once and to marge them together, notation like `[3:7,10]` is to be used. In this way, you achieve loading scans 3 to 7 and the scan 10 and merging them together into one data-set.

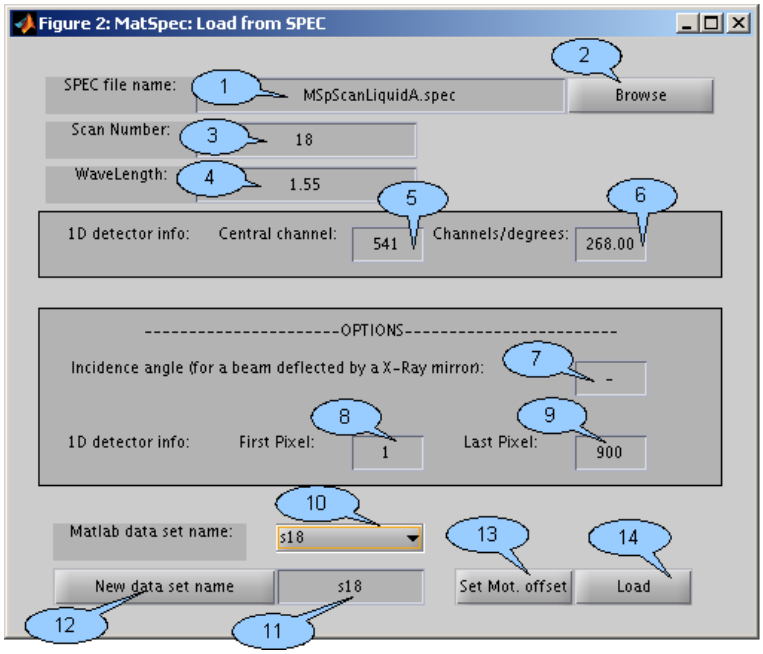

Figure 2: Sub-menu: Load from SPEC.

(4) The wavelength of x-rays used for the experiment in Å.

(5) A central channel of the PSD, i.e., index of a channel c_0 of the PSD to which the direct beam impinges when the detector is in the origin of the angular coordinate system. For ID10B beamline the detector arm moves around a horizontal and a vertical axes by angles δ and γ, respectively. Thus, central channel c_0 should be determined at $\delta = 0$, $\gamma = 0$.

(6) Number of channels by which PSD moves with respect to the direct beam when rotated by 1° around an axis perpendicular to the PSD orientation. (Should be calibrated during an experiment!)

(7) (Optional) An angle (in deg) by which the primary beam is deflected from the horizontal plane for purpose of experiments at liquid surfaces (see 4.1). For ID10B beam-line at the ESRF the deflection is done by an x-ray mirror in a optical hutch and the angle is not explicitly written to SPEC files. Thus it needs to be entered manually.

(8) (Optional) A bottom limit (index of a pixel) of PSD from which PSD data should be loaded.

(9) (Optional) An upper limit (index of a pixel) of PSD up to which PSD data should be loaded.

(10) A drop-down list with names of data-set to which the SPEC data can be loaded. Choose required data-set. The loaded data are referred by the chosen name in other sub-menus.

(11) A text box to input a new name of a data-set.

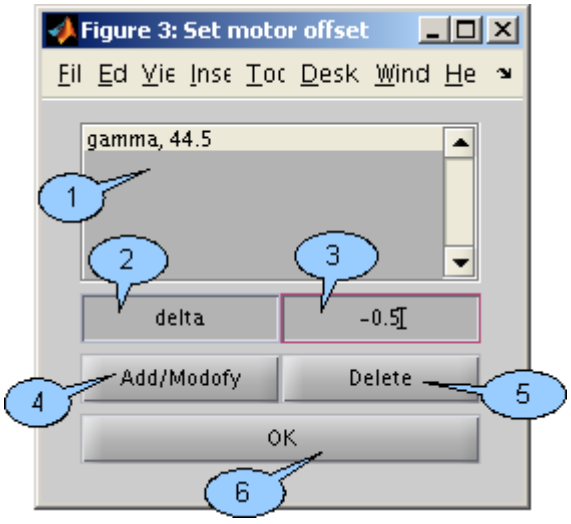

Figure 3: Sub-menu: Set motor offset

(12) A button to add a new data-set name (11) to drop-down list (10). The just added name becomes the current one.

(13) A button for opening a sub-menu to set motor offsets (see Sec. 3.2.1) at the moment of loading data.

(14) A button to load data and closing the sub-menu.

3.2.1 Setting motors offsets

The sub-menu "Set motor offset" (see Fig. 3) serves for correcting differences between physical position of motors and corresponding "software" values written in a SPEC file. In other words, it allows for a change of the origin of the set of coordinates. The indicated value of the offset is substracted from values written in a SPEC file whenever data are loaded to MatSpecGUI, i.e., $\text{motor}_{\text{MatSpecGUI}} = \text{motor}_{\text{SPEC}} - \text{offset}$.

Elements of the sub-menu are as follows:

(1) A list box with pairs motor name, motor offset.

(2) A text box for a motor name for which the offset should be set.

(3) A text box for a value of the offset which should be set.

(4) A button to add the pair motor name (2), motor offset (3) to the list box (1).

(5) Deletes a selected pair motor name, offset from the listbox (1).

(6) Confirms the required motor offset settings and hand it over to the "Load from SPEC" sub-menu.

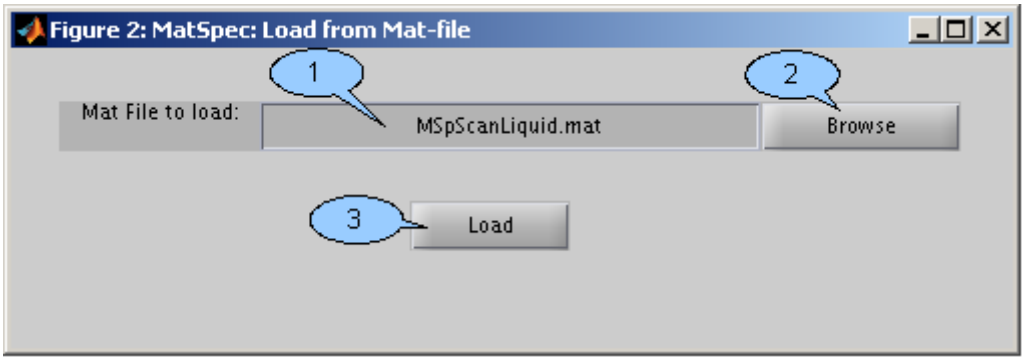

Figure 4: Sub-menu: Load from MAT-file.

