

Version 2.3.4

User Manual

January 1, 2025

Wolfgang Sturhahn



<https://www.nrixs.com>

support@nrixs.com

Contents

1	License information	4
2	Getting started	4
2.1	The GUI	4
2.2	The CLI	5
3	Using the GUI	5
3.1	Open an existing project	5
3.2	Create a new project	6
3.3	Find a project	6
3.4	Fit parameters	6
4	The CLI: standard input files	6
4.1	Syntax rules	6
4.2	Assigning fit parameters	6
4.2.1	Dependent fit parameters	7
4.3	Specifying priors	7
5	Equations-of-state: seos	8
5.1	Thermal Equation of State	12
5.1.1	Grüneisen models	14
5.2	Spin Equation of State	14
5.3	Parameter fitting	15
5.3.1	Fisher information	16
5.3.2	Temperature errors	16
5.4	Bulk modulus	17
5.5	Volume	17
5.6	p-T array (option A)	18
5.7	Covariance matrix (option C)	18
5.8	Spin entropy (option E)	18
5.9	Spin free energy (option F)	19
5.10	Specific heat (option H)	19
5.11	Internal energy (option I)	19
5.12	Error ellipse (option L)	19
5.13	f-F plot (option M)	20
5.14	Thermal and spin pressure (option P)	20
5.15	Spin state populations (option S)	21
5.16	Thermal expansion (option T)	21
5.17	Unpaired electrons (option U)	21
5.18	$K_{\beta'}$ intensity (option X)	21
5.19	Crystal field splitting and spin-pairing energy (option Z)	21
5.20	Free energy landscape (option Z)	22

6	Melting and diffusion: simx	23
6.1	Temperature profile	26
6.2	Temperature from spectral analysis	26
6.3	X-ray profile	27
6.4	Nuclear forward scattering signal	27
6.5	NRIXS signal	28
6.6	Parameter fitting	29
6.6.1	Fisher information	30
6.7	Output files	30
6.7.1	Contours	30
6.7.2	X-ray intensity distributions	30
6.7.3	Temperature	30
6.7.4	NFS related	31
7	Linear mixing model: smix	32
7.1	Linear mixing	33
8	Sound velocities: svec	34
8.1	Sound velocity calculation	35
8.2	Elastic moduli	36
8.3	Sound velocity averages	36
8.4	Anisotropy corrections	37
8.4.1	Nuclear resonant IXS	38
8.4.2	Momentum-resolved IXS	38
8.5	Output files	39
8.5.1	Sound velocity distributions (options A, L, T)	39
8.5.2	Sound velocity surfaces (option S)	39
9	XRD Background Subtraction: sbck	40
9.1	Background detection procedure	42
9.1.1	Normalization refinement	42
9.1.2	Skew angle	43
9.1.3	Shear angle	43
9.1.4	Background averaging	43
9.1.5	Background refinement	43
9.1.6	Quality assessment	44
9.1.7	Negativity quench	45
9.2	Pattern analysis	45
9.2.1	Strain analysis	45
9.3	Guide peak analysis	46
9.4	Burst mode	46
9.5	Output files	46
9.5.1	Option B	46
9.5.2	Option E	46
9.5.3	Option N	46
9.5.4	Option P	47
9.5.5	Option R	47
9.5.6	Option S	47

10 General function fitting: sfit	48
10.1 Comb function	49
10.2 Exponential function	49
10.3 External function	49
10.4 Pattern function	49
10.5 Polynomial function	50
10.6 Power function	50
10.7 Sigmoid function	51
10.8 Sine function	51
10.9 Voigt function	51
10.10Parameter fitting	51
10.10.1Fisher information	52
10.11Error ellipse (option L)	52
11 Command line options	54

1 License information

MINUTI-2.3.4 is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

MINUTI-2.3.4 is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

Any use of results obtained using MINUTI-2.3.4 in related or unrelated publications has to be properly acknowledged by reference to the name of the package, to the name of the developer(s), and to the NRIXS software site.

2 Getting started

The MINUTI (MINeral physics UTILities) software is a collection of scientific applications to aid mineral physicists to evaluate experimental p-V-T data (module seos), to calculate sound velocities from elastic constants (module svec), to evaluate nuclear forward scattering melting spectra (module simx), to develop linear mixing models of mineral assemblages (module smix), to perform background subtraction for a related set of XRD 2θ patterns (module sbck), and to fit a general function including XRD patterns (module sfit).

Prior to using the program follow the instructions described in the installation guide. A MINUTI program installation produces the five executables sbck, seos, simx, smix, svec, and the graphical user interface (GUI) named Minuti.app (MacOS) or minuti (linux or UNIX). For some linux types a desktop launcher named Minuti should also exist.

The MINUTI programs can be run using either the GUI (the recommended method) or via command line interface (CLI). Output files are the same for GUI and CLI versions – both use the same executables.

2.1 The GUI

This is the faster and for many users more intuitive way to interact with the MINUTI executables. Proceed as follows

- MacOS

With Finder locate the Application folder into which the GUI Minuti.app was installed. This is either '~/Applications' for a local install or '/Applications' for a system-wide install. Start the GUI by double-click on Minuti.app but depending on the OS version you might have to defeat the Gate Keeper mechanism.

Defeating the Gate Keeper: right-click on Minuti.app to get pull-down menu; select open; acknowledge to open.

Defeating the Gate Keeper on MacOS 10.12 and higher: double-click on Minuti.app; acknowledge that it can't be opened; select Apple -> System Preferences -> Security & Privacy -> General; at the bottom right click open Minuti.app. The app might still be translocated by the security mechanism, so that some functionality could be lost.

Preventing the translocation: quit Minuti.app; open Terminal.app; then type 'xattr ~/Applications/Minuti.app' or 'xattr /Applications/Minuti.app'; if you see the string 'com.apple.quarantine' the app has been quarantined and only runs in translocation mode; to lift the quarantine type 'sudo xattr -dr com.apple.quarantine ~/Applications/Minuti.app' or 'sudo xattr -dr com.apple.quarantine /Applications/Minuti.app' which requires admin privileges.

The best way to prevent translocation is the removal of the quarantine flag *before* installation: please see the Installation Guide section 4.1.

- Linux

If a desktop launcher named Minuti was created during installation start that. Otherwise, open terminal window and type 'minuti' on the command line. If this fails the location of the MINUTI executables may not be known permanently to your computer. Correct this by modification of the 'path' setting in the login resource file in your home directory, usually something like '.bash_profile', '.profile', or '.login'. If DIR was the installation directory the directory DIR/bin should be part of the 'path'. You have to logout and login again to update the 'path' settings.

Select MINUTI -> Show Examples from the menu at the top. A list of examples shows up in lower middle panel. Select an item by left-click, then right-click to see a menu of options, select 'Open As Project'. Left-click on 'Run' button on top right to run the example. The runtime output is displayed in the large center panel. A list of the produced output files is shown as thumbnails on the lower left and as a list in the upper right panel. Tooltips become visible after hovering for about one second over a GUI element. The input fields correspond to valid lines of the standard input file (SIF) which are explained in sections 5, 6, and 8.

The GUI can be adapted to your visual and behavioral preferences: select Edit -> Preferences from the menu at the top.

2.2 The CLI

Working with a CLI requires to open a terminal window first. Commands are typed text strings followed by pressing the return key. Examples ready for use with the command-line-interface (CLI) are provided with the MINUTI package. The location of the example directories depends on how the MINUTI was installed. Under MacOS this is '\$HOME/Library/NRIXS/MINUTI/examples' or '/Library/NRIXS/MINUTI/examples' for local or system-wide installs, respectively. Under other Unix systems this is 'DIR/.NRIXS/MINUTI/examples' where DIR is the installation location if owned by the installer (\$HOME is the default). If DIR is a system directory such as /usr/local the examples are located in DIR/share/NRIXS/MINUTI/examples. Your computer system can only access the MINUTI executables if the directory that they are located in is part of the 'path' defined for your system at the time of login.

Now enter the directory with the examples, e.g., by 'cd ~/.NRIXS/MINUTI/examples/seos_quartz'. You need write permissions in the examples directory: type 'ls -ld' and the output string should start with 'drwx'. If not you don't have sufficient permission, so just copy the whole directory to your desktop or home directory and the change to that copy. Run the seos executable by typing 'seos'.¹ The executable reads the input file in_seos and the data file quartz.dat, produces copious output in the terminal window, and creates various output files. If your installation supports visualization a display of some of the output appears on the screen.

The functionality of the MINUTI programs is adapted to your needs by changing the standard input files that control the program flow. It is best to use the files in_seos, in_simx, and in_svec, which can be found in the examples folder, as blueprints. A detailed description of each executable's purpose and their input and output files follows.

The MINUTI software contains three executables, each for a specific task. Calculation results depend on input parameters specified in the standard input file (SIF). Output files are ASCII formatted with space or comma separated columns of numbers. They are readily imported into plotting programs such as Grace and spreadsheets such as OpenOffice or Excel.

In addition to output files with numerical results, each program creates a protocol file <prefix>-ptl.txt with a list of input parameters and main results.

3 Using the GUI

The use of the GUI is often self-explanatory or helped by the tooltips. The MINUTI GUI's 'memory' are project directories: input and output files including data for fit projects are saved here. Each example that shows via MINUTI -> Show Examples is a project directory. Henceforth, we identify a project with its project directory and just talk about the project.

3.1 Open an existing project

There are several possibilities: via MINUTI -> Open...; select an item with extension '.prc' in the lower right directory listing, popupMenu -> Open As Project; via Edit -> Find Project as explained below.

¹

- If this fails the location of the MINUTI executables may not be known permanently to your computer. Correct this by modification of the 'path' setting in the login resource file in your home directory, usually something like '.bash_profile', '.profile', or '.login'. If DIR was the installation directory the directory DIR/bin should be part of the 'path'. You have to logout and login again to update the 'path' settings.

3.2 Create a new project

There are two ways to create a new project: copy an existing project and edit it; create a default project and fill in appropriate values. In the first case, open an existing project as described above. Then use MINUTI -> Save As to create a new project with identical parameters. In the second case, via MINUTI -> New Project use the project wizzard to create a new project with default parameters.

3.3 Find a project

Finding MINUTI project on your computer is easy. Select Edit -> Find Project and define your search criteria in the window that opened. After the search you may select projects from the list to either open them or copy them as new.

3.4 Fit parameters

Right-click on an entry field to see if the parameter is fitable. If yes then a popup menu allows you to select either fitting or linking to another fit parameter. The assignment mechanism is different for fit parameters in the sfit module.

4 The CLI: standard input files

Each MINUTI executable requires a standard input file (SIF). SIFs are plain text files containing no more than 200 characters per line (excess characters will be ignored). Such files are easily edited with text editors such as emacs, xemacs, textedit, or gedit. Do not use word processing software such as MSword or OpenOffice to edit the SIFs because such programs may add special characters that would be misinterpreted by a MINUTI executable.

4.1 Syntax rules

The contents of a line in the SIF is ignored if: '*' is the first character, '@' is the first character, it is empty, or it is filled with spaces only. Lines starting with '*' are comments lines and can be added anywhere in the SIF to permit the user to make comments and notes in the SIF. Lines starting with '@' are substitution directives and may appear anywhere in the SIF. The type of input line has four parts separated by spaces: '@' character, parameter name, assignment symbol ':=' , parameter value. The parameter name has a length eight characters or less and must not contain the '@' character. The parameter name directly preceeded by the '@' character is then substituted by its value as defined in the last preceeding substitution directive. Substitution directives are useful if values are substituted at several locations in the SIF. The use of comment lines and substitution directives in the SIF is optional.

The remaining lines in the SIF are read and interpreted by the executable in order of their appearance. Each of these lines has three parts separated by double colons '::'. The first and the third part contain comments. The second part contains data. Number values are read format free, e.g., data can be specified as 45 or 45. or 4.5E1 etc. Multiple data values are separated by one or more spaces. The second double colon and the second comment (the third part of the line) are optional.

4.2 Assigning fit parameters

Modules seos, simx, and sfit also accept a data input file and allow parameter fitting to minimize the weighted mean-square deviation between calculation and data. Any input parameter may be promoted to a fit parameter by adding a percent symbol (%) to the beginning of the input line. For example, the bulk modulus parameter in the following input line

```
(12) bulk modulus (GPa) :: 160
```

turns into a fit parameter via

```
% (12) bulk modulus (GPa) :: 160
```


where the value is now the start value for the fitting procedure. The same outcome can be achieved by use of substitution directives

```
% @ BulkMod := 160
.... (other lines) ....
(12) bulk modulus (GPa) :: @BulkMod
```

Fit parameters may be grouped by use of substitution directives

```
% @ BulkMod := 160
.... (other lines) ....
(12) high-spin bulk modulus (GPa) :: @BulkMod
.... (other lines) ....
(15) low-spin bulk modulus (GPa) :: @BulkMod
```

4.2.1 Dependent fit parameters

Module sfit for fitting of more general functions supports complicated relationships between parameters. The syntax has to follow rules according to the `expr` command of the `tclsh` scripting language. For example, two parameters that are fitted determine the value of a third parameter

```
% @ parm1 := 10
% @ parm2 := 3
@ parm3 := @parm1*exp(@parm2/@parm1)
```

A parameter that is dependent on other parameters cannot be a fit parameter. All parameters that are used in a dependent parameter must have been defined beforehand.

4.3 Specifying priors

In multiparameter fits, parameters are often considered to be determined by the data set at hand only. However, in many cases, previous information is available or a test of the compatibility of the data at hand with a given set of parameters is of interest. A prior is specified as follows

```
% (12) bulk modulus (GPa) :: 160 5
```

where the first value is the prior and the second value is the uncertainty of the prior. The extreme cases of infinitely large and zero value uncertainty correspond to freely fitted and constant parameters, respectively. The option of prior specification permits a quantitative test for incompatibility of data with a set of given parameters (for example from a different experiment). Priors enter the fitting process as for example shown in (30).

5 Equations-of-state: seos

The seos program calculates the isothermal equation-of-state (EOS) including spin-transition effects with a set of parameters defined in the standard input file (SIF) and optionally permits fitting of parameters to experimental data. The EOS is constructed as a sum of Helmholtz free energies for relevant parts of the material under consideration. The inclusion of spin-transition effects is optional. Line numbers refer to the SIF. The following sections describe the content of each line of the SIF, the algorithms used to implement the EOS and the fitting procedure. Furthermore the contents of the output files is explained.

The SIF of the seos executable has at least 14 lines of data. If a spin crossover equation-of-state (EOS) is used at least 20 lines of data are required. The content of these lines is discussed in order of appearance.

1. Unit formula – two values :

The first value gives the composition of the material in the form $A\{n\}B\{m\}$ where A,B are element symbols and n,m are their abundances in the unit cell. The total number of atoms has to be consistent with the zero-pressure volume specified (or fitted) in line 11. For example, the high-pressure phase of magnesium-silicate perovskite with a volume of about 160 \AA^3 would be specified as $\text{Mg}\{4\}\text{Si}\{4\}\text{O}\{12\}$. Fractional values for the abundances are also supported.

The second value is optional and specifies the spin crossover element symbol. For example, the entry ' $\text{Fe}\{1.4\}\text{Mg}\{2.6\}\text{O}\{4\} \text{Fe}$ ' gives the composition first and then the element symbol of the spin crossover element, Fe in this case. If a spin crossover element is specified lines 15-20 must also be specified.

2. Name of measured data file – many values :

The first value gives the name of a measured data file.² The following values define the assignment of column numbers in the data file to physical parameters. Each assignment is a combination of one letter and a number, e.g., P1 or v15. The letter refers to the physical meaning according to Table 1 on page 8. Data in the measured data file between pressures p_1 and p_2 may be excluded from fitting by adding the line '#Mask p_1 p_2 ' to the file. Several lines of this type may be added to exclude specific regions of data.

letter	physical meaning	units	comment
P	pressure	GPa	required
V	volume	\AA^3	required
T	temperature	K	optional
p	pressure uncertainty	GPa	optional
v	volume uncertainty	\AA^3	required
t	temperature uncertainty	K	optional

Table 1: Column assignments for seos experimental data input file. The units refer to what is expected from input data.

3. Output file prefix and options – many values :

The first value gives the prefix for output file names.³ Several output files listed on the top in Tables 2 and 3 are created by default. Output files are plain text files. If option ',' is specified csv-formatting is provided. If option G is specified a header for the xmgrace graphical program is added. Additional output files are created according to options following the file name prefix. With exception of options F, L, and M, the first value is the pressure in units of GPa. The second value depends on the option specified, and the third value is the statistical error of the second value. In cases without spin crossover, the options have physical meaning according to Table 2 on page 9. Options L and R produce output only if a data file was specified in line 2. The measured volume output is reduced to the temperature given in line 4. All other output is produced at the temperature given in line 4. In cases with spin crossover, the options have physical meaning according to Table 3 on page 10. A detailed description of the procedures used to produce the output is provided in sections 5.6 on page 18 to 5.20 on page 22.

²The input file is a plain text file containing no more than 300 characters per line (excess characters are ignored). Each line in the file contains a sequence of data values called columns separated by either spaces, tabs, or commata (csv format). Lines that do not begin with a number are ignored (after removing leading spaces, tabs, commata).

³Names may be defined either relative to the working directory or absolute: Prefix creates files in the working directory, ../Prefix creates files in the parent directory, myDir/Prefix creates files in the directory myDir (must exist) in the working directory, /Users/-data/Prefix creates files in the directory /Users/data (must exist and be writeable).

option	column 1	column 2,3, ...	y units	file name
	pressure	isothermal bulk modulus, K	GPa	<prefix>_bmt.<ext>
	pressure	adiabatic bulk modulus	GPa	<prefix>_bma.<ext>
	pressure	pressure derivative, K'		<prefix>_bdp.<ext>
	pressure	temp. derivative dK/dT	MPa/K	<prefix>_bdt.<ext>
	pressure	seismic velocity, v_ϕ	km/s	<prefix>_smv.<ext>
	pressure	calculated density, ρ	g/cm ³	<prefix>_dns.<ext>
	pressure	calculated volume, V	Å ³	<prefix>_vol.<ext>
	pressure	measured volume	Å ³	<prefix>_vrd.<ext>
	pressure	normalized fit residuals		<prefix>_rsd.<ext>
	pressure	thermal Grüneisen parameter		<prefix>_tgp.<ext>
a,A	pressure	temperature, density	K, g/cm ³	<prefix>_dns.<ext1>
	pressure	temperature, bulk modulus	K, GPa	<prefix>_bmt.<ext1>
	pressure	temperature, thermal pressure	K, GPa	<prefix>_prt.<ext1>
c,C	pressure	V, K, K', covariance matrix	various	<prefix>_vkk.csv
	pressure	ρ , v_ϕ covariance matrix	various	<prefix>_dsv.csv
	volume	isothermal bulk modulus	GPa	<prefix>_VK/
	volume	pressure derivative, K'		<prefix>_VKp/
	isothermal bulk modulus	pressure derivative, K'		<prefix>_KKp/
	density	seismic velocity	km/s	<prefix>_dsv/
h,H	pressure	specific heat	k_B /atom	<prefix>_ccv.<ext>
i,I	pressure	internal energy	eV/atom	<prefix>_int.<ext>
l,L	parameter n	parameter m	variable	<prefix>_p<n>v<m>_eel.<ext>
	n-m, normalized, centered			<prefix>_p<n>v<m>_nel.<ext>
m,M	f of data	F of data	GPa	<prefix>_ffm.<ext>
	f of theory	F of theory	GPa	<prefix>_fft.<ext>
p,P	pressure	thermal pressure	GPa	<prefix>_prt.<ext>
t,T	pressure	thermal expansion	ppm/K	<prefix>_thx.<ext>

Table 2: Output options for thermal EOS calculations. <prefix> is the file prefix specified in line 3. The extension <ext> is either 'csv' for comma-separated-value files (option ', ' in line 3) or 'dat' otherwise. Extensions <ext1> are 3D.gz for compressed ViSit files, ppm for image files, and mbf for binary data files.

In addition, the protocol file '<prefix>_ptl.txt' and, for fits to data, the results file '<prefix>_ite.csv' are created.

4. Temperature polynomial – four values :

The first value is required and gives the temperature in units of K. The next three values a_n are optional and define a pressure dependent temperature via $T(p) = T_1 + a_1p + a_2p^2 + a_3p^3$ where T_1 is the first value and p is pressure in units of GPa. If the A-option is specified in line 3 then values two and three are required, and value four is obsolete. With option A, the first and second value give a temperature range and the third value the number of steps for calculations.

