DEXA User Manual

October 2, 2009

Contents

1 Introduction **3**

2 Basic Operation **4**
2.1 Compiling *DEXA* 4
2.1.1 Compiling on Linux type systems 4
2.2 Preparing an input script 5
2.3 Starting *DEXA* 6

3 Advanced Features **8**
3.1 Fitting Data from Multiple Edges 8
3.2 Linking Parameters 9
3.3 Using Sub-structures 9
3.4 A Complete Example 11

4 Command Reference **14**
4.1 Global Commands 14
4.1.1 `create` * 14
4.1.2 `magnetistion` * 14
4.1.3 `polarisation` * 15
4.1.4 `numavesteps` 15
4.1.5 `startfit` 15
4.1.6 `saveparams` 16
4.2 Spectrum Commands 16
4.2.1 `spectrum` * 16
4.2.2 `feffinp` * 17
4.2.3 `pathsdat` * 17
4.2.4 `addpath` * 17
4.2.5 `updatepath` 18
4.2.6 `setparameter` 18
4.2.7 `fitparameter` 19
4.2.8 `preforientation` 20

4.2.9 `savespectrum` 20
4.2.10 `saveexperiment` 21
4.2.11 `symmetry` 21
4.2.12 `filter` 21

Chapter 1

Introduction

Chapter 2

Basic Operation

2.1 Compiling *DEXA*

2.1.1 Compiling on Linux type systems

To compile *DEXA* from the raw source files, you must first make sure that you have:

1. A working C++ compiler. By default, *DEXA* will use GNU `g++`. If you have another compiler, modify the first line of the `Makefile` so that `CC = <your compiler>`.
2. The GNU Scientific Library (GSL). Both the libraries and the include files must be installed to a location where the compiler can find them.
3. The Minuit library, which is available from http://seal.web.cern.ch/seal/snapshot/work-packages/mathlibs/minuit/index.html. The latest version tested with *DEXA* is Minuit 1.7.9.
4. GNU `make` in order to run the `Makefile` included with *DEXA* .

Assuming these are all present, just open a standard terminal window, and enter

```
user@host: cd <directory where you unpacked DEXA>
user@host: make
```

This will compile the standard version of *DEXA* and create the binary `dexa`. If you want to make these available system-wide (and assuming you have the privilages to do this), enter something like

```
user@host: cp dexa /usr/local/bin/
```

2.2 Preparing an input script

DEXA will analyse a given DiffXAS problem from instructions given in an input script. This script must do such things as tell *DEXA* how many spectra need fitting, where to find the experimental and *ab initio* data, what type of sample was used and how it was mounted in the beam, and finally, what DiffXAS parameters must be fitted.

Table 2.1 gives the full list of valid commands for use in the input, script along with their syntax. They are split into three sections: Global commands that affect the entire analysis environment and all spectra and structures contained within, spectrum commands affecting only a specific spectrum object, and structure commands for handling a given spectrum sub-structure. The distinction between these three classes is largely irrelevent for simple analyses, but becomes important for the more complex tasks described in Chapter 3. A full command reference is given in Chapter 4. All commands are **case-sensitive**.

The first task of the input script is to create a spectrum object that will be then be developed into a representation of the studied DiffXAS within *DEXA* . This is done quite simply with

```
create <spectrum name>
```

where *<spectrum name>* is replaced with any desired name for the spectrum, which may include any alpha-numeric character but no spaces.

Once such a blank spectrum has been created, it must be developed into the studied DiffXAS signal by loading both *ab initio* information from *FEFF* and the experimental data. Assuming all the necessary data files exist in the same directory from which *DEXA* was executed, this can be done with

```
create FeSpectrum
FeSpectrum.spectrum FeDiffXAS.dat 0 1 2
FeSpectrum.feffinp  feff.inp
FeSpectrum.pathsdat paths.dat
```

The experimental DiffXAS is assumed to be in a file called `FeDiffXAS.dat`, and is loaded with `.spectrum`. The three numeric arguments at the end of the command specify, respectively, how many lines to discard at the top of the file for its header, and then the column indicies for the x- and y-axis data, with the first column being number 1. `.feffinp` and `.pathsdat` then load the *FEFF* `feff.inp` and `paths.dat` files in order to firstly obtain the crystallographic co-ordinates and species of each of the atoms in the model structure, and then the information describing the important scattering paths.

2.3 Starting *DEXA*

Once an input script has been created, *DEXA* may be started from a standard terminal window by calling `dexa`, followed by the name of the script to be processed. For example, entering

user@host: `dexa script.dex`

will start *DEXA* , and process the contents of `script.dex`. As *DEXA* progresses through the script it will output useful information to the terminal, including any error or warning messages. If you wish to save this information, just use a standard terminal redirection, such as

user@host: `dexa script.dex > output.txt`

In this case, nothing will be output to the terminal, with the information instead being written to `output.txt`.