3.3 Loading from MAT file

Data-sets can be saved from MatSpecGUI to a MAT-file (see Sec. 3.9), which is a Matlab binary file format for storing data. The sub-menu "load from MAT-file" (see Fig. 4) allows for loading a data-set stored in a MAT-file back to MatSpecGUI. The required content of the editable box and function of buttons of the sub-menu is as follows:

(1) A text box for input of path and name of the required MAT-file.

(2) A button to open a browse dialog to input a path and a name of the required MAT-file into the text box (1).

(3) A button to load the file to MatSpecGUI. Sub-menu is closed afterwards the main-menu is reopened.

3.4 Normalize intensities

The sub-menu "Normalize intensities" (see Fig. 5) allows to normalize intensity measured by a detector (usually a PSD) for beam attenuators, direct beam monitor, and to multiply them by a normalization factor.

A beam-line set-up is assumed here, where intensity of the primary beam is detected by a detector referred here as the monitor. Down-stream from the monitor there is an attenuator wheel or an attenuator box for attenuating the primary beam intensity by inserting required number of attenuation foils into the beam. Down-stream from attenuators there is sample and further from the x-ray source there is a detector detecting scattered intensity (see Fig. 6). Then, the normalized intensity is then calculated according to formula:

$$I_{\text{out}} = I_{\text{in}} * A^{n_{\text{att}}} / I_{\text{mon}} * C, \tag{1}$$

where I_{out}, I_{in}, A, n_{att}, I_{in}, and C are resulting normalized intensity, intensity measured directly with a detector, attenuation factor of one attenuation foil, number of attenuation foils inserted into the primary beam path, intensity on the monitor, and a multiplication factor.

The meaning of elements of the sub-menu in Fig. 5 is as follows:

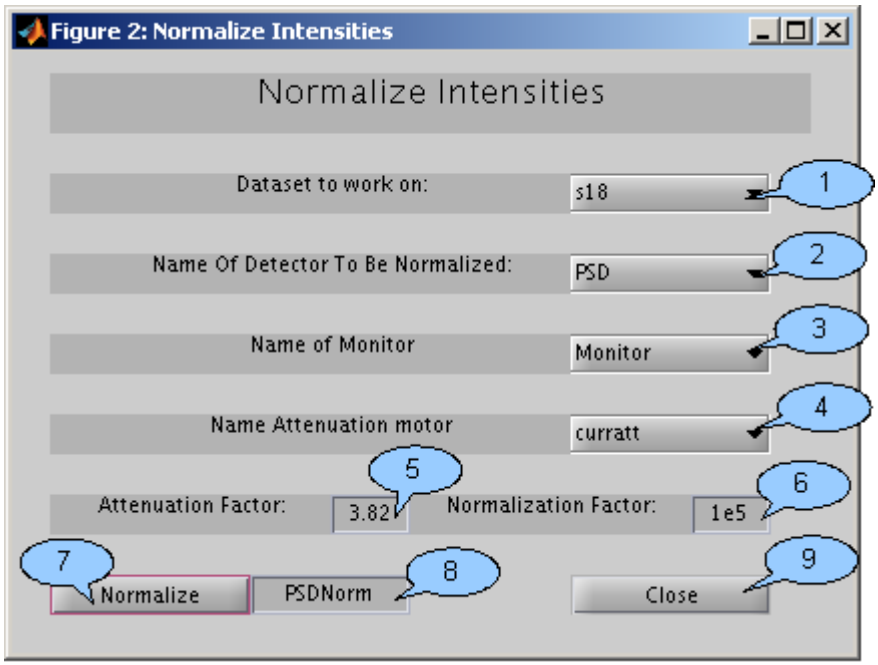

Figure 5: Sub-menu: Normalize intensities.

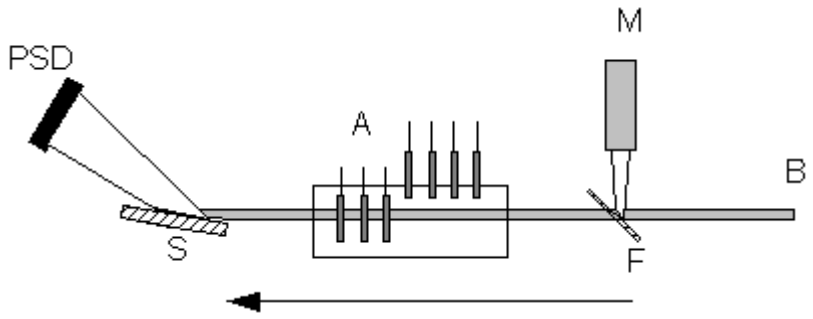

Figure 6: Setup assumed in the "Normalize intensities" sub-menu. Legend: B — the primary beam, F — a foil scattering a part of the primary beam to the monitor, A — an attenuator box with attenuation foils insertable into the primary beam, S — a sample, and PSD — detector. The arrow indicates down-stream direction.

(1) A data-set for which to normalize intensity.

(2) A name of the detector for which to normalize the intensity. Most commonly, user the variable `PSD` is normalized, where data from a 1D detector are stored. Data of the variable corresponds to I_{in} in Eq. (1).

(3) A name of the primary beam monitor. For ID10B beam-line at the ESRF, it is `Monitor.`

(4) A name of a motor or a variable where number of attenuators inserted into the beam n_{att} are stored. For ID10B beam-line at the ESRF, it is `curratt.`

(5) The attenuation factor A.

(6) The normalization factor C.

(7) A button to perform the normalization.

(8) A non-editable text-box with the name of output variable, where normalized intensity I_{out} will be stored.

(9) A button to close the sub-menu.

3.5 Plot rough data

The menu allows for a quick inspection of rough data as loaded from a SPEC file. In a resulting plot, intensity is color-scaled and it is plotted as a function of the motor position of a motor which was scanned and channels of PSD. The motor coordinates are on the x-axis and channels of PSD on the y-axis.

Elements of the sub-menu (see Fig. 7) are as follows:

(1) A data-set to deal with.

(2) A name of a motor to plot on the x-axis.

(3) A name of a variable with intensities to be plotted. Should be `PSD` or `PSDNorm.`

(4) A non-editable text box indicating the minimum value of intensity data to be plotted.

(5) A non-editable text box indicating the maximum value of intensity data to be plotted.

(6) A text box for setting the minimal intensity to be color-scaled.

(7) A text box for setting the maximal intensity to be color-scaled. (6) and (7) are to be used to enhance visibility of various intensity features in experimental data.

(8) A check-box switching between linear and logarithmic scaling of intensity.

(9) A check-box to indicate whether the resulting plot should be also printed into a file.

(10) A root of the filename of the output file.

