



Version 3.0.1

User Manual

January 1, 2025

Wolfgang Sturhahn



<https://www.nrixs.com>

support@nrixs.com

Contents

1	License information	3
2	Getting started	3
2.1	The GUI	3
2.2	The CLI	4
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
4	The CLI: standard input files	6
4.1	Syntax rules	6
4.2	Assigning fit parameters	7
4.2.1	Specifying priors	7
4.2.2	Specifying parameter boundaries	8
4.3	Assigning sampling parameters	8
4.4	Parameter dependencies	8
5	Forward scattering: kfor	10
5.1	Standard input file	10
5.2	Material data input file	10
5.2.1	Isotope data file	12
5.2.2	Relaxation parameters	13
5.2.3	Pre-defined distributions	14
5.2.4	User-defined distributions	15
5.3	Output files	15
6	Bragg/Laue diffraction: kref	17
6.1	Standard input file	17
6.2	Material data input file	18
6.2.1	Atomic positions	20
6.3	Output files	21
7	Specular reflection: kgin	22
7.1	Standard input file	22
7.2	Output files	23
8	Fourier transformations: kmix	24
8.1	Standard input file	24
8.2	Output files	26

9 Intensities: kfit	27
9.1 Standard input file	27
9.2 Output files	30
10 Fitting data: kctl	31
10.1 Standard input file	31
10.2 Output files	31
11 Exploring parameter space: kmco	33
11.1 Standard input file	33
11.2 Output files	33
12 Quicklooks: kfmf, kgmf, krmf	35
13 Deconvolution: kdec	36
13.1 Standard input file	36
13.2 Output files	37
14 Pure nuclear reflections: kpnr	38
14.1 Standard input file	38
14.2 Output files	38
15 Electric field gradient axes: kvzz	39
15.1 Standard input file	39
15.2 Output files	39
16 Command line options	40
17 Scattering geometry and angle definitions	41

1 License information

CONUSS-3.0.1 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.

CONUSS-3.0.1 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.

A copy of the GNU General Public License is distributed with CONUSS-3.0.1 and can also be seen at <http://www.gnu.org/licenses/gpl.html>.

Any use of results obtained using CONUSS-3.0.1 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 <https://www.nrixs.com>.

2 Getting started

The CONUSS (COherent Nuclear resonant Scattering by Single crystals) software is a collection of scientific applications to aid the evaluation of experimental nuclear resonant scattering data. The software supports simulations of and fits to time and energy spectra from Synchrotron Mössbauer Spectroscopy (SMS), Nuclear Forward Scattering (NFS), conventional Mössbauer Spectroscopy (MBS), Nuclear Bragg/Laue Scattering (NBS), and Nuclear resonant Grazing Incidence Scattering (NGIS).

Prior to the use the program, follow the instructions described in the installation guide. The CONUSS program installation produces several executables, some of them working interdependently: `kctl`, `kdec`, `kfan`, `kfit`, `kfor`, `kfmf`, `kgin`, `kgmf`, `kmco`, `kmix`, `kpnr`, `kref`, `krmf`, `kvzz`, and the graphical user interface (GUI) named `Conuss.app` (MacOS) or `conuss_gui` (Linux). For some Linux types a desktop launcher named `Conuss` should also exist.

The CONUSS 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 CONUSS executables. Proceed as follows

- MacOS

With Finder locate the Application folder into which the GUI app was installed. This is either `'$HOME/Applications'` for a local install or `'/Applications'` for a system-wide install. Start the GUI by double-click on `Conuss.app` but you might have to defeat the Gate Keeper if the app doesn't start. Defeating the Gate Keeper¹: double-click on `Conuss.app`; acknowledge that it can't be opened; select Apple -> System Preferences -> Security & Privacy -> General; at the bottom right click open `Conuss.app`. Later MacOS versions may have different mechanisms.

If the app still does not start error output is saved into file `'$HOME/Library/NRIXS/CONUSS/app.log'` or `'$HOME/CONUSSapp.log'`.

- Linux

Use Desktop launcher or open terminal and type `'conuss_gui'` on the command line.

¹The Gatekeeper mechanism varies for different MacOS versions.

Select CONUSS → Show Examples. A list of examples shows up in lower right panel; right-click on an item and select 'Open As Project'; click on 'run' button on top of middle panel to execute the calculation. Inspect output by left-click on thumbnail in lower left or by menu actions in upper right panel.

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

2.2 The CLI

The executables need input files containing control parameters and sometimes experimental data to run in a meaningful way. It is a good start to check out the examples directory, study the input files, and inspect the results folder for expected output.

CONUSS works with a command-line-interface, i.e., to run any executable you need to open a terminal window first. Commands that you have to type always need to be followed by pressing the return key. For starters, change to the directory with the examples, e.g., by

```
> cd ~/CONUSS-3.0.1/examples/kctl-NFS1
> stat -f "%Su %Sp" .
> 'owner' drwxr-xr-x
```

The 'stat' command checks on ownership and permissions of the current directory. Make sure that you have write permissions in the examples directory. Otherwise just copy the whole directory to your desktop or home directory.

Note that your computer system can only access the CONUSS executables if the directory that they are located in is part of the 'path' defined for your system at the time of login. You can test this, e.g., by trying something like

```
> cd ~
> which kctl
/usr/local/bin/kctl
```

If the second command doesn't succeed the location of the CONUSS executables can be made known permanently to your computer. This is done by modification of the 'path' setting in the login resource file in your home directory, usually something like '.bash.profile', '.profile', or '.login'. You have to logout and login again to update the 'path' settings.

Run the kctl executable with command 'kctl'. The executable reads the standard input files (SIFs) in_kctl, in_kfor, in_kmix, in_kfit, the material data input file Fe.mif, the isotope data file Fe57.mbt, and the data file Fe.dat, produces copious output in the terminal window, and creates various output files. If your installation supports visualization some graphical output is displayed on the screen.

	kgin	kctl	kmco	kfmf	kgmf	krmf
kfor	✓	✓	✓	✓	✓	
kref		✓	✓			✓
kmix		✓	✓	✓	✓	✓
kfit		✓	✓	✓	✓	✓
kgin	✓	✓	✓		✓	

Table 1: Dependency matrix. Executables listed in columns depend on executables listed in rows, e.g., program kgin depends on kfor and itself. The dependency implies that corresponding standard input files (SIFs) have to be supplied to run the program, e.g., to run kctl the SIFs in_kctl, in_kmix, in_kfit, and one of in_kfor, in_kgin, or in_kref have to be prepared. Executables not listed as columns only depend on themselves.

problem	program	SIF	example directory
fitting data	kctl	in_kctl	
forward scattering, time spectra		in_kfor	kctl-NFS1, kctl-NFS2
dual fit		in_kfor	kctl-NFS3
forward scattering, energy spectrum		in_kfor	kctl-NFS4
forward scattering, TDI spectrum		in_kfor	kctl-TDI
Mössbauer spectroscopy		in_kfor	kctl-MBS1, kctl-MBS2
grazing incidence		in_kgin	kctl-GINS
Bragg/Laue diffraction		in_kref	kctl-NBS1, kctl-NBS2
explore parameter space	kmco	in_kmco	
forward scattering or Mössbauer		in_kfor	kmco-NFS
grazing incidence		in_kgin	kmco-GINS
Bragg/Laue diffraction		in_kref	kmco-NBS
calculate spectra	kfmf		
forward scattering or Mössbauer		in_kfor	kfmf-NFS, kfor-NFS
grazing incidence		in_kgin	kgmf-GINS, kgmf-GIS
Bragg/Laue diffraction		in_kref	krmf-NBS1, krmf-NBS2

Table 2: Problems in nuclear resonant scattering and programs for solving them. In addition to the given standard input files, the SIFs `in_kmix` and `in_kfit` as well as material data input file(s) are required. The last column gives names of directories located in `CONUSS-3.0.1/examples` that contain all needed input files for a specific problem.

The functionality of the CONUSS executables is adapted to specific tasks by changing the SIFs that control the program flow. It is best to use the files `in_kctl`, `in_kfor`, `in_kref`, `in_kmix`, `in_kfit`, `in_kgin`, `in_kmco`, `in_kfan`, `in_kpnr`, `in_kvzz` which can be found in the examples folder, as blueprints. See Table 1 on page 4 for dependencies between individual executables and their SIFs. Output files are often ASCII formatted with two to four columns of numbers. They are easily imported into plotting programs such as Grace and spreadsheets such as LibreOffice, OpenOffice, or Excel. In addition to output files with numerical results, each executable creates a protocol file with a list of input parameters and main results. A detailed description of each executable’s purpose and their input and output files follows.

The CONUSS software offers great flexibility in solving problems and extracting information from nuclear resonant scattering spectra. It contains several examples to demonstrate functionality and input files. Table 2 shows which example to start with for some common problems. The examples `kctl-NFS2`, `kctl-NFS3`, `kctl-NFS4`, and `kctl-TDI` require a binary file named `ss.bin` which has to be generated first using example `kfor-NFS`. The file `ss.bin` then needs to be copied or linked into directories `kctl-NFS2`, `kctl-NFS3`, `kctl-NFS4`, and `kctl-TDI`.

3 Using the GUI

The use of the GUI is often self-explanatory or helped by the tooltips. The CONUSS GUI’s ‘memory’ are project directories: input and output files including data for fit projects are saved here. Each example that shows via `CONUSS → 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 `CONUSS → Open...`; select an item with extension ‘.prc’ in the lower right directory listing, `popupMenu → Open As Project`; via `Edit → Find Project` as explained below.

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 CONUSS → Save As to create a new project with identical parameters. In the second case, via CONUSS → New Project use the project wizzard to create a new project with default parameters.

3.3 Find a project

Finding CONUSS 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 to a new location.

4 The CLI: standard input files

Each CONUSS executable requires a standard input file (SIF). In addition, the kfor, kgin, and kref executables require a material data input file (MIF) the name of which is defined in the SIF. The MIF and the 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 could be misinterpreted by a CONUSS executable. ⇐

The MIF requires an isotope data file that defines properties of the nuclear transition. Table 3 lists isotope data files that are provided with the CONUSS-3.0.1 distribution. These files are located in the directory CONUSS-3.0.1/etc and, after installation, also in the directory DIR/CONUSS/etc where DIR depends on the installation directory and your computing system (see section 5 of the Installation Guide for details). Isotope data files can be added to those locations or to a present working directory.