Global Commands		
General syntax: *<command> <arguments>*		
Command	Synopsis	Arguments
`create`	Creates a new spectrum object for analysis	*<spectrum>*
`magnetisation`	Sets the two sample magnetisation vectors	*<x1> <y1> <z1> <x2> <y2> <z2>*
`polarisation`	Sets the x-ray polarisation vector	*<x> <y> <z>*
`preforientation`	Adds a crystallite preferential orientation	*<θ_1> <θ_2> <θ_3> <degree>*
`numavesteps`	Sets the number of configurational averaging steps	*<steps>*
`startfit`	Starts the fit	*[<iterations>]*
`saveparams`	Saves the current state of all fit parameters	*<file>*
Spectrum Commands		
General syntax: *<spectrum>.<command> <arguments>*		
Command	Synopsis	Arguments
`addstructure`	Creates a new sub-structure for the given spectrum	*<structure>*
`feffinp`	Loads a *FEFF* feff.inp data file	*<file>*
`pathsdat`	Loads a *FEFF* paths.dat data file	*<file>*
`spectrum`	Loads the experimental DiffXAS data	*<file> <header> <xcol> <ycol>*
`filter`	Fourier filters the experimental data	*<kmin> <kmax> <dk> <rmin> <rmax> <dr>*
`saveexperiment`	Saves the experimental DiffXAS data	*<file>*
`savespectrum`	Saves the theory DiffXAS spectrum	*<file>*
Structure Commands		
General syntax: *<spectrum>[.<structure>].<command> <arguments>*		
Command	Synopsis	Arguments
`symmetry`	Defines the structural symmetry of the sample	*<symmetry>*
`addpath`	Adds a new *FEFF* scattering path	*<file> <header>*
`updatepath`	Modifies a given scattering path property	*<path> <property> <value>*
`setparameter`	Sets the value of a given DiffXAS parameter	*<parameter> <value> [<path>]*
`fitparameter`	States that a given parameter should be fitted	*<name> <parameter> [<path>]*

Table 2.1: The list of valid commands for use in a *DEXA* input script.

Chapter 3

Advanced Features

3.1 Fitting Data from Multiple Edges

By treating each spectrum as separate entities, it is quite straightforward to use *DEXA* to simultaneously analyse multiple spectra from different absorption edges in a sample. Simply use the `create` command to create one spectrum object for each of the data sets to be included in the analysis. For example, entering

```
create FeSpectrum
FeSpectrum.spectrum FeCoDiffXAS_FeK.dat 0 1 2
FeSpectrum.feffinp  feffFeK.inp
FeSpectrum.pathsdat pathsFeK.dat

create CoSpectrum
CoSpectrum.spectrum FeCoDiffXAS_CoK.dat 0 1 2
CoSpectrum.feffinp  feffCoK.inp
CoSpectrum.pathsdat pathsCoK.dat
```

would create two spectra for some supposed Fe and Co data, which are then managed individually using the appropriate spectrum name as a prefix. Later in the script, when `startfit` is called, *DEXA* will take all created spectrum objects and fit their parameters simultaneously.

The only other point to note is that when dealing with two or more spectra, it becomes important to appreciate the difference between global commands and those applying to spectra or structures (as listed in Table 2.1). Global commands, of which `create` is one, are *not* prefixed with a spectrum or structure name. This class of command covers aspects of the fit

that, for instance, relate to the mounting of the sample in the beam, or the basic properties of the sample, and so must apply to *all* spectrum objects. By contrast, Spectrum commands affect only a single spectrum, and so *must* be prefixed with the name of a previously created spectrum object. A further distinction comes into play when multiple structures are defined, as described in Section 3.3.

3.2 Linking Parameters

When dealing with more complex DiffXAS analyses, and especially when dealing with a simultaneous fit at multiple absorption edges, there may be some physical necessity to link parameters within a spectrum or across multiple spectra. This would be the case, for instance, when looking at strain in an A-B type bond from both the A and B element absorption edges. The strain from the point of view of one should be reciprocated by the other.

Linking parameters in this way is done with the `fitparameter` command. When adding a new fit parameter in the input script, the parameter must be assinged a *<name>*. There are no constrains on what name to choose, but each independent fit parameter must have a unique name. To link two or more parameters, simply assign them the same name. For example, taking the multiple edge script extract given in Section 3.1, and adding

```
FeSpectrum.fitparameter lg2 lambdaG2
CoSpectrum.fitparameter lg2 lambdaG2
```

would link the $\lambda^{\gamma,2}$ parameters of both `FeSpectrum` and `CoSpectrum` since both have been given the name `lg2`. Consequently, at the end of the fitting process, both spectra will have the same $\lambda^{\gamma,2}$. Note that, currently, it is only possible to do direct one-to-one linking such as this. It is not possible to enter some mathematical expression to define a more complex relation between fit parameters.