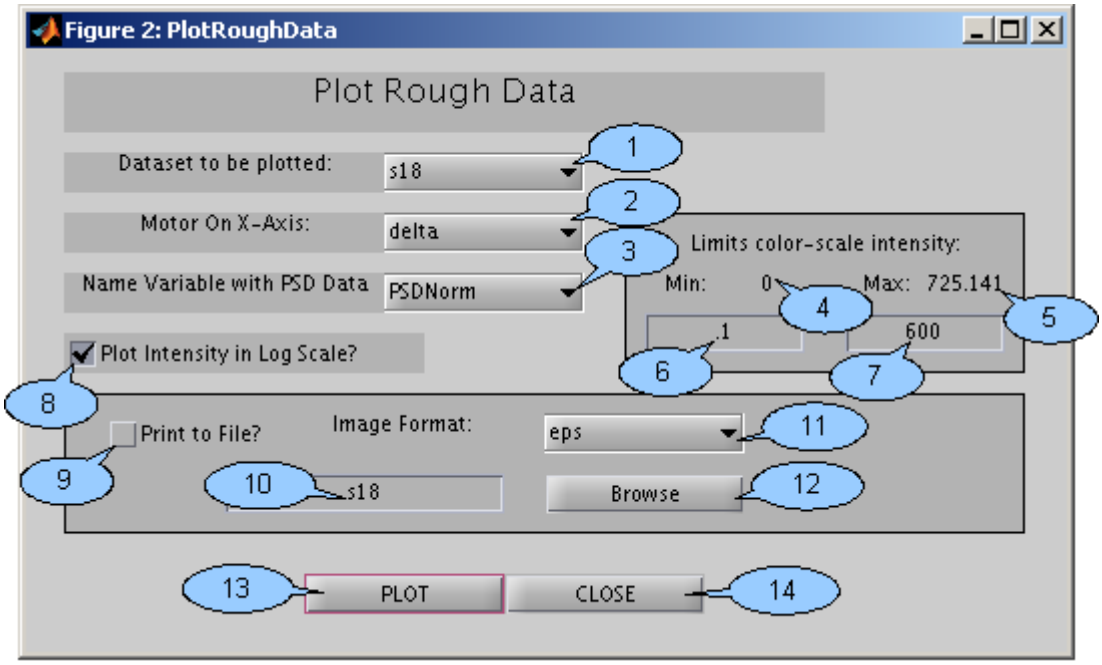

Figure 7: Sub-menu: Plot Rough Data.

(11) The required format of the output file.

(12) A button to open browse dialog to select the output file.

(13) A button to plot the data and to save the resulting plot, if (9) is checked on.

3.6 Transform to Q-space

The menu allows for transforming angular space coordinates to Q-space. MatSpecGUI comes with transformation functions for 3 experimental set-ups, which are described in Sec. 4. Commonly, it is sufficient to choose a transformation function corresponding to the experimental set-up in the bottom part of the sub-menu (see Fig. 8), load the function and its parameter list to the upper part of the menu and perform the transformation. In that case sub-menu elements (2), (4), and (7) are automatically filled according to the configuration.

In addition, the sub-menu also makes it possible to extend the list of available configurations in the bottom part of the menu. For this purpose, the elements (2), (4), and (7) are editable and their content can be saved as a new set-up to a MatSpecGUI configuration file via the button (5). Elements (6), (8), and (9) serve for modifying the list box (4).

Elements of the sub-menu (see Fig. 8) are as follows:

(1) A data-set to deal with.

(2) A name of transformation function to be applied. The text box is filled in when a configuration is loaded.

(3) A button to perform the transformation and to close the sub-menu.

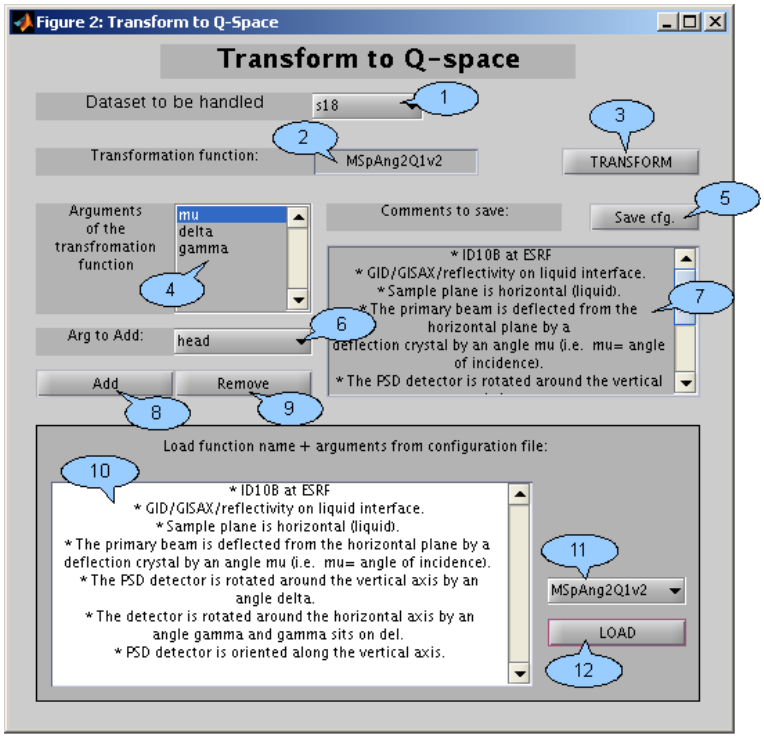

Figure 8: Sub-menu: Transform to Q-space.

(4) A list box containing a list of motor names in the order in which they are send as arguments to the transformation function (see Sec. 4.3).

(5) A button to save a new set-up parameters from elements (2), (4), and (7) to a configuration file.

(6) A drop-down list of variable's names, which can be added to the motor name list (4).

(7) A description of the current set-up.

(8) A button to add a motor name from (6) at the end of motor name list (4).

(9) A button to remove the selected motor name from the motor name list (4).

(10) A description of the experimental set-up currently selected for loading.

(11) A drop-down list with available transformation functions. One transformation function with different arguments can correspond to several experimental set-ups. So, one should watch the description in the text box (10).

(12) A button to load the configuration.