5. Pressure ranges (GPa) – seven values :

The first and second values give the start and end of the pressure range for subsequent calculations. The third value specifies the number of steps in this range. The second triple of values is optional and applies to the generation of pressure-dependent error ellipses. If values are provided and output option C is specified the first and second values give the start and end of the pressure range, and the third value specifies the number of ellipses to generate in this range. The last value if given specifies the generation of pressure-dependent error ellipses at α levels of 1, 2, and 3 (see section 5.12 on page 19 for more information).

6. Reference temperature (K) – one value :

This value specifies the temperature at which the elastic parameters in lines 11-16 are defined. If the reference temperature and the temperature in line 4 (no polynomial) or temperatures given in a data input file are identical values in lines 7-9 have no influence on calculations. The search for hysteresis behavior of the spin EOS is enabled for temperatures defined in line 4 that are smaller than the reference temperature.

option	column 1	column 2,3, ...	y units	file name
	pressure	bulk modulus, K	GPa	<prefix>_bmt.<ext>
	pressure	adiabatic bulk modulus	GPa	<prefix>_bma.<ext>
	pressure	bulk mod., low-pressure spin state	GPa	<prefix>_bml.<ext>
	pressure	bulk mod., high-pressure spin state	GPa	<prefix>_bmh.<ext>
	pressure	rel. spin contr. to bulk mod.	%	<prefix>_bms.<ext>
	pressure	pressure derivative, K'		<prefix>_bdp.<ext>
	pressure	temp. derivative	MPa/K	<prefix>_bdt.<ext>
	pressure	seismic velocity, v_ϕ	km/s	<prefix>_smv.<ext>
	pressure	calculated density, ρ	g/cm ³	<prefix>_dns.<ext>
	pressure	calculated volume, V	Å ³	<prefix>_vol.<ext>
	pressure	volume, low-pressure spin state	Å ³	<prefix>_vll.<ext>
	pressure	volume, high-pressure spin state	Å ³	<prefix>_vlh.<ext>
	pressure	relative volume, spin only	%	<prefix>_vls.<ext>
	pressure	measured volume	Å ³	<prefix>_vrd.<ext>
	pressure	normalized fit residuals		<prefix>_rsd.<ext>
	pressure	thermal Grüneisen parameter		<prefix>_tgp.<ext>
a,A	pressure	temperature, density	K, g/cm ³	<prefix>_dns.<ext1>
	pressure	temperature, bulk modulus	K, GPa	<prefix>_bmt.<ext1>
	pressure	temperature, thermal pressure	K, GPa	<prefix>_prt.<ext1>
	pressure	temperature, unpaired electrons	K	<prefix>_unp.<ext1>
	temperature	spin transition pressure	GPa	<prefix>_stp.<ext>
	temperature	spin transition width	GPa	<prefix>_stw.<ext>
c,C	pressure	V, K, K', covariance matrix	various	<prefix>_vkk.csv
	pressure	ρ , v_ϕ covariance matrix	various	<prefix>_dsv.csv
	volume	bulk modulus, K	GPa	<prefix>_VK/
	volume	pressure derivative, K'		<prefix>_VKp/
	bulk modulus, K	pressure derivative, K'		<prefix>_KKp/
	density	seismic velocity	km/s	<prefix>_dsv/
e,E	pressure	spin entropy	k_B /atom	<prefix>_etp.<ext>
f,F	pressure	spin free energy	eV/atom	<prefix>_fre.<ext>
h,H	pressure	specific heat	k_B /atom	<prefix>_ccv.<ext>
i,I	pressure	internal energy	eV/atom	<prefix>_int.<ext>
	pressure	spin internal energy	eV/atom	<prefix>_ins.<ext>
l,L	parameter n	parameter m	variable	<prefix>_p<n>v<m>_eel.<ext>
	n-m, normalized, centered			<prefix>_p<n>v<m>_nel.<ext>
m,M	f of data	F of data	GPa	<prefix>_ffm.<ext>
	f of theory	F of theory	GPa	<prefix>_fft.<ext>
p,P	pressure	thermal pressure	GPa	<prefix>_prt.<ext>
	pressure	spin pressure	GPa	<prefix>_prs.<ext>
s,S	pressure	spin populations, state n		<prefix>_<n>_ssp.<ext>
t,T	pressure	thermal expansion	ppm/K	<prefix>_thx.<ext>
u,U	pressure	unpaired electrons		<prefix>_unp.<ext>
x,X	pressure	$K_{\beta'}$ intensity		<prefix>_xes.<ext>
z,Z	pressure	$(\Delta - \Pi)$	eV	<prefix>_dpp.<ext>
	V/V_0	$(\Delta - \Pi)$	eV	<prefix>_dpv.<ext>
	pressure	free energy landscape	meV/atom	<prefix>_frl.<ext>

Table 3: Output options for spin EOS calculations. <prefix> is the file prefix specified in line 3. The extension <ext> is either 'csv' for comma-separated-value files (option ',' in line 3) or 'dat' otherwise. Extensions <ext1> are 3D.gz for compressed ViSit files, ppm for image files, and mbf for binary data files.

7. Debye temperature (K) – one value :

This value specifies the Debye temperature Θ of the material at zero pressure.

8. Debye-Grüneisen parameter – two values :

The first value specifies the Debye-Grüneisen parameter $\gamma_0 = (\partial \ln \Theta / \partial \ln V)$ at zero pressure. The second value gives the model for the volume dependence of the Debye-Grüneisen parameter. Specify zero for standard model $\gamma = \gamma_0 (V/V_0)^q$. Specify one for the Altschuler model $\gamma = \gamma_\infty + (\gamma_0 - \gamma_\infty) (V/V_0)^q$. Specify three for the Speziale/Jeanloz model $\gamma = \gamma_0 \exp[-q/s \{1 - (V/V_0)^s\}]$.

9. Debye-Grüneisen parameter exponents – two values :
The first value specifies the exponent q of the previously selected model. The second value gives γ_∞ for the Altschuler model or s for the Speziale/Jeanloz model.
10. Electron temperature (K) – one value :
This value specifies the characteristic conduction electron temperature T_{con} of the material at zero pressure.
11. Electron scaling parameter – one value :
This value specifies the scaling parameter of the electron temperature $\Gamma_0 = (\partial \ln T_{con} / \partial \ln V)$ at zero pressure.
12. Electron scaling parameter exponent – one value :
This value specifies the exponent $Q = (\partial \ln \Gamma / \partial \ln V)$ to determine the volume dependence of the Electron scaling parameter Γ . This is also expressed as $\Gamma = \Gamma_0 (V/V_0)^Q$.
13. Type of elastic equation-of-state – two values :
The first value specifies the type of EOS for the elastic part of the total EOS. The choices are BM (3rd order Birch-Murnaghan), BM4 (4th order Birch-Murnaghan), Vinet, log (3rd order logarithmic), log4 (4th order logarithmic). In case of BM4 or log4, the second value specifies the second pressure derivative of the bulk modulus at zero pressure in units of GPa^{-1} .
14. Volume at zero pressure (\AA^3) – one value :
This value specifies the volume of the composition unit given in line 1 at zero pressure and temperature given in line 6.
15. Bulk modulus at zero pressure (GPa) – one value :
This value specifies the isothermal bulk modulus of the material at zero pressure and temperature given in line 6.
16. Pressure derivative of bulk modulus at zero pressure (GPa) – one value :
This value specifies the isothermal pressure derivative of the bulk modulus at zero pressure and temperature given in line 6.
17. Iteration control – three values :
The first value gives the maximum number of iteration steps. The other values are optional. The second value specifies the Levenberg-Marquardt parameter λ that is used to create a multiplier m to the diagonal of the second-derivative matrix of the normalized χ^2 . For iteration step n , the multiplier is given by $m = 1 + \lambda \max [0, (\chi_{n-1}^2 - \chi_n^2) / \chi_n^2]$. Larger values for λ have a dampening effect and steer the iteration process toward the gradient method. The Newton method is obtained for $\lambda = 0$ which is also the default value. The third value gives the required relative change of χ^2 value for an iteration step to stop the iteration.

The following input lines only need to be specified in the case of a spin crossover. The values in lines 11-13 now describe the low-pressure phase which for Fe is typically the high-spin region.

18. High-pressure phase, volume at zero pressure (\AA^3) – one value :
This value specifies the volume of the high-pressure phase at zero pressure.
19. High-pressure phase, bulk modulus (GPa) – one value :
This value specifies the isothermal bulk modulus of the high-pressure phase at zero pressure and temperature given in line 6.
20. High-pressure phase, pressure derivative of bulk modulus (GPa) – one value :
This value specifies the isothermal pressure derivative of the bulk modulus of the high-pressure phase at zero pressure and temperature given in line 6.
21. This line is reserved for future use.

22. Spin transition pressure estimate (GPa) – one value :
This value specifies an estimate of the pressure at which crystal-field splitting Δ and spin-pairing energy Π are identical. This is approximately the pressure at which volume collapse becomes noticeable.
23. Spin state energies ($\Delta - \Pi$) – several values :
These values specify the energies of the spin states in units of ($\Delta - \Pi$), the difference of crystal-field splitting Δ and spin-pairing energy Π . The number of values given also specifies the number of spin states. The following input lines have to specify the same number of values.
24. Orbital degeneracies – several values :
These values specify the orbital degeneracy for each spin state.
25. Unpaired electrons – several values :
These values specify the number of unpaired electrons for each spin state.

5.1 Thermal Equation of State

The free energy or Helmholtz energy is written as sum of a term describing the elastic deformation F_{el} and a temperature-dependent term describing lattice vibrations and conduction electrons

$$F(V, T) = F_{el}(V) + F_{th}(V, T) - F_{th}(V, T_0) \quad . \quad (1)$$

The elastic part of (1) can be selected to be of type Birch-Murnaghan, Vinet, or logarithmic (line 10), and is given by

- Birch-Murnaghan (3rd order)

$$F_{el}(V) = \frac{9}{2} K_0 V_0 f^2 \{1 + f (K'_0 - 4)\} \quad \text{with} \quad f = \frac{1}{2} \left\{ \left(\frac{V_0}{V} \right)^{2/3} - 1 \right\} \quad (2)$$

- Birch-Murnaghan (4th order)

$$F_{el}(V) = \frac{9}{2} K_0 V_0 f^2 \left\{ 1 + f (K'_0 - 4) + f^2 \frac{3}{4} \left(K_0 K''_0 + (K'_0 - 4)(K'_0 - 3) + \frac{35}{9} \right) \right\} \quad (3)$$

- Vinet

$$F_{el}(V) = \frac{4K_0 V_0}{(K'_0 - 1)^2} \{ (x - 1) e^x + 1 \} \quad \text{with} \quad x = \frac{3}{2} (K'_0 - 1) \left\{ 1 - \left(\frac{V}{V_0} \right)^{1/3} \right\} \quad (4)$$

- logarithmic or natural strain (3rd order)

$$F_{el}(V) = \frac{9}{2} K_0 V_0 f^2 \{1 + f (K'_0 - 2)\} \quad \text{with} \quad f = \frac{1}{3} \ln \left(\frac{V_0}{V} \right) \quad (5)$$

- logarithmic or natural strain (4th order)

$$F_{el}(V) = \frac{9}{2} K_0 V_0 f^2 \left\{ 1 + f (K'_0 - 2) + f^2 \frac{3}{4} (K_0 K''_0 + (K'_0 - 2)(K'_0 - 1) + 1) \right\} \quad (6)$$

In these relations, V_0 is the volume at zero pressure (line 14), K_0 is the isothermal bulk modulus at zero pressure (line 15), and K'_0 is the pressure derivative of the isothermal bulk modulus at zero pressure (line 16), and K''_0 is the second pressure derivative of the isothermal bulk modulus at zero pressure (line 13). All values are assumed at a reference temperature T_0 (line 6). Therefore at the reference temperature the EOS is given by the elastic part only.

The thermal part of the free energy has two contributions, a quasi-harmonic vibrational term and a conduction electron term that vanishes for insulators

$$F_{th}(V, T) = F_{qh}(V, T) + F_{con}(V, T) \quad . \quad (7)$$

For the quasi-harmonic lattice vibrations, the free energy is expressed by the phonon density of states

$$F_{th}(V, T) = k_B T \int \ln(2 \sinh \frac{\omega}{2k_B T}) D(\omega, V) d\omega \quad . \quad (8)$$

The seos program uses the Debye approximation for the phonon density of states which features one volume dependent parameter, the Debye temperature Θ , and we have for $\omega \leq k_B \Theta$

$$D(\omega) = \frac{9}{k_B \Theta} \left(\frac{\omega}{k_B \Theta} \right)^2 \quad . \quad (9)$$

With this approximation, the free energy of the Debye phonon gas is given by

$$F_{qh}(V, T) = 3k_B T \ln \left(2 \sinh \frac{\Theta}{2T} \right) - k_B T I_3 \left(\frac{\Theta}{T} \right) - \frac{3}{8} k_B \Theta \quad . \quad (10)$$

with the third-order Debye function

$$I_3(t) = \frac{3}{t^3} \int_0^t \frac{x^3 dx}{e^x - 1} \quad . \quad (11)$$

The quasi-harmonic pressure now takes the form

$$p_{qh}(V, T) = - \left(\frac{\partial F_{qh}}{\partial V} \right)_T = \frac{\gamma}{V} \left(3k_B T I_3 \left(\frac{\Theta}{T} \right) + \frac{9}{8} k_B \Theta \right) \quad , \quad (12)$$

where $\gamma = -\partial \ln \Theta / \partial \ln V$ is the Debye-Grüneisen parameter. The seos program supports several models for its volume dependence as discussed in section 5.1.1.

For the electrons, the free energy is expressed as

$$F_{con}(V, T) = \frac{k_B T^2}{T_{con}(V)} \quad , \quad (13)$$

where T_{con} is the volume dependent electron scaling temperature. The conduction electron pressure now takes the form

$$p_{con}(V, T) = - \left(\frac{\partial F_{con}}{\partial V} \right)_T = \frac{\Gamma}{V} \frac{k_B T^2}{T_{con}} \quad , \quad (14)$$

where $\Gamma = -\partial \ln T_{con} / \partial \ln V$ is the electron scaling parameter which is expressed in the seos program using a scaling law

$$\Gamma(V) = \Gamma_0 \left(\frac{V}{V_0} \right)^Q \quad , \quad (15)$$

where Γ_0 (line 11), Q (line 12), and V_0 (line 14) are input to the program. The expression for the electron scaling temperature follows via integration of $\Gamma = -\partial \ln T_{con} / \partial \ln V$ to

$$T_{con}(V) = T_{con0} \exp \left[\frac{\Gamma_0 - \Gamma(V)}{Q} \right] \quad , \quad (16)$$

where T_{con0} (line 10) is input to the seos program.

For a given pressure p and temperature T , the volume is calculated by solving the thermal EOS

$$p = - \frac{\partial F_{el}}{\partial V} + p_{th}(V, T) - p_{th}(V, T_0) \quad \text{with} \quad p_{th}(V, T) = p_{qh}(V, T) + p_{con}(V, T) \quad . \quad (17)$$

5.1.1 Grüneisen models

Three models for volume dependence of the Grüneisen parameter are implemented: standard model; Altshuler model; Speziale-Jeanloz model. For the standard model, the Debye-Grüneisen parameter is expressed using a scaling law

$$\gamma(V) = \gamma_0 \left(\frac{V}{V_0} \right)^q, \quad (18)$$

where γ_0 (line 8), q (line 9), and V_0 (line 14) are input to the program. The expression for the Debye temperature follows via integration of $\gamma = -\partial \ln \Theta / \partial \ln V$ to result in

$$\Theta(V) = \Theta_0 \exp \left[\frac{\gamma_0 - \gamma(V)}{q} \right], \quad (19)$$

where Θ_0 (line 7) is input to the seos program.

For the Altshuler model, the Debye-Grüneisen parameter is expressed using a modified scaling law

$$\gamma(V) = \gamma_\infty + (\gamma_0 - \gamma_\infty) \left(\frac{V}{V_0} \right)^q, \quad (20)$$

where γ_0 (line 8), γ_∞ (line 9), q (line 9), and V_0 (line 14) are input to the program. The expression for the Debye temperature follows via integration to result in

$$\Theta(V) = \Theta_0 \left(\frac{V}{V_0} \right)^{-\gamma_\infty} \exp \left[\frac{\gamma_0 - \gamma_\infty}{q} \left\{ 1 - \left(\frac{V}{V_0} \right)^q \right\} \right], \quad (21)$$

The Speziale-Jeanloz model uses

$$\gamma(V) = \gamma_0 \exp \left[-\frac{q_0}{q_1} \left\{ 1 - \left(\frac{V}{V_0} \right)^{q_1} \right\} \right], \quad (22)$$

where γ_0 (line 8), q_0 (line 9), q_1 (line 9), and V_0 (line 14) are input to the program. The expression for the Debye temperature has to be numerically integrated

$$\Theta(V) = \Theta_0 \exp \left[-\int_{V_0}^V \gamma(V) \frac{dV}{V} \right]. \quad (23)$$

5.2 Spin Equation of State

The formalism to develop a simple description of additional contributions to the free energy for atoms with spin states was given before (Sturhahn et al., GRL 2005). Assume a set of spin states described by number of unpaired electrons u_i , volume-dependent energy E_i , and orbital degeneracy g_i . The free energy per spin-state atom is then given by

$$F_s(V, T) = -k_B T \ln Z_s = -k_B T \ln \left\{ \sum_i G_i \exp \left[-\frac{E_i(V)}{k_B T} \right] \right\}, \quad (24)$$

where $G_i = g_i(1 + u_i)$ are the combined degeneracies, orbital and spin, and Z_s is the spin partition function. The seos program constrains the E_i values as follows: at zero pressure the spin contribution is absorbed in the elastic free energy; at very high pressure the spin contribution is also absorbed in the elastic free energy but with different parameters; the E_i values are multiples of the same volume-dependent function, i.e., $E_i = \epsilon_i \Omega$ with dimensionless coefficients ϵ_i and the difference of crystal-field splitting and spin-pairing energy Ω .

