

SXRDcalc

February 26, 2007

The program `sxrdcalc` was developed to analyse the surface X-ray diffraction data. It can be used to calculate X-ray diffraction intensities from a surface layer, to calculate a Patterson map of experimental data, and to fit a model structure to experimental data using either a Levenberg-Marquardt least-squares fit or simulated annealing.

The program has been written in the C programming language. It uses routines from the GNU scientific library (GSL) for the least-squares fitting. It should be possible to compile it with any C compiler. Tests were made with the GNU `gcc` compiler (versions 3.4.3 and 4.1.2) and the Intel `icc` compiler (version 8.1).

The program is started by typing

```
sxrdcalc inputfile > logfile
```

on the command line. `inputfile` is the name of the main input file, which contains the parameters and the names of the files defining the structural models etc. This is explained in the next section. The output of the program including the fit results like optimized parameter values are printed to standard output only, usually one would want to save the output to a file `logfile` like above.

1 Main input file

The main input file contains the information telling `sxrdcalc` what to do. The input file is a simple text file, each line contains an assignment to a variable in the following form:

```
variable = value
```

The possible variables and values are explained below. The input file is case-sensitive. If the same variable is defined more than once, then the last definition is used. Variables that are not needed, empty lines and additional whitespace are ignored. A `#` anywhere on a line introduces a comment, everything to the right of the `#` will be ignored.

The most important variable is the `do` variable. This tells the program which task it should do. For example,

```
do = patterson
```

means that the Patterson map should be calculated from experimental intensities. Possible values for `do` are:

<code>patterson</code>	– Calculate Patterson map.
<code>theo_range</code> and <code>l</code> .	– Calculate structure factors for a range of h , k , and l .
<code>theo_data</code> as in input file.	– Calculate structure factors for same h , k , and l as in input file.
<code>ls_fit</code>	– Structure refinement by least-squares fit.
<code>sa_fit</code>	– Structure refinement by simulated annealing fit.

The other variables are explained in the following.

The names of the additional input files are defined with:

<code>f_in_file</code>	– Input file for structure factors.
<code>struc_in_file1</code>	– Input file for model structure (domain 1).
<code>struc_in_fileN</code>	– Input file for model structure (domain N).
<code>fit_struc_in_file1</code>	– Structure input file for structure refinement (domain 1).
<code>fit_struc_in_fileN</code>	– Structure input file for structure refinement (domain N).

Output files are defined with:

<code>patterson_file</code>	– Output file for Patterson map.
<code>f_out_file</code>	– Output file for structure factors.
<code>fit_coord_out_file1</code>	– Output file for fitted coordinates (domain 1).
<code>fit_coord_out_fileN</code>	– Output file for fitted coordinates (domain N).

The maximum number of domains with different structures is 20. The intensities of the domains are added to get the total intensity. The number of domains and relative occurrence are defined with the following variables:

<code>nr_domains</code>	– Number of domains.
<code>domain_occ1</code>	– Occupancy for domain 1.
<code>domain_occN</code>	– Occupancy for domain N .

For the calculation of the Patterson map, the symmetry of the surface has to be given with:

<code>symmetry</code>	– Symmetry of the surface used for Patterson map.
-----------------------	---

If `do` is set to `theo_range`, then the h , k and l for which the structure factors are calculated are selected with:

<code>minh</code>	– Minimum h
<code>maxh</code>	– Maximum h
<code>steph</code>	– Step size for h
<code>mink</code>	– Minimum k
<code>maxk</code>	– Maximum k
<code>stepk</code>	– Step size for k

<code>minl</code>	– Minimum l
<code>maxl</code>	– Maximum l
<code>stepl</code>	– Step size for l

The structure factors that are calculated when `do` is `theo_range` or `theo_data` are scaled with the value of

<code>theo_scale</code>	– Scale for structure factors.
-------------------------	--------------------------------

For the structure refinement, one should use several different starting parameters to make sure that the global minimum is found. The starting parameters can be varied randomly with:

<code>n_fit</code>	– Number of fits
<code>par_var_width</code>	– Width of the gaussian function for varying the parameters
<code>rng_seed</code>	– Seed for random number generator

For the least-squares structure refinement the following input variables can be used:

<code>max_iteration</code>	– Maximal number of iterations.
<code>delta_abs</code>	– Fit stops if the changes Δr_i of the positions r_i is smaller than <code>delta_abs</code>

The simulated annealing structure refinement is controlled with the following variables:

<code>n_tries</code>	– The number of points to try for each step
<code>iters_fixed.T</code>	– The number of iterations at each temperature
<code>step_size</code>	– The maximum step size in the random walk
<code>sa_k</code>	– Boltzmann K
<code>t_initial</code>	– Initial temperature
<code>mu_t</code>	– damping factor for temperature
<code>t_min</code>	– Minimum temperature
<code>sa_seed</code>	– Seed for random number generator

Information about the current fit can be printed at each step of the iteration or only at the end of each fit:

<code>print_intermediate</code>	– Print intermediate information at each step (yes or no).
---------------------------------	--

Comments can be inserted at the beginning of the output file with the following variable:

<code>comment</code>	– Comment for output file. This uses the whole line.
----------------------	--

2 Other input files

The format of the input file for the experimental structure factors for the calculation of the Patterson map or the structure refinement is the following:

```
h k l F σ
h k l F σ
...
...
```

h , k and l are the reciprocal lattice coordinates of the reflection, F is the structure factor and σ the standard deviation of F .