3.7 Map data to grid

The sub-menu allows to map data unevenly sampled in Q-space to a regular grid. Data-points (i.e., measured intensities) are averaged over areas of rectangles around

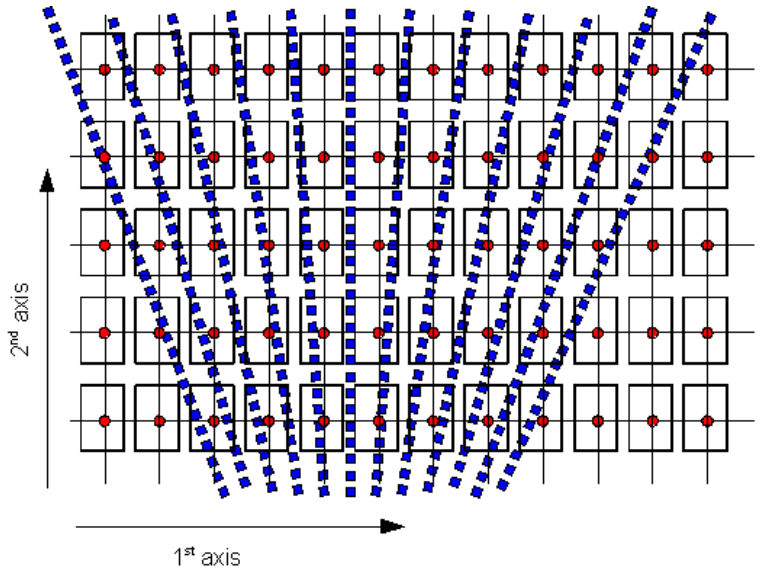

Figure 9: Illustration of the method of mapping data to a regular grid. Legend: circles — grid-points regularly space along axes 1 and 2, squares — measured data-points, thick rectangles demarcate averaging areas.

grid points (see Fig. 9) in the process of mapping. The mapping is performed by an external program `gridq1_01`. The mapped data are intended for plotting in the "Plot data" sub-menu. Elements of the sub-menu in Fig. 10 are:

(1) A data-set to deal with.

(2) Coordinates along the 1st-axis.

(3) Coordinates along the 2nd-axis.

(4) A variable with the data-points to be mapped. Usually it is `PSDNorm` or `PSD`.

(5) A definition of the grid points along the 1st-axis in the form `[the first point coordinate : step between points : the last point coordinate]`.

(6) A definition of the grid points along the 2nd-axis in the same form as in (5).

(7) The width of the averaging area along the 1st-axis.

(8) The width of the averaging area along the 2nd-axis.

(9) A non-editable text box with a name of the output variable which will contain positions of grid points along the 1st-axis after the mapping.

(10) The same as in (9) for the 2nd-axis.

(11) A name of the output variable with mapped data.

(12) A button to perform an automatic estimation of the grid parameters, i.e., grid point positions and widths of the averaging areas. The results are filled in editable boxes (5)–(8). Results of mapping of data should be inspected in

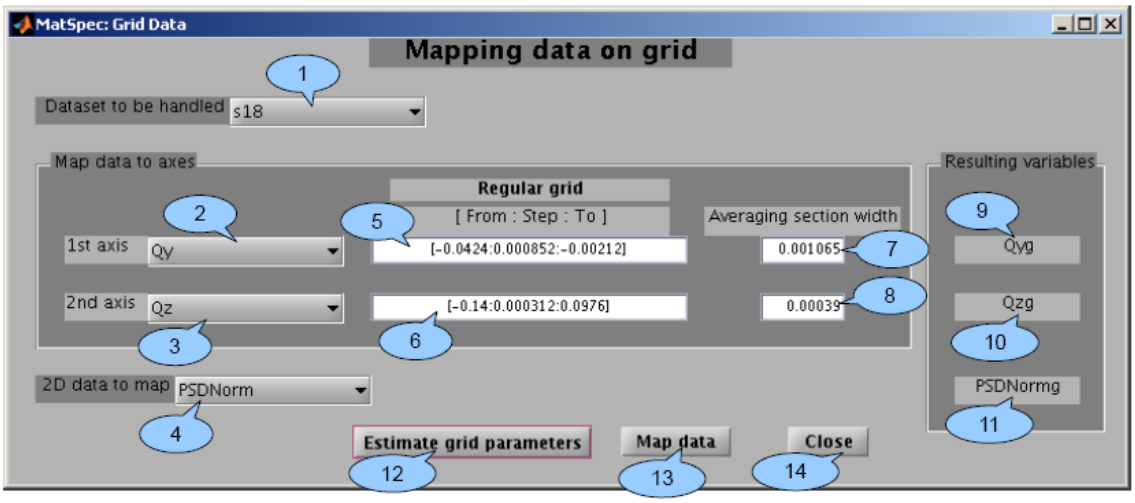

Figure 10: Sub-menu: Mapping data to grid.

the "Plot data" sub-menu. If the results are unsatisfactory, it may be needed to tune widths of averaging areas and grid-points positions in boxes (5)–(8) manually.

(13) Button to perform the mapping.

(14) Button to close the sub-menu.

3.8 Plot data

The primary function of the sub-menu "Plot data" (see Fig. 11) is to plot data in Q-space. Intensity is plotted as a function of two coordinates. The plotting can be done in two ways. First, data mapped to a grid can be plotted with continuous color-scaling via the Matlab function `imagesc` (see Fig. 12(a)). Second, plotting data as a filled-color-plot via the Matlab function `contourf` (see Fig. 12(b)) is supported. The former method allows for a more just image of the intensity distribution. The latter method, in contrast to the former, provide a plot where for example axes could be changed to the logarithmical scale.

Elements of the sub-menu in Fig. 10 are:

(1) A data-set to deal with.

(2) Data to plot on the horizontal axis.

(3) Data to plot on the vertical axis.

(4) Data to be color-scaled. Usually, PSDNormg is plotted color-scaled.

(5) A label on the horizontal axis.

(6) A label on the vertical axis.

(7) A label of the color-bar.