At low pressures, i.e., $V \approx V_0$, one spin state (for Fe typically the high-spin state) has the lowest energy by far and $F_s = E_1 - k_B T \ln G_1$ in excellent approximation. At very high pressures, i.e., $V \ll V_0$, also one spin state (for Fe typically the low-spin state) has the lowest energy by far and $F_s = E_2 - k_B T \ln G_2$ in excellent approximation. The term that is added to (1) in the seos program is given by

$$-n k_B T \ln Z_s - n (\epsilon_1 \Omega(V) - k_B T \ln G_1), \quad (25)$$

where n is the number of spin-state atoms in the volume V_0 . For low pressures, this term disappears and the EOS of material behaves as outlined in the previous section. For very high pressures, the EOS behaves similar but with a different elastic term. This behavior is properly modeled by a difference of crystal-field splitting and spin-pairing energy Ω that is given in terms of elastic free energies for the low- and high-pressure phases

$$\Omega(V) = \Omega_0(T) + \frac{1}{n(\epsilon_2 - \epsilon_1)} \left(F_{el}^{(2)}(V) - F_{el}^{(1)}(V) \right) + \frac{k_B T \ln(G_2/G_1)}{n(\epsilon_2 - \epsilon_1)} . \quad (26)$$

The volume-indepent parameter Ω_0 determines the volume at the spin transition which occurs close to $\Omega = 0$, i.e., the crystal-field splitting equals the spin-pairing energy. The input parameter transition pressure estimate (line 22) is used to estimate the transition volume V_{tr} using the average of the EOS value for low- and high-pressure phases. The estimated transition volume is then used to calculate Ω_0 in (26) resulting in

$$\Omega(V) = \frac{1}{n(\epsilon_2 - \epsilon_1)} (\delta F_{el}(V) - \delta F_{el}(V_{tr})) , \quad (27)$$

where $\delta F_{el} = F_{el}^{(2)} - F_{el}^{(1)}$. The spin pressure needed for the spin EOS is now calculated as

$$\begin{aligned} p_s(V, T) &= - \left(\frac{\partial F_s}{\partial V} \right)_T \\ &= \frac{1}{(\epsilon_2 - \epsilon_1)} \left(\frac{\partial F_{el}^{(2)}}{\partial V} - \frac{\partial F_{el}^{(1)}}{\partial V} \right) \left(\epsilon_1 - Z_s^{-1} \sum_i G_i \epsilon_i \exp \left[-\frac{\epsilon_i \Omega(V)}{k_B T} \right] \right) \end{aligned} \quad (28)$$

Spin-state specific parameters that are specified in the SIF are: zero-pressure volume of the high-pressure phase (line 18); isothermal bulk modulus at zero pressure of the high-pressure phase (line 19); pressure derivative of the isothermal bulk modulus at zero pressure of the high-pressure phase (line 20); transition pressure estimate (line 22); spin-state energies ϵ_i (line 23); spin-state orbital degeneracies (line 24); spin-state unpaired electrons (line 25). For a given pressure p and temperature T , the volume is calculated by solving the spin EOS

$$p = -\frac{\partial F_{el}^{(1)}}{\partial V} + p_{th}(V, T) - p_{th}(V, T_0) + p_s(V, T) . \quad (29)$$

5.3 Parameter fitting

Input data (EIF given in line 2) are specified as N sets $\{p_i, V_i, T_i, \delta p_i, \delta V_i, \delta T_i\}$ consisting of pressure, volume, temperature, and their uncertainties. The EOS is fitted to these data by variation of n parameters $\{x_j\}$ by minimizing the normalized mean-square deviation (method of weighted least squares)

$$\chi^2 = \frac{1}{(N + p - n)} \left\{ \sum_{i=1}^N w_i (V_i - V(p_i, T_i, \{x_j\}))^2 + \frac{N}{n} \sum_{k=1}^p \frac{(x_k - X_k)^2}{\delta^2 X_k} \right\} \quad \text{with} \quad N + p - n > 0 . \quad (30)$$

Some number p of the fit parameters may have priors X_k with uncertainties δX_k . The weights w_i are determined by uncertainties of the data via

$$w_i^{-1} = \delta^2 V_i + \left(\frac{\partial V}{\partial p} \right)_T^2 \delta^2 p_i + \left(\frac{\partial V}{\partial T} \right)_p^2 \delta^2 T_i = \delta^2 V_i + V_i^2 \left(\frac{\delta^2 p_i}{K_i^2} + \alpha_i^2 \delta^2 T_i \right) , \quad (31)$$

where K_i and α_i are bulk modulus and thermal expansion calculated from the EOS at $\{p_i, T_i\}$, respectively. If data and priors constrain the fit parameters reasonably well a solution with minimal χ^2 is obtained. These optimal parameters have errors δx_j related to data variation and prior uncertainties. The errors are estimated as variances (square errors) by

$$\delta x_j = \sqrt{\sigma_{jj}} \quad \text{with} \quad \sigma_{jj'} = \chi_{min}^2 \left[\sum_{i=1}^N w_i \frac{\partial V(p_i, T_i)}{\partial x_l} \frac{\partial V(p_i, T_i)}{\partial x_{l'}} + \delta_{ll'} \frac{N}{n} \frac{1}{\delta^2 X_l} \right]_{jj'}^{-1} , \quad (32)$$

where $\sigma_{jj'}$ is the covariance matrix and $[\dots]^{-1}$ denotes matrix inversion. The derivatives are calculated with the optimal parameters. The fit-parameter correlation matrix is given by

$$c_{jj'} = \frac{\sigma_{jj'}}{\sqrt{\sigma_{jj}\sigma_{j'j'}}} \quad . \quad (33)$$

The variance of a function F of the fit parameters is calculated as

$$\delta^2 F(\{x_j\}) = \sum_{jj'=1}^n \sigma_{jj'} \frac{\partial F}{\partial x_j} \frac{\partial F}{\partial x_{j'}} \quad , \quad (34)$$

5.3.1 Fisher information

Fit parameters can be highly correlated leading to overinterpretation of data. The Fisher-information matrix can help identify parameters that are weakly defined by the data. In the context of the minimum χ^2 as maximum likelihood estimator, the Fisher information is calculated as

$$I_{jj'} = \sqrt{\sigma_{jj}\sigma_{j'j'}} \frac{\partial^2 \chi^2}{\partial x_j \partial x_{j'}} \quad , \quad (35)$$

where parameter values are obtained by minimizing χ^2 . The smallest eigenvalues of the Fisher information matrix indicate fit parameter combinations that are least important for minimizing χ^2 . An individual fit parameter x_k is tested by calculating a modified covariance matrix

$$\sigma'_{jj'} = \chi_{min}^2 \left[\sum_{i=1}^N w_i \frac{\partial V(p_i, T_i)}{\partial x_l} \frac{\partial V(p_i, T_i)}{\partial x_{l'}} + \delta_{ll'} \frac{1}{\delta^2 X_l} + \delta_{lk} \delta_{l'k} \frac{\lambda_k}{\sigma_{kk}} \right]_{jj'}^{-1} \quad , \quad (36)$$

where $\lambda_k = 20000$ basically fixes the parameter. The modified Fisher information matrix is then

$$I'_{jj'} = \sqrt{\sigma'_{jj}\sigma'_{j'j'}} \frac{\partial^2 \chi^2}{\partial x_j \partial x_{j'}} + \delta_{jk} \delta_{j'k} \lambda_k \quad . \quad (37)$$

If fit parameter x_k is only weakly determined by the data the smallest eigenvalues of the modified Fisher information matrix are significantly larger than the smallest eigenvalues of the full Fisher information matrix given in (35).

5.3.2 Temperature errors

Equation (31) includes temperature errors provided in the data set. These temperature errors are often estimated, and seos performs an estimate of what these errors should be to be consistent with the fit to the data set. The volumes $V(p_i, T_i)$ are calculated from the data $\{p_i, V_i, T_i\}$ with the optimized fit parameters. In accordance with uncertainties in the data, the differences $\Delta V_i = V_i - V(p_i, T_i)$ do not vanish. Now we assume that the deviation is caused solely by incorrect temperature values. The temperature corrections ΔT_i^\pm that satisfy

$$V_i - V(p_i, T_i \pm \Delta T_i^\pm) = 0 \quad (38)$$

should then be comparable to the provided temperature errors. Here the following averages are calculated for comparison

$$\overline{\Delta T} = \frac{1}{4} \sqrt{\langle (\Delta T^+ - \Delta T^-)^2 \rangle - \langle \Delta T^+ - \Delta T^- \rangle^2} \quad \text{and} \quad \overline{\delta T} = \langle \delta T \rangle \quad , \quad (39)$$

where $\langle \dots \rangle$ are averages over all data points. The values of $\overline{\Delta T}$ and $\overline{\delta T}$ should not deviate significantly from each other. Otherwise incorrectly specified temperatures or temperature errors can be suspected.

5.4 Bulk modulus

The isothermal bulk modulus is calculated by direct differentiation of the EOS

$$K(p, T) = - \left(\frac{\partial p}{\partial \ln V} \right)_T . \quad (40)$$

The bulk modulus in the low-pressure and high-pressure states is calculated from (17) with corresponding elastic parameters, respectively.

The relative spin contribution to the bulk modulus is calculated as

$$\frac{K_s}{K - K_s} = \left(\frac{\partial p_s}{\partial \ln V} \right)_T \left(\frac{\partial(p - p_s)}{\partial \ln V} \right)_T^{-1} . \quad (41)$$

The seismic velocity is calculated via

$$v_\phi(p, T) = \sqrt{(1 + \gamma_{th} \alpha T) K V / m_0} , \quad (42)$$

where m_0 is the mass contained in the cell volume V_0 , and α is the thermal expansion coefficient of (67). The thermal Grüneisen parameter is calculated as

$$\gamma_{th} = \alpha K V \frac{1}{c_V} , \quad (43)$$

with specific heat c_V of (53). The pressure derivative of the bulk modulus is calculated as

$$K' = \left(\frac{\partial K}{\partial p} \right)_T = K^2 \left(\frac{\partial^2 \ln V}{\partial p^2} \right)_T . \quad (44)$$

The temperature derivative of the bulk modulus is calculated as

$$\left(\frac{\partial K}{\partial T} \right)_p = K^2 \left(\frac{\partial \alpha}{\partial p} \right)_T , \quad (45)$$

where the thermal expansion α is obtained from (67).

If fit parameters are defined the variances are calculated using (34) with the derivatives

$$\frac{1}{K} \frac{\partial K}{\partial x_j} = \frac{1}{V} \frac{\partial V}{\partial x_j} + \frac{K}{V} \frac{\partial}{\partial p} \left(\frac{\partial V}{\partial x_j} \right) \quad (46)$$

and

$$\frac{\partial K'}{\partial x_j} = (K' + 1) \left\{ \frac{1}{V} \frac{\partial V}{\partial x_j} + \frac{2K}{V} \frac{\partial}{\partial p} \left(\frac{\partial V}{\partial x_j} \right) \right\} + \frac{K^2}{V} \frac{\partial^2}{\partial p^2} \left(\frac{\partial V}{\partial x_j} \right) \quad (47)$$

and

$$\frac{1}{v_\phi} \frac{\partial v_\phi}{\partial x_j} = \frac{1}{V} \frac{\partial V}{\partial x_j} + \frac{K}{2V} \frac{\partial}{\partial p} \left(\frac{\partial V}{\partial x_j} \right) . \quad (48)$$

All partial derivatives are taken at constant temperature.

5.5 Volume

The volume V is calculated based on the EOS, (29), for various pressures and temperatures as specified in the SIF. The volume in the low-pressure and high-pressure states is calculated from (17) with corresponding elastic parameters, respectively. The relative spin contribution to the volume is calculated as

$$\delta_s = \frac{V(p) - V(p - p_s)}{V(p - p_s)} . \quad (49)$$

If fit parameters are defined the temperature reduced volumes V_{ri} are calculated from the data $\{p_i, V_i, T_i\}$ as

$$V_{ri} = V_i - V(p_i, T_i) + V(p_i, T_0) \quad , \quad (50)$$

where T_0 is the reference temperature specified in line 6 of the SIF. A plot of $V(p_i, T_0)$ and V_{ri} versus pressure provides an easy visual assessment of deviations between data and model.

The density is derived from the calculated volume as $\rho(p) = \rho_0 V_0 / V(p)$. The density at zero pressure, ρ_0 , is calculated from the unit formula defined in line 1 of the SIF.

After successful optimization of χ^2 given by (30) the residuals consists of pairs $\{p_i, \sqrt{w_i}(V_i - V(p_i, T_i))\}$ calculated for the optimal parameters. The weights w_i are given by (31).

5.6 p-T array (option A)

The output file with extension '`<val>.3D`' contains four columns of numbers: normalized pressure; normalized temperature; normalized density; value. Value, `<val>` pairs are: density, dns; isothermal bulk modulus, bmt; thermal pressure, prt; unpaired electrons, unp. The normalization is a linear map from $[\text{min}, \text{max}] \rightarrow [0, 1]$. These files can for example be visualized with the VisIt software of Lawrence Livermore National Laboratory.

Files with the extension 'mbf' are binary containing the same arrays as above. The binary stream is

bytes	type	meaning
80	char	file magic and description
8	float double prec	P_min
8	float double prec	P_max
4	signed integer	N_P
8	float double prec	T_min
8	float double prec	T_max
4	signed integer	N_T
8*N_P	float double prec	values for P_min ... P_max at T_min
...		
8*N_P	float double prec	values for P_min ... P_max at T_max

The files with extension 'ppm' are images with same content as above. The values of density, isothermanl bulk modulus, and unpaired electrons are color-coded. Color scale N and contour width w values are appended to the output option as 'a_N.w'. Values of N are integer and range between 0 and 10. Values 0 and 1 correspond to gray and inverse gray, respectively. The next values select Imola, Bamako, Oslo, Hawaii, Acton, and Batlow scales which are adapted from Fabio Cramer. Then follow Inferno, Kindlmann, and Rainbow scales. The value of w lies between 0 and 1 with value 0 turning off contours.

The spin transition pressure and width are determined for each temperature from the 50 % and 20/80 % level of the unpaired electrons, respectively. At high temperatures these levels might lie outside the specified pressure range. Then the corresponding values for transition pressure and width are omitted from the output data.

5.7 Covariance matrix (option C)

The pressure dependent covariance matrix of volume, bulk modulus, and pressure derivative of the bulk modulus is calculated with parameter derivatives given by Eqs. 46 and 47 inserted into Eq. 34.

In addition, the pressure dependent covariance matrix of density and seismic velocity is calculated with parameter derivatives given by $\partial\rho/\partial x_j = -\rho \partial \ln V / \partial x_j$ and Eq. 48 inserted into Eq. 34.

These covariances are used to calculate pressure-dependent error ellipses using the procedure outlined in section 5.12 on the next page.

5.8 Spin entropy (option E)

The spin entropy per spin-state atom is calculated from the spin partition function given in (24) according to

$$S_s(V, T) = k_B \left(\ln Z_s + \frac{\partial \ln Z_s}{\partial \ln T} \right) = \frac{1}{T} \{ F_s(V, T) - U_s(V, T) \} \quad , \quad (51)$$

where F_s and U_s are spin free energy and spin internal energy, respectively. See also Eqs. (52) and (54).

5.9 Spin free energy (option F)

The spin Helmholtz free energy per spin-state atom is calculated from the spin partition function given in (24) and energies given by (26) according to

$$F_s(V, T) = -k_B T \ln \left\{ \sum_i G_i \exp \left[-\frac{1}{k_B T} \frac{\epsilon_i}{n(\epsilon_2 - \epsilon_1)} (\delta F_{el}(V) - \delta F_{el}(V_{tr})) \right] \right\} . \quad (52)$$

5.10 Specific heat (option H)

The specific heat at constant volume is calculated as the temperature derivative of (55) while keeping the volume V unchanged

$$c_V(V, T) = \left(\frac{\partial}{\partial T} \left\{ U_s(V, T) + 3k_B T I_3\left(\frac{\Theta(V)}{T}\right) + \frac{k_B T^2}{T_{con}} \right\} \right)_V . \quad (53)$$

5.11 Internal energy (option I)

The spin internal energy per spin-state atom is calculated from the spin partition function given in (24) and energies given by (26) according to

$$\begin{aligned} U_s(V, T) &= -k_B T \frac{\partial \ln Z_s}{\partial \ln T} \\ &= k_B T Z_s^{-1} \sum_i G_i \epsilon_i \exp \left[-\frac{1}{k_B T} \frac{\epsilon_i}{n(\epsilon_2 - \epsilon_1)} (\delta F_{el}(V) - \delta F_{el}(V_{tr})) \right] . \end{aligned} \quad (54)$$

The total internal energy is obtained by addition of elastic and thermal contributions

$$U(V, T) = F_{el}^{(1)}(V) + U_s(V, T) + 3k_B T I_3\left(\frac{\Theta(V)}{T}\right) + \frac{k_B T^2}{T_{con}(V)} , \quad (55)$$

where the Debye integral $I_3(\Theta/T)$, the Debye temperature $\Theta(V)$, and the conduction electron temperature $T_{con}(V)$ are given by Eqs. (11), (19), (21), (23), and (16), respectively.

5.12 Error ellipse (option L)

The error ellipsoid is defined as the hypersurface for which $\chi^2 = (1 + \alpha)\chi_{min}^2$. where $\alpha > 0$ determines the probability that the area of the ellipsoid contains the true parameter values. The error ellipse is a visualization of the correlation between two parameters. They are based on an expansion of χ^2 near its minimal value. With optimal parameter values \bar{x}_j we may write for small deviations $|h_j| \ll 1$

$$\begin{aligned} \chi^2((1 + h_j)\bar{x}_j) &= \chi_{min}^2 + \frac{1}{2} \sum_{jj'=1}^n h_j h_{j'} \bar{x}_j \bar{x}_{j'} \frac{\partial^2 \chi^2}{\partial x_j \partial x_{j'}} \\ &= \chi_{min}^2 + \chi_{min}^2 \sum_{jj'=1}^n h_j h_{j'} D_{jj'} . \end{aligned} \quad (56)$$

The matrix $D_{jj'}$ is symmetric and approximated by

$$D_{jj'} = \frac{\bar{x}_j \bar{x}_{j'}}{\chi_{min}^2} \left\{ \sum_{i=1}^N w_i \frac{\partial V(p_i, T_i)}{\partial x_j} \frac{\partial V(p_i, T_i)}{\partial x_{j'}} + \delta_{jj'} \frac{1}{\delta^2 X_j} \right\} = \bar{x}_j \bar{x}_{j'} [\sigma]_{jj'}^{-1} , \quad (57)$$

where $[\sigma]^{-1}$ is the inverse of the covariance matrix. With $\chi^2 = (1 + \alpha) \chi_{min}^2$ the equation defining the error ellipsoid is obtained

$$\sum_{jj'=1}^n h_j h_{j'} D_{jj'} = \alpha^2 \quad . \quad (58)$$

Visualization is afforded by singling out groups of two parameters h_l and $h_{l'}$. The corresponding reduced 2×2 covariance matrix is then used to produce the defining equation for the l, l' error ellipse

$$x_l^2 [\sigma]_{ll}^{-1} + 2x_l x_{l'} [\sigma]_{ll'}^{-1} + x_{l'}^2 [\sigma]_{l'l'}^{-1} = \alpha^2 \quad , \quad (59)$$

where $[\sigma]^{-1}$ is the inverse of the reduced covariance matrix. Now we introduce polar coordinates (r, ϕ) with the transformation $x_l = \sqrt{\sigma_{ll}} \alpha r \cos \phi$ and $x_{l'} = \sqrt{\sigma_{l'l'}} \alpha r \sin \phi$ to result in

$$r^2 = \frac{1 - c_{ll'}^2}{1 - c_{ll'} \sin 2\phi} \quad , \quad (60)$$

where $c_{ll'}$ is the correlation parameter defined by (33). The parametric form of the error ellipse follows accordingly

$$x_l = \alpha \cos \phi \left\{ \frac{\sigma_{ll} (1 - c_{ll'}^2)}{1 - c_{ll'} \sin 2\phi} \right\}^{1/2} \quad \text{and} \quad x_{l'} = \alpha \sin \phi \left\{ \frac{\sigma_{l'l'} (1 - c_{ll'}^2)}{1 - c_{ll'} \sin 2\phi} \right\}^{1/2} \quad . \quad (61)$$

The circumscribing box of this ellipse has the size $\alpha \sqrt{\sigma_{ll}} \times \alpha \sqrt{\sigma_{l'l'}}$. The scaling factor α determines the probability that the area of the ellipse contains the true parameter values. This probability is given by integration of the gaussian probability distribution over the ellipse area. The result for this two-dimensional case is $P(\alpha) = 1 - \exp[-\alpha^2/2]$ and gives 39 %, 86 %, and 99 % for α values of 1, 2, and 3, respectively. Here $\alpha = 1$ is selected to produce a circumscribing box of size $\sqrt{\sigma_{ll}} \times \sqrt{\sigma_{l'l'}}$. Values of 2 and 3 are used for optional pressure-dependent error ellipses.

5.13 f-F plot (option M)

The f-F plot permits a visual quality assessment for an EOS, where a normalized stress is plotted against a finite strain f . Based on the f-F plot one can often determine the compressional behavior of a material. For the Birch-Murnaghan EOS, the finite strain and the normalized stress are calculated as

$$f = \frac{1}{2} \left\{ \left(\frac{V_0}{V} \right)^{2/3} - 1 \right\} \quad \text{and} \quad F = \frac{p}{3f(1+2f)^{5/2}} \quad . \quad (62)$$

For the Vinet EOS, the finite strain is the same but the normalized stress is calculated as

$$F = \frac{p(1+2f)^{1/2}}{3(1+2f)((1+2f)^{1/2}-1)} \exp \left[-\frac{9(1+2f)^{1/2}}{2((1+2f)^{1/2}-1)} \right] \quad . \quad (63)$$

For the Logarithmic EOS, the finite strain and the normalized stress are calculated as

$$f = \frac{1}{3} \ln \left(\frac{V_0}{V} \right) \quad \text{and} \quad F = \frac{p}{3f(1+3f)} \exp[-3f] \quad . \quad (64)$$

This plot is most commonly used to test a 3rd order Birch-Murnaghan EOS for which one obtains a linear relationship between F and f with a slope directly related to K'_0

$$F = K_0 \left\{ 1 + \frac{3}{2}(K'_0 - 4)f \right\} \quad . \quad (65)$$

For the other EOS types, the above linear relation is approximate for small values of f .

5.14 Thermal and spin pressure (option P)

The thermal pressure and the spin pressure are calculated according to Eqs. (12) and (28), respectively.

5.15 Spin state populations (option S)

The probability η_j that spin state j in a particular spin-state atom is occupied is calculated as

$$\eta_j = Z_s^{-1} G_j \exp \left[-\frac{1}{k_B T} \frac{\epsilon_j}{n(\epsilon_2 - \epsilon_1)} (\delta F_{el}(V) - \delta F_{el}(V_{tr})) \right] . \quad (66)$$

5.16 Thermal expansion (option T)

Thermal expansion is calculated according to

$$\alpha = \frac{1}{V} \left(\frac{\partial V}{\partial T} \right)_p = -\frac{1}{V} \left(\frac{\partial S}{\partial p} \right)_T . \quad (67)$$

The entropy S has two contributions, thermal and spin state, and is given by

$$S(V, T) = S_s(V, T) - 3k_B \ln \left(2 \sinh \frac{\Theta}{2T} \right) + 4k_B I_3 \left(\frac{\Theta}{T} \right) + k_B \frac{3\Theta}{2T} + k_B \frac{2T}{T_{con}} , \quad (68)$$

where Θ and T_{con} are Debye and conduction electron temperatures. S_s and I were defined in Eqs. (51) and (11), respectively. If fit parameters are defined the variance of α is calculated using (34) with the derivative

$$\frac{\partial \alpha}{\partial x_j} = -\frac{\alpha}{V} \frac{\partial V}{\partial x_j} + \frac{1}{V} \frac{\partial}{\partial p} \left(\frac{\partial S}{\partial x_j} \right) . \quad (69)$$

5.17 Unpaired electrons (option U)

The expectation value of the number of unpaired electrons $\langle u \rangle$ is calculated with the spin-state occupation probabilities in (66) as

$$\langle u \rangle = \sum_i \eta_i u_i , \quad (70)$$

where u_i is the number of unpaired electrons in spin state i as defined in line 20 of the SIF.