For the calculation of the structure factors of a model structure, the atomic positions are given in a file with the following structure:

```
Comment
a b c α β γ
atomsymbol r1 r2 r3 DWfactor occupancy
atomsymbol r1 r2 r3 DWfactor occupancy
...
...
```

In the first line a comment, for example the name of the structure, can be given. The second line defines the unit cell. a , b and c are the length of the unit cell vectors and α , β and γ the angles of the unit cell. The definition of the unit cell should correspond to the reciprocal space unit cell used for the structure factors. The following lines define the atoms. *atomsymbol* is the chemical symbol of the atom, e.g. Si. r_1 , r_2 and r_3 are the reduced coordinates of the atom in the unit cell defined in the second line. *DWfactor* is the Debye-Waller factor of the atom (in Å²). *occupancy* is the fractional occupancy of the atom. The occupancy is optional, if it is not present, an occupancy of 1.0 will be assumed.

The input file for the structure refinement defines the atomic positions and, in addition, the fit parameters for each atom. The fit parameters can be for each atom three different displacement vectors, a isotropic Debye-Waller factor and the occupancy of the atom. An overall scale factor is also included into the fit automatically. The format of the structure refinement input file is:

```
Comment
a b c α β γ
pos atomsymbol r1 r2 r3 DWfactor occupancy
displ1 parnumber dr1 dr2 dr3
displ2 parnumber dr1 dr2 dr3
displ3 parnumber dr1 dr2 dr3
dw_par parnumber scale
occ parnumber
pos atomsymbol r1 r2 r3 DWfactor occupancy
displ1 parnumber dr1 dr2 dr3
displ2 parnumber dr1 dr2 dr3
displ3 parnumber dr1 dr2 dr3
dw_par parnumber scale
occ parnumber
...
```

```

...
start_par 0 value
start_par parnumber value
...

```

The first two lines are the same as for the input file for a structural model above. The `pos` keyword indicates the definition of an atom. The meaning of each term is the same as in the input file for a structural model. `displ1`, `displ2` and `displ3` give the displacements vectors that are allowed for that atom. *parnumber* is the number of the fit parameter assigned to this displacement. dr_1 , dr_2 and dr_3 are the components of the displacement vector in the coordinate system defined in the second line of the input file. `dw_par` assigns the fit parameter with number *parnumber* to the Debye-Waller factor of the atom. The Debye-Waller factors are multiplied by *scale* in the calculation of the structure factors for the structure refinement. Setting *scale* to 10 or 100 reduces the importance of the Debye-Waller factor for the fit, which improves the convergence behavior of the fit. `occ` assigns the occupancy of the atom to the parameter with the number *parnumber*. For each atom only a line with the `pos` keyword has to be given, the other keywords are optional. The *parnumber* gives the number of the fit parameter used for each keyword. *parnumber* can be the same for different atoms, for example if the atoms are symmetry-related or if they are expected to have a similar Debye-Waller factor. Starting values for the fit parameters are given with the `start_par` keyword. The fit parameter with number 0 is always a scale factor between the experimental and calculated intensities.

The symmetry of the structure is not included automatically in the fit. The symmetry constraints have to be given explicitly by using appropriate displacement vectors. For example, if an atom sits on a high-symmetry site and is only allowed to move in the direction perpendicular to the surface (r_3 direction), then an appropriate displacement vector would be

```

pos Si 0.5 0.5 0.0 0.0 1.0
displ1 0.0 0.0 1.0

```

The symmetry relations between two atoms can be incorporated by giving them appropriate displacement vectors with the same parameter number. For example,

```

pos Si 0.7 0.5 0.0 0.0 1.0
displ1 1 1.0 0.0 0.0
displ2 2 0.0 1.0 0.0
displ3 3 0.0 0.0 1.0
pos Si 0.3 0.5 0.0 0.0 1.0
displ1 1 -1.0 0.0 0.0
displ2 2 0.0 -1.0 0.0
displ3 3 0.0 0.0 1.0

```

constraints the positions of the two atoms to be point symmetric around (0.5 0.5) in the surface plane with the same height.

3 Output

The output file from the calculation of the Patterson map looks like this:

```
r1 r2 value
r1 r2 value
...
...
```

r_1 and r_2 are the reduced coordinates in the unit cell define in the structure input file, and *value* is the value of the Patterson function.

The output file of the structure factors calculated for a model structure has the following format:

```
comment
h k l F 0.1
h k l F 0.1
...
...
```

with h , k and l the reciprocal lattice coordinates of the reflection, and F the structure factor. The *comment* is the one defined in the main input file. The output file of the structure factors can be used as an input file for e. g. calculation of the Patterson map, if the comment in the first line is deleted.

The output of the structure optimization is printed to standard output, so this should be saved to some file. For each fit, χ^2 and R are printed, then the optimized parameters are printed. The errors for each parameter are calculated from the square root of the diagonal terms of the covariance matrix. The complete covariance matrix is printed as well.

An output file with the optimized atomic positions is made for the structural optimization as well. The format is the same as the input file for the calculation of the structure factors. The output file corresponds to the last fit, if several fits are made at the same time.