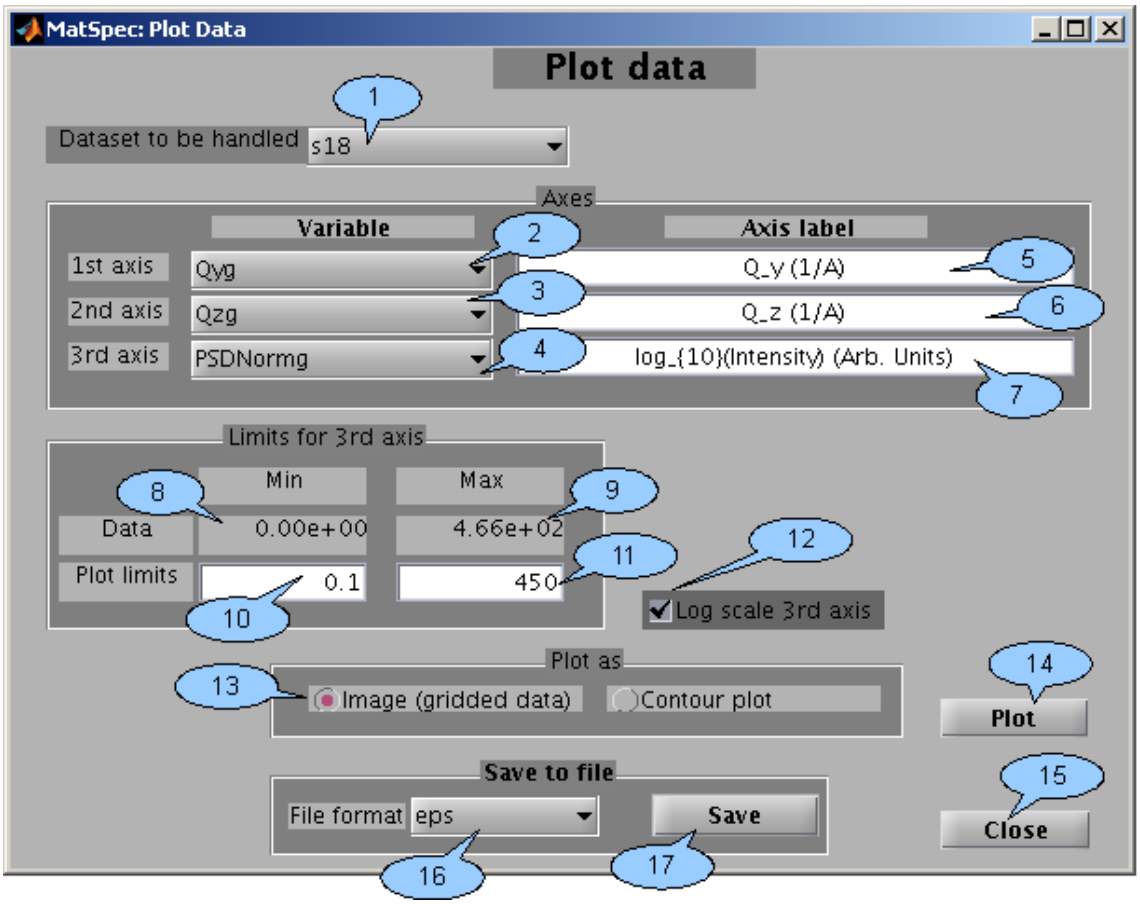

Figure 11: Sub-menu: Plot data.

(a) (b)

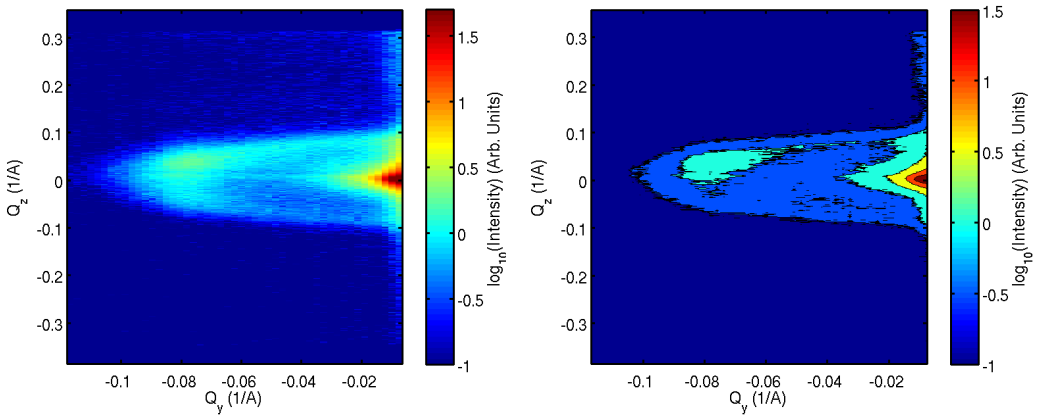

Figure 12: Small angle x-ray scattering data from thiol protected Au nano-particles ordered at water/toluene interface. Data were plotted via (a) continuous color-scaling, (b) filled-contour-plot.

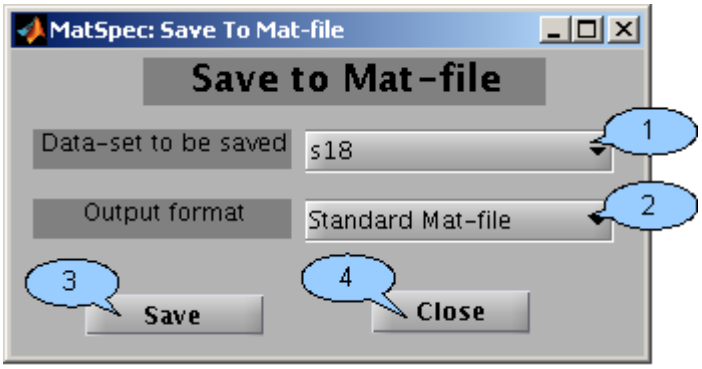

Figure 13: Sub-menu: Save to MAT-file.

(8) A non-editable text box indicating minimum value of intensity data to be plotted.

(9) A non-editable text box indicating maximum value of intensity data to be plotted.

(10) A text box for setting minimal intensity to be color-scaled.

(11) A text box for setting maximal intensity to be color-scaled. (10) and (11) are to be used to enhance visibility of different features in experimental data.

(12) A check-box switching between linear and logarithmic scaling of data on the 3^{rd}-axis.

(13) A radio buttons to switch between a continuous color-scaling and a filled-contour-plot plotting style.

(14) A button to perform the plotting.

(15) A button to close the sub-menu.

(16) A drop-down list with possible formats of a output image file.

(17) A button to save the current plot to a image-file. A dialog-window to select the path and filename of the file is opened after pressing the button.

3.9 Save to MAT-file

The sub-menu allows to save a selected data-set to a MAT-file, i.e., to Matlab format file designed for storage of data.

Elements of the sub-menu in Fig. 13 are:

(1) A data-set to deal with.

(2) An output format. For Matlab version newer than 7.3, inclusively, there is a choice to save in the HDF5 format.

(3) A button to open a dialog to choose a path and a filename for the output file and to save the file.