3.3 Using Sub-structures

The macroscopic description of crystal strain, through such things as Neumann's Principle and crystal tensors, works in the language of unit cells and lattices, and concerns the long range order of the sample structure. This can be transferred to DiffXAS, but it is often more meaningful to describe strain in terms of changes in specific bond lengths; to relax crystallographic arguments and deal explicitly with the sample's short range order.

This is especially true when attempting to fully exploit the chemical selectivity of DiffXAS in determining the strain surrounding different atomic environments in a multicomponent system.

Take, for instance, the magnetostriction of rare-earth iron alloys such as Fe_2Tb. These systems form in a cubic phase, and would macroscopically be described with two coefficients - one for strain along the 100 lattice vector, and one for strain along the 111. A similar approach is possible with *DEXA* by loading information pertaining to each bond with `addpath`, declaring the sample is cubic with the `symmetry` command, and then fitting a single $\lambda^{\gamma,2}$ and $\lambda^{\epsilon,2}$ to the entire structure.

However, locally, the spin-orbit coupling of the rare-earth atom is considerably larger than that of the Fe atom, such that one would expect a significant difference in the response of Fe-Fe and Fe-Tb bonds in an applied magnetic field. It would therefore be desirable to handle each independently; to assign separate coefficients to the sub-lattice comprising the Fe-Fe bonds, and that composed of Fe-Tb bonds.

Thus, whilst it is possible to work within the language of crystals - with the strain in specific bonds defined by some overall crystal strain and the crystallographic position of given atoms - *DEXA* also allows 'sub-structures' to be created to divide the macroscopic structure into a number of units that are expected to exhibit different short-range behaviour.

Each unit is created within a spectrum object using the `addstructure` command. For example, entering

```
FeSpectrum.addstructure FeFe
FeSpectrum.addstructure FeTb
```

would create two sub-structures; one called `FeFe`, and the other, `FeTb`. The 'Structure commands' listed in Table 2.1, can then work on each of the sub-structures by giving the spectrum name *and* structure name in the command prefix. For example, entering

```
FeSpectrum.FeFe.addpath feff0001.dat 10
FeSpectrum.FeTb.addpath feff0002.dat 10
FeSpectrum.FeFe.addpath feff0003.dat 10
FeSpectrum.FeTb.addpath feff0004.dat 10
```

would load `feff0001.dat` and `feff0003.dat` into the `FeFe` structure, and `feff0002.dat` and `feff0004.dat` into `FeTb`, where it is assumed that *FEFF*

produced an Fe-Fe type path at indicies 1 and 3 in its `paths.dat`, and an Fe-Tb type path at indicies 2 and 4. Subseqeuently, separate magnetostrictive strains may be applied to each sub-structure with

```
FeSpectrum.FeFe.fitparameter lg2FeFe lambdaG2
FeSpectrum.FeTb.fitparameter lg2FeTb lambdaG2
```

As with complete crystal structures, individual parameters may be linked by assigning the same parameter name to all parameters that should have the same value.

Note, however, that at present, sub-structures should only be used with single-scattering paths. *DEXA* does not yet decompose multiple-scattering paths into their individual components in order to assure that each leg is assigned to the correct sub-structure. Thus, by including multiple-scattering paths in one structure or other could cause one or more identical legs to be assigned different properties at different parts of the DiffXAS calculation.