5.18 $K_{\beta'}$ intensity (option X)

The intensity of the $K_{\beta'}$ x-ray emission line depends on the spin-state of the fluorescing atom, e.g., Fe. This intensity is modeled here by

$$I_K = \sum_i \frac{\eta_i u_i}{u_i + 1} , \quad (71)$$

where η_i and u_i are occupation probability and number of unpaired electrons of spin state i , respectively.

5.19 Crystal field splitting and spin-pairing energy (option Z)

The difference between crystal field splitting and spin-pairing energy is calculated as

$$\Delta - \Pi = \frac{1}{n(\epsilon_2 - \epsilon_1)} (\delta F_{el}(V) - \delta F_{el}(V_{tr})) \quad (72)$$

versus pressure and volume.

5.20 Free energy landscape (option Z)

For a given temperature, the free energy function

$$A(V, T) = pV + F_{el}(V, T) + F_{th}(V, T) - F_{th}(V, T_0) + F_s(V, T) \quad (73)$$

describes a surface in p - V space which has local minima along the volume direction. If spin contributions are present two local minima may occur in the spin transition region under certain conditions, e.g., at low temperature. The lowest minimum is the thermodynamically stable ground state and is equivalent to the EOS of (29). However, in practice the system could be kinematically trapped in the larger minimum leading to a hysteresis loop in the EOS. If two minima exist the free energy function versus volume for several pressures in the spin transition region is saved.

6 Melting and diffusion: simx

The simx program either calculates the nuclear forward scattering intensity as a function of temperature for experimental situations using melting induced by Laser heating or simulates the temperature average of nuclear resonant inelastic x-ray scattering spectra in Laser heating scenarios. The calculations are controlled by a set of parameters defined in the standard input file (SIF) as outlined below. The program optionally permits fitting of parameters to experimental melting data.

The SIF of the simx executable has 31 lines of data. The content of these lines is discussed in order of appearance.

1. Peak temperature (K) – one value :
This value gives the peak temperature of the Laser-heated spot.
2. Background temperature (K) – one value :
This value gives temperature of the material away from the heated area.
3. FWHM of profile (μm) – one value :
This value gives the full width at half maximum of the temperature profile created by Laser heating.
4. Shape coefficient of profile – one value :
This value defines the shape of the temperature profile created by Laser heating. The profile is described by a radially symmetric function of distance r from the point of peak temperature, $\exp[-(r/w)^\alpha]$, where α is the shape coefficient and $2w = \text{FWHM}/(\ln 2)^{1/\alpha}$ is given by the FWHM specified in line 3.
5. Radial range (μm) – two values :
The first value specifies the range used for calculations involving the temperature profile. If specified as negative or zero it will be calculated as $\text{FWHM}/2 \cdot 20^{1/\alpha}$ with FWHM from line 3 and shape coefficient α from line 4. This is the case for the GUI version. The second parameter specifies the number of steps for creation of the profile. It is set to 201 in the GUI version.
6. Spectrograph wavelength window (nm) – three values :
The two values specify the wavelength window that is considered for evaluation of the emission spectrum of the Laser-heated spot. The third value specifies the number of points in this window used for calculations.
7. Spectrograph aperture (μm) – one value :
This value gives the diameter of the aperture placed center on the temperature profile to evaluate the emission spectrum of the Laser-heated spot.
8. Horizontal offset of x-rays (μm) – one value :
This value gives the horizontal offset of the x-ray focal spot with respect to the temperature profile.
9. Horizontal FWHM of x-rays (μm) – one value :
This value gives the horizontal full width at half maximum of the x-ray focal spot.
10. Horizontal shape coefficient of x-ray profile – one value :
This value defines the horizontal shape of the x-ray profile. The profile is described by a symmetric function of distance x from the center of the x-ray focal spot, $\exp[-(x/w_h)^{\alpha_h}]$, where α_h is the shape coefficient and $2w_h = \text{FWHM}/(\ln 2)^{1/\alpha_h}$ is given by the FWHM specified in line 9.
11. Vertical offset of x-rays (μm) – one value :
This value gives the vertical offset of the x-ray focal spot with respect to the temperature profile.
12. Vertical FWHM of x-rays (μm) – one value :
This value gives the vertical full width at half maximum of the x-ray focal spot.
13. Vertical shape coefficient of x-ray profile – one value :
This value defines the vertical shape of the x-ray profile. The profile is described by a symmetric function of distance y from the center of the x-ray focal spot, $\exp[-(y/w_v)^{\alpha_v}]$, where α_v is the shape coefficient and $2w_v = \text{FWHM}/(\ln 2)^{1/\alpha_v}$ is given by the FWHM specified in line 12.
14. Transition energy (keV) – one value :
This value gives the nuclear resonant transition energy.

15. Recoil energy (keV) – one value :
This value gives the recoil energy of the free nucleus. The recoil energy is given by nuclear transition energy E_0 and isotope mass M via $E_0^2/(2Mc^2)$, where c is the speed of light.
16. Level width (neV) – one value :
This value gives the width of the nuclear excited state.
17. Execute melting simulation – one value :
Enter 'yes' to perform a melting simulation or fit to melting data. If not 'yes' then values in lines 18 to 27 are ignored.
18. Simulator engine – one value :
This value is not used at present, but is intended at some point in the future to engage an external program to perform the calculation of nuclear forward scattered intensity.
19. Lamb-Mössbauer temperature (K) – two values :
The first value gives the Lamb-Mössbauer temperature T_{LM0} of the material at the start of the temperature range T_0 given in line 25. The second value is optional and gives the temperature derivative of the Lamb-Mössbauer temperature T'_{LM} . The Lamb-Mössbauer factor is calculated as $\ln F_{LM}(T) = -T/(T_{LM0} + T'_{LM}(T - T_0))$.
20. Melting temperature (K) – one value :
This value specifies the melting temperature of the Laser-heated material containing the nuclear resonant atoms.
21. Self-diffusion constant (m^2/s) – one value :
This value specifies the self-diffusion coefficient of the nuclear resonant atom at infinitely large temperature.
22. Activation energy for self-diffusion (melting temperature) – one value :
This value specifies the activation energy ϵ for the self-diffusion process in units of the melting temperature T_m . The self-diffusion coefficient is then given by $D(T) = D_\infty \exp[-\epsilon T_m/T]$.
23. Effective thickness – one value :
This value specifies the effective thickness⁴ of the Laser-heated material containing the nuclear resonant atoms at the start of the temperature range given in line 25.
24. Time integration window (ns) – three values :
The first and second values give the start and end of the time window used for integration of the nuclear forward scattering signal. The third value specifies the number of steps for the integration.
25. Temperature scan range (K) – three values :
The first and second values give the start and end of temperature range used in the melting scan. The third value specifies the number of steps for the calculation.
26. Name of measured data file – many values :
The first value gives the name of a measured data file.⁵ The following values define the assignment of column numbers in the data file to physical parameters. Each assignment is a combination of one letter and a number, e.g., T1 or c15. The letter refers to the physical meaning according to Table 4 on page 25. The units refer to what is expected from input data. Data in the measured data file between temperatures T_1 and T_2 may be excluded from fitting by adding the line '#Mask T_1 T_2 ' to the file. Several lines of this type may be added to exclude specific regions of data.
27. Iteration control – three values :
The first value gives the maximum number of iteration steps. The other values are optional. The second value specifies the Levenberg-Marquardt parameter λ that is used to create a multiplier m to the diagonal

⁴The effective thickness η is defined as the product $\eta = F_{LM}\rho\sigma_0d$, where F_{LM} is the Lamb-Mössbauer factor, ρ is the number density of the nuclear resonant atoms, σ_0 is the nuclear resonant cross section, and d is the physical thickness. The effective thickness is dimensionless.

⁵The input file is a plain text file containing no more than 300 characters per line (excess characters will be ignored). Each line in the file contains a sequence of data values called columns separated by either spaces, tabs, or commata (csv format). Lines that do not begin with a number are ignored (after removing leading spaces, tabs, commata).

letter	physical meaning	units	comment
T	temperature	K	required
C	signal		required
t	temperature uncertainty	K	optional
c	signal uncertainty		required
s,S	thickness scaling		optional

Table 4: Column assignments for simx measured data input file.

of the second-derivative matrix of the normalized χ^2 . For iteration step n , the multiplier is given by $m = 1 + \lambda \max [0, (\chi_{n-1}^2 - \chi_n^2)/\chi_n^2]$. Larger values for λ have a dampening effect and steer the iteration process toward the gradient method. The Newton method is obtained for $\lambda = 0$ which is also the default value. The third value gives the required relative change of χ^2 value for an iteration step to stop the iteration.

28. Execute NRIXS simulation – one value :
Enter 'yes' to perform a NRIXS simulation. If not 'yes' then values in lines 29 to 32 are ignored.
29. Simulator engine – one value :
This value specifies the name of the psth executable. This executable is included in the PHOENIX software which is also available at <https://www.nrixs.com>. PHOENIX version 2.1.0 or higher has to be installed for this simulation to function.
30. Input data file name – four values :
The first value specifies the name of the input data file to be used by the psth executable.⁶ The file contains a phonon density of states (DOS) spectrum. The second and third values are optional. They give the column number of energy values and DOS values, respectively. If given both values must be specified. The defaults are one and two. The fourth value is optional and specifies the column number of DOS errors. If omitted the errors are calculated as square-root of the DOS values.
31. Energy range (meV) – three values :
The first and second values give the start and end of the energy range for psth calculations. The third value specifies the number of steps on this energy scale.
32. This line is reserved for future use.
33. Output file prefix and options – two values :
The first value gives the prefix for output file names.⁷ The second value is optional. If specified a value of ',' provides csv-formatted output files. The value 'G' produces headers readable by the 'xmgrace' graphics program. See Table 5 on page 26 for a summary of created output files. Not all files are created for the same calculation. File creation depends on input parameters.
In addition, the protocol file '<prefix>.ptl.txt' and, for fits to data, the results file '<prefix>_ite.csv' are created.

⁶The input file is a plain text file containing no more than 300 characters per line (excess characters will be ignored). Each line in the file contains a sequence of data values called columns separated by either spaces, tabs, or commata (csv format). Lines that do not begin with a number are ignored (after removing leading spaces, tabs, commata).

⁷Names may be defined either relative to the working directory or absolute: Prefix creates files in the working directory, ../Prefix creates files in the parent directory, myDir/Prefix creates files in the directory myDir (must exist) in the working directory, /Users/-data/Prefix creates files in the directory /Users/data (must exist and be writeable).

content	column 1	column 2,3	y units	file name
contours	x-coordinate	y-coordinate	$\mu\text{m}, \mu\text{m}$	<prefix>_con.<ext>
self-diffusion	temperature	self-diffusion coefficient	$\text{K}, \text{m}^2/\text{s}$	<prefix>_dff.<ext>
intensity distrib.	eff. thickness	$dI/d\eta$	NA	<prefix>_eff.<ext>
reduced data	temperature	counts, errors	K, NA	<prefix>_exp.<ext>
fitted theory	temperature	counts	K, NA	<prefix>_fit.<ext>
NRIXS spectrum	energy	excitation prob.	$\text{meV}, \text{eV}^{-1}$	<prefix>_ixs.<ext>
NFS signal	spectrogr. temp.	intensity	K, Γ	<prefix>_mls.<ext>
NFS signal	peak temp.	intensity	K, Γ	<prefix>_mlt.<ext>
fit residuals	temperature	data-fit	K, NA	<prefix>_rsd.<ext>
time spectrum	time	intensity	$\text{ns}, \Gamma/\text{ns}$	<prefix>_sms.<ext>
average temp.	radius	temperature	$\mu\text{m}, \text{K}$	<prefix>_tav.<ext>
temp. profile	radius	temperature	$\mu\text{m}, \text{K}$	<prefix>_tpr.<ext>
spectrograph temp.	radius	temperature	$\mu\text{m}, \text{K}$	<prefix>_tsp.<ext>
intensity distrib.	temperature	dI/dT	K, K^{-1}	<prefix>_xdt.<ext>

Table 5: Output created by simx calculations. <prefix> is the file prefix specified in line 33. The extension <ext> is either 'csv' for comma-separated-value files (option ',' in line 33) or 'dat' otherwise. NA means specification not applicable.

6.1 Temperature profile

The Laser-heated area on the sample is described by a radially symmetric temperature field

$$T(r) = T_b + (T_p - T_b)\Theta(r) \quad \text{with} \quad \Theta(r) = \exp[-(r/w)^\alpha], \quad (74)$$

with center maximum temperature T_p and background temperature T_b given in lines 1 and 2, respectively. Here α is the shape coefficient given in line 4, and $2w = \text{FWHM}/(\ln 2)^{1/\alpha}$ is given by the FWHM specified in line 3. The average temperature in a disc of radius R is then given by

$$T(R) = \frac{2}{R^2} \int_0^R r T(r) dr = T_b + (T_p - T_b) \frac{2}{R^2} \int_0^R r \Theta(r) dr \quad . \quad (75)$$

6.2 Temperature from spectral analysis

Ideally a segment of the heated sample at a particular temperature T emits black-body radiation described by a Planck intensity distribution

$$\frac{dP}{dA dE d\Omega} = \frac{2E^3}{h^3 c^2} \left(\exp \left[\frac{E}{k_B T} \right] - 1 \right)^{-1} \quad , \quad (76)$$

where dP is the power per energy dE emitted by area dA into solid angle $d\Omega$ normal to the area, and k_B is the Boltzmann constant. The Laser-heated area then emits the following power per energy and solid angle into an aperture of radius R

$$\frac{dP}{dE d\Omega} = \frac{2E^3}{h^3 c^2} \pi R^2 G(E) \quad \text{with} \quad G(E) = 2 \int_0^1 \left(\exp \frac{E}{k_B T(\rho R)} - 1 \right)^{-1} \rho d\rho \quad . \quad (77)$$

This function is fitted to a Planck distribution with temperature T_S which is then considered the spectroscopically measured average temperature inside a disc with radius R . This measured temperature is implicitly given as a solution of the optimization procedure

$$\int_{E_1}^{E_2} dE E^6 \left\{ G(E) - A \left(\exp \left[\frac{E}{k_B T_S} \right] - 1 \right)^{-1} \right\}^2 \rightarrow \text{Min}(A, T_S) \quad , \quad (78)$$

where A and T_S are varied. The spectral range for analysis is determined by E_1 and E_2 given in line 6. In practice, the spectral range endpoints are between one and two eV and temperatures involved are typically around 3000 K

corresponding to energies of about 0.25 eV. Then the exponentials in Eqs. (77) and (78) are large, and we may approximate

$$\left(\exp \left[\frac{E}{k_B T} \right] - 1 \right)^{-1} \approx \exp \left[-\frac{E}{k_B T} \right] . \quad (79)$$

We introduce the parameter $\beta = 1/(k_B T_S)$ and rewrite the approximate optimization problem in (78) as follows

$$Z(\beta, A) = \int_{E_1}^{E_2} dE E^6 \{G(E) - A e^{-\beta E}\}^2 \quad \text{with} \quad \frac{\partial Z}{\partial \beta} = 0 \wedge \frac{\partial Z}{\partial A} = 0. \quad (80)$$

The evaluation of the derivatives gives

$$\begin{aligned} \frac{\partial Z}{\partial \beta} &= 2A \int_{E_1}^{E_2} dE E^7 e^{-\beta E} \{G(E) - A e^{-\beta E}\} \\ \frac{\partial Z}{\partial A} &= -2 \int_{E_1}^{E_2} dE E^6 e^{-\beta E} \{G(E) - A e^{-\beta E}\} , \end{aligned}$$

and we obtain the optimal value for $T_S = 1/(k_B \beta)$ by eliminating A from above equations and solving

$$\int_{E_1}^{E_2} E^7 e^{-2\beta E} dE \int_{E_1}^{E_2} G(E) E^6 e^{-\beta E} dE = \int_{E_1}^{E_2} E^6 e^{-2\beta E} dE \int_{E_1}^{E_2} G(E) E^7 e^{-\beta E} dE . \quad (81)$$

The optical aperture of the spectrograph system as specified in line 7 is given by $2R$, and the solutions of the previous equation determine the average temperature obtained by fitting the optical emission spectrum to a Planck distribution.

6.3 X-ray profile

X-rays incident on the Laser-heated sample are described by a symmetric intensity profile $I_0 P(x, y)$ peaked at x_0, y_0 with

$$P(x, y) = \frac{\alpha_h \alpha_v}{w_h w_v \Gamma(1/\alpha_h) \Gamma(1/\alpha_v)} \exp[-|(x - x_0)/w_h|^{\alpha_h} - |(y - y_0)/w_v|^{\alpha_v}] , \quad (82)$$

where x_0 is the horizontal offset (line 8), $2w_h = \text{FWHM}_h/(\ln 2)^{1/\alpha_h}$ is given by the horizontal FWHM (line 9), and α_h is the horizontal shape coefficient (line 10). In analogy, y_0 is the vertical offset (line 11), $2w_v = \text{FWHM}_v/(\ln 2)^{1/\alpha_v}$ is given by the vertical FWHM (line 12), and α_v is the vertical shape coefficient (line 13). The distribution $P(x, y)$ is area-normalized, and Γ is the gamma function.

The x-ray intensity profile intersects the temperature profile given in (74). For simulations of temperature dependent x-ray experiments, we need to know the x-ray intensity associated with a given temperature. The radial positions corresponding to a given temperature T are obtained by inverting (74), i.e., $r(T) = w \ln^{1/\alpha}[(T_p - T_b)/(T - T_b)]$. The x-ray intensity integrated over this circle then leads to the normalized intensity distribution according to

$$\begin{aligned} X(T) &= -r \frac{dr}{dT} \int_0^{2\pi} P(r \cos \phi, r \sin \phi) d\phi \\ &= \frac{w^{\alpha} r^{(2-\alpha)}}{\alpha (T - T_b)} \int_0^{2\pi} P(r \cos \phi, r \sin \phi) d\phi . \end{aligned} \quad (83)$$

6.4 Nuclear forward scattering signal

The time-dependent nuclear forward scattering (NFS) intensity or time spectrum of a single-line absorber is given by

$$I(t) = I_0 \frac{\eta \tau}{t} J_1^2(\sqrt{\eta t / \tau}) \exp[-t/\tau] , \quad (84)$$

where J_1 is the first-order Bessel function, η is the effective thickness, and the nuclear lifetime $\tau = \hbar/\Gamma$ is derived from the nuclear level width Γ (line 16). The time-integrated NFS intensity is then obtained as

$$N(T) = \int_{t_1}^{t_2} I(t) dt = I_0 \eta(T) \tau \int_{t_1/\tau}^{t_2/\tau} J_1^2(\sqrt{\eta(T)z}) e^{-z} \frac{dz}{z} , \quad (85)$$

where $[t_1, t_2]$ is the time window specified in line 24, and the temperature dependence of the effective thickness via the Lamb-Mössbauer factor is modeled by

$$\eta(T) = \begin{cases} \eta_0 \exp[-\frac{T-T_0}{T_{LM0}+T'_{LM}(T-T_0)}] & \text{if } T < T_m \\ 0 & \text{if } T \geq T_m \end{cases} . \quad (86)$$

Here T_0 is the start of the temperature range (line 25), η_0 and T_{LM0} are effective thickness and Lamb-Mössbauer temperature at T_0 (lines 23 and 19), respectively, and T'_{LM} gives the temperature derivative of the Lamb-Mössbauer temperature. At the melting temperature T_m , the effective thickness and thus the NFS intensity suddenly drop to zero.

In case of self-diffusion of the nuclear resonant atoms, the nuclear lifetime $\tau = \hbar/\Gamma$ in (85) has to be replaced with

$$\tau(T) = \frac{\hbar}{\Gamma} \left\{ 1 + \frac{2E_0^2}{\hbar c^2 \Gamma} D(T) \right\}^{-1} , \quad (87)$$

where E_0 is the nuclear resonant transition energy (line 14), and the temperature dependent diffusion constant is given by

$$D(T) = D_0 \exp \left[-\epsilon \frac{T_m}{T} \right] . \quad (88)$$

Here D_0 (line 21) is the self-diffusion constant at infinite temperature, and ϵ (line 22) specifies the activation energy for the self-diffusion process in units of the melting temperature T_m (line 20).