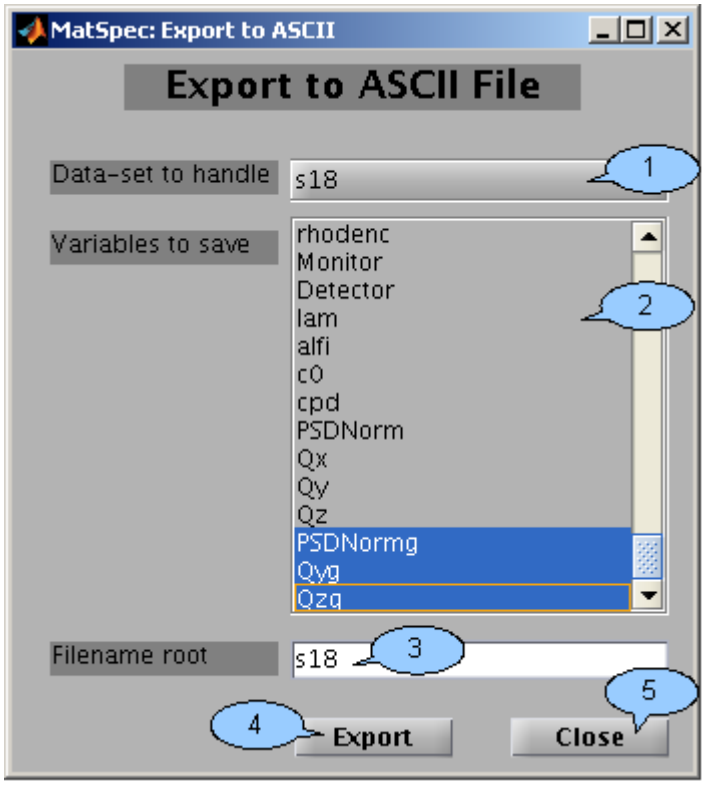

Figure 14: Sub-menu: Save to ASCII-file.

(4) A button to close the sub-menu.

3.10 Save to ASCII-file

The sub-menu allows for saving the data in the ASCII format. Supported types of variables to be saved are vectors, matrices (e.g., position of a motor or data from a PSD) and strings (for saving headers of scans). If PSD data are saved, data from subsequent PSD shots are stored row by row in a matrix in an output file, i.e., different rows are correspond to different positions of a motor which was scanned. The files are composed of a filename root, entered by the user, a name of the exported variable, and the extension `dat` as `root_VariableName.dat`.

Elements of the sub-menu in Fig. 13 are:

(1) A data-set to deal with.

(2) A list-box allowing the user to choose one or more variables to be saved. Multiple variables can be chosen by clicking on variable names while holding the `Ctrl` button.

(3) A root of the output filename.

(4) A button to open the dialog for entering a output-path. After finishing the dialog the output file(s) are saved.

(5) A button to close the sub-menu.

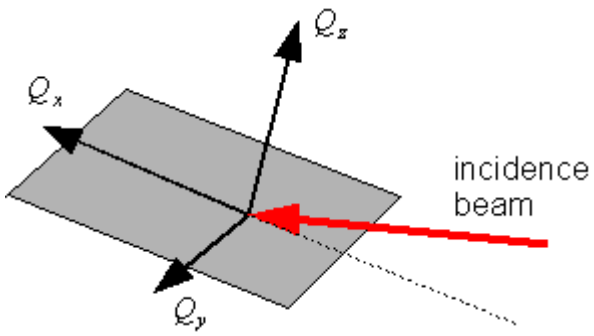

Figure 15: Orientation of axes in the Q-space with respect to the sample surface and to the incidence beam.

4 Experimental set-ups and transformation functions

MatSpecGUI comes with predefined functions for transformations from angular space to Q-space. This section describes experimental set-ups at the ID10B beamline of the ESRF corresponding to the functions. Additionally, structure of transformation functions and their linking to the graphical user interface is described here.

For all types of set-ups, the Q-coordinates axes are oriented as follows (see Fig. 15): the Q_x, Q_y axes lie in the plane of the sample surface. Q_z axis is perpendicular to the sample surface. Q_x axis lies in the plane of incidence of the beam. Q_y axis is perpendicular to Q_x axis.

4.1 Scattering at liquid/air and liquid/liquid interfaces

In this type of set-ups the incoming beam is deflected from the horizontal plane. There are two alternatives of the beam deflection at the ID10B beamline of the ESRF. First, beam can be deflected by a deflection crystal in the experimental hutch by rotating a deflection crystal, which is tilted to the Bragg condition, around the beam. The set-up allows for varying the incidence angle of the beam. Second, the beam can be tilted by mirrors in the optical hutch. In that case, the incidence angle is fixed during the experiment. The transformation MatSpecGUI function for this type of set-ups is `MSpAng2Q1v2.m`. The transformation function covers scattering experiments both in the coplanar geometry (i.e., specular reflectivity and coplanar diffraction) and in the non-coplanar geometry (i.e., GID and GISAXS).

4.1.1 Beam deflected by a deflection crystal

The corresponding set-up is shown in Fig. 16. The primary beam impinges on a sample at an incidence angle μ. PSD detector is oriented along the vertical plane. A detector arm can be rotated by an angle δ around the vertical axis and by an angle γ around the horizontal axis. The motor γ sits on the motor δ.

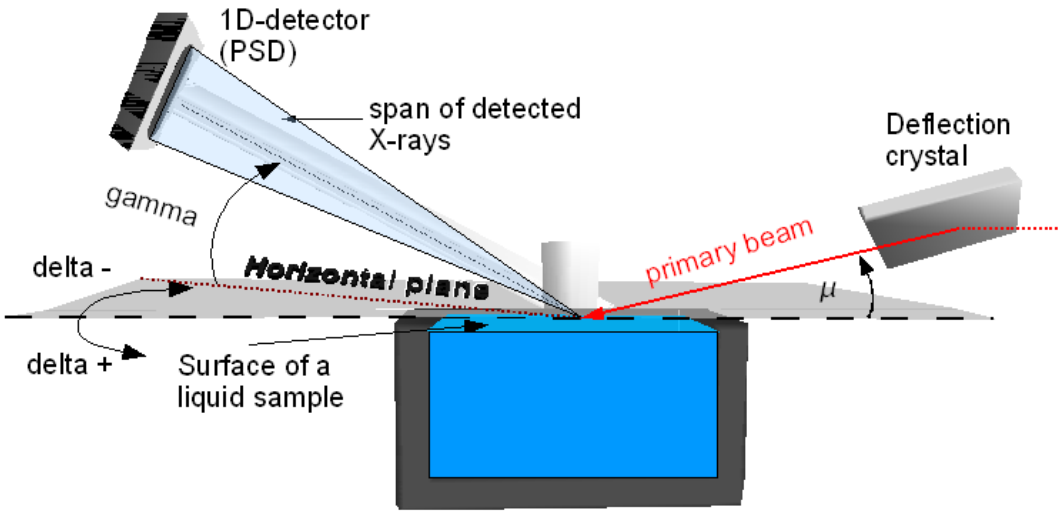

Figure 16: Set-up for scattering experiments at liquid/air and liquid/liquid interfaces at the ID10B beamline of the ESRF.

4.1.2 Beam deflected by a mirror

The beam incidence angle α_i is fixed and it is determined by a tilt angle of a x-ray mirror in the optical hutch. Since the incidence angle value is not explicitly written in the SPEC file, one needs to enter it manually during loading a scan (see Sec. 3.2). The PSD is oriented along the vertical plane. Detector arm is rotated by an angle δ around the vertical axis and by an angle γ around the horizontal axis. The motor γ sits on the motor δ.