3.4 A Complete Example

```
*-------------------------------------------------------
* Prepare the spectrum objects
*
create FeSpectrum
FeSpectrum.spectrum FeCoDiffXAS_FeK.dat 0 1 2
FeSpectrum.feffinp  feffFeK.inp
FeSpectrum.pathsdat pathsFeK.dat
FeSpectrum.filter 3 10 1.0 1.8 2.8 0.1

create CoSpectrum
CoSpectrum.spectrum FeCoDiffXAS_CoK.dat 0 1 2
CoSpectrum.feffinp  feffCoK.inp
CoSpectrum.pathsdat pathsCoK.dat
CoSpectrum.filter 3 10 1.0 1.8 2.8 0.1

*-------------------------------------------------------
* Define the sample environment
*
magnetisation 1 0 0 0 1 0
polarisation 1 0 0
preforientation 0 0 0 1.0
```

```
*-----------------------------------------------------
* Create the spectrum sub-structures and load the
* individual path information from FEFF
*
FeSpectrum.addstructure FeFe
FeSpectrum.addstructure FeCo
CoSpectrum.addstructure CoFe
CoSpectrum.addstructure CoCo

FeSpectrum.FeFe.addpath feff0001Fe.dat 10
FeSpectrum.FeCo.addpath feff0002Fe.dat 10
CoSpectrum.CoFe.addpath feff0001Co.dat 10
CoSpectrum.CoCo.addpath feff0002Co.dat 10

*-----------------------------------------------------
* Update the reference structure
*
* updatepath would be used here to insert the results
* from a conventional EXAFS fit, and so define the
* reference structure for the DiffXAS.

*-----------------------------------------------------
* Define the fit parameters and link the Fe-Co bond.
*
FeSpectrum.FeFe.fitparameter lg2FeFe lambdaG2
FeSpectrum.FeCo.fitparameter lg2FeCo lambdaG2
CoSpectrum.CoFe.fitparameter lg2FeCo lambdaG2
CoSpectrum.CoCo.fitparameter lg2CoCo lambdaG2

*-----------------------------------------------------
* Perform the fit and save the results
*
FeSpectrum.saveexperiment FeK_Filtered.dat
CoSpectrum.saveexperiment CoK_Filtered.dat

startfit

saveparams FitResults.dat
FeSpectrum.savespectrum FittedFeK.dat
CoSpectrum.savespectrum FittedCoK.dat
```

Chapter 4

Command Reference

4.1 Global Commands

The commands in this section act on the entire *DEXA* fitting environment rather than on one specific spectrum. Commands marked with a '*' are compulsory and must appear at least once in your *DEXA* input script. All commands are case-sensitive and so must be entered exactly as they are written here.

4.1.1 `create` *

Syntax:

```
create <spectrum>
```

Creates a new spectrum object called *<spectrum>*. This spectrum's properties, given in section 4.2, may then be modified using the '.' delimiter in the general format *<spectrum>.<command> <options>*.

It is necessary to create at least one spectrum object at the start of the *DEXA* input file in order to perform an analysis. If more than one spectrum is created, all will be fitted simultaneously. This makes it possible to link parameters that are common to more than one spectrum, such as when viewing an A-B bond from both the A and B absorption edges. See section **??** for more details on linking parameters.

4.1.2 `magnetistion` *

Syntax:

```
magnetisation <x1> <y1> <z1> <x2> <y2> <z2>
```

Sets the two DiffXAS sample magnetisation vectors. The first vector is defined by the first three arguments (*<x1>*, *<y1>*, and *<z1>*), and the second by the last three (*<x2>*, *<y2>*, and *<z2>*). Both must be unit vectors such that $(x^2 + y^2 + z^2)^{1/2} = 1$. For the standard DiffXAS configuration, where all coordinates are specified in the beamline frame with the beam propogating along the +z direction, the first vector would be $(1, 0, 0)$ and the second $(0, 1, 0)$. This command must be called at some point before `startfit`.

4.1.3 `polarisation` *

Syntax:

```
polarisation <x> <y> <z>
```

Sets the x-ray polarisation vector. It must be specified as a unit vector such that $(x^2 + y^2 + z^2)^{1/2} = 1$. For the standard DiffXAS configuration, where all coordinates are specified in the beamline frame with the beam propogating along the +z direction, the polarisation vector would typically be $(1, 0, 0)$. This command must be called at some point before `startfit`.

4.1.4 `numavesteps`

Syntax:

```
numavesteps <steps>
```

Sets the number individual crystal orientations to use when averaging crystallite contributions about some preferential axis. The larger the value of *<num steps>*, the closer the average will be to a true random distribution about the axis, but the longer *DEXA* will take to perform the averaging. The default value is 64. When averaging crystallites with no preferential orientation, contributions are averaged by *<num steps>* about a circle, and *<num steps>*/2 circles to cover the complete surface of a sphere.

It is recommended to leave this parameter alone unless you are using a single crystal sample. In this case, no crystallite averaging is needed, so *<num steps>* should be set to 1. A `preforientation` of *<degree>* 1.0 should then be used to orientate the crystal correctly in the beamline frame.

4.1.5 `startfit`

Syntax:

```
startfit [<iterations>]
```

Once all spectrum objects have been prepared and the important experiment vectors defined (sample magnetisation and x-ray polarisation), this command starts the fitting process that attempts to optimise the declared parameters. The optional argument *[<iterations>]* may be used to specify the maximum number of iterations allowed by the minimisation routine. If it is not specified, the algorithm itself will only quit once it has reached an acceptable minimum.

If the specified fit parameters are assigned a value before calling `startfit`, that value will be used as a first guess by the optimisation algorithm. Once `startfit` has finished, each fit parameter will contain its optimised value.

Whilst this command isn't compulsory, it must be called for any optimisation of fit parameters to take place.

4.1.6 `saveparams`

Syntax:

`saveparams` *<