4.1 Syntax rules

The contents of a line in the SIF/MIF is ignored if: “*” is the first character, it is empty, or it is filled with spaces only. Lines starting with “*” are comments lines, can be added anywhere in the SIF/MIF, and permit users to make comments and notes.

Lines starting with “@” are substitution directives which declare variables. They are interpreted and may appear anywhere in the SIF/MIF. This type of input line has four parts separated by spaces as can be seen in this example

`@ Bhyper := 33`

file name	isotope	transition (keV)
Fe57.mbt	⁵⁷ Fe	14.4125
Dy161.mbt	¹⁶¹ Dy	25.655
Eu151.mbt	¹⁵¹ Eu	21.5414
Gd157.mbt	¹⁵⁷ Gd	63.917
Ir193.mbt	¹⁹³ Ir	74.04
Sn119.mbt	¹¹⁹ Sn	23.88
Ta181.mbt	¹⁸¹ Ta	6.2157
Tm169.mbt	¹⁶⁹ Tm	8.41

Table 3: Pre-installed isotope data files.

The @ character is followed by variable name, assignment symbol :=, and variable value. The variable name may be up to 32 characters long and must not contain spaces, tabs, non-printable characters, and most special characters. Permissible are alphanumeric characters and the special character underscore. Variable names are case sensitive. In the lines that follow the substitution directive, the variable name directly preceded by @ is substituted by its value as declared earlier. Substitution directives are useful if values are substituted at several locations in the SIF/MIF and for the definitions of boundaries, see 4.2.2. This is an example.

```
@ Bhyper := 33
.... (other lines) ....
hyperfine field site 1 :: @Bhyper
.... (other lines) ....
hyperfine field site 2 :: @Bhyper
```

Lines in the SIF/MIF that are not comments, empty, or substitution directives 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., a real number 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

The executables kctl accepts data input files and allows 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 hyperfine field parameter in the following input line

```
(7.1.4) magnetic hyperfine field / T :: 33
```

turns into a fit parameter via

```
% (7.1.4) magnetic hyperfine field / T :: 33
```

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

```
% @ Bhyper := 33
.... (other lines) ....
(7.1.4) magnetic hyperfine field / T :: @Bhyper
```

4.2.1 Specifying priors

In multi-parameter fits, parameters are often considered to be determined by the data set at hand only. However, in some 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. An expected value for a fit parameter (prior) is specified as follows

```
% (7.1.4) magnetic hyperfine field / T :: 33 0.1
```

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).

4.2.2 Specifying parameter boundaries

Some fit parameters have physical boundaries that should not be violated by the fit procedure. For example, the weight of a site or the width of a hyperfine field distribution must be positive. Also cyclic parameters, such as angles, may lead to instabilities if allowed to take any value. A fit parameter may be constrained to a range in the following way

```
% @ [0,0.2] weight1 := 0.1
.... (other lines) ....
(7.1.2) weight of the site :: @weight1
```

Note how the permitted range for the fit parameter is specified by inserting [value1,value2] between the @ and the name of the variable. The space separators are required here.

4.3 Assigning sampling parameters

Executable kmco also accepts data input files and allows parameter sampling to minimize the weighted mean-square deviation between calculation and data. Any input parameter may be promoted to a sampling parameter in the same manner as described for fit parameters. However, the definition of a sampling parameter requires an additional value for the size of the sampling box and is achieved via

```
% (7.1.4) magnetic hyperfine field / T :: 33 1
```

where the sampling box is now between 32 T and 34 T.

In addition to the simple specification of a sampling parameter shown above, more complicated shapes of the sampling box can be defined.

```
% @ [-2,2] Bhf := 33 g_1
.... (other lines) ....
(7.1.4) magnetic hyperfine field / T :: @Bhf
```

This example defines a gaussian sampling box with half width at half maximum of 1. The support of the gaussian box is given by the numbers in brackets. It means that the box ranges from 31 to 35, i.e., it is of finite size. Other options are the following.

```
* an exponential box with HWHM of 0.5
% @ [-2,2] Bhf := 33 e_0.5
*
* a rectangular box with HWHM of 1, no need for the support
% @ Bhf := 33 r_1
*
* a U-shaped gaussian box with HWHM of 1
% @ [-2,2] Bhf := 33 u_1
```

4.4 Parameter dependencies

Fit or sampling parameters can become dependent through substitution directives using mathematical expressions. The expression syntax is governed by the rules of the **expr** command of the tcsh scripting language (see here). For example

```
% @ Bhyper := 33
% @ deltaB := 1.2
@ Bhyper_1 := @Bhyper - @deltaB
@ Bhyper_2 := @Bhyper + @deltaB
@ Bhyper_3 := 0.5*@Bhyper + 0.8*@deltaB
.... (other lines) ....
hyperfine field site 1 :: @Bhyper_1
.... (other lines) ....
hyperfine field site 2 :: @Bhyper_2
.... (other lines) ....
hyperfine field site 3 :: @Bhyper_3
```

Here the fields of three sites depend on only two independent parameters which can be fitted or sampled. The dependent parameters are now varied via their dependencies and must not be declared fit/sampling parameters themselves.

5 Forward scattering: kfor

The kfor program calculates the energy-polarization dependent transmission amplitudes of nuclear resonant scatterers in the forward scattering geometry. Optionally the amplitudes can be calculated at various thicknesses of the sample. The executable is run in two ways: standalone or via another executable such as kctl, kgin, kmco, kfmf, kgmf. The results of a calculation are either saved to a binary file if run as standalone or are internally forwarded otherwise.

5.1 Standard input file

The SIF of the kfor executable has 6 lines of data. The content of these lines is discussed in order of appearance.

1. project name – one value :
This value gives a name to the calculation that is performed.
2. name of MIF – one value :
This value gives the name of a material data input file.² The MIF is a plain text file containing no more than 200 characters per line (excess characters will be ignored). The content of the MIF is discussed in section 5.2.
3. name of output file – one value :
This value gives the name of a binary output file readable to kmix. If kfor is run via another executable the value is ignored.
4. direction of incident x-ray – three values :
These values give the direction of the x-ray incident on the sample in cartesian coordinates. See Figure 1 on page 41 for illustration.
5. energy range – three values :
The first value gives the energy range of the calculation in units of natural linewidths Γ . The energy range is centered around zero. The following two values are optional. The second value gives the number of energy points used in the calculation. The third value is a non-blank string which causes a bypass of energy range verification. Specifying this value can be useful for Mössbauer spectra that do not require an extended energy range.
6. thickness range – three values :
The first value gives the thickness of the material in units of μm . The following two values are optional. If the second value is specified the first and second value give start and end of the thickness range, respectively. The third value gives the number of thickness points used in the calculation.

5.2 Material data input file

The MIF of the kfor executable has at least 20 lines of data. The content of these lines is discussed in order of appearance.

1. material composition – one value :
This value gives the composition of the material in the form $A\{n\}B\{m\}$ where A,B are element

²Names may be defined either relative to the working directory or absolute: file.mif must exist in the working directory, ../file.mif must exist in the parent directory, myDir/file.mif must exist in the directory myDir in the working directory, /Users/data/file.mif must exist in the directory /Users/data .

symbols and n,m are their abundances. For example, a particular stainless steel would be specified as Fe{0.55}Cr{0.25}Ni{0.2}, and tin monoxide would be given as Sn{1}O{1}.

2. material density – one value :

This value gives the density of the material in units of g/cm³.

3. name of isotope data file – one value :

This value gives the name of the file containing information about the nuclear resonant isotope. The content is explained in 5.2.1. Files provided with the CONUSS distribution are listed in Tab.3. As an alternative to the file name, the inline specification of isotope parameters is possible

```
(3) isotope data :: <!
.... (13 valid lines for the isotope parameters) ....
!
```

4. abundance – one value :

This value gives the abundance of the nuclear resonant isotope. Its value lies between 0 and 1.

5. Lamb-Mössbauer factor – one value :

This value gives the Lamb-Mössbauer factor. Its value lies between 0 and 1.

6. number of sites – one value :

This value gives the number of sites occupied by the nuclear resonant isotope with different hyperfine interactions.

The following 14 lines specify the properties of one site. For each site one block of 14 lines must be specified. Additional lines are needed if relaxation parameters or distributions are defined. These are described in sections 5.2.2 on page 13 and 5.2.3 on page 14.

7. name of site – one value :

This value gives a name to this site.

8. weight – one value :

This value gives the weight of this site which must be positive. The weights of all sites will be normalized to one.

9. isomer shift – one value :

This value gives isomer shift for resonant isotopes occupying this site in units of mm/s.

10. magnetic hyperfine field – one value :

This value gives magnetic hyperfine field for resonant isotopes occupying this site in units of Tesla.

11. magnetic hyperfine field direction, polar angle – one value :

This value gives the polar angle for the direction of the magnetic hyperfine field in units of degrees. See Figure 1 on page 41 for illustration.

12. magnetic hyperfine field direction, azimuthal angle – one value :

This value gives the azimuthal angle for the direction of the magnetic hyperfine field in units of degrees. See Figure 1 on page 41 for illustration.

13. quadrupole splitting – one value :

This value gives the quadrupole splitting of the excited nuclear state in units of mm/s and is defined by $\Delta = eqV_{zz}/2\sqrt{1+\eta^2/3}$, with elementary charge e , quadrupole moment of the excited nuclear state q , the eigenvalue of largest magnitude V_{zz} of the electric field gradient tensor at the nuclear position, and the asymmetry parameter η (see next input field).

14. asymmetry parameter – one value :

This value gives the deviation of the electric field gradient tensor from axial symmetry. If V_{xx}, V_{yy}, V_{zz} are the eigenvalues of the electric field gradient tensor and $|V_{zz}| \geq |V_{yy}| \geq |V_{xx}|$ then the asymmetry parameter is defined as $(V_{xx} - V_{yy})/V_{zz}$. It takes values between 0 and 1. In case of axial symmetry, i.e., the asymmetry parameter is zero, the Euler angles β and α are the polar and azimuthal angles that define the direction of the eigenvector corresponding to eigenvalue V_{zz} , and Euler angle γ becomes irrelevant.

15. Euler angle α – one value :

This value gives the first Euler angle for rotation of the electric field gradient eigenvectors into the cartesian reference frame. See Figure 5 on page 43 for illustration.

16. Euler angle β – one value :

This value gives the second Euler angle for rotation of the electric field gradient eigenvectors into the cartesian reference frame. See Figure 5 on page 43 for illustration.

17. Euler angle γ – one value :

This value gives the third Euler angle for rotation of the electric field gradient eigenvectors into the cartesian reference frame. See Figure 5 on page 43 for illustration.