4.2 Scattering on a solid sample on the horizontal sample stage

For this set-up, a solid sample resides on the horizontal stage of the goniometer (see Fig. 17), which is tilted by an angle χ around the horizontal axis perpendicular to the primary beam. PSD detector is oriented along the vertical plane. Detector arm can be rotated by an angle δ around the vertical axis and by an angle γ around the horizontal axis. The motor γ sits on the motor δ. The transformation MatSpecGUI function for this type of set-ups is `MSpAng2Q2v2.m`. The transformation function covers scattering experiments both in the coplanar geometry (i.e., specular reflectivity and coplanar diffraction) and in the non-coplanar geometry (i.e., GID and GISAXS).

4.3 Structure of transformation functions and their linking to GUI

Transformations from angular space to Q-space in MatSpecGUI are ensured by Matlab functions. To link a transformation function with MatSpecGUI the first argument of the transformation function has to be a structure with SPEC data (i.e., a data-set, see Sec. 5). Next, the function should accept arbitrary number of string

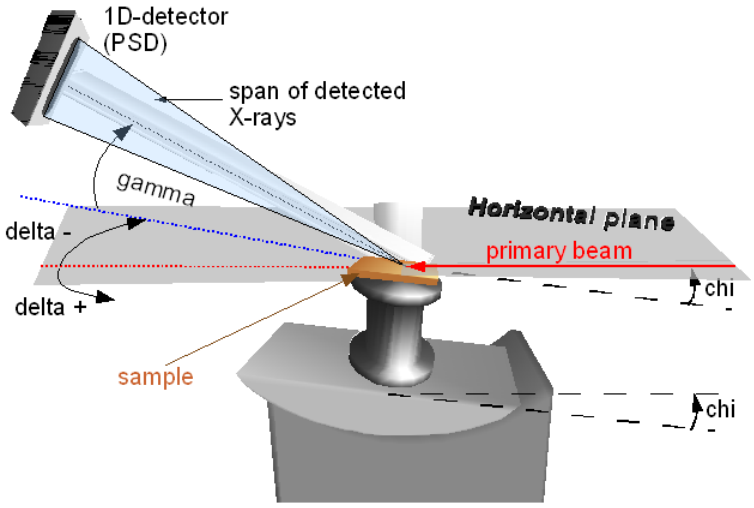

Figure 17: Set-up for experiments on solid samples residing on the horizontal stage of the goniometer.

variables with motor names (usually motors which rotate PSD and the sample). The output of the transformation function should be a structure, which contains original fields of the input structure and additionally reciprocal space coordinates Q_x, Q_y, and Q_z corresponding to measured data-points. The calculated fields with reciprocal space coordinates are matrices of the same size as the PSD data, i.e., Q-coordinates are calculated for each pixel of the PSD and each PSD shot.

As an example, we list here the transformation function `MSpAng2Q1v2.m`, performing transformation from the angular space to the Q-space for scattering experiments at liquid/liquid or liquid/air interfaces (see Sec. 4.1).

```
function scn= MSpAng2Q1v2(sco, NameAi, NameDel, NameGam)
% sco - input structure from MatSpecGUI
% NameAi - a name (i.e., a string) of motor (or a varible)
%          for the beam tilt
% NameDel - a name of motor for the inplane detector rotation
% NameGam - a name of motor for the vertical detector rotation
% scn - a sctructure returned to MatSpecGUI at the function exit

d2r=pi/180;
% alpha_i, delta, gamma in radians
ai=sco.(NameAi)*d2r;
del=sco.(NameDel)*d2r;
gam0=sco.(NameGam)*d2r;

% PSD information, cpd - channels per degree, c0 - central channel
cpd=sco.cpd;
c0=sco.c0;
channels=sco.channs;
nChannels=size(sco.PSD,2);
```

```
% Ensure correct size of vectors
nai=length(ai);
ndel=length(del);
ngam0=length(gam0);
nmax=max([nai,ndel,ngam0]);
if (nai<nmax),
if (nai~=1) error('Wrong number of scan points for %s\n',NameAi);
end;
ai=repmat(ai,nmax,1);
end;
if (ndel<nmax),
if (ndel~=1) error('Wrong number of scan points for %s\n',NameDel);
end;
del=repmat(del,nmax,1);
end;
if (ngam0<nmax),
if (ngam0~=1),
error('Wrong number of scan points for %s\n',NameGam);
end;
gam0=repmat(gam0,nmax,1);
end;

% Transform PSD channel's positions to radians
gamd=atan((channels-c0)*d2r/cpd);

% Calculate the unit vector coordinates for the exit wave
csgam0=cos(gam0); sngam0=sin(gam0);
csgamd=cos(gamd); sngamd=sin(gamd);
csgam=csgam0*csgamd-sngam0*sngamd;
sngam=csgam0*sngamd+sngam0*csgamd;
csdel=cos(del); sndel=sin(del);

% Calculate the unit vector coordinates for the incoming wave
csai=cos(ai); snai=sin(ai);

% Ensure proper size of the matrix with incoming wavevector
% components
if size(csai,2)==1,
csai=repmat(csai,1,nChannels);
snai=repmat(snai,1,nChannels);
end;

% Calculate the scattering vector components
Qx=repmat(csdel,1,nChannels).*csgam-csai;
Qy=repmat(sndel,1,nChannels).*csgam;
Qz=sngam+snai;
```

```
% Copy the input structure to the output structure
scn=sco;
K0=2*pi/sco.lam;
% Add scattering vector components to to the output structure
scn.Qx=Qx*K0;
scn.Qy=Qy*K0;
scn.Qz=Qz*K0;
```

To perform the transformation for experimental data measured in the set-up shown in Fig. 16, the function above would be called in Matlab as
`s18=MSpAng2Q1(s18,'mu','delta','gamma').`
Here `s18` is a data-set containing data from a SPEC file loaded via MatSpecGUI.

The calling of transformation functions is ensured by the sub-menu "Transform to Q-space" (see Fig. 8) in MatSpecGUI. The elements (2) and (4) of the sub-menu have to be filled with the transformation function name (e.g., MSpAng2Q1v2) and name of the motors in the order as they are submitted to the function, respectively.