Finally the time- and temperature-integrated NFS intensity or NFS signal is obtained by combining Eqs. (83) and (85)

$$\begin{aligned} \langle N \rangle &= \int_{T_b}^{T_p} N(T) X(T) dT \\ &= I_0 \frac{w^\alpha}{\alpha} \int_{T_b}^{T_p} dT \frac{\eta \tau r^{(2-\alpha)}}{(T-T_b)} \int_{t_1/\tau}^{t_2/\tau} J_1^2(\sqrt{\eta z}) e^{-z} \frac{dz}{z} \int_0^{2\pi} P(r \cos \phi, r \sin \phi) d\phi , \end{aligned} \quad (89)$$

where $r(T) = w \ln^{1/\alpha}[(T_p - T_b)/(T - T_b)]$, and the temperature dependences of η and τ are given by Eqs. (86) and (87), respectively.

6.5 NRIXS signal

Temperature dependent NRIXS spectra are calculated using the psth module of the PHOENIX program package. PHOENIX version 2.1.0 or higher has to be installed for this calculation to work. For temperature T , the NRIXS spectrum $S(E, T)$ is calculated from the given density-of-states $g(E)$ via

$$\begin{aligned} S(E, T) &= F_{LM}(T) \left\{ a R(E) + \sum_{n=1}^{\infty} \int G_n(E', T) R(E - E') dE' \right\} \\ F_{LM}(T) &= \exp \left[-E_R \int \frac{g(E)}{E} \coth \frac{\beta E}{2} dE \right] \\ G_{n+1}(E) &= \frac{1}{n+1} \int G_n(E') G_1(E - E') dE' \\ G_1(E) &= \frac{E_R g(E)}{E(1 - e^{-\beta E})} , \end{aligned} \quad (90)$$

where E_R is the recoil energy (line 15), and $\beta = 1/(k_B T)$ is the inverse temperature. In the above equations, $g(E)$ is normalized, i.e., $\int g(E) dE = 1$, and $g(E) = 0$ for $E \leq 0$. The value of a is set to 0.2, and the resolution function is given by

$$R(E) = \frac{\alpha}{w \Gamma(1/\alpha)} \exp[-|E/w|^\alpha] \quad , \quad (91)$$

where $\alpha = 1.7$ and $w = 0.62$ meV are chosen to produce a reasonable shape and a FWHM of 1 meV. The temperature average is performed using the x-ray intensity distribution of (83)

$$\begin{aligned} \langle S(E) \rangle &= \int_{T_b}^{T_p} S(E, T) X(T) dT \\ &= \frac{w^\alpha}{\alpha} \int_{T_b}^{T_p} dT \frac{F_{LM}(T) r^{(2-\alpha)}}{(T - T_b)} \int_0^{2\pi} P(r \cos \phi, r \sin \phi) d\phi \times \\ &\quad \left\{ a R(E) + \sum_{n=1}^{\infty} \int G_n(E', T) R(E - E') dE' \right\} \quad , \end{aligned} \quad (92)$$

where $r(T) = w \ln^{1/\alpha}[(T_p - T_b)/(T - T_b)]$,

6.6 Parameter fitting

For melting runs, Input data (EIF given in line 26) are specified as N sets $\{T_i, C_i, s_i, \delta T_i, \delta C_i\}$ consisting of temperature, counts (measured signal), their uncertainties, and effective thickness scaling factor. The NFS signal given by (89) is fitted to these data by variation of n parameters $\{x_j\}$ by minimizing the normalized mean-square deviation (method of weighted least squares)

$$\chi^2 = \frac{1}{(N + p - n)} \left\{ \sum_{i=1}^N w_i (C_i - x_1 S(T_p(T_i), s_i \eta_0, \{x_j\}))^2 + \sum_{k=1}^p \frac{(x_k - X_k)^2}{\delta^2 X_k} \right\} \quad . \quad (93)$$

Here the scaling parameter x_1 is always fitted. The peak temperature of the Laser-heated area T_p corresponding to a measured temperature T_i is obtained by inversion of (81). The effective thickness (line 23) is scaled by the s_i where $s_1 = 1$ for T_1 as the lowest temperature is enforced. Some number p of the fit parameters may have priors X_k with uncertainties δX_k . The weights w_i are determined by uncertainties of the data via

$$w_i^{-1} = \delta^2 C_i + \left(x_1 \frac{\partial S}{\partial T} \right)^2 \delta^2 T_i \quad , \quad (94)$$

where K_i and α_i are bulk modulus and thermal expansion calculated from the EOS at $\{p_i, T_i\}$, respectively. If data and priors constrain the fit parameters reasonably well a solution with minimal χ^2 is obtained. These optimal parameters have errors related to data variation and prior uncertainties. The errors are estimated as variances (square errors) by

$$\delta^2 x_j = \sqrt{\sigma_{jj}} \quad \text{with} \quad \sigma_{jj'} = \chi_{min}^2 \left[\sum_{i=1}^N w_i \frac{\partial(x_1 S(T_i, s_i))}{\partial x_l} \frac{\partial(x_1 S(T_i, s_i))}{\partial x_{l'}} + \delta_{jj'} \frac{1}{\delta^2 X_j} \right]_{jj'}^{-1} \quad , \quad (95)$$

where $\sigma_{jj'}$ is the covariance matrix and $[\dots]^{-1}$ denotes matrix inversion. The derivatives are calculated with the optimal parameters. The fit-parameter correlation matrix is given by

$$c_{jj'} = \frac{\sigma_{jj'}}{\sqrt{\sigma_{jj} \sigma_{j'j'}}} \quad . \quad (96)$$

The variance of a function F of the fit parameters can be calculated as

$$\delta^2 F(\{x_j\}) = \sum_{jj'=1}^n \sigma_{jj'} \frac{\partial F}{\partial x_j} \frac{\partial F}{\partial x_{j'}} \quad , \quad (97)$$

6.6.1 Fisher information

Fit parameters can be highly correlated leading to overinterpretation of data. The Fisher-information matrix can help identify parameters that are weakly defined by the data. In the context of the minimum χ^2 as maximum likelihood estimator, the Fisher information is calculated as

$$I_{jj'} = \sqrt{\sigma_{jj}\sigma_{j'j'}} \frac{\partial^2 \chi^2}{\partial x_j \partial x_{j'}} \quad , \quad (98)$$

where parameter values are obtained by minimizing χ^2 . The smallest eigenvalues of the Fisher information matrix indicate fit parameter combinations that are least important for minimizing χ^2 . An individual fit parameter x_k is tested by calculating a modified covariance matrix

$$\sigma'_{jj'} = \chi_{min}^2 \left[\sum_{i=1}^N w_i \frac{\partial V(p_i, T_i)}{\partial x_l} \frac{\partial V(p_i, T_i)}{\partial x_{l'}} + \delta_{ll'} \frac{1}{\delta^2 X_l} + \delta_{lk} \delta_{l'k} \frac{\lambda_k}{\sigma_{kk}} \right]_{jj'}^{-1} \quad , \quad (99)$$

where $\lambda_k = 20000$ basically fixes the parameter. The modified Fisher information matrix is then

$$I'_{jj'} = \sqrt{\sigma'_{jj}\sigma'_{j'j'}} \frac{\partial^2 \chi^2}{\partial x_j \partial x_{j'}} + \delta_{jk} \delta_{j'k} \lambda_k \quad . \quad (100)$$

If fit parameter x_k is only weakly determined by the data the smallest eigenvalues of the modified Fisher information matrix are significantly larger than the smallest eigenvalues of the full Fisher information matrix given in (98).

6.7 Output files

The file prefix specified in line 33 of the SIF is used for output files containing calculation results. Table 5 on page 26 shows a summary of output files created by the `simx` executable.

6.7.1 Contours

Temperature and x-ray intensity profiles are given by Eqs. (74) and (82), respectively. They are plotted together as contours on a x-y grid. Also plotted is the spectrograph aperture as a circle with diameter given in line 7 of the SIF.

6.7.2 X-ray intensity distributions

The temperature-dependent x-ray intensity distribution as given in (83) is provided in the `'_xdt'` file. The `'_eff'` file contains the effective thickness dependent x-ray intensity distribution defined by

$$\begin{aligned} X'(\eta) &= -r \frac{dr}{d\eta} \int_0^{2\pi} P(r \cos \phi, r \sin \phi) d\phi \\ &= \frac{\eta w^\alpha r^{(2-\alpha)} x^3}{\alpha T_{LM0} ((T_0 - T_b)x + T_{LM0} \ln \eta_0 / \eta)} \int_0^{2\pi} P(r \cos \phi, r \sin \phi) d\phi \end{aligned} \quad (101)$$

with $x(\eta) = 1 - T'_{LM} \ln \eta_0 / \eta$ and

$$r(\eta) = w \ln^{1/\alpha} \left[(T_p - T_b) / (T_0 - T_b + \frac{T_{LM0}}{x} \ln \eta_0 / \eta) \right] \quad . \quad (102)$$

6.7.3 Temperature

The `'_tpr'` file contains the radially symmetric temperature profile of the Laser-heated area and is calculated according (74). The average temperature in a disc of specified radius is given by (75) and saved in the `'_tav'` file. The `'_tsp'` file contains the temperature of the Laser-heated area as seen by an apertured spectrograph according to (81) versus the radius of the aperture. It is assumed that the aperture is centered on the temperature profile.

6.7.4 NFS related

The '_mlt' and '_mls' files contain the NFS signal calculated by (89) versus peak temperature and spectrographic temperature, respectively. The '_sms' file gives a time spectrum according to (84) with effective thickness and diffusion coefficient taken at the start temperature T_0 specified in line 25 of the SIF.

If experimental data is fitted three additional files are created: the fit result in the '_fit' file; the residuals in the '_rsd' file; the reduced experimental data in the '_exp' file. The latter is different from the input experimental data only if effective scaling is specified in line 26 of the SIF. The reduced experimental data is calculated as

$$Y_i = C_i + x_1 \{S(T_p(T_i), s_i \eta_0, \{x_j\}) - S(T_p(T_i), \eta_0, \{x_j\})\} \quad . \quad (103)$$

The '_ixs' file contains the temperature averaged NRIXS signal calculated by (92).

7 Linear mixing model: smix

The smix program takes properties of a set of minerals and tries to match them to target properties.

The SIF of the smix executable has at least 10 lines of data. The content of the 10 input lines is discussed in order of appearance.

1. Output file prefix and option – two values :
The first value gives the prefix for output file names.⁸ Generated output files are listed in Table 6. Output files are plain text files. If option ',' is specified csv-formatting is provided. If option G is specified a header for the xmgrace graphical program is added. A detailed description of the procedures used to produce the output is provided in sections 5.6 on page 18 to 5.20 on page 22.
In addition, the protocol file '<prefix>_ptl.txt' and the results file '<prefix>_ite.csv' are created.
2. Iteration control – four values :
The first value gives the maximum number of iteration steps. The other values are optional. The second value specifies the Levenberg-Marquardt parameter λ that is used to create a multiplier m to the diagonal of the second-derivative matrix of the normalized χ^2 . For iteration step n , the multiplier is given by $m = 1 + \lambda \max [0, (\chi_{n-1}^2 - \chi_n^2)/\chi_n^2]$. Larger values for λ have a dampening effect and steer the iteration process toward the gradient method. The Newton method is obtained for $\lambda = 0$ which is also the default value. The third value gives the required relative change of χ^2 value for an iteration step to stop the iteration. The fourth value is either 'yes' or 'no' to indicate whether mineral concentration value should be restricted to physical values in $[0,1]$.
3. Target property names – many values :
A space-separated list of names for the target properties.
4. Target property meaning – many values :
A space-separated list of integers to assign physical meaning to target properties: 0, density (g/cm³); 1, bulk modulus (GPa); 2, shear modulus (GPa); 3, inverse Bulk modulus (1/GPa); 4, inverse shear modulus (1/GPa); other, undefined.
5. Target property values – many values :
A space-separated list of values for the target properties.
6. Target property uncertainties – many values :
A space-separated list of uncertainties of values for the target properties.

column 1	column 2	x units	y units	file name
parameter n	parameter m	variable	variable	<prefix>_p<n>v<m>_eel.<ext>
density	P-wave velocity	g/cm ³	km/s	<prefix>_vP_ds_eel.<ext>
density	S-wave velocity	g/cm ³	km/s	<prefix>_vS_ds_eel.<ext>
P-wave velocity	S-wave velocity	km/s	km/s	<prefix>_vS_VP_eel.<ext>

Table 6: Output options for mineral property mixing calculations. <prefix> is the file prefix specified in line 1. The extension <ext> is either 'csv' for comma-separated-value files (option ',' in line 1) or 'dat' otherwise.

From here on each mineral is defined by a block of four input lines. These four lines consist of

1. Mineral name – one values :
The name of the mineral.
2. Mineral property values – many values :
A space-separated list of property values for this mineral.
3. Mineral property uncertainties – many values :
A space-separated list of uncertainties of property values for this mineral.

⁸Names may be defined either relative to the working directory or absolute: Prefix creates files in the working directory, ../Prefix creates files in the parent directory, myDir/Prefix creates files in the directory myDir (must exist) in the working directory, /Users/-data/Prefix creates files in the directory /Users/data (must exist and be writeable).

4. Mineral property concentration – one values :
The concentration of this mineral in the assemblage.

7.1 Linear mixing

The mixture contains $(N + 1)$ individual minerals. The concentrations $p_j \geq 0$ of the minerals in the mixture are normalized by

$$\sum_{j=1}^{N+1} p_j = 1 \quad . \quad (104)$$

The minerals have properties ϕ_{jk} , where j and k number mineral types and properties, respectively. The average properties are then

$$\langle \phi_k \rangle = \sum_{j=1}^{N+1} p_j \phi_{jk} = \phi_{N+1,k} + \sum_{j=1}^N p_j (\phi_{j,k} - \phi_{N+1,k}) \quad , \quad (105)$$

where the concentration p_{N+1} was eliminated by use of (104). The measure for matching these average properties and the target properties Φ_k is defined as

$$M(\{p_j\}) = \frac{1}{P + L - n} \left\{ \sum_{k=1}^P w_k (\langle \phi_k \rangle - \Phi_k)^2 + \sum_{k=1}^L \frac{(p_k - X_k)^2}{\delta^2 X_k} \right\} \quad . \quad (106)$$

where P is the number of properties and n is the number of fitted concentrations. Some number L of the fitted concentrations may have priors X_k with uncertainties δX_k . The weights w_k are taken as

$$w_k = \left(\delta \Phi_k^2 + \sum_{j=1}^{N+1} p_j^2 \delta \phi_{jk}^2 \right)^{-1} \quad , \quad (107)$$

where $\delta \Phi_k^2$ and $\delta \phi_{jk}^2$ are the variances of properties Φ_k and ϕ_{jk} , respectively. Concentration p_n is constrained to physical parameter space by adding the terms $\alpha \Theta(p_n - 1) (p_n - 1)^2$ and $\alpha \Theta(-p_n) p_n^2$ to the measure. Here $\Theta()$ is the Heavyside step function, and $\alpha = 10^4$ for steep constraints.

The measure (106) depends on N independent concentrations, but the number of properties P that determine the measure create constraints for the process $M \rightarrow \min_{\{p_j\}}$. For the smix program, N must be smaller than $P + L$, e.g., three target properties allow only two mineral concentrations without priors to be fitted.

The fit procedure is equivalent to that described in 5.3 on page 15 with χ^2 replaced with the measure (106).

8 Sound velocities: svec

The svec program performs sound velocity calculation and averages based on the elastic tensor of a material.

The SIF of the svec executable has 10 lines of data. The content of the 10 input lines is discussed in order of appearance.

1. Crystal system – one value :
This value defines the symmetry of the unit cell of the material relevant to the elastic tensor. Supported values are cubic, hexagonal, trigonal, tetragonal, orthorhombic, monoclinic, triclinic.
2. Density (g/cm^3) – one value :
This value specifies the density of the material.
3. Elastic constant (GPa) – one value :
This value specifies C_{11} .
4. Elastic constants (GPa) – two values :
These values specify C_{21} and C_{22} .
5. Elastic constants (GPa) – three values :
These values specify C_{3n} with $n = 1 \dots 3$.
6. Elastic constants (GPa) – four values :
These values specify C_{4n} with $n = 1 \dots 4$.
7. Elastic constants (GPa) – five values :
These values specify C_{5n} with $n = 1 \dots 5$.
8. Elastic constants (GPa) – six values :
These values specify C_{6n} with $n = 1 \dots 6$.
9. Angle grid – two values :
The first value specifies the number of grid points N for the polar angle θ . The second value is optional and specifies the number of grid points M for the azimuthal angle φ . If the second value is not given then $M = N$. The GUI version has three settings: coarse, fine, very fine. Depending on the crystal system they translate to N, M values.
10. Velocity resolution (km/s) – one value :
This value specifies the size of the velocity bins for the calculation of velocity probability distributions. The GUI version assigns 0.05, 0.02, or 0.01 to this value for settings coarse, fine, very fine, respectively.
11. name of output file prefix and options – many values :
The first value gives the prefix for output file names.⁹ Output files are plain text files. If option 'c' is specified csv-formatting is provided. If option G is specified a header for the xmgrace graphical program is added. Output files are created according to options following the file name prefix. With exception of options S, the first value is the sound velocity in units of km/s . The other values depend on the option specified and their physical meaning is listed in Table 7 on page 35. Ascii output files created with the S option are compressed if the 'gzip' program is available. A detailed description of the procedures used to produce the output is provided in the following sections.
In addition, the protocol file '<prefix>_ptl.txt' is created.

⁹Names may be defined either relative to the working directory or absolute: Prefix creates files in the working directory, ../Prefix creates files in the parent directory, myDir/Prefix creates files in the directory myDir (must exist) in the working directory, /Users/-data/Prefix creates files in the directory /Users/data (must exist and be writeable).

option	description	units	file name
a,A	sound velocity probability distribution	km/s, (km/s) ⁻¹	<prefix>_asv.<ext>
l,L	probability distribution, longitudinal	km/s, (km/s) ⁻¹	<prefix>_lsv.<ext>
t,T	transverse, polar type	km/s, (km/s) ⁻¹	<prefix>_t1v.<ext>
	transverse, azimuthal type	km/s, (km/s) ⁻¹	<prefix>_t2v.<ext>
s,S	sound velocity surface, longitudinal	km/s	<prefix>_L.3D.gz
	azimuthal orthographic projection		<prefix>_A.L.ppm
	mercator projection		<prefix>_M.L.ppm
	transverse, type 1	km/s	<prefix>_T1.3D.gz
	azimuthal orthographic projection		<prefix>_A_T1.ppm
	mercator projection		<prefix>_M_T1.ppm
	transverse, type 2	km/s	<prefix>_T2.3D.gz
	azimuthal orthographic projection		<prefix>_A_T2.ppm
	mercator projection		<prefix>_M_T2.ppm
	seismic velocity	km/s	<prefix>_SV.3D.gz
	azimuthal orthographic projection		<prefix>_A_SV.ppm
	mercator projection		<prefix>_M_SV.ppm

Table 7: Output options for sound velocity calculations. <prefix> is the file prefix specified in line 11. The extension <ext> is either 'csv' for comma-separated-value files (option ', ' in line 11) or 'dat' otherwise.

8.1 Sound velocity calculation

Assume E_{ijkl} is the cartesian representation of the fourth-rank elastic tensor of the crystal, and ρ is the density. For a given propagation direction $\mathbf{q} = (q_1, q_2, q_3)$, sound velocities v_s and normalized polarization vectors $\mathbf{e}^{(s)}(\mathbf{q}) = (e_1^{(s)}, e_2^{(s)}, e_3^{(s)})$ are obtained as solutions of the eigenvalue problem

$$\rho v_s^2 e_i^{(s)} = \sum_{jkl} E_{ijkl} q_j q_k e_l^{(s)} \quad i, j, k, l, s = 1 \dots 3 \quad . \quad (108)$$

For each direction in the crystal, (108) provides three sound velocities and polarization vectors, and the symmetry of E_{ijkl} assures that the polarization vectors are mutually orthogonal. The elastic constants C_{ij} specified in input lines 3-8 represent the elastic tensor using the condensed Voigt notation, i.e., $C_{11} = E_{1111}$, $C_{12} = E_{1122}$, $C_{13} = E_{1133}$, $C_{44} = E_{1212}$, etc.

Sound velocities and their polarization vectors are calculated on a discrete angular grid of spherical coordinates $\eta = \cos \theta$ and φ given by

$$\eta_n = \frac{n-1}{N-1} \quad \varphi_m = \frac{m-1}{M} \varphi_0 \quad , \quad (109)$$

where $1 \leq n \leq N$, $1 \leq m \leq M$ and N, M are specified in line 9 of the SIF. The polar angle ranges are determined by the crystal symmetry specified in line 1 of the SIF: cubic, hexagonal, $\varphi_0 = \pi/2$; orthorhombic, tetragonal, $\varphi_0 = \pi$; trigonal, monoclinic, triclinic $\varphi_0 = 2\pi$. Using the spherical coordinates, directions of sound waves are expressed by

$$q_1 = \sin \theta_n \cos \varphi_m \quad q_2 = \sin \theta_n \sin \varphi_m \quad q_3 = \cos \theta_n \quad . \quad (110)$$

The propagation direction of \mathbf{q} together with the transverse polar and azimuthal unit vectors \mathbf{t}_1 and \mathbf{t}_2 , respectively, given by

$$\mathbf{t}_1 = \begin{pmatrix} \cos \theta_n \cos \varphi_m \\ \cos \theta_n \sin \varphi_m \\ -\sin \theta_n \end{pmatrix} \quad \text{and} \quad \mathbf{t}_2 = \begin{pmatrix} -\sin \varphi_m \\ \cos \varphi_m \\ 0 \end{pmatrix} \quad . \quad (111)$$

form a reference frame for sound waves. Sound velocities modes are characterized according to the projections of their polarization vectors onto the basis $(\mathbf{q}, \mathbf{t}_1, \mathbf{t}_2)$. The longitudinal sound (or p-wave) is defined by $\max_s [|\mathbf{q} \cdot \mathbf{e}^{(s)}|]$, Transverse sound (or s-waves) of polar and azimuthal type is defined by $\max_s [|\mathbf{t}_1 \cdot \mathbf{e}^{(s)}|]$ and $\max_s [|\mathbf{t}_2 \cdot \mathbf{e}^{(s)}|]$, respectively. The $N \cdot M$ number of velocity values are arranged into probability distributions (or binned). The probability distribution of sound velocities is calculated as

$$D(v) = \frac{1}{NM \delta v} \sum_{v \leq v' < v + \delta v} \sum_s \left(\mathbf{a} \cdot \mathbf{e}^{(s)}(v') \right)^2 \quad , \quad (112)$$

where δv is the velocity bin size given in line 10 of the SIF, and $\mathbf{a} \in (\mathbf{q}, \mathbf{t}_1, \mathbf{t}_2)$.

8.2 Elastic moduli

The bulk modulus is calculated as

$$K = \left(\sum_{ij=1}^3 [C^{-1}]_{ij} \right)^{-1},$$

where C^{-1} is the inverse of the elastic constant matrix. Voigt bounds of bulk and shear moduli are calculated as

$$K_V = \frac{1}{3} (\langle E \rangle_{1111} + 2\langle E \rangle_{1122}) \quad G_V = \langle E \rangle_{2323},$$

where $\langle E \rangle$ is the spatial average of the fourth-rank elastic tensor of the crystal. Reuss bounds of bulk and shear moduli are calculated using the inverse of the spatial average of the fourth-rank compliance tensor $\langle E^{-1} \rangle^{-1}$ and are given by

$$K_R = \frac{1}{3} ([\langle E^{-1} \rangle^{-1}]_{1111} + 2[\langle E^{-1} \rangle^{-1}]_{1122}) \quad G_R = [\langle E^{-1} \rangle^{-1}]_{2323}.$$

Voigt-Reuss-Hill bounds of the bulk and shear moduli are the averages $K_{VRH} = (K_V + K_R)/2$ and $G_{VRH} = (G_V + G_R)/2$. The universal elastic anisotropy is calculated as

$$A_U = \frac{K_V}{K_R} + 5 \frac{G_V}{G_R} - 6.$$

Bounds of the sound velocities are calculated as

$$v_P = \sqrt{\frac{1}{\rho} \left(K + \frac{4}{3} G \right)} \quad v_S = \sqrt{\frac{G}{\rho}},$$

where v_P and v_S are the compressional and shear sound velocities, respectively, and K, G are the respective bounds of the elastic moduli.

8.3 Sound velocity averages

With the set of sound velocities and polarization vectors calculated using (108) various projected averages of powers of the p-wave sound velocities are calculated via

$$\langle v_P^\alpha \rangle_j = \sum_{\mathbf{q}, s} \left(\mathbf{q} \cdot \mathbf{e}^{(s)} \right)^2 (e_j^{(s)})^2 v_s^\alpha(\mathbf{q}) \left\{ \sum_{\mathbf{q}, s} \left(\mathbf{q} \cdot \mathbf{e}^{(s)} \right)^2 (e_j^{(s)})^2 \right\}^{-1},$$

where the sum is over all polarizations and directions, and $\alpha \in (-3, 1, 2)$. The averages of powers of the s-wave sound velocities are calculated in analogy using the transverse unit vectors of (111)

$$\langle v_S^\alpha \rangle_j = \sum_{\mathbf{q}, s} \left(\left(\mathbf{t}_1 \cdot \mathbf{e}^{(s)} \right)^2 + \left(\mathbf{t}_2 \cdot \mathbf{e}^{(s)} \right)^2 \right) (e_j^{(s)})^2 v_s^\alpha(\mathbf{q}) \left\{ \sum_{\mathbf{q}, s} \left(\left(\mathbf{t}_1 \cdot \mathbf{e}^{(s)} \right)^2 + \left(\mathbf{t}_2 \cdot \mathbf{e}^{(s)} \right)^2 \right) (e_j^{(s)})^2 \right\}^{-1}.$$

The Debye sound velocity is calculated according to its definition via

$$v_{Dj} = \left\{ \frac{1}{3} \langle v_P^{-3} \rangle_j + \frac{2}{3} \langle v_S^{-3} \rangle_j \right\}^{-1/3}.$$

Averages over the projections j are obtained by

$$\langle v_P^\alpha \rangle = \frac{1}{NM} \sum_{\mathbf{q}, s} \left(\mathbf{q} \cdot \mathbf{e}^{(s)} \right)^2 v_s^\alpha(\mathbf{q}) \quad , \quad (113)$$

$$\langle v_S^\alpha \rangle = \frac{1}{2NM} \sum_{\mathbf{q},s} \left(\left(\mathbf{t}_1 \cdot \mathbf{e}^{(s)} \right)^2 + \left(\mathbf{t}_2 \cdot \mathbf{e}^{(s)} \right)^2 \right) v_s^\alpha(\mathbf{q}) \quad , \quad (114)$$

$$v_D = \left\{ \frac{1}{3} \langle v_P^{-3} \rangle + \frac{2}{3} \langle v_S^{-3} \rangle \right\}^{-1/3} = \left\{ \frac{1}{3NM} \sum_{\mathbf{q},s} v_s^{-3}(\mathbf{q}) \right\}^{-1/3} .$$

The rms values for the sound velocity distributions are obtained in the standard manner as $\text{rms}_v = \sqrt{\langle v^2 \rangle - \langle v \rangle^2}$. Peak and rms seismic sound velocity anisotropies are provided for p- and s-waves as

$$A_{P,S}^{(peak)} = \frac{\max v_{P,S} - \min v_{P,S}}{\max v_{P,S} + \min v_{P,S}} \quad \text{and} \quad A_{P,S}^{(rms)} = \sqrt{\frac{\langle v_{P,S}^2 \rangle}{\langle v_{P,S} \rangle^2} - 1} \quad .$$

The sensitivity of each sound velocity average with respect to an independent element of the elastic constant matrix C_{ij} is calculated numerically as the logarithmic derivative, e.g.,

$$S_{ij}(v_D) = \frac{\partial \ln v_D}{\partial \ln C_{ij}} \quad .$$

The result of this analysis is given in the protocol file '<prefix>_ptl.txt'.

8.4 Anisotropy corrections

Nuclear resonant and momentum resolved IXS techniques are used to extract aggregate compressional and shear sound velocities but generally assume elastic isotropy. In isotropic materials, sound propagation is described by the compressional sound velocity v_P and the shear sound velocity v_S independent of propagation direction. They are related to the Debye sound velocity v_D by

$$\frac{3}{v_D^3} = \frac{1}{v_P^3} + \frac{2}{v_S^3} \quad . \quad (115)$$

In addition, we have the following relationship with the seismic velocity v_ϕ originating from the Christoffel equation

$$v_\phi^2 = v_P^2 - \frac{4}{3} v_S^2 \quad . \quad (116)$$

The Debye sound velocity is obtained from nuclear resonant IXS spectra, and the seismic velocity is often derived from compression data from x-ray diffraction experiments. Studies using momentum-resolved IXS typically measure the average compressional sound velocity. Eqs. (115) and (116) are then used to extract v_P and v_S (NRIXS) or v_S (mrIXS). For anisotropic materials, this procedure produces systematic errors which are quantified in the following. A replacement of (116) can be found by using the averages in Eqs. (113) and (114) with $\alpha = 2$ in combination with (108)

$$\langle v_P^2 \rangle - \frac{4}{3} \langle v_S^2 \rangle = \frac{1}{\rho} \sum_{ijkl} E_{ijkl} \left\langle q_i q_j q_k q_l - \frac{2}{3} t_{1,i} q_j q_k t_{1,l} - \frac{2}{3} t_{2,i} q_j q_k t_{2,l} \right\rangle = \frac{K_V}{\rho} \quad . \quad (117)$$

The calculation of the averaged tensor on the right side of this expression is tedious but straightforward with the result

$$K_V = \frac{1}{9} \sum_{ij} E_{iijj} \quad i, j = 1 \dots 3 \quad ,$$

where K_V is also the Voigt bound of the bulk modulus.

8.4.1 Nuclear resonant IXS

In this case, we know the Debye sound velocity v_D , the adiabatic bulk modulus K , and the density ρ from experimental study. We introduce average sound velocities $\bar{v}_P = \langle v_P^{-3} \rangle^{-1/3}$ and $\bar{v}_S = \langle v_S^{-3} \rangle^{-1/3}$ which by definition satisfy the relation

$$\frac{3}{v_D^3} = \frac{1}{\bar{v}_P^3} + \frac{2}{\bar{v}_S^3} . \quad (118)$$

Equation (116) then takes the form

$$v_\phi^2(1 + \delta_K) = (1 + \delta_P)\bar{v}_P^2 - \frac{4}{3}(1 + \delta_S)\bar{v}_S^2 , \quad (119)$$

where $v_\phi^2 = K/\rho$ and the parameters $\delta_{P,S} = \langle v_{P,S}^2 \rangle \langle v_{P,S}^{-3} \rangle^{2/3} - 1$ and $\delta_K = K_V/K - 1$ quantify the effects of averaging over the material-specific velocity distributions. These parameters vanish for isotropic materials. In addition, δ_K vanishes for cubic crystal symmetry. If Eqs. (115) and (116) are applied they provide solutions v_P and v_S that deviate from the correct values, and we encounter errors ξ_P and ξ_S defined by $\bar{v}_P = (1 + \xi_P)v_P$ and $\bar{v}_S = (1 + \xi_S)v_S$. The calculation of the ξ values quantitatively describes the consequence of the elastic anisotropy on the procedure of sound velocity determination by nuclear resonant IXS. A combination of Eqs. (115), (116), (118), and (119) gives

$$\begin{aligned} \gamma^3 \{ (1 + \xi_P)^{-3} - 1 \} + 2 \{ (1 + \xi_S)^{-3} - 1 \} &= 0 \\ \{ (1 + \xi_P)^2(1 + \delta_P) - 1 \} - \frac{4}{3}\gamma^2 \{ (1 + \xi_S)^2(1 + \delta_S) - 1 \} &= \alpha^2 \delta_K , \end{aligned} \quad (120)$$

where $\gamma = v_S/v_P$ and $\alpha = v_\phi/v_P$. These equations are solved numerically for two situations: the provided bulk modulus K represents the Voigt or the Reuss bound, respectively. The values for ξ_P and ξ_S give the methodological error resulting from application of Eqs. (115) and (116) to anisotropic materials. The procedure provides us with an average compression sound velocity defined by $\bar{v}_P = \langle v_P^{-3} \rangle^{-1/3}$ and an average shear sound velocity defined by $\bar{v}_S = \langle v_S^{-3} \rangle^{-1/3}$. These values are compared to Voigt or Reuss bounds of aggregate sound velocities described in section 8.2.

8.4.2 Momentum-resolved IXS

In this case, we assume to know the average compressional sound velocity $\bar{v}_P = \langle v_P \rangle$, the adiabatic bulk modulus K , and the density ρ from experimental study. We now introduce the average shear sound velocity $\bar{v}_S = \langle v_S \rangle$ which satisfies (117) for anisotropic materials

$$v_\phi^2(1 + \delta_K) = (1 + \epsilon_P)\bar{v}_P^2 - \frac{4}{3}(1 + \epsilon_S)\bar{v}_S^2 , \quad (121)$$

where $v_\phi^2 = K/\rho$ and the parameters $\epsilon_{P,S} = \langle v_{P,S}^2 \rangle \langle v_{P,S} \rangle^{-2} - 1$ and $\delta_K = K_V/K - 1$ quantify the effects of averaging over the material-specific velocity distributions. Again these parameters vanish for isotropic materials. The solution v_S of the isotropic (116) deviates from the correct value of \bar{v}_S by a certain amount η_S defined by $\bar{v}_S = (1 + \eta_S)v_S$. The calculation of η_S quantitatively describes the consequence of the elastic anisotropy on the procedure of sound velocity determination by momentum-resolved IXS. The combination of (116) and (121) gives the result

$$\eta_S = \left(\frac{1 + \epsilon_P - \gamma^2(1 + \delta_K)}{(1 + \epsilon_S)(1 - \gamma^2)} \right)^{1/2} - 1 , \quad (122)$$

where $\gamma = v_\phi/\bar{v}_P$. The correction η_S for the shear velocities is calculated for two situations: the provided bulk modulus represents either the Voigt or the Reuss limit. The values for η_S give the methodological error resulting from application of (116) to anisotropic materials. The procedure provides us with an average shear sound velocity defined by $\bar{v}_S = \langle v_S \rangle$. These values are compared to Voigt or Reuss bounds of aggregate sound velocities described in section 8.2.

8.5 Output files

The file prefix specified in line 11 of the SIF is used for output files containing calculation results. Table 7 on page 35 shows a summary of output files created by the svec executable.

8.5.1 Sound velocity distributions (options A, L, T)

The probability distribution for p-waves is calculated using (112) with $\mathbf{a} = \mathbf{q}$ and saved into the '_lsv' file. For polar-type and azimuthal type s-waves, (112) is used with $\mathbf{a} = \mathbf{t}_1$ and $\mathbf{a} = \mathbf{t}_2$, respectively, and the results are saved into files '_tpv' and '_tav'. The average sound velocity distribution $D(v) = (D_P(v) + 2D_S(v))/3$ is saved into file '_asv'. In all cases, the probability distributions are normalized as $\int D(v) dv = 1$.

8.5.2 Sound velocity surfaces (option S)

Sound velocity surfaces are constructed from calculated sound velocities of a particular type (p-wave, s-wave polar or azimuthal) on an evenly spaced angular grid. They represent sound velocity vectors of a certain type.

The file '<prefix>_L.3D' contains four columns of numbers: the cartesian coordinates (v_{P1}, v_{P2}, v_{P3}) and the length $|\mathbf{v}_P|$ of the longitudinal sound velocity vector. The files '<prefix>_T1.3D' and '<prefix>_T2.3D' contain the two transverse sound velocities in the same manner. The file '<prefix>_SV.3D' contains seismic velocity values that are calculated as $v_\phi = \sqrt{v_P^2 - 2(v_{Sp}^2 + v_{Sa}^2)}/3$ from p-wave sound velocities v_P and s-wave sound velocities v_{Sp} (polar) and v_{Sa} (azimuthal). The '.3D' files are compressed and appear as '.3D.gz' files if the 'gzip' program is available. These files can for example be visualized with the VisIt software of Lawrence Livermore National Laboratory.

The files with extension 'ppm' are images with same content as above. The directional dependence of the sound velocities is mapped onto the images by mercator and azimuthal orthographic projections, respectively. The values of the sound velocities are color-coded. The color scale is selected by an integer N appended to the output option as 's_N'. Values of N range between 0 and 21. Even values give unmodulated and odd values modulated color scales. The modulation can enhance variations for small intensities. Values 0 and 2 correspond to gray and inverse gray, respectively. The next values select Imola, Bamako, Oslo, Hawaii, Acton, and Batlow scales which are adapted from Fabio Cramer. Then follow Inferno, Kindlmann, and Rainbow scales.

9 XRD Background Subtraction: sbck

The sbck program takes a stack of XRD patterns and applies background subtraction procedures. Two types of input are accepted, azimuthally integrated two-theta patterns in multi-column format and stratified two dimensional patterns (caked XRD image) from formatted output files of the DIOPTAS software. All files in a stack must be of the same type.

, determines the background function, and subtracts it from the input patterns. Within the assumptions of the program, the result is a set of background-less diffraction patterns that are ready to be analyzed by programs such as GSAS or FullProf. No further background correction “by hand” should be needed.

The SIF of the sbck executable has at least 14 lines of data. The content of the input lines is discussed in order of appearance.

1. Output file prefix and options – many values :

The first value gives the prefix for output folder and file names.¹⁰ Generated output is listed in Table 8. With the exception of the ppm-formatted image files, all output files are plain text files. If option ‘,’ is specified csv-formatting is provided. If option G is specified a header for the xmgrace graphical program is added. A detailed description of the procedures used to produce the output is provided in sections 5.6 on page 18 to 5.20 on page 22. In addition, the protocol file ‘<prefix>_ptl.txt’ is created.

option	description	units (x,y)	file name
b,B	background function	deg, none	<prefix>_bkg.<ext>
e,E	diffraction ring deformation	index, %	<prefix>_sas.<ext>
	strain ellipse angle	index, deg	<prefix>_saa.<ext>
	geometry distortion	index, %	<prefix>_sag.<ext>
n,N	normalization factor	index, 1/intensity	<prefix>_nrm.<ext>
p,P	guide peak position	index, deg	<prefix>_gpp.<ext>
	guide peak width	index, deg	<prefix>_gpw.<ext>
	guide peak intensity	index, intensity	<prefix>_gpi.<ext>
	guide peak mode skewness	index, none	<prefix>_gps.<ext>
	guide peak fit quality	index, %	<prefix>_gpq.<ext>
	relative guide peak position	index, deg	<prefix>_gpr.<ext>
	integrated decompressed stack	deg, none	<prefix>_dci.<ext>
r,R	background-corrected stack	deg, intensity	<prefix>_/*
	normalized background-corrected stack	deg, none	<prefix>_nrm/*
	FullProf formatted stack	NA	<prefix>_fpr/*
s,S	background-corrected stack image	NA	<prefix>_stk.ppm
	decompressed stack image	NA	<prefix>_sdc.ppm

Table 8: Output options for background calculations. <prefix> is the file prefix specified in line 1. The extension <ext> is either ‘csv’ for comma-separated-value files (option ‘,’ in line 1) or ‘dat’ otherwise.

2. Diffraction angle range – two values :

The 2θ range of the input stack to be used for calculations of the background.

3. Correction angles – two values :

The first value give the skew angle that is applied to the stack. The second value provides a shear angle that is applied to the stack which is needed for stacks with no or little change of diffraction line positions along the stack index.

4. Background averaging – five values :

The first value specifies the 2θ interval over which the background function is averaged. A running average is applied if the second value is not given. If the second value is a non-space character the third value gives the

¹⁰Names may be defined either relative to the working directory or absolute: Prefix creates files in the working directory, ../Prefix creates files in the parent directory, myDir/Prefix creates files in the directory myDir (must exist) in the working directory, /Users/-data/Prefix creates files in the directory /Users/data (must exist and be writeable).

order of a polynomial that is fitted to the background function and then replaces it, the fourth value gives a coefficient to overweigh negative deviations, and the fifth value gives a coefficient to straighten the polynomial curve.

5. Background refinement level – two values :
The first value specifies the level of the normalized background-corrected stack above which peaks are recognized as such. If this value is zero refinements to normalization factors and background function are not applied. If this value is negative refinements to normalization factors and background function are automatically determined for each pattern of the stack. The second value is optional and only used in the case of a positive first value. It specifies the amount of corrective shift that is applied to offset too dominant negativities. A value of zero leads to the most aggressive background refinement.
6. Image scaling type – two values :
The first value gives the type of scaling used for production of stack images: linear, logarithmic, and natural. The second value gives the dynamic range in case of the latter two options. Type “natural” implies a scaling of diffraction intensities by $(2\theta)^{2.5}$ before logarithmic scaling is applied.
7. Image color scale – one value :
This value gives an integer between 0 and 21 to select the color scale for stack images. Even values give unmodulated and odd values modulated color scales. The modulation can enhance variations for small intensities. Values 0 and 2 correspond to gray and inverse gray, respectively. The next values select Imola, Bamako, Oslo, Hawaii, Acton, and Batlow scales which are adapted from Fabio Crameri. Then follow Inferno, Kindlmann, and Rainbow scales.
8. Image stretch factor, stack index – one value :
This value gives a stretch factor for image creation in the direction of the stack index.
9. Image stretch factor, diffraction angle – one value :
This value gives a stretch factor for image creation in the direction of the angle.
10. Guide peak fit range – three values :
The first two values specify the 2θ range in which a guide peak is identified and fitted for each pattern of the stack. The fitted position of the guide peak is used to generate a decompressed stack. If the third value is “yes” then the shape of the guide peak is fitted in addition to position, width, and height. To suppress guide peak fitting omit all three values.
11. Iteration control for guide peak fitting – three values :
The first value gives the maximum number of iteration steps. The other values are optional. The second value specifies the Levenberg-Marquardt parameter λ that is used to create a multiplier m to the diagonal of the second-derivative matrix of the normalized χ^2 . For iteration step n , the multiplier is given by $m = 1 + \lambda \max [0, (\chi_{n-1}^2 - \chi_n^2)/\chi_n^2]$. Larger values for λ have a dampening effect and steer the iteration process toward the gradient method. The Newton method is obtained for $\lambda = 0$ which is also the default value. The third value gives the required relative change of χ^2 value for an iteration step to stop the iteration.
12. Burst mode – two values :
A first value of yes selects burst mode for which the stack consists of an alternating sequence of patterns from the cold or heated sample. The second value specifies if the alternating sequence begins with a hot or a cold pattern. The default is hot, specify “cold” for the other case.
13. Negativity quench, analysis type – five values :
A first value of yes selects a quench of negative intensity values in the background-subtracted output stacks. The procedure aims at maintaining a smooth background function and tries to avoid peak-shape distortions for negativities close to diffraction peaks. The second value selects the analysis type of caked-pattern input patterns. Specify “r” for root-mean-square, “n” for normalized root-mean-square, and “i” for intensity analysis. If the third value is “yes” then a deformation analysis of a diffraction ring is performed with the fourth and fifth values giving the 2θ range for a particular diffraction ring. Values two to five are ignored for column input files.
14. Name of data files with XRD patterns – many values :
The first value gives the name of a data file. The following values define the assignment of column numbers

in the data file to physical parameters. Each assignment is a combination of one letter and a number, e.g., a1. The letter refers to the physical meaning: a,A for diffraction angle values; d,D for detector intensity values; b,B for an optional normalization factor correction; c,C for a caked-pattern input file¹¹. The B option does not pair with a column number but specifies a value. For example, B0.01 means that the internally determined normalization factor is modified by 1.01. Caked pattern and column-number files cannot be mixed. Automatic file name expansion is supported for a numerical sequence of data file names. For example, the syntax 'myFile_<10-12>.dat' results in files 'myFile_10.dat', 'myFile_11.dat', and 'myFile_12.dat' to be read with the same set of assignments. The sequence of XRD patterns as specified here constitutes the stack. The stack index starts with 1 for the first specified file. Add more lines for additional data files.

9.1 Background detection procedure

The central problem that is addressed in the sbck program is to determine and remove the background, so we are left with only the diffraction peaks. The input stack is an array of XRD patterns I_{nk} with a range of diffraction angles $2\theta_k$ as specified in input line 2 and the stack index n . Each pattern is a sum of diffraction peaks and background, and our central assumption is that the shape of the background is common to all patterns

$$I_{nk} = P_{nk} + A_n B_k \quad , \quad (123)$$

where B_k is the background function, and A_n are normalization factors. The normalization factors are approximated as

$$A_n = (1 + b_n) \frac{1}{K} \sum_k (I_{nk} - P_{nk}) \approx (1 + b_n) \frac{1}{K} \sum_k I_{nk} \quad , \quad (124)$$

where K is the number of 2θ angle points, and the normalization corrections b_n are specified in input lines 14ff. The approximation is justified for large backgrounds or weak peaks which occur in experiments with substantial parasitic scattering from sample containers. The background function is then given by

$$B_k = \frac{1}{A_n} (I_{nk} - P_{nk}) \quad , \quad (125)$$

which expresses our key assumption Eq. (123) as invariance with respect to the stack index. We still do not know P_{nk} , but we assume that a given diffraction peak changes its position with changing stack index. This is for example the case in high-pressure experiments if stack index is equivalent to pressure. Then $\min_n P_{nk} = 0$ and therefore

$$B_k = \min_n \frac{I_{nk}}{A_n} \quad . \quad (126)$$

The diffraction patterns follow from the background function and the normalization factors

$$P_{nk} = I_{nk} - A_n \min_n \frac{I_{nk}}{A_n} \quad . \quad (127)$$

9.1.1 Normalization refinement

In Eq. (124), we approximated the normalization factors because the diffraction peaks P_{nk} were unknown. The direct use of Eq. (127) would be circular, and therefore we clip the normalized diffraction patterns at level α which is specified in input line 5 to reach a normalization refinement as

$$A_n^{(r)} = (1 + b_n) \frac{1}{K} \sum_k \left\{ I_{nk} - P_{nk} \Theta \left(\frac{P_{nk}}{A_n} - \alpha \right) \right\} \quad , \quad (128)$$

where Θ is Heaviside's step function, and P_{nk} is obtained via Eq. (127). The refined normalization factors are then used to obtain the refined diffraction patterns

$$P_{nk}^{(r)} = I_{nk} - A_n^{(r)} \min_n \frac{I_{nk}}{A_n^{(r)}} \quad . \quad (129)$$

¹¹Caked-pattern files have to be generated by the Dioplas pattern analysis software

9.1.2 Skew angle

Sometimes the background function in Eq. (126) has a small overall dependence on the stack index. This can often be empirically corrected by skewing the stack via

$$I_{nk} \longrightarrow \left(1 + \sin \xi \frac{2k - K}{K} \frac{2n - N}{N} \right) I_{nk} \quad ,$$

where K is the number of 2θ angle points, N is the number of patterns in the stack, and ξ is the skew angle specified in input line 3.

9.1.3 Shear angle

For stacks that contain patterns without an intrinsic dependence on the stack index the procedure defined in Eq. (126) does not work because $\min_n P_{nk} \neq 0$. This situation is accommodated by shearing the stack via

$$I_{nk}^{(S)} = \frac{1}{2} (I_{n,k+j} + I_{n,k-j}) \quad \text{with} \quad j = (n-1) \frac{\eta}{\Delta\theta} \quad ,$$

where $\Delta\theta$ is the step size on the 2θ scale, and η is the shear angle specified in input line 3. If the shear angle is sufficiently large we can achieve $\min_n P_{nk}^{(S)} = 0$ leading to

$$\frac{1}{2} (B_{k+j} + B_{k-j}) = \min_n \frac{I_{nk}^{(S)}}{A_n^{(r)}} \quad . \quad (130)$$

We now assume that the background function's dependence on diffraction angle is weak such that

$$\frac{1}{2} (B_{k+j} + B_{k-j}) \approx B_k + \frac{1}{2} j^2 B_k'' \quad \Rightarrow \quad B_k \approx \min_n \frac{I_{nk}^{(S)}}{A_n^{(r)}} \quad , \quad (131)$$

where B_k'' is the second derivative of the background function. This emphasizes that the shear angle should be chosen as small as possible but also not too small. The background function obtained this way is then applied to the unsheared stack.

9.1.4 Background averaging

The background function obtained either from Eq. (126) or Eq. (131) can be averaged using a running average

$$\overline{B}_k = \frac{1}{2j+1} \sum_{l=k-j}^{k+j} B_l \quad \text{with} \quad 2j+1 = \frac{\delta}{\Delta\theta} \quad , \quad (132)$$

where δ is the averaging window specified in input line 4, and $\Delta\theta$ is the step size on the 2θ scale.

Alternately the background function can be fitted to a polynomial with order specified in input line 4. The fitted polynomial becomes the new background function. This often looks good to the eye but leads to inferior background subtraction.

9.1.5 Background refinement

Ideally the normalized diffraction peak pattern given by

$$p_{nk} = \frac{I_{nk}}{A_n^{(r)}} I_{nk} - \min_n \frac{I_{nk}^{(S)}}{A_n^{(r)}} \quad (133)$$

would be clear of all background contributions. In practice, the outlined procedures remove most but not all of the background. To potentially improve the background subtraction, we create two broadly averaged stacks

$$S_{nk} = \frac{1}{2j+1} \sum_{l=k-j}^{k+j} \Theta(\alpha - p_{nl}) \Theta(p_{nl}) p_{nl} \quad \text{and} \quad T_{nk} = \frac{1}{2j+1} \sum_{l=k-j}^{k+j} \Theta(\alpha - p_{nl}) \Theta(p_{nl}) \frac{1}{p_{nl}} \quad , \quad (134)$$

where $2j + 1 = K/10$ with K as the number of 2θ angle points, Θ is Heaviside's step function, and the cut-off value α is specified in input line 5. A correction to the background subtraction is then obtained by

$$p_{nk}^{(r)} = p_{nk} - C_{nk} + \beta \min_k C_{nk} \quad \text{with} \quad C_{nk} = \frac{1}{2j+1} \sum_{l=k-j}^{k+j} \min(S_{nl}, \frac{1}{T_{nl}}) \quad . \quad (135)$$

This refinement procedure is purely empirical and does not guarantee an improved background subtraction. Its quality strongly depends of the choice of the cut-off value α . The shift fraction β controls how aggressive the refinement is applied with a zero-value as most aggressive.

The cut-off value can be determined automatically using the following procedure. A test value τ for the cut-off value is chosen and the number of crossings with the normalized diffraction peak pattern is determined

$$\psi(\tau) = \sum_k \Theta(\tau - p_{n,k-1}) \Theta(p_{nk} - \tau) \quad . \quad (136)$$

The function $\psi(\tau)$ is then smoothed, and the position τ_n of its maximum is determined. The cut-off value is then set to $\alpha_n = 1.5 \tau_n$. The shift fraction β is set to zero.

9.1.6 Quality assessment

The overall quality of the background subtraction can be assessed qualitatively by visual inspection of the p_{nk} image. If the image looks uniform the background subtraction still could have been too aggressive leading to many negative values of p_{nk} . The negativity is quantified by the average negativity

$$Q_0 = \frac{1}{q} \sum_{n,k} \Theta(-p_{nk}) p_{nk} \quad \text{with} \quad q = \sum_{n,k} \Theta(-p_{nk}) \quad , \quad (137)$$

the diluted negativity

$$Q_d = \frac{1}{NK} \sum_{n,k} \Theta(-p_{nk}) p_{nk} \quad , \quad (138)$$

the anisotropy of negativity

$$Q_a = \frac{\min \left[M_{11} + M_{22}, \sqrt{(M_{11} - M_{22})^2 + 4 M_{12}^2} \right]}{\max \left[M_{11} + M_{22}, \sqrt{(M_{11} - M_{22})^2 + 4 M_{12}^2} \right]} \quad , \quad (139)$$

and the twist of negativity

$$Q_t = Q_a \frac{2M_{12}}{|M_{11} - M_{22}| + \sqrt{(M_{11} - M_{22})^2 + 4 M_{12}^2}} \quad . \quad (140)$$

The moments are defined by

$$M_0 = \sum_{n,k} \Theta(-p_{nk}) p_{nk} \quad (141)$$

$$M_1 = \frac{1}{M_0} \sum_{n,k} n \Theta(-p_{nk}) p_{nk} \quad (142)$$

$$M_2 = \frac{1}{M_0} \sum_{n,k} k \Theta(-p_{nk}) p_{nk} \quad (143)$$

$$M_{11} = \frac{1}{M_0 N^2} \sum_{n,k} (n - M_1)^2 \Theta(-p_{nk}) p_{nk} \quad (144)$$

$$M_{22} = \frac{1}{M_0 K^2} \sum_{n,k} (k - M_2)^2 \Theta(-p_{nk}) p_{nk} \quad (145)$$

$$M_{12} = \frac{1}{M_0 NK} \sum_{n,k} (n - M_1) (k - M_2) \Theta(-p_{nk}) p_{nk} \quad . \quad (146)$$

The anisotropy describes the left/right and top/down evenness of the distribution of negative regions of p_{nk} . It is by definition invariant with respect to the sign of M_{12} and therefore a mirrored stack gives the same anisotropy value. On the other hand, the twist quantifies a stack that is not invariant with respect to an individual mirror operation, but only to an inversion. The twist value is weighted with anisotropy to emphasize its lesser importance for small anisotropies. The values of anisotropy and twist can be minimized by prudent choices for skew and shear angles as well as refinement level.

9.1.7 Negativity quench

After the above options to obtain a well-balanced background subtraction have been exhausted, the remaining negative values in the background-subtracted patterns can be completely removed with this procedure. The negative values of a background-subtracted pattern are convoluted with a normalized Gaussian G_k with a rms value of one degree. The resulting function is subtracted from the background-subtracted pattern, and the procedure is repeated 100 times

$$p_{nk}^{(j+1)} = p_{nk}^{(j)} - \sum_k G_k p_{nk}^{(j)} \Theta(-p_{nk}^{(j)}) \quad . \quad (147)$$

The iteration leads to very small remaining negativities that are set to zero. Due to the convolution and thus broadening the negativities are re-distributed over the vicinity of negative regions, and the smoothness of the background function is mostly maintained.

9.2 Pattern analysis

Input files that are provided as caked XRD patterns can be analyzed in three ways: azimuthal integration; azimuthal root-mean-square value; normalized azimuthal root-mean-square value. In the first case, the background in the resulting 2θ patterns is treated as described in section 9.1.

Let I_{kj} be the intensity of the caked pattern in a range of diffraction angles $2\theta_k$ as specified in input line 2 and the azimuthal angle ϕ_j . Then the root-mean-square pattern is given by

$$\delta I_k = \sqrt{\frac{1}{N} \sum_j I_{kj}^2 - \left(\frac{1}{N} \sum_j I_{kj} \right)^2} \quad , \quad (148)$$

where N is the number of azimuthal data points that are not exactly zero. This expression emphasizes 2θ areas of the pattern that have large fluctuations which are typically spotty diffraction peaks. The background is quite smooth leading to much smaller rms values with little variation over the 2θ range. This method of background reduction does not depend on the presence of a stack of data.

The normalized root-mean-square pattern is given by

$$\delta I_k = \left\{ \frac{1}{N} \sum_j I_{kj}^2 - \left(\frac{1}{N} \sum_j I_{kj} \right)^2 \right\}^{1/2} \left(\frac{1}{N} \sum_j I_{kj} \right)^{-1/3} \quad , \quad (149)$$

where the normalization is chosen somewhat empirically to emphasize small diffraction peaks while avoiding saturation of intense diffraction peaks.

9.2.1 Strain analysis

Strain analysis requires caked-pattern input files. For each 2θ value in the range given in input line 13, the azimuthal range of 360° is divided into 18 sections each of which are summed up into one value representative of this section. Care is taken to avoid data points that are exactly zero. Each of the 18 2θ patterns is fit to a symmetric gaussian shown in Eq. 151 with $\eta_{\pm} = 2$ and $\xi_- = \xi_+$. The fitted peak positions ξ_{1j} are then fitted to the eccentricity function

$$f(\phi) = \xi_0 \{1 + \xi_1 \cos(\phi - \xi_2) + \epsilon \cos 2(\phi - \alpha)\} \quad . \quad (150)$$

The second term, variables ξ_1 and ξ_2 , describe the effect of uncentered diffraction rings and is usually very close to zero for proper generation of the caked pattern. The third term describes an elliptical distortion of the diffraction ring which can be caused by non-hydrostatic stress acting on the sample.

9.3 Guide peak analysis

A diffraction peak in the first pattern of the stack can be selected for fitting. In this way, the change of position, width, intensity, and shape of this peak can be monitored as a function of the stack coordinate. The guide peak should be reasonably well isolated from other features in the pattern. The fit range for the peak is selected in input line 10. The fit function is defined as

$$f(2\theta) = \xi_0 \left\{ \Theta(\xi_1 - 2\theta) \exp \left[-\ln 2 \left| \frac{2\theta - \xi_1}{\xi_-} \right|^{\eta_-} \right] + \Theta(2\theta - \xi_1) \exp \left[-\ln 2 \left| \frac{2\theta - \xi_1}{\xi_+} \right|^{\eta_+} \right] \right\} \quad , \quad (151)$$

where the full width at half maximum is given by $\xi_- + \xi_+$. The parameters ξ_0 , ξ_1 , ξ_+ , and ξ_- are always fitted. Fitting of the shape, parameters η_- and η_+ , is optional and can be selected in input line 10. When shape is not fitted the fit function defaults to gaussian, i.e., $\eta_- = \eta_+ = 2$. The fit procedure minimizes the mean square deviation in the 2θ range defined in input line 10

$$\sum_{k=k_1}^{k_2} \left\{ p_{nk}^{(r)} - f(2\theta_k) \right\}^2 \rightarrow \min \quad . \quad (152)$$

The position of the guide peak along the stack index is used to create a “decompressed” stack, a description chosen from the situation in which the stack index represents pressure. The 2θ values of each pattern in the stack are scaled with the fitted position of the guide peak

$$D_{nk} = p_{nj}^{(r)} \quad \text{with} \quad j = k \frac{\xi_{11}}{\xi_{1n}} \quad . \quad (153)$$

In the decompressed stack, the position of the guide peak is independent of the stack index. Diffraction peaks with the same compression behavior also appear at constant 2θ values. Unrelated diffraction peaks, e.g. of another material, can be identified this way.

9.4 Burst mode

XRD experiments with short hot/cold cycles, e.g. Laser heating in combination with diamond anvil cells, are accommodated with the burst mode setting in input line 12. In this mode, the stack I_{nk} is separated into hot and cold stacks. The normalization procedure in Eq. (128) is only applied to the cold stack. Two resulting normalization factors are used to normalize the enclosed hot spectra. The determination of the background function also uses the cold spectra only and usually requires shearing of the stack as described in section 9.1.3 on page 43. The use of only the cold stack for these procedures enables the detection of diffuse scattering in the hot patterns which otherwise might appear alike to background.

9.5 Output files

The amount and content of output files is briefly described in Table 8. Output options specified in input line 1 specify specific files.

9.5.1 Option B

The normalized background as determined by Eqs. (126), (131), and (132) is saved. It is given as function of the diffraction angle 2θ .

9.5.2 Option E

This option only produces output for cake-pattern input files. The elliptical distortion ϵ and angle α as well as the geometry distortion ξ_1 resulting from fits using Eq. 150 are saved. They are given as function of the stack index.

9.5.3 Option N

The inverse normalization factors $1/A_n^{(r)}$ as determined by Eq. (128) are saved. They are given as function of the stack index.

9.5.4 Option P

This option causes the results of a guide peak fit procedure using fit function Eq. (151) to be saved. It is ignored if guide peak fitting is disabled. Position, width, intensity, and quality of fit are saved. The width is calculated with the best-fit function f as the scaled 2θ rms value

$$w = \frac{2\sqrt{2 \ln 2}}{\langle f \rangle} \sqrt{\langle 4\theta^2 f \rangle \langle f \rangle - \langle 2\theta f \rangle^2} \quad , \quad (154)$$

where $\langle \dots \rangle$ indicates the sum over the fit range. For a gaussian shape, this value gives the full width at half maximum. If the shape is fitted the mode skewness gives a measure of the asymmetry of the fitted peak

$$s = \frac{1}{w} \left\{ \frac{\langle 2\theta f \rangle}{\langle f \rangle} - \xi_1 \right\} \quad , \quad (155)$$

where ξ_1 is the fitted peak position. The quality is defined as the normalized mean square deviation

$$Q = \sum_{k=k_1}^{k_2} \left\{ p_{nk}^{(r)} - f(2\theta_k) \right\}^2 \left(\sum_{k=k_1}^{k_2} \left\{ p_{nk}^{(r)} - \frac{1}{2} (p_{nk_1}^{(r)} + p_{nk_2}^{(r)}) \right\}^2 \right)^{-1} \quad . \quad (156)$$

Not converged fits are identified by a quality that is set to -1.

All fit parameters are given as function of the stack index. If burst mode is specified guide peak fitting occurs separately for hot and cold stacks. Also the relative guide peak position defined as the difference of hot minus cold peak positions between adjacent patterns is saved.

In addition, the decompressed diffraction patterns are integrated and saved, $D_k = 1/N \sum_n D_{nk}$ using Eq. (153). The rms fluctuation of peak intensity are provided as error values in the data file.

9.5.5 Option R

The output is sorted into three directories: background corrected stack $A_n^{(r)} p_{nk}^{(r)}$ from Eqs. (128) and (135); normalized background corrected stack $p_{nk}^{(r)}$ from Eq. (135); background corrected stack but formatted for the FullProf XRD refinement software.

9.5.6 Option S

The background corrected stack $p_{nk}^{(r)}$ is converted into a color-coded image with vertical and horizontal pixels representing stack index and 2θ values, respectively. Since often $k \gg n$ the aspect ratio can be modified by the stretch factors in input lines 8 and 9. If both stretch factors are one each (k, n) pair is mapped onto one pixel. The intensity scale type and the color scale given in input lines 6 and 7 specify how the patterns' intensities are mapped onto color values. For a logarithmic scale, the lowest value is limited by $\alpha \max p_{nk}^{(r)}$ with the dynamic range α specified in input line 6. For a natural scale, the stack is scaled by $(2\theta)^{5/2}$ before logarithmic scaling is applied. This will enhance visibility of diffraction peaks at higher angles which are naturally weaker. In the case of linear scaling, negative values of the stack are enhanced by a factor of $1/\min p_{nk}^{(r)}$ and emphasized with a linear red color scale. This way the distribution of negative values of the stack becomes visible. In case of guide peak fitting, the decompressed stack is imaged in the same manner.

In case of burst mode operation, hot and cold stacks are combined into single images. Top and bottom halves of the images represent hot and cold stacks, respectively.

10 General function fitting: sfit

The sfit calculates a function specified in the SIF and optionally fits its parameters to a data set.

The SIF of the sfit executable has at least 4 lines of data. Lines 4 and higher contain function specifications. The content of the input lines is discussed in order of appearance.

1. Input file or range— many values :

The first value gives the name of a measured data file.¹² The following values define the assignment of column numbers in the data file to physical parameters. Each assignment is a combination of one letter and a number, e.g., X1 or y15. The letter refers to the physical meaning according to X: x-values, Y: y-values, x: x-value uncertainties, and y: y-value uncertainties. Only the specification for the x-value uncertainties is optional. Data in the measured data file between x-values x_1 and x_2 may be excluded from fitting by adding the line '#Mask x_1 x_2 ' to the file. Several lines of this type may be added to exclude specific regions of data.

If the first value is not a data file a calculation is performed. Then first and second value give x-values for start and end of the calculation range. The third value provide the number of points on an equidistant x-scale. In case of a XRD pattern calculation or fit the x-values are diffraction angles in degrees.

2. Output file prefix and options – many values :

The first value gives the prefix for output file names.¹³ Several output files listed on the top in Table 9 are created by default. Output files are plain text files. If option ',' is specified csv-formatting is provided. If option G is specified a header for the xmgrace graphical program is added. Additional output files are created according to options following the file name prefix. With exception of option L, the first value is the x-value. In case of a XRD pattern calculation or fit the x-values are diffraction angles in degrees. The second value depends on the option specified, and the third value is the statistical error of the second value. In addition, the protocol file '<prefix>.ptl.txt' and, for fits to data, the results file '<prefix>.ite.csv' are created. In case of a XRD pattern calculation or fit a list of reflections is saved into '<prefix>.xrd.csv'.

option	column 1	column 2,3, ...	file name
	x-value	calculated function values	<prefix>_fct.<ext>
	x-value	calculated function with gaussian noise	<prefix>_nsy.<ext>
	x-value	input data without masked values	<prefix>_dat.<ext>
	x-value	best fit to input data	<prefix>_fit.<ext>
	x-value	normalized residuals of best fit	<prefix>_rsd.<ext>
l,L	parameter n	parameter m	<prefix>_p<n>v<m>_eel.<ext>
		n-m, normalized, centered	<prefix>_p<n>v<m>_nel.<ext>

Table 9: Output options for fits and calculations. <prefix> is the file prefix specified in line 2. The extension <ext> is either 'csv' for comma-separated-value files (option ',' in line 3) or 'dat' otherwise. In case of a XRD pattern calculation or fit the x-values are diffraction angles in degrees.

3. Iteration control – three values :

The first value gives the maximum number of iteration steps. The other values are optional. The second value specifies the Levenberg-Marquardt parameter λ that is used to create a multiplier m to the diagonal of the second-derivative matrix of the normalized χ^2 . For iteration step n , the multiplier is given by $m = 1 + \lambda \max [0, (\chi_{n-1}^2 - \chi_n^2)/\chi_n^2]$. Larger values for λ have a dampening effect and steer the iteration process toward the gradient method. The Newton method is obtained for $\lambda = 0$ which is also the default value. The third value gives the required relative change of χ^2 value for an iteration step to stop the iteration.

4. Function specification – many values :

The first value specifies the type of function to be added to the total function to be calculated or used for fitting. Table 10 shows a list of supported function types and needed parameters. Functions are defined in sections 10.2 to 10.4.

¹²The input file is a plain text file containing no more than 300 characters per line (excess characters are ignored). Each line in the file contains a sequence of data values called columns separated by either spaces, tabs, or commata (csv format). Lines that do not begin with a number are ignored (after removing leading spaces, tabs, commata).

¹³Names may be defined either relative to the working directory or absolute: Prefix creates files in the working directory, ../Prefix creates files in the parent directory, myDir/Prefix creates files in the directory myDir (must exist) in the working directory, /Users/-data/Prefix creates files in the directory /Users/data (must exist and be writeable).

function type	number of parameters	parameters
comb	6	scale, position, period, peak type, FWHM, shape
exponential	6	scale, position, left HWHM & shape, right HWHM & shape
external	≥ 1	name of external executable or script, parameter string
pattern	≥ 6	symgrp, unit cell, peak type, peak parms
polynomial	$\geq 1, \leq 11$	polynomial coefficients, increasing order
power	6	scale, position, left HWHM & shape, right HWHM & shape
sigmoid	3	scale, position, slope
sine	5	scale, position, frequency, phase, damping
voigt	4	scale, position, FWHM, shape

Table 10: List of supported function types.

10.1 Comb function

The comb function type requires five numerical parameters $p_1 \dots p_3$ and p_5, p_6 and is defined as follows

$$f(x) = p_1 \sum_{j=0} P(x - p_2 - j p_3, p_5, p_6) \quad . \quad (157)$$

The peak function P can be symmetric exponential, symmetric power, or Voigt. In either case, parameters p_5 and p_6 are FWHM and shape. The number of peaks considered depends on the period p_3 and the calculation range.

10.2 Exponential function

The exponential function type requires six parameters $p_1 \dots p_6$ and is defined as follows

$$f(x) = \begin{cases} p_1 \exp \left[-\ln 2 \left| \frac{x-p_2}{p_3} \right|^{p_4} \right] & \text{for } x < p_2 \\ p_1 \exp \left[-\ln 2 \left| \frac{x-p_2}{p_5} \right|^{p_6} \right] & \text{for } x > p_2 \end{cases} \quad . \quad (158)$$

All parameters are fitable using substitution directives as outlined in section 4.2.

10.3 External function

The external function type requires the name of an executable or script performing the calculation. Parameters following the name declaration are passed as argument list. The sfit module creates a formatted file named '.ZZapi.in' that contains x-values, one value per line, and calls the external function with argument list. The external function is expected to provide a formatted file named '.ZZapi.out' that contains the calculated function values, one value per line.

The location of the files '.ZZapi.in' and '.ZZapi.out' is the location in which sfit was called. The location of the executable or script follows the rules of path evaluation and can be either referred to with absolute path (starts with '/') or otherwise is searched for in directories provided by the 'PATH' shell environment.

10.4 Pattern function

The pattern function type requires multiple parameters the numbers of which are interdependent. The first parameter flowing the function-type specification is the symmetry group of the material producing the XRD pattern. Either the symmetry group number or Hermann–Mauguin notation are accepted. For example, group number 198 is equivalent to P2₁3 and has to be given as 198 or P.2.1.3 in the function specification. The next parameters specify the minimal set of unit cell (see Fig. 1) dimensions and angles which depends on the lattice system belonging to the symmetry group. Dimensions and angles are expected in units of wavelength and degrees, respectively. The set

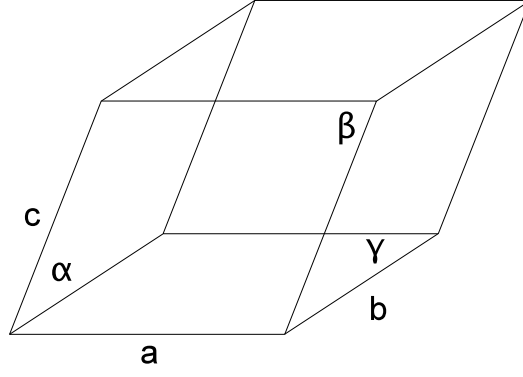


Figure 1: Definition of unit cell parameters. The lengths of the unit cell edges are given by (a,b,c) in units of wavelength. The angles opposite to edges a,b,c are given by α, β, γ in units of degrees.

of diffraction angles possible in the angle range either specified in line 1 or from a measured data file is calculated as

$$2\theta_{hkl} = \arcsin \sqrt{\frac{(h' \sin \alpha)^2 + (k' \sin \beta)^2 + (l' \sin \gamma)^2 + 2h'k'g(\alpha, \beta, \gamma) + 2k'l'g(\beta, \gamma, \alpha) + 2h'l'g(\alpha, \gamma, \beta)}{1 + 2\cos \alpha \cos \beta \cos \gamma - \cos^2 \alpha - \cos^2 \beta - \cos^2 \gamma}} \quad , \quad (159)$$

with $h' = h/a$, $k' = k/b$, $l' = l/c$, and $g(\alpha, \beta, \gamma) = \cos \alpha \cos \beta - \cos \gamma$. Values for Miller indices hkl are determined from the symmetry group and the relevant angle range. Non-unique sets of reflections are reduced to one reflection. The set of unique diffraction angles defines the positions of diffraction peaks in the calculated pattern.

Each reflection is expressed with a peak function specified as first parameter following the unit cell specification. Supported peak functions are exponential followed by four parameters for and voigt followed by two parameters defining width and shape in the order given in Table 10.

The peak function type specification and corresponding parameters are followed by a list of values that specify the intensities of the diffraction peaks. Intensity values are assigned to peaks in ascending order until the list is exhausted. If the list is short of intensity values intensities of one are assigned to the remaining peaks.

10.5 Polynomial function

The power function type uses up to 11 parameters $p_1 \dots p_{11}$ and is defined as follows

$$f(x) = \sum_{n=1}^{11} p_n x^n \quad . \quad (160)$$

All parameters are fitable using substitution directives as outlined in section 4.2.

10.6 Power function

The power function type requires six parameters $p_1 \dots p_6$ and is defined as follows

$$f(x) = \begin{cases} p_1 \left\{ 1 + \left| \frac{x-p_2}{p_3} \right|^{p_4} \right\}^{-1} & \text{for } x < p_2 \\ p_1 \left\{ 1 + \left| \frac{x-p_2}{p_5} \right|^{p_6} \right\}^{-1} & \text{for } x > p_2 \end{cases} \quad . \quad (161)$$

All parameters are fitable using substitution directives as outlined in section 4.2.

10.7 Sigmoid function

The sigmoid function type requires 3 parameters $p_1 \dots p_3$ and is defined as follows

$$f(x) = p_1 \tanh[-p_3(x - p_2)] \quad . \quad (162)$$

All parameters are fitable using substitution directives as outlined in section 4.2.

10.8 Sine function

The sine function type requires 5 parameters $p_1 \dots p_5$ and is defined as follows

$$f(x) = p_1 \exp[-p_5(x - p_2)] \sin[p_3(x - p_2) + p_4] \quad . \quad (163)$$

All parameters are fitable using substitution directives as outlined in section 4.2.

10.9 Voigt function

The voigt function type requires 4 parameters $p_1 \dots p_4$ and is defined as follows

$$f(x) = p_1 \int \frac{\exp[-\xi^2]}{(g(\alpha)(x - p_2)/p_3 - \xi)^2 + \alpha} d\xi \quad \text{with} \quad \alpha = \sqrt{\ln 2} p_4 \quad . \quad (164)$$

where $g(\alpha)$ was empirically determined as

$$g(\alpha) = 2\alpha + \frac{0.424051}{(1 + 0.08806786\alpha)^{1.6963}} + \frac{1.24391}{(1 + 0.08806786\alpha)^{8.40126}} \quad . \quad (165)$$

All parameters are fitable using substitution directives as outlined in section 4.2.

10.10 Parameter fitting

Input data (EIF given in line 1) are specified as N sets $\{x_i, y_i, \delta x_i, \delta y_i\}$ consisting of x- and y-values and their uncertainties. The specified total function is fitted to these data by variation of n parameters $\{p_j\}$ by minimizing the normalized mean-square deviation (method of weighted least squares)

$$\chi^2 = \frac{1}{(N + m - n)} \left\{ \sum_{i=1}^N w_i (y_i - f(x_i, p_i))^2 + \frac{N}{n} \sum_{k=1}^m \frac{(p_k - P_k)^2}{\delta^2 P_k} \right\} \quad \text{with} \quad N + m - n > 0 \quad . \quad (166)$$

Some number m of the fit parameters may have priors P_k with uncertainties δP_k . The weights w_i are determined by uncertainties of the data via

$$w_i^{-1} = \delta^2 y_i + \left(\frac{\partial f}{\partial x} \right)^2 \delta^2 x_i \quad . \quad (167)$$

If data and priors constrain the fit parameters reasonably well a solution with minimal χ^2 is obtained. These optimal parameters have errors δp_j related to data variation and prior uncertainties. The errors are estimated as variances (square errors) by

$$\delta p_j = \sqrt{\sigma_{jj}} \quad \text{with} \quad \sigma_{jj'} = \chi_{min}^2 \left[\sum_{i=1}^N w_i \frac{\partial f(x_i)}{\partial p_i} \frac{\partial f(x_i)}{\partial p_{i'}} + \delta_{ll'} \frac{N}{n} \frac{1}{\delta^2 P_l} \right]_{jj'}^{-1} \quad , \quad (168)$$

where $\sigma_{jj'}$ is the covariance matrix and $[\dots]^{-1}$ denotes matrix inversion. The derivatives are calculated with the optimal parameters. The fit-parameter correlation matrix is given by

$$c_{jj'} = \frac{\sigma_{jj'}}{\sqrt{\sigma_{jj}\sigma_{j'j'}}} \quad . \quad (169)$$

The variance of a function F of the fit parameters is calculated as

$$\delta^2 F(\{p_j\}) = \sum_{jj'=1}^n \sigma_{jj'} \frac{\partial F}{\partial p_j} \frac{\partial F}{\partial p_{j'}} \quad , \quad (170)$$

10.10.1 Fisher information

Fit parameters can be highly correlated leading to overinterpretation of data. The Fisher-information matrix can help identify parameters that are weakly defined by the data. In the context of the minimum χ^2 as maximum likelihood estimator, the Fisher information is calculated as

$$I_{jj'} = \sqrt{\sigma_{jj}\sigma_{j'j'}} \frac{\partial^2 \chi^2}{\partial p_j \partial p_{j'}} \quad , \quad (171)$$

where parameter values are obtained by minimizing χ^2 . The smallest eigenvalues of the Fisher information matrix indicate fit parameter combinations that are least important for minimizing χ^2 . An individual fit parameter p_k is tested by calculating a modified covariance matrix

$$\sigma'_{jj'} = \chi_{min}^2 \left[\sum_{i=1}^N w_i \frac{\partial f(x_i)}{\partial p_l} \frac{\partial f(x_i)}{\partial p_{l'}} + \delta_{ll'} \frac{N}{n} \frac{1}{\delta^2 P_l} + \delta_{lk} \delta_{l'k} \frac{\lambda_k}{\sigma_{kk}} \right]_{jj'}^{-1} \quad , \quad (172)$$

where $\lambda_k = 20000$ basically fixes the parameter. The modified Fisher information matrix is then

$$I'_{jj'} = \sqrt{\sigma'_{jj}\sigma'_{j'j'}} \frac{\partial^2 \chi^2}{\partial p_j \partial p_{j'}} + \delta_{jk} \delta_{j'k} \lambda_k \quad . \quad (173)$$

If fit parameter p_k is only weakly determined by the data the smallest eigenvalues of the modified Fisher information matrix are significantly larger than the smallest eigenvalues of the full Fisher information matrix given in (171).

10.11 Error ellipse (option L)

The error ellipsoid is defined as the hypersurface for which $\chi^2 = (1 + \alpha)\chi_{min}^2$. where $\alpha > 0$ determines the probability that the area of the ellipsoid contains the true parameter values. The error ellipse is a visualization of the correlation between two parameters. They are based on an expansion of χ^2 near its minimal value. With optimal parameter values \bar{p}_j we may write for small deviations $|h_j| \ll 1$

$$\begin{aligned} \chi^2((1 + h_j)\bar{p}_j) &= \chi_{min}^2 + \frac{1}{2} \sum_{jj'=1}^n h_j h_{j'} \bar{p}_j \bar{p}_{j'} \frac{\partial^2 \chi^2}{\partial p_j \partial p_{j'}} \\ &= \chi_{min}^2 + \chi_{min}^2 \sum_{jj'=1}^n h_j h_{j'} D_{jj'} \quad . \end{aligned} \quad (174)$$

The matrix $D_{jj'}$ is symmetric and approximated by

$$D_{jj'} = \frac{\bar{p}_j \bar{p}_{j'}}{\chi_{min}^2} \left\{ \sum_{i=1}^N w_i \frac{\partial y(x_i)}{\partial p_l} \frac{\partial y(x_i)}{\partial p_{l'}} + \delta_{ll'} \frac{N}{n} \frac{1}{\delta^2 P_l} \right\} = \bar{p}_j \bar{p}_{j'} [\sigma]_{jj'}^{-1} \quad , \quad (175)$$

where $[\sigma]^{-1}$ is the inverse of the covariance matrix. With $\chi^2 = (1 + \alpha)\chi_{min}^2$ the equation defining the error ellipsoid is obtained

$$\sum_{jj'=1}^n h_j h_{j'} D_{jj'} = \alpha^2 \quad . \quad (176)$$

Visualization is afforded by singling out groups of two parameters h_l and $h_{l'}$. The corresponding reduced 2×2 covariance matrix is then used to produce the defining equation for the l, l' error ellipse

$$x_l^2 [\sigma]_{ll}^{-1} + 2x_l x_{l'} [\sigma]_{ll'}^{-1} + x_{l'}^2 [\sigma]_{l'l}^{-1} = \alpha^2 \quad , \quad (177)$$

where $[\sigma]^{-1}$ is the inverse of the reduced covariance matrix. Now we introduce polar coordinates (r, ϕ) with the transformation $x_l = \sqrt{\sigma_{ll}} \alpha r \cos \phi$ and $x_{l'} = \sqrt{\sigma_{l'l}} \alpha r \sin \phi$ to result in

$$r^2 = \frac{1 - c_{ll'}^2}{1 - c_{ll'} \sin 2\phi} \quad , \quad (178)$$

where $c_{ll'}$ is the correlation parameter defined by (33). The parametric form of the error ellipse follows accordingly

$$x_l = \alpha \cos \phi \left\{ \frac{\sigma_{ll} (1 - c_{ll'}^2)}{1 - c_{ll'} \sin 2\phi} \right\}^{1/2} \quad \text{and} \quad x_{l'} = \alpha \sin \phi \left\{ \frac{\sigma_{l'l'} (1 - c_{ll'}^2)}{1 - c_{ll'} \sin 2\phi} \right\}^{1/2} . \quad (179)$$

The circumscribing box of this ellipse has the size $\alpha\sqrt{\sigma_{ll}} \times \alpha\sqrt{\sigma_{l'l'}}$. The scaling factor α determines the probability that the area of the ellipse contains the true parameter values. This probability is given by integration of the gaussian probability distribution over the ellipse area. The result for this two-dimensional case is $P(\alpha) = 1 - \exp[-\alpha^2/2]$ and gives 39 %, 86 %, and 99 % for α values of 1, 2, and 3, respectively. Here $\alpha = 1$ is selected to produce a circumscribing box of size $\sqrt{\sigma_{ll}} \times \sqrt{\sigma_{l'l'}}$.

11 Command line options

Each of the executables recognizes the following command line options.

- geometry=<X×Y+U+W>** defines the window geometry for the graphical display, width X, height Y, horizontal position U, vertical position W.
- help** display a list of options and exit.
- infile=<file>** use <file> as input file, the default input file is in_<name of the module>
- nographics** disable runtime graphics.
- quiet** suppress most screen printout.
- version** display the version number and exit.
- vtool** list supported visualization tools.
- vtool=<exe>** use the executable <exe> as data visualization program.