18. relaxation parameters – one value :

This value is optional but if specified must appear as follows.

```
(7.1.12) relaxation parameters :: <!  
...  
    (parameter definitions)  
...  
!
```

The parameter definitions and other needed modifications to the MIF are described in section 5.2.2 on the following page.

19. distribution description – one value :

This value is optional but if specified appears as follows.

```
(7.1.13) distribution description :: <!  
...  
    (parameter definitions)  
...  
!
```

An alternative form is

```
(7.1.13) distribution description :: <file_name>
```

where the file contains the parameter definitions. The parameter definitions are described in sections 5.2.3 on page 14 and 5.2.4 on page 15.

20. texture coefficient – one value :

This value gives the complement of fraction of randomly oriented hyperfine fields in units of %. For example, complete randomness is given by a texture coefficient of 0.

5.2.1 Isotope data file

The isotope data file has 13 lines of data. The content of these lines is discussed in order of appearance.

1. element – one value :

This value gives the element symbol of the nuclear resonant isotope.

2. mass number – one value :
This value gives the mass number of the nuclear resonant isotope.
3. spin – one value :
This value gives the spin of the nuclear ground state.
4. g-factor – one value :
This value gives the gyro-magnetic ratio of the nuclear ground state.
5. quadrupole moment – one value :
This value gives the quadrupole moment of the nuclear ground state in units of barn ($= 10^{-24} \text{ cm}^2$).
6. spin – one value :
This value gives the spin of the nuclear excited state.
7. g-factor – one value :
This value gives the gyro-magnetic ratio of the nuclear excited state.
8. quadrupole moment – one value :
This value gives the quadrupole moment of the nuclear excited state in units of barn ($= 10^{-24} \text{ cm}^2$).
9. half-life time – one value :
This value gives the half-life time of the nuclear excited state in units of ns.
10. internal conversion factor – one value :
This value gives the internal conversion factor of the nuclear transition.
11. energy – one value :
This value gives the nuclear transition energy in units of keV.
12. multipolarity – three values :
The first value gives the major multipolarity of the nuclear transition, e.g., M1 or E2 specify magnetic dipole and electric quadrupole transitions, respectively. The following two values are optional. The second value gives the minor multipolarity of the nuclear transition, and the third value specifies the multipole mixing ratio.
13. interference coefficient – one value :
This value gives the interference between nuclear and electronic currents for a given multipolarity.

5.2.2 Relaxation parameters

The relaxation parameters are supplied in form of the relaxation matrix. For N relaxation states, N lines with N values need to be specified. The values are given in units of jumps per nuclear lifetime. Note that these lines do not contain any double colons '::'. An example for two relaxation states is given below.

```
(7.1.12)  relaxation parameters :: <!
      -50  50
       10 -10
      !
```

The hyperfine parameters of the first relaxation state are given by lines preceeding the relaxation parameter definition. In addition, hyperfine parameters for the remaining N-1 relaxation states must be specified. For this purpose, N-1 blocks consisting of 10 parameter definitions equivalent to lines 7, 8-17 have to added after the “texture coefficient” specification in line 20.

An example for relaxation is provided in directory CONUSS-3.0.1/examples/kfmf-NFS.

5.2.3 Pre-defined distributions

The input lines defining hyperfine field distributions do not contain any double colons '::'. The syntax for specification of a pre-defined distribution is

```
(7.1.13)  distribution  description  ::  <!
          Target  <field>
          Attach  <otherField> <gradient>
          Make  <type>_<shape> <points>  <FWHM1> <FWHM2>
          !
```

The 'Target' line specifies the distributed hyperfine parameter. The permitted values of <field> and corresponding distributed quantities are

mag magnetic hyperfine field
efg electric quadrupole splitting
iso isomer shift
theta polar angle of the magnetic hyperfine field
phi azimuthal angle of the magnetic hyperfine field

The 'Attach' line is optional. If specified it perfectly correlates the distribution of <field> with <otherField> using the relation

$$\Delta < \text{otherField} > = < \text{gradient} > \cdot \Delta < \text{field} > \quad ,$$

where Δ symbolizes deviation from the center value.

The 'Make' line gives the functional shape of the distribution function which depends on these values.

<type> Exponential or Power
<shape> positive number to determine the shape of the distribution
<points> number of grid points, has to be adapted to the width, usually 50-100,
 distributions with long tails may require more points
<FWHM1> full width at half maximum of low value side of the distribution
<FWHM2> optional - full width at half maximum of high value side of the distribution,
 if omitted <FWHM1> becomes FWHM of distribution

Values for the widths are absolute. The exponential function type is defined by

$$f(x) = \exp \left[-\ln 2 \left(\frac{2|x-h|}{w_{\leq}} \right)^s \right] \quad \text{for } x \leq h \quad ,$$

where h is the value of the distributed hyperfine field, w_{\leq} are the full-width-at-half-maximum values, and s is the shape. For the power function type, we have

$$f(x) = \left\{ 1 + \left(\frac{2|x-h|}{w_{\leq}} \right)^s \right\}^{-1} \quad \text{for } x \leq h \quad .$$

Examples for pre-defined hyperfine field distributions are provided in directories kctl-GINS, kctl-MBS2, kfor-NFS, kmco-GINS, and kmco-NFS all of which can be found in the example directory CONUSS-3.0.1/examples.

Besides the generic distributions there are two special distributions: directions of the magnetic hyperfine field evenly distributed in a plane; the orientations of an axially symmetric electric field gradient evenly distributed over a sphere. The first case is defined by

```
(7.1.13)  distribution  description  ::  <!
          Target  mag
          Make  plane <points>  <theta> <phi>
          !
```


where $\langle \text{points} \rangle$ is the number of grid points to generate the distribution. The angles $\langle \text{theta} \rangle$ and $\langle \text{phi} \rangle$, given in degrees, are the spherical coordinates of the plane normal in the reference system, see Figs. 1, 2, and 4.

The second case is defined by

```
(7.1.13) distribution description :: <!  
    Target efg  
    Make grid <points>  
    !
```

where $\langle \text{points} \rangle$ is the number of grid points for one angle. The total number of grid points on the sphere is the square of $\langle \text{points} \rangle$. This averaging process uses a Gauss-Legendre tiling of the sphere.

5.2.4 User-defined distributions

The input lines defining hyperfine field distributions do not contain any double colons '::'. The syntax for specification of a generic user-defined distribution is

```
(7.1.13) distribution description :: <!  
    Target <field>  
    Make custom <total_weight1>  
    <value1> <weight1>  
    <value2> <weight2>  
    ....  
    Make custom <total_weight2>  
    <value1> <weight1>  
    <value2> <weight2>  
    ....  
    !
```

The 'Target' line specifies the distributed hyperfine parameter. The permitted values of $\langle \text{field} \rangle$ and corresponding distributed quantities are

mag	magnetic hyperfine field
efg	electric quadrupole splitting
iso	isomer shift
theta	polar angle of the magnetic hyperfine field
phi	azimuthal angle of the magnetic hyperfine field

Each "Make custom" block provides a distribution in the form of value-weight pairs. The total weight of each distribution block is normalized to the corresponding value of $\langle \text{total_weight}^* \rangle$. If only one "Make custom" block is provided $\langle \text{total_weight1} \rangle$ is ignored.

5.3 Output files

The kfor executable produces one binary file and several text files. Existing files are overwritten without notice. The name of the binary file is taken from line 3 of the SIF. The unformatted write statements can be gleaned from source code file savtrs.F located in directory CONUSS-3.0.1/src/for/. The text files are

<MIF>_kfor_log.txt log-file providing a summary of input parameters;

<MIF>_kfor_ptl.txt protocol file with input parameters, intermediate and final results, e.g., directional definitions. The amount of reported results can be specified in file kforParms.txt;

<MIF>_kfor_let.dat local effective thickness versus energy. A single-line scatterer with effective thickness of one has a maximum local effective thickness of one as well;

<MIF>_site<N>_<type>_dist.dat probability distribution of hyperfine parameter <type> of site <N>. This is a two-column ASCII file with first column value of distributed hyperfine parameter and second column value probability density.

Here <MIF> represents the name of the material data input file after removal of the extension.

6 Bragg/Laue diffraction: kref

The kref executable calculates the energy-polarization-angle dependent reflection/transmission amplitudes of nuclear resonant scatterers for Bragg/Laue scattering geometries. The executable is run in two ways: standalone or via another executable such as kctl, kmco, krmf. The results of a calculation are either saved to a binary file if run as standalone or are internally forwarded otherwise.

6.1 Standard input file

The SIF of the kref executable has 9 lines of data. The content of these lines is discussed in order of appearance.

1. project name – one value :
This value gives a name to the calculation that is performed.
2. name of MIF – one value :
This value gives the name of a material data input file.³ The MIF is a plain text file containing no more than 200 characters per line (excess characters will be ignored). The content of the MIF is discussed in section 5.2.
3. name of output file – one value :
This value gives the name of a binary output file readable to kmix. If kfor is run via another executable the value is ignored.
4. run mode – one value :
This value specifies the type of data to produce: r for calculating reflected field amplitudes; t for calculating transmitted field amplitudes.
5. Miller indices – three values :
These values give the Miller indices of the Bragg/Laue reflection that is calculated.
6. direction of surface normal – three values :
These values give the orientation of the crystal surface in crystal coordinates. See Figure 4 on page 42 for illustration.
7. direction of incident x-rays – one value :
This value gives the angle between the incident x-rays projected onto the crystal surface and a reference vector in units of degrees. For a symmetric reflection, the reference vector is given by unit cell base vector 1 projected onto the crystal surface. Otherwise the reference vector is defined by the intersection of surface and netplanes. See Figure 4 on page 42 for illustration.
8. crystal thickness – one value :
This value is optional. If specified it gives the thickness of the crystal perpendicular to the surface in units of μm . If the value is not specified, zero, or negative the crystal is assumed semi-infinite. This is meaningful only for Bragg geometries.
9. energy range – two values :
The first value gives the energy range of the calculation in units of natural linewidths Γ . The energy range is centered around zero. The second value is optional and gives the number of energy points used in the calculation.

³Names may be defined either relative to the working directory or absolute: file.mif must exist in the working directory, ../file.mif must exist in the parent directory, myDir/file.mif must exist in the directory myDir in the working directory, /Users/data/file.mif must exist in the directory /Users/data .

10. angle range — three values :

The first two values give beginning and end of the angle range in units of μrad . The third value is optional and gives the number of angle points used in the calculation. The angles define the deviation of the incident x-ray direction from the kinematic Bragg angle.

6.2 Material data input file

The MIF of the kref executable has at least 23 lines of data. The content of these lines is discussed in order of appearance.

1. name of isotope data file — one value :

This value gives the name of the file containing information about the nuclear resonant isotope. Content and availability are explained in 5.2.1. As an alternative to the file name, the inline specification of isotope parameters is possible

```
(1) isotope data :: <!  
.... (13 valid lines for the isotope parameters) ....  
!
```

2. abundance — one value :

This value gives the abundance of the nuclear resonant isotope. Its value lies between 0 and 1.

3. Lamb-Mössbauer factor, forward — one value :

This value gives the Lamb-Mössbauer factor for scattering in the forward direction. Its value lies between 0 and 1.

4. Lamb-Mössbauer factor, reflected — one value :

This value gives the Lamb-Mössbauer factor for scattering in the reflection direction. Its value lies between 0 and 1.

5. length of unit cell edges — three values :

These values give the length of the three different edges of the unit cell in units of Å. See Figure 3 on page 42 for illustration.

6. angles between unit cell base vectors — three values :

These values give the angles between vectors 2 and 3, 3 and 1, 1 and 2 in units of degrees, respectively. See Figure 3 on page 42 for illustration.

7. number of sites — one value :

This value gives the number of sites occupied by the nuclear resonant isotope with different hyperfine interactions.

The following 16 lines specify the properties of one resonant site. For each resonant site, one block of at least 16 lines must be specified. Additional lines may be needed for atomic positions, relaxation parameters, or distributions. The latter are described in sections 5.2.2 on page 13 and 5.2.3 on page 14.

8. name of site — one value :

This value gives a name to this site.

9. weight — one value :

This value gives the weight of this site which must be positive.

10. ionization number — one value :

This value gives the ionization number of the resonant atom.

11. atomic positions – one value :

This value must appear as follows.

```
(7.1.4)  atomic positions :: <!  
        ...  
        (positions)  
        ...  
        !
```

The position definitions are described in section 6.2.1 on the next page.

12. isomer shift – one value :

This value gives isomer shift for resonant isotopes occupying this site in units of mm/s.

13. magnetic hyperfine field – one value :

This value gives magnetic hyperfine field for resonant isotopes occupying this site in units of Tesla.

14. magnetic hyperfine field direction, polar angle – one value :

This value gives the polar angle for the direction of the magnetic hyperfine field in units of degrees. See Figure 4 on page 42 for illustration.

15. magnetic hyperfine field direction, azimuthal angle – one value :

This value gives the azimuthal angle for the direction of the magnetic hyperfine field in units of degrees. See Figure 4 on page 42 for illustration.

16. quadrupole splitting – one value :

This value gives the quadrupole splitting of the excited nuclear state and is defined by $\Delta = eqV_{zz}/2$, with elementary charge e , quadrupole moment of the excited nuclear state q , and the eigenvalue of largest magnitude V_{zz} of the electric field gradient tensor at the nuclear position.

17. asymmetry parameter – one value :

This value gives the deviation of the electric field gradient tensor from axial symmetry. If V_{xx}, V_{yy}, V_{zz} are the eigenvalues of the electric field gradient tensor and $|V_{zz}| \geq |V_{yy}| \geq |V_{xx}|$ then the asymmetry parameter is defined as $(V_{xx} - V_{yy})/V_{zz}$. It takes values between 0 and 1. In case of axial symmetry, i.e., the asymmetry parameter is zero, the Euler angles β and α are the polar and azimuthal angles that define the direction of the eigenvector corresponding to eigenvalue V_{zz} , and Euler angle γ becomes irrelevant.

18. Euler angle α – one value :

This value gives the first Euler angle for rotation of the electric field gradient eigenvectors into the cartesian reference frame. See Figure 5 on page 43 for illustration.

19. Euler angle β – one value :

This value gives the second Euler angle for rotation of the electric field gradient eigenvectors into the cartesian reference frame. See Figure 5 on page 43 for illustration.

20. Euler angle γ – one value :

This value gives the third Euler angle for rotation of the electric field gradient eigenvectors into the cartesian reference frame. See Figure 5 on page 43 for illustration.

21. relaxation parameters – one value :

This value is optional but if specified must appear as follows.

```
(7.1.12)  relaxation parameters :: <!  
        ...  
        (parameter definitions)  
        ...  
        !
```

The parameter definitions and other needed modifications to the MIF are described in section 5.2.2 on page 13.

22. distribution description – one value :

This value is optional but if specified must appear as follows.

```
(7.1.13)  distribution description :: <!  
          ...  
          (parameter definitions)  
          ...  
          !
```

An alternate form is

```
(7.1.13)  distribution description :: <file_name>
```

where the file contains the parameter definitions. The parameter definitions are described in sections 5.2.3 on page 14 and 5.2.4 on page 15.

23. texture coefficient – one value :

This value gives the complement of fraction of randomly oriented hyperfine fields in units of %. For example, complete randomness is given by a texture coefficient of 0.

The following line defines non-resonant sites. For each non-resonant site, one block of at least 4 lines must be specified. Additional lines may be needed for atomic positions.

24. number of sites – one value :

This value gives the number of sites occupied by non-resonant atoms. If zero no additional input lines are needed.

25. type of atom – one value :

This value gives the element symbol of the non-resonant atom.

26. weight – one value :

This value gives the weight of this site which must be positive.

27. ionization number – one value :

This value gives the ionization number of the non-resonant atom.

28. atomic positions – one value :

This value must appear as follows.

```
(9.1.4)  atomic positions :: <!  
          ...  
          (positions)  
          ...  
          !
```

The position definitions are described next.

6.2.1 Atomic positions

The position coordinates are supplied as number triplets, two per input line. The number of given triplets defines the number of atoms in a particular site. Note that these lines do not contain any double colons '::'. An example for a site with two atoms is given below.

```
(9.1.4)  atomic positions :: <!  
          0 0 0  0.5 0.5 0.5  
          !
```

The coordinates are understood in the reference frame of the unit cell and are potentially non-cartesian. Let $\mathbf{a}, \mathbf{b}, \mathbf{c}$ present vectors along the edges of the unit cell with lengths and included angles given by lines 5 and 6 of the MIF. Then the coordinates u, v, w place the atom at the position $\mathbf{r} = u\mathbf{a} + v\mathbf{b} + w\mathbf{c}$ in the unit cell. Examples for single-crystal problems are listed in Tab. 2.

6.3 Output files

The kref executable produces one binary file and several text files. Existing files are overwritten without notice. The name of the binary file is taken from line 3 of the SIF. The unformatted write statements can be gleaned from source code file savref.F located in directory CONUSS-3.0.1/src/ref/. The text files are

<MIF>_kref_log.txt log-file providing a summary of input parameters;

<MIF>_kref_ptl.txt protocol file with input parameters, intermediate and final results, e.g., directional definitions. The amount of reported results can be specified in file krefParms.txt;

<MIF>_kref_uc.mld extended unit cell atomic positions in MOLDAN format which can be visualized using the VESTA software;

<MIF>_site<N>_<type>_dist.dat probability distribution of hyperfine parameter <type> of site <N>. This is a two-column ASCII file with first column value of distributed hyperfine parameter and second column value probability density.

Here <MIF> represents the name of the material data input file after removal of the extension.

7 Specular reflection: kgin

The kgin executable calculates the energy-polarization-angle dependent reflection amplitudes of nuclear resonant scatterers for specular reflection under grazing incidence. The executable is run in two ways: standalone or via another executable such as kctl, kmco, kgmf. The results of a calculation are either saved to a binary file if run as standalone or are internally forwarded otherwise.

The scatterers are combinations of laterally smooth layers consisting of materials that may contain nuclear resonant scatterers. The nuclear resonant scattering properties are calculated using the kfor executable. All fit parameters pertaining to material data input files for the resonant scatterers must be declared via substitution directive in the SIF. They are global to all material data input files. See Figure 2 on page 41 for illustration of scattering geometry and polarization vectors.

7.1 Standard input file

The SIF of the kgin executable has at least 9 lines of data. The content of these lines is discussed in order of appearance.

1. project name – one value :
This value gives a name to the calculation that is performed.
2. name of output file – one value :
This value gives the name of a binary output file readable to kmix. If kgin is run via another executable the value is ignored.
3. energy range – three values :
The first value gives the energy range of the calculation in units of natural linewidths Γ . The energy range is centered around zero. The following two values are optional. The second value gives the number of energy points used in the calculation. The third value is a non-blank string which causes a bypass of energy range verification. Specifying this value can be useful for Mössbauer spectra that do not require an extended energy range.
4. angle range – three values :
The first value specifies the angle of the incident x-rays with respect to the surface in units of mrad. The following two values are optional. If the second value is specified the first and second value give start and end of an angle range, respectively. The third value gives the number of angle points used in the calculation.
5. x-ray energy – one value :
This value specifies the energy of the incident x-rays in units of keV. This value is superseded by the nuclear transition energy if resonant layers are part of the layer system.
6. layer system definition – multiple values :
The first value specifies the geometrical arrangement of thin layers that form the layer system from top to bottom. The syntax for a layer system definition is

```
(6) layer system :: layer1 + <N>*layer2 + layer3 + ..  
Define layer1  
  Ag 10.49 10 0  
Define layer3  
  W 19.25 -1 0  
Define layer2  
  fe.mif 8 16 0
```


The optional multiplier $\langle N \rangle$ indicates repetition and can be used to define multi-layers. The '+' and '*' characters are operators. Each part of the layer system, 'layer1', 'layer2', and 'layer3' in the example above, is defined separately following input line 6. Here 'layer1' consists of a 10 Å thick layer (col. 3) of silver (col. 1) with density 10.49 g/cm³ (col. 2) and no roughness (col. 4). 'layer2' contains a Mössbauer isotope and is described by a material input data file 'fe.mif' of type described in section 5.2 on page 10. The density, thickness, and roughness of this resonant layer are 8 g/cm³, 16 Å, and 0 Å, respectively. The bottom 'layer3' is of semi-infinite thickness indicated by a negative number.

The definition of a layer can be more complex, e.g., two resonant iron layers with differing hyperfine interactions that are sandwiched between chromium would be specified as

Define combined-layer			
fe1.mif	8	10	0
Cr	7.19	10	0
fe2.mif	8	10	0
Cr	7.19	10	0

This definition could then be used in line 6 in combination with a multiplier $\langle N \rangle$ to create a resonant multilayer. This example is provided in directory 'examples/kgmf-GINS'.

7.2 Output files

The kgin executable produces one binary file and several text files. Existing files are overwritten without notice. The name of the binary file is taken from line 2 of the SIF. The unformatted write statements can be gleaned from source code file savgrf.F located in directory CONUSS-3.0.1/src/gin/. The text files are

kgin_ptl.txt protocol file with input parameters and layer system summary;

$\langle \text{MIF} \rangle$ **_kfor_log.txt** log-file providing a summary of input parameters;

$\langle \text{MIF} \rangle$ **_kfor_ptl.txt** protocol file with input parameters, intermediate and final results, e.g., directional definitions. The amount of reported results can be specified in file kforParms.txt;

$\langle \text{MIF} \rangle$ **_kfor_let.dat** local effective thickness versus energy. A single-line scatterer with effective thickness of one has a maximum local effective thickness of one as well;

$\langle \text{MIF} \rangle$ **_site $\langle N \rangle$ _ $\langle \text{type} \rangle$ _dist.dat** probability distribution of hyperfine parameter $\langle \text{type} \rangle$ of site $\langle N \rangle$. This is a two-column ASCII file with first column value of distributed hyperfine parameter and second column value probability density.

Here $\langle \text{MIF} \rangle$ represents the name of each material data input file occurring in the layer system definition after removal of the extension.

8 Fourier transformations: kmix

The kmix executable is run in two ways: standalone or via another executable such as kctl, kgin, kmco, kfmf, kgmf, krmf. The kmix program calculates the Fourier transform of the energy-polarization-thickness or energy-polarization-angle dependent nuclear resonant scattering amplitudes and calculates scattered intensities. The optional Fourier transformation leads to time spectra otherwise energy-dependent intensities such as Mössbauer spectra are created. If run as standalone the results of a calculation are saved to a binary file, otherwise the results are internally forwarded and no binary output file is created.

8.1 Standard input file

The SIF of the kmix executable has 12 lines of data. The content of these lines is discussed in order of appearance.

1. name of input file – one value :

This value gives either the name or a list of names of binary input files created by standalone execution of kfor, kgin, or kref. If kmix is run via kctl, kgin, kmco, kfmf, kgmf, or krmf the file with tag value 'def' is replaced with the internal calculation. This name is a placeholder.

The multi-line syntax for specification of a list of filenames for kmix standalone execution is

```
(1) list of input files :: <!  
    <file1> <tag1>  
    <file2> <tag2>  
    ...  
    !
```

Alternately the inline syntax may be used

```
(1) list of input files :: <file1> <tag1> // <file2> <tag2>
```

The syntax for specification of a list of filenames for other invocations of kmix is

```
(1) list of input files :: <!  
    <file1> <tag1>  
    placeholder def  
    <file2> <tag2>  
    ...  
    !
```

The specifier <tag?> is optional. Possible values are 'def', 'scan', and 'tdi'. The file with specifier 'def' provides energy, time, thickness, and angle scales. If omitted the first file is used. Specifier tags can only be used once. If an absorber scan is requested in field 4 the 'scan' tag specifies the absorber to be scanned. If omitted the last file is used. If time-domain-interferometry is requested in field 4, the 'tdi' tag specifies the absorber after the non-resonant scatterer. The 'scan' and 'tdi' tags are mutually exclusive.

Each file represents a scattering matrix, and therefore the order in which file names are specified reflects the spatial order of the scatterers. The first file describes the scatterer that is encountered first by the x-rays.

2. name of output file – one value :

This value gives the name of a binary output file. If kmix is not run standalone the value is ignored.

3. run mode – five values :

The first value is either 'energy' or 'time' and determines if energy or time spectra should be created, respectively. The second value is optional. If not specified scattered intensities are calculated. Otherwise recognized values are: 'p', calculate the phase of the scattered x-ray field; 'r', calculate

the Faraday rotation of the polarization of the scattered field (only meaningful if polarizations are specified as linear in lines 5-8); 'filter', calculate the energy spectrum of the scattered radiation corresponding to a particular time window. If the second value is 'filter' the third and fourth values must be specified to provide the beginning and end of the time window in units of ns. An optional fifth value specifies the region of edge softening in units of ns.

4. scan/tdi mode – three values :

The first value can be 1, 2, 3, or 4. Other values disable scan and tdi modes.

Values 1 to 3 specify scan modes, and the run mode must be 'energy filter'. Mode 1 positions the time filter after all scatterers. Mode 2 positions the time filter with the scanned element and assumes that the scanned element is spatially positioned first in the chain of scatterers. Mode 3 positions the scanned element in a crossed-polarizer setup and first in the chain of scatterers. Time filtering is applied after all scatterers in mode 3. The second field is optional and gives the energy range for scanning in units of Γ . The third field is optional and gives the requested number of points for the scan.

A first value of 4 specifies tdi mode, and the run mode must be 'time'. The second value is optional and a non-zero value specifies a scale factor of the energy range for tdi averaging. The third value is optional and a non-zero value selects adaptive energy step size for tdi averaging. The default is to use an automatic scale factor and adaptive averaging.

5. time structure of synchrotron radiation (SR) – three or more values :

The first value gives the time separation between SR bursts in units of ns. The second value gives the number of bunches in each SR burst. The third value gives the time separation of the bunches in each SR burst in units of ns. The following values specify the intensities of each bunch in a SR burst. Intensities will be normalized automatically. All values are optional. If none is specified the SR time structure consists of a single bunch. If the second and third values are omitted a SR burst consists of a single bunch, and the SR time structure consists of an infinite series of single bunches. If intensities are omitted all bunches are of equal intensity.

6. this line is not used but must be present.

7. polarization degree – one value :

This value gives the polarization degree of the incident x-rays in units of %.

8. mixing angle – one value :

This value gives an angle that defines the polarization type of the incident x-rays in units of degrees. An angle of zero selects linear polarization. Angles of 90° and -90° select left and right circular polarization. Angle values in-between select elliptical polarization.

9. canting angle – one value :

This value gives an angle that rotates the polarization reference of the incident x-rays in units of degrees. The polarization reference is defined by the executable that creates data feeding into kmix as the σ polarization vector, see Figure 1 on page 41 for kfor, Figure 2 on page 41 for kgin, and Figure 4 on page 42 for kref.

10. filter efficiency – one value :

This value gives the filter or selection efficiency of a detecting device that analyses the polarization of scattered x-rays in units of %.

11. mixing angle – one value :

This value gives an angle that selects the polarization type of the scattered x-rays in units of degrees, see also line 8.

12. canting angle – one value :

This value gives an angle that rotates the polarization reference of the scattered x-rays in units of degrees, see also line 9.

Lines 13 to 17 are only needed for tdi mode, i.e., a value of 4 in line 4. The time-dependent intensity for tdi mode is given by

$$I(t) = F_{DW} I_{BA}(0, t) + (1 - F_{DW}) (1 - F) I_A(t) + (1 - F_{DW}) F \int \tilde{Q}(\omega) I_{BA}(\omega, t) d\omega \quad , \quad (1)$$

where absorbers before and after the quasi-elastic scatterer are indexed as A and B, respectively. The Debye-Waller factor F_{DW} , the non-ergodicity factor F , and the shape of the distribution function $\tilde{Q}(\omega)$ are specified in lines 13 to 17.

13. distribution function– one value :

Accepted choices are 'Kohlrausch', 'exponential', and 'power'.

For the 'Kohlrausch' option, we have $F_{DW} = 0$ and $\tilde{Q}(\omega) = \frac{1}{\pi} \int_0^\infty \cos[\omega t] \exp[-(t/\tau)^\beta] dt$ with two parameters β and τ .

For the 'exponential' option, we have

$$\tilde{Q}(\omega) = A \exp \left[-\ln 2 \left| \frac{2\omega}{w} \right|^\beta \right] \quad \text{with} \quad 1/A = 4 w \alpha (\ln 2)^\alpha \Gamma(\alpha) \quad ,$$

where $\alpha = 1/\beta$ and parameters full-width-half-maximum w and shape β .

For the 'power' option, we have

$$\tilde{Q}(\omega) = A \left\{ 1 + \left| \frac{2\omega}{w} \right|^\beta \right\}^{-1} \quad \text{with} \quad 1/A = w \frac{\alpha}{2} \{ \psi(\alpha + 1/2) - \psi(\alpha) \} \quad ,$$

where $\alpha = 1/(2\beta)$ and parameters full-width-half-maximum w and shape β . The psi-function is related to the gamma-function and its derivative via $\psi(x) = \Gamma'(x)/\Gamma(x)$.

14. Debye-Waller factor– one value :

The value of F_{DW} in Eq. 1, ignored for 'Kohlrausch' distribution.

15. non-ergodicity factor– one value :

The value of F in Eq. 1.

16. width parameter– one value :

The value of τ for 'Kohlrausch' or else value of w as explained above.

17. shape parameter– one value :

The value of β as explained above.

8.2 Output files

The kmix executable produces one binary file and one text file. Existing files are overwritten without notice. The name of the binary file is taken from line 2 of the SIF. The unformatted write statements can be gleaned from source code file savmix.F located in directory CONUSS-3.0.1/src/mix/. The text file is named kmix_ptl.txt and contains a protocol with input parameters, intermediate and final results.

9 Intensities: kfit

The kfit executable is run in two ways: standalone or via another executable such as kctl, kgin, kmco, kfmf, kgmf, krmf. The kfit program calculates averages over output data arrays of executable kmix and optionally compares the resulting calculation to a measured data set. Parameters in lines 10-15 may be declared fit parameters.

9.1 Standard input file

The SIF of the kfit executable has 17 lines of data. The content of these lines is discussed in order of appearance.

1. name of input file – two values :

The first value gives the name of a binary input file created by standalone execution of kmix. The file name value is ignored if kfit is run via kctl, kgin, kmco, kfmf, kgmf, krmf. The second value specifies the angle/thickness grid to be used. It is optional and if specified has to be 1 (default), 2, or 3. The meaning depends on the executable feeding data into kfit: value 1, thickness (kfor, kfmf), angle of incidence (kgin, kgmf), angular deviation from kinematic angle of incidence (kref, krmf); value 2, effective thickness (kfor, kfmf), angle of incidence (kgin, kgmf), angular deviation from kinematic Bragg angle (kref, krmf); value 3, undefined (kfor, kfmf), angle of incidence (kgin, kgmf), angular deviation from kinematic exit angle (kref, krmf).

2. name of measured data file – many values :

The first value gives either the name or a list of names of measured data files.⁴ The values following the filename 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 v15. The letter refers to the physical meaning according to Table 4 on page 28. Data in time or velocity intervals $[t_1, t_2]$ or $[v_1, v_2]$ may be excluded from fitting by adding the line '#Mask $t_1 t_2$ ' or '#Mask $v_1 v_2$ ' to the file, respectively. Several lines of this type may be added to exclude specific regions of data.

The dual-fit option of kctl requires two data files, and two syntax rules are supported for specification of a list of filenames

```
(2) list of input files :: <!  
    <file1> <..> <..>  
    <file2> <..> <..>  
    !
```

and the in-line syntax

```
(2) list of input files :: <file1> <..> <..> // <file2> <..> <..>
```

Both data files must have the same time/velocity axis, and data masking is only recognized in the first file.

⁴The input file is a plain text file containing no more than 500 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
C	counts		required
T	time	ns	optional
V	velocity	mm/s	optional
c	counts uncertainty		optional
t	time uncertainty	ns	optional
v	velocity uncertainty	mm/s	optional

Table 4: Column assignments for kfit experimental data input file. Either a time or a velocity column must be specified. If 'counts uncertainty' is not assigned to a column the uncertainties of the input data counts are calculated as square root of the counts. The units refer to what is expected from input data.

3. name of output file prefix and options – two values :

The first value gives the prefix for output file names.⁵ The second value is optional and influences the formatting of the output files according to Table 5 on page 28.

option	output format	comment
3c	ASCII, two or three column	default
csv	ASCII, comma speparated fields	for Excel
grace	ASCII, 2 or 3 column, grace header	for xmgrace
3D	ASCII, four column, compressed	for VisIt
3DL <range>	ASCII, four column, compressed	for VisIt, log-scale
image <color> <contour>	binary, ppm, P6	linear scale
image1 <range> <color> <contour>	binary, ppm, P6	log scale

Table 5: Formats for kfit data output files. For log-scale presentation, <range> specifies the ratio of smallest to largest displayed value. For image options, <color> specifes the color scale to use: 0, gray; 1, x-ray; 2 Imola; 3 Bamako; 4 Oslo; 5 Hawaii; 6 Acton; 7 Batlow; 8 Inferno; 9 Kindlmann; 10 Rainbow. <contour> specifes the width of contour lines and is optional. Color scales 2 to 7 are adapted from Fabio Crameri.

4. start of angle or thickness range – one value :

This value is optional and gives the start of an angle or thickness range depending on the executable feeding data into kfit: depending on the option in line 1 kfor, kfmf provide either a thickness scale in units of μm or an effective thickness scale; kgin, kgmf provide an angle scale in units of mrad; kref, krmf provide an angle scale in units of μrad . If omitted the smallest possible value is selected.

5. end of angle or thickness range – one value :

This value is optional and gives the end of an angle or thickness range depending on the executable feeding data into kfit. If omitted the largest possible value is selected.

6. start of time or energy range – one value :

This value is optional and gives the start of a time or energy range in units of ns or Γ , respectively, depending on the value of line 3 of the SIF of kmix. If omitted the smallest possible value is selected.

⁵Names 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 in the working directory, /Users/data/Prefix creates files in the directory /Users/data. Directories must exist and be writeable.

7. end of time or energy range – one value :
This value is optional and gives the end of a time or energy range in units of ns or Γ , respectively, depending on the value of line 3 of the SIF of kmix. If omitted the largest possible value is selected.
8. thickness or angle distribution function – three values :
The first value specifies the type of distribution function: *exponential* or *power* for built-in distributions; file name for an externally defined distribution shape. For built-in functions, the second and third values are optional and give the shape coefficient and the left/right asymmetry, respectively. The file name specification has to be an alone value. The file must contain space, comma, or tab separated columns of numbers. The first column specifies thickness or angle, the second column specifies probability density.
9. time or energy distribution function – three values :
The first value specifies the type of distribution function: *exponential* or *power* for built-in distributions; file name for an externally defined distribution shape. For built-in functions, the second and third values are optional and give the shape coefficient and the left/right asymmetry, respectively. The file name specification has to be an alone value. The file must contain space, comma, or tab separated columns of numbers. The first column specifies time or energy, the second column specifies probability density.
10. constant background – one value :
This value specifies a constant that is added to scaled calculated values in case of fitting experimental data.
11. scaling factor – one value :
This value specifies a scaling factor of calculated values in case of fitting experimental data.
12. scale shift – one value :
This value specifies a shift of the time/energy scale of experimental data.
13. thickness/angle distribution width or stretch factor – one value :
This value specifies the FWHM of a built-in thickness/angle distribution defined in line 8. For externally defined distributions, the value gives a stretch factor that is applied to the thickness or angle axis.
14. time/energy distribution width or stretch factor – one value :
This value specifies the FWHM of the time/energy distribution defined in line 9. For externally defined distributions, the value gives a stretch factor that is applied to the time or energy axis.
15. thickness/angle position – one value :
This value specifies the center of thickness/angle distribution in case of fitting experimental data.
16. create rocking curves – one value :
This value is relevant only without experimental data treatment. If 'yes' the calculation produces data as function of angle (rocking curve) or thickness at a particular energy or time specified in line 15.
17. rocking curve position – one value :
This value is relevant only without experimental data treatment. It specifies the energy or time at which rocking or thickness curves are produced.

9.2 Output files

The kfit executable produces several text files. Existing files are overwritten without notice. The protocol file with input parameters, intermediate and final results is named 'kfit_ptl.txt'. All other file names are generated with the prefix from line 3 of the SIF and are given in Table 6.

content	column 1	column 2,3	x,y units	file names
data	time	counts	ns	<pfx>_exp.<ext> <pfx>_exp_ref.<ext>
	velocity	counts	mm/s	<pfx>_exp.<ext> <pfx>_exp_ref.<ext>
theory, fitted	time	scaled theory	ns	<pfx>_fit.<ext> <pfx>_fit_ref.<ext>
	velocity	scaled theory	mm/s	<pfx>_fit.<ext> <pfx>_fit_ref.<ext>
fit residuals	time	counts	ns	<pfx>_rsd.<ext> <pfx>_rsd_ref.<ext>
	velocity	counts	mm/s	<pfx>_rsd.<ext> <pfx>_rsd_ref.<ext>
theory from kfor	time	intensity	ns, Γ /ns	<pfx>/thickness<n1>.<ext>
	energy	rel. intensity	Γ , %	<pfx>/thickness<n1>.<ext>
theory from kref,kgin	time	intensity	ns, Γ /ns	<pfx>/angle<n1>.<ext>
	energy	rel. intensity	Γ , %	<pfx>/angle<n1>.<ext>
time integrated theory	thickness	integrated intens.	μm , Γ	<pfx>/IntegratedNRS.<ext>
	angle	integrated intens.	μrad , mrad, Γ	<pfx>/IntegratedNRS.<ext>
thickness curves	thickness	intensity	μm , Γ /ns,%	<pfx>_<n2>.<ext>
rocking curves	angle	intensity	μrad , mrad, Γ /ns,%	<pfx>_<n2>.<ext>

Table 6: Output files generated by kfit. The prefix <pfx> is specified in line 3 of the standard input file. The extension <ext> is csv for comma separated output format and dat otherwise. The '_ref' files are only created in dual-fit mode. The variable <n1> refers to thickness or angle index of the preceeding calculation. The variable <n2> refers to either time or energy index derived from line 17 of the SIF.

10 Fitting data: kctl

The kctl executable is designed to optimize the match between measured data and calculation by intelligent variation of input parameters. The measure for agreement is the normalized χ^2 value. Parameter variation is performed iteratively using Newton, Levenberg-Marquardt, or gradient methods. kctl uses one of kfor, kref executables and executables kmix, kfit.

10.1 Standard input file

The SIF of the kctl executable has 7 lines of data. The content of these lines is discussed in order of appearance.

1. name of input file – one value :
This value gives the name of a SIF for the kfor or kgin or kref executables.
2. name of input file – one value :
This value gives the name of the SIF for the kmix executable.
3. name of input file – one value :
This value gives the name of the SIF for the kfit executable.
4. name of input file – one value :
This value is optional and gives the name of a binary reference file. If specified a binary output file created by the kfor executables must be present. The kctl executable then enters dual-fit mode for which two measured spectra are fitted simultaneously. These spectra must be defined in line 2 of the SIF of executable kfit. The first spectrum represents the sample, and the second spectrum represents sample plus a reference absorber. The reference absorber is fully described by the binary file which is obtained by a calculation using executable kfor. The energy ranges as specified in line 5 of the SIF of kfor must be identical.
5. iteration steps and Levenberg-Marquardt parameter – two values :
The first value gives the maximum number of iteration steps. The second value is optional. Specify *Newton*, *gradient*, or 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.
6. iteration stop criterium – one value :
This value gives the required relative change of χ^2 value for an iteration step to stop the iteration.
7. name of input file – one value :
This value gives the name of the prior correlation file. If correlated priors were specified for any fit parameters this file contains the corresponding correlation matrix. The number of columns and rows must correspond to the number and order of priors.

10.2 Output files

The kctl executable produces several text files. Existing files are overwritten without notice. The use of kfor, kref, kgin, and kfit generates files as described in sections 5.3, 6.3, 7.2, and 9.2, respectively. In addition, the following files are created.

<prefix>_kctl.csv list of fit parameters and χ^2 values for all iteration steps;

<prefix>_kctl_ptl.txt protocol file with input parameters and final results, such as best-fit parameters, errors, and correlations;

<prefix>_BestFit/in_<exe> SIF of <exe> created with best-fit parameters;

<prefix>_BestFit/<MIF>.mif MIF created with best-fit parameters;

<prefix>_ErrEllipses/<prefix>_p<p1>v<p2>_eel.dat error ellipses representing correlations between fit parameters <p1> and <p2>;

<prefix>_ErrEllipses/<prefix>_p<p1>v<p2>_nel.dat normalized error ellipses representing correlations between fit parameters <p1> and <p2>.

Here <prefix> represents the file prefix as specified in line 3 of the SIF of kfit, and <MIF> is the name of the material data input file specified in the SIF of kfor, kref, or kgin after removal of the extension.

11 Exploring parameter space: kmco

The kmco executable is designed to explore landscape of the normalized χ^2 value in parameter space. Parameters are sampled in pre-defined intervals by random choice. The collection of sampling parameters and intervals creates an initial search box. After a certain number of samplings, the smallest χ^2 values are identified and used to create a set of smaller sampling boxes, the next level. The procedure is then repeated. kmco uses one of kfor, kgin, kref executables and executables kmix, kfit.

11.1 Standard input file

The SIF of the kctl executable has 9 lines of data. The content of these lines is discussed in order of appearance.

1. name of input file – one value :
This value gives the name of a SIF for the kfor or kgin or kref executables.
2. name of input file – one value :
This value gives the name of the SIF for the kmix executable.
3. name of input file – one value :
This value gives the name of the SIF for the kfit executable.
4. levels – one value :
This value is optional and gives the number of levels.
5. samples – one value :
This value gives the number of parameter samples for each level.
6. size reduction – one value :
This value gives the reduction factor (each dimension) of the sampling box for each level.
7. selection tolerance – one value :
This value gives the tolerance in the selection of the smallest χ^2 values for a level. With value $q > 1$, parameter vectors leading to χ^2 values in the interval $[\chi^2, q \chi^2]$ are used to create a set of smaller sampling boxes for the next level.
8. random seed – one value :
This value is optional and gives the seed for the used random number generator. If provided the set of created random numbers is always the same.
9. name of input file – one value :
This value is optional and gives the name of a binary reference file. If specified a binary output file created by the kfor executables must be present. The kmco executable then enters dual-sampling mode for which two measured spectra are sampled simultaneously. These spectra must be defined in line 2 of the SIF of executable kfit. The first spectrum represents the sample, and the second spectrum represents sample plus a reference absorber. The reference absorber is fully described by the binary file which is obtained by a calculation using executable kfor. The energy ranges as specified in line 5 of the SIF of kfor must be identical.

11.2 Output files

The kmco executable produces several text files. Existing files are overwritten without notice. The use of kfor, kref, and kgin generates files as described in sections 5.3, 6.3, and 7.2, respectively. In addition, the following files are created.

<prefix>_kmco.csv list of parameters and χ^2 values for all samplings;
 <prefix>_kmco_ptl.txt protocol file with input parameters and final results, such as best-sampling parameters;
 <prefix>_chi2/chi2_hist.dat histogram of χ^2 values for all samplings;
 <prefix>_chi2/chi2_seq.dat sequence of χ^2 values for all samplings;
 <prefix>_chi2/chi2_ave.dat average and rms of χ^2 values for each level;
 <prefix>_parms/parm_hist.dat histogram of normalized parameter distances for all samplings;
 <prefix>_parms/parm<n>.dat χ^2 vs. sampling parameter <n> for all samplings;
 <prefix>_BestSampling/in_<exe> SIF of <exe> created with best-sampling parameters;
 <prefix>_BestSampling/<MIF>.mif MIF(s) created with best-sampling parameters;
 <prefix>_BestSampling/<data_file> file with experimental data as specified in the SIF of kfit;
 <MIF>_<exe>.log.txt log-file of <exe> created with best-sampling parameters;
 <MIF>_<exe>_site<N>_<type>_dist.dat probability distribution of hyperfine parameter <type> of site <N> created with best-sampling parameters;
 <MIF>_kfor_let.dat local effective thickness versus energy (only for forward scattering geometries).

Here <prefix> represents the file prefix as specified in line 3 of the SIF of kfit, and <MIF> is the name of the material data input file(s) as specified in the SIF of kfor, kref, or kgin after removal of the extension.

12 Quicklooks: kfmf, kgmf, krmf

The kfmf, kgmf, and krmf executables offer a convenient way to run sequences (kfor,kmix,kfit), (kgin,kmix,kfit), and (kref,kmix,kfit), respectively. Expected input file names default to in_kfor, in_kgin, in_kref, in_kmix, and in_kfit. No binary output files are created. Text output files are generated as described in sections about the individual executables.

13 Deconvolution: kdec

The kdec executable deconvolutes measured data with a specified source profile. In energy scanning type experiments at synchrotron radiation sources, the source profile is often not simply a Lorentzian. Data measured with multiline sources and complicated lineshapes can be quickly deconvoluted to obtain the absorber response. The kdec code implements singular-value-decomposition of the source profile mapped onto a Toeplitz matrix. Eigenvalue cut-off enables a convenient way to control the deconvolution process of the usually noisy input data.

13.1 Standard input file

The SIF of the kfan executable has 6 lines of data. The content of these lines is discussed in order of appearance.

1. file name of the source profile – three values :
The first value gives the file name.⁶ The values following the filename 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. The letter here has to be either X for x-values or Y for y-values. Specifications X and Y are required.
2. file name of the data to be deconvoluted – three to five values :
The first value gives the file name.⁷ The values following the filename 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. The letter refers to X for x-values, Y for y-values, x for x-value uncertainties, and y for y-value uncertainties. Specifications X, Y, and y are required. Specification x is optional.
3. name of output file prefix and options – two values :
The first value gives the prefix for output file names.⁸ The second value is optional and influences the formatting of the output files according to Table 7 on page 36.

option	output format	comment
3c	space separated two/three column ASCII	default
csv	comma separated two/three column ASCII	for Excel
grace	space separated two/three column ASCII	for xmgrace

Table 7: Formats for kdec data output files.

4. Eigenvalue cut-off – one value :
This value gives the smallest acceptable eigenvalue. Smaller eigenvalues are ignored in the construction of the pseudo inverse of the source-profile Toeplitz matrix.
5. Averaging windows – one or two values :
If one value is specified it gives the window width for a running average of the deconvoluted data.
If two values are specified they give the smallest and largest window width for an adaptive running average of the deconvoluted data.

⁶The input file is a plain text file containing no more than 500 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).

⁷see above

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 in the working directory, /Users/data/Prefix creates files in the directory /Users/data. Directories must exist and be writeable.

6. Lorentzian width— one value :

This value gives the FWHM of a Lorentzian broadening used to produce a pseudo Mössbauer absorption spectrum from the deconvoluted data.

13.2 Output files

The kdec executable produces several text files. Existing files are overwritten without notice. The protocol file contains input parameters, intermediate and final results. All file names are generated with the prefix from line 3 of the SIF and are given in Table 8 on page 37.

content	column 1	column 2,3	x,y units	file name
protocol				<prefix>_kdec.ptl.txt
deconvoluted data	velocity	absorption	mm/s, %	<prefix>_dec.<ext>
deconvoluted, smoothed data	velocity	absorption	mm/s, %	<prefix>_dds.<ext>
deconvoluted source profile	velocity	intensity	mm/s	<prefix>_dsc.<ext>
pseuso Mössbauer spectrum	velocity	transmission	mm/s, %	<prefix>_mbs.<ext>
eigenvalues	index	normalized eigenvalue		<prefix>_eig.<ext>

Table 8: Output files generated by kdec. The prefix <prefix> is specified in line 3 of the standard input file. The extension <ext> is csv for comma separated format and dat otherwise.

14 Pure nuclear reflections: kpnr

The kpnr executable provides a list of electronically allowed and electronically forbidden but nuclear allowed Bragg/Laue reflections.

14.1 Standard input file

The SIF of the kpnr executable has 5 lines of data. The content of these lines is discussed in order of appearance.

1. project name – one value :
This value gives a name to the calculation that is performed.
2. name of MIF – one value :
This value gives the name of a material data input file.⁹ The MIF is a plain text file containing no more than 200 characters per line (excess characters will be ignored). The MIF format is identical to the MIF required for executable kref as discussed in section 6.2.
3. start of calculation range – three values :
These values give the Miller indices of the highest negative order of reflection. If left empty the maximum range is assumed.
4. end of calculation range – three values :
These values give the Miller indices of the highest positive order of reflection. If left empty the maximum range is assumed.
5. reflectivity level cut-off – one value :
This value gives the reflectivity in % below which reflections are considered electronically forbidden.
6. name of output file prefix and options – two values :
The first value gives the prefix for output file names.¹⁰ The second value is optional and influences the formatting of the output files according to Table 7 on page 36.

14.2 Output files

The kpnr executable produces several text files. Existing files are overwritten without notice. The following files are created.

<prefix>_kpnr_ptl.txt protocol file with input parameters and list of reflections;

<prefix>_sigma_ref.dat peak reflectivity of reflections assuming incident sigma polarization;

<prefix>_pi_ref.dat peak reflectivity of reflections assuming incident pi polarization;

<prefix>_sigma_int.dat energy-integrated reflectivity of reflections assuming incident sigma polarization;

<prefix>_pi_int.dat energy-integrated reflectivity of reflections assuming incident pi polarization.

Here <prefix> represents the file prefix as specified in line 5 of the SIF of kpnr.

⁹Names may be defined either relative to the working directory or absolute: file.mif must exist in the working directory, ../file.mif must exist in the parent directory, myDir/file.mif must exist in the directory myDir in the working directory, /Users/data/file.mif must exist in the directory /Users/data .

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 in the working directory, /Users/data/Prefix creates files in the directory /Users/data. Directories must exist and be writeable.

15 Electric field gradient axes: kvzz

The kvzz executable calculates the directions of the main axes of the electric field gradient from local symmetry of point charges around the nuclear resonant isotope.

15.1 Standard input file

The SIF of the kvzz executable has 4 lines of data. The content of these lines is discussed in order of appearance.

1. project name – one value :
This value gives a name to the calculation that is performed.
2. name of MIF – one value :
This value gives the name of a material data input file.¹¹ The MIF is a plain text file containing no more than 200 characters per line (excess characters will be ignored). The MIF format is identical to the MIF required for executable kref as discussed in section 6.2.
3. radius for efg calculation – one value :
This value gives the radius up to which the crystal lattice is expanded around each nuclear resonant isotope in units of Å. Atoms are assumed to be point charges with values specified in the MIF.
4. largest allowed distance defining nearest neighbors – one value :
This value gives the distance relative to each nuclear resonant isotope up to which surrounding atoms are considered nearest neighbors in units of Å.
5. name of output file prefix and options – two values :
The first value gives the prefix for output file names.¹² The second value is optional and influences the formatting of the output files according to Table 7 on page 36.

15.2 Output files

The kvzz executable produces several text files. Existing files are overwritten without notice. The following files are created.

<prefix>_kvzz_ptl.txt protocol file with input parameters and list of reflections;

<prefix>_nn.mld atomic positions of nearest neighbors in MOLDAN format which can be visualized using the VESTA software;

<prefix>_uc.mld extended unit cell atomic positions in MOLDAN format which can be visualized using the VESTA software;

<prefix>_site-<n>-<type>.dat distances between resonant atoms on site <n> and neighbors of type <type>.

Here <prefix> represents the file prefix as specified in line 5 of the SIF of kvzz.

¹¹Names may be defined either relative to the working directory or absolute: file.mif must exist in the working directory, ../file.mif must exist in the parent directory, myDir/file.mif must exist in the directory myDir in the working directory, /Users/data/file.mif must exist in the directory /Users/data .

¹²Names 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 in the working directory, /Users/data/Prefix creates files in the directory /Users/data. Directories must exist and be writeable.

16 Command line options

All executables recognize the following command line options.

- geometry=<X×Y+U+W>** specify the window geometry for run-time graphics, 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.
- nographics** disable run-time graphics.
- quiet** suppress most printout.
- version** display the version number and exit.
- vtool** list supported visualization programs.
- vtool=<exe>** use the executable <exe> as data visualization program.

The kmco executable supports acceleration of calculations by the command line option

- cores=<val>** use <val> physical cores for computations.

17 Scattering geometry and angle definitions

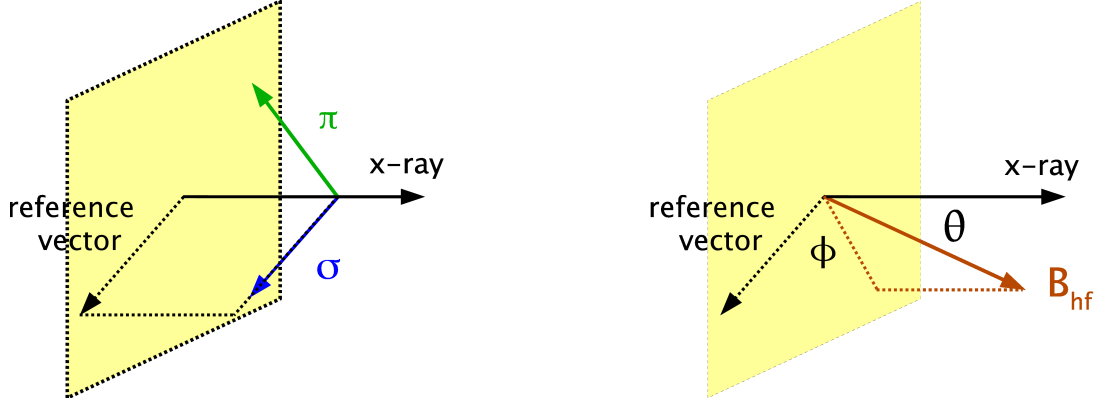


Figure 1: Executable kfor: directions of linear polarization vectors (left panel) and magnetic hyperfine field (right panel). The direction of the x-ray is specified in line 4 of the SIF of kfor and assumed to be perpendicular to and pointing into the surface. The reference vector is the projection of base vector (100) onto the surface and defines σ polarization. The polar and azimuthal angles θ and ϕ are specified for each MB-site.

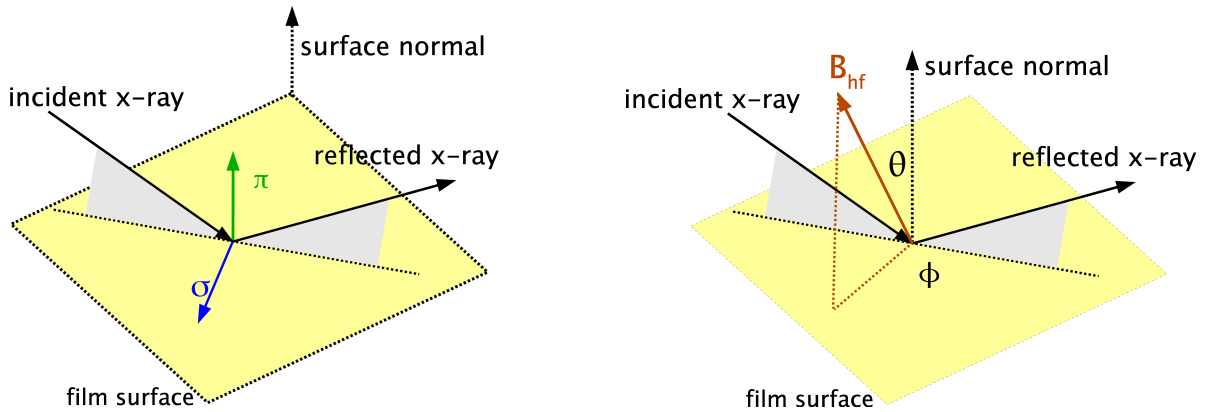


Figure 2: Executable kgin: directions of linear polarization vectors (left panel) and magnetic hyperfine field (right panel). The direction of the surface normal is given by (001). The projections of incident and reflected x-ray onto the surface are given by (100). σ polarization is defined to be perpendicular to the scattering plane and common to incident and reflected x-rays. The very small scattering angle permits use of a common π polarization in the direction of the surface normal. The polar and azimuthal angles θ and ϕ are specified for each MB-site.

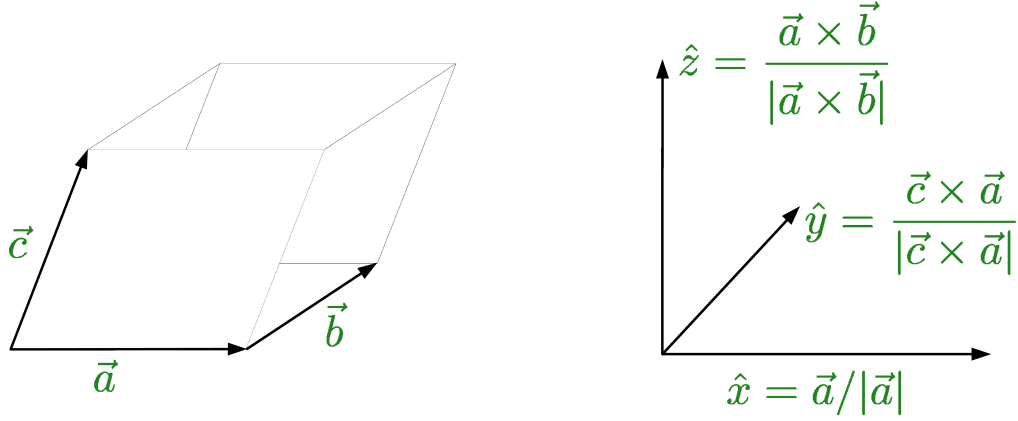


Figure 3: Executable kref: definition of the cartesian reference frame. The left panel shows the unit cell as defined in lines 5 and 6 of the MIF. The right panel illustrates the mathematical relationship to the orthonormal triad $(\hat{x}, \hat{y}, \hat{z})$ that forms the base for the cartesian reference frame.

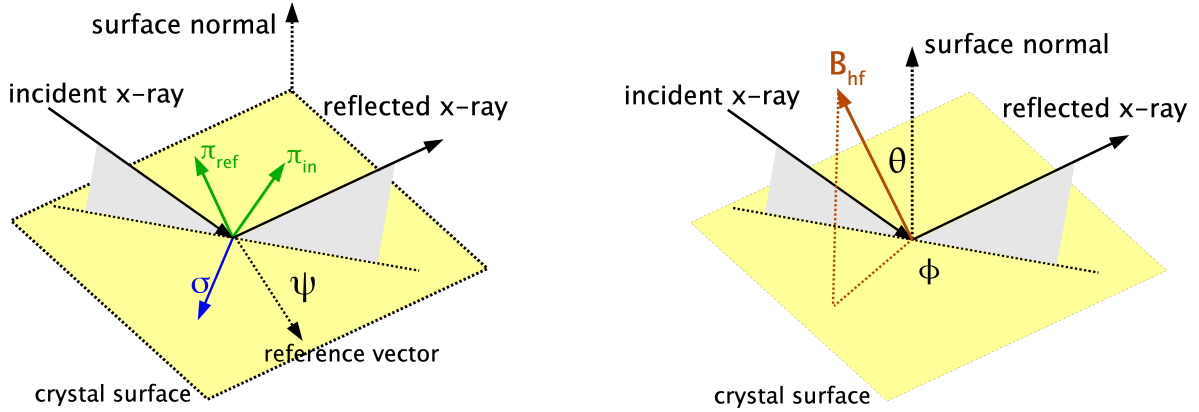


Figure 4: Executable kref: directions of linear polarization vectors (left panel) and magnetic hyperfine field (right panel). The direction of the surface normal is specified in line 6 of the SIF. The reference vector is given by the cross product of netplane and surface normals. In case of a symmetric Bragg reflection, the reference vector is the projection of base vector (100) onto the surface. The angle ψ between surface projection of the incident x-ray and reference vector is specified in line 7 of the SIF. σ polarization is defined to be perpendicular to the scattering plane and common to incident and reflected x-rays. The polar and azimuthal angles θ and ϕ are specified for each MB-site.

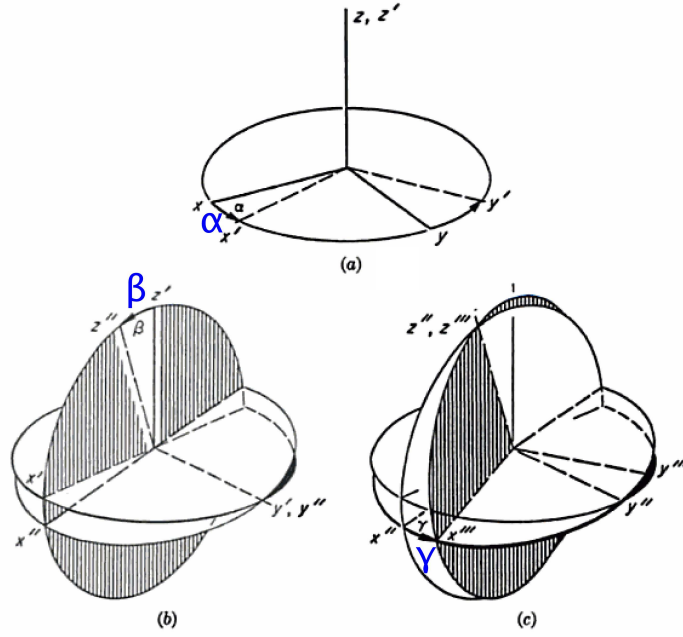


Figure 5: Definition of the Euler angles α , β , γ that rotate the electric field gradient tensor into the cartesian reference frame. The Euler angles are specified for each MB-site. Observe the order of rotations, zyz , after the convention of M.E. Rose, *Elementary Theory of Momentum* (Wiley, New York, 1957). For an axially symmetric efg, i.e., the asymmetry parameter vanishes, the angle γ is obsolete.