The transformation function listed above perform calculation of Q-coordinates (see Fig. 15) according to the formula:

$$\mathbf{Q} = K_0(\mathbf{n}_f - \mathbf{n}_i), \text{ where} \tag{2}$$

$K_0 = 2\pi/\lambda$, $\mathbf{n}_\mathrm{f}$, and $\mathbf{n}_\mathrm{i}$ are size of the vacuum wavevector of x-rays, and unit vectors in the directions of the outgoing and incoming rays, respectively. The unit vector in the direction of the outgoing ray $\mathbf{n}_\mathrm{f}$ is calculated as

$$\mathbf{n}_\mathrm{f} = \hat{M}_2 \cdot \hat{M}_1 \cdot \mathbf{n}'_\mathrm{f}, \tag{3}$$

where

$$\mathbf{n}'_\mathrm{f} = \begin{pmatrix} \cos\gamma_\mathrm{d} \\ 0 \\ \sin\gamma_\mathrm{d} \end{pmatrix}, \tag{4a}$$

$$\hat{M}_1 = \begin{pmatrix} \cos\gamma & 0 & -\sin\gamma \\ 0 & 1 & 0 \\ \sin\gamma & 0 & \cos\gamma \end{pmatrix}, \tag{4b}$$

$$\hat{M}_1 = \begin{pmatrix} \cos\delta & -\sin\delta & 0 \\ \sin\delta & \cos\delta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \text{ and} \tag{4c}$$

$\gamma_\mathrm{d} = \arctan((c - c_0)/\mathrm{cpd})$ is an angle between the channel c_0 to which direct beam impinges when the PSD is in the origin of the coordinate system and the primary beam is not tilted, i.e., for $\alpha_\mathrm{i} = 0$, $\delta = 0$, $\gamma = 0$, and the channel c, for which the Q-coordinates are calculated. The vertex of the angle γ_d lies in the center of the rotation of the goniometer. Hereafter, cpd is number of channels between two rays intersecting the PSD and spanning an angle of 1° and originating in the center of the rotation of the goniometer. Finally, $\mathbf{n}_\mathrm{i} = (\cos\alpha_\mathrm{i}, 0, \sin\alpha_\mathrm{i})^\mathrm{T}$ is the unit vector in the direction of the outgoing ray and α_i is the angle of incidence of the primary beam.

5 Data storage and data-set manipulation outside MatSpecGUI

5.1 Data storage

Data loaded to MatSpecGUI from a SPEC file are stored in variables of the type structure. They are referred as data-sets within MatSpecGUI. Normally, data-sets are visible only within MatSpecGUI and not in the interactive Matlab console. The names of the fields of a data-set (i.e., a structure) are names of instrument motors and of detectors as they are read from a SPEC file. The fields contain values of corresponding instrument elements during a scan. Additionally, fields `head`, `c0`, `cpd`, `channs`, and `PSD` are created in the moment of loading of data. The fields contain the head(s) of the loaded scans, index of the channel of origin of the PSD, number of channels corresponding to a PSD rotation by 1° around an axis perpendicular to the PSD orientation, indices of PSD channels from which data were loaded and PSD data (i.e., intensities in individual channels for each shot of the PSD), respectively. The values are scalars, vectors or matrices of the type float or single, if possible, except for the field `head,` which is of the type string of chars. New fields are added to the data-sets as they are produced by operations within MatSpecGUI. In the `PSD` field, individual shots of the PSD are stored along the rows of the matrix.

As an example, a contents of a data-set (i.e., MatSpecGUI structure) is listed here:

```
s18 =
        head: [1x876 char]
       delta: [58x1 single]
       gamma: 0.02
       omega: 0
       theta: 0
          mu: 0.02
       sigma: 9.765
      sigmat: 9.8591
          xt: -8.0038
          zt: 1.4645
         zt1: 92.105
         thd: 4.9245
        chid: 1.8151
        rhod: -0.11691
          xd: -0.88
          yd: 0.023
        PhiD: 1.2018
          zd: 1.6153
        arcf: -0.02
          zf: 2.0103
        att0: 0
      gslitT: 10.5
```

```
 gslitB: 10
 gslitR: 0.019975
 gslitF: 0.019975
gslitho: -2.2204e-16
gslithg: 0.03995
gslitvo: 0.25
gslitvg: 20.5
  phigH: 0.15
  chigH: 0.15
    xgH: 0
    ygH: 0
    zgH: -0.2709
  phigV: 0
  chigV: 0
    xgV: 0
    ygV: 0
    zgV: -130
   s0hg: 0.30004
   s0ho: 0.13957
   s0vo: 0.35126
   s0vg: 0.0099899
 slit1T: 0.22348
 slit1B: 0.17614
 slit1R: 4.1828
 slit1F: 5.8172
slit1ho: -0.81723
slit1vo: 0.023674
slit1hg: 10
slit1vg: 0.39962
slit2hg: 10
slit2vg: 9.0001
    TRT: -4.7448
   chia: 0
    tha: -14.351
   ttha: -28.215
  picou: 0
  picod: 0
     pi: 0
   rien: 1
 trough: 0.0024405
    hv1: 0
 channs: [1x1024 double]
    PSD: [58x1024 single]
      H: 0.0009494
      K: [58x1 single]
      L: [58x1 single]
  Epoch: [58x1 single]
```

```
    Seconds: 150
     Ion_m1: [58x1 single]
     Ion_m2: 0
       all0: [58x1 single]
       psd0: [58x1 single]
       dir0: [58x1 single]
      refl0: [58x1 single]
    yoneda0: [58x1 single]
       foil: [58x1 single]
      srcur: [58x1 single]
    curratt: 0
    detcorr: [58x1 single]
      ratio: [58x1 single]
     mcaLt0: [58x1 single]
     ccdint: 0
     thdenc: 0
    rhodenc: 0
    Monitor: [58x1 single]
   Detector: [58x1 single]
        lam: 1.55
       alfi: []
         c0: 541
        cpd: 268
```

5.2 Data-set manipulation outside MatSpecGUI

Data from MatSpecGUI can be manipulated also outside MatSpecGUI within Matlab in the interactive mode. In this way, a large variety of mathematical operations over the data are possible by use of the Matlab programming language. Data-sets are made visible outside MatSpecGUI by making them global in the interactive session by the command `global`, i.e., use

```
>> global VariableName.
```

The fields of the data-sets are referred via the standard dot notation `DataSetName.FieldName`. Results of operations over data can be further used within MatSpecGUI, if they are stored as a new field in the original MatSpecGUI data-set. In other words, since a data-set is made global, MatSpecGUI see results of operations on them made in the interactive Matlab mode.

As an example, to operate with the in-plane components of Q-coordinates Q_r of a data-set `s18` within MatSpecGUI, one can write a script like

```
% script to transform Qx, Qy to inplane coordinates
global s18
% calculate distance in cartesian coordinates, .^ means make quadrat of
% a matrix element by element
s18.Qr=sqrt(s18.Qx.^2+s18.Qy.^2)
```

to a file, e.g., `inplane_s18.m`, in the current Matlab directory. After execution of the script in the Matlab interactive session

```
>> inplane_s18,
```

the field `Qr` is accessible also in a MatSpecGUI session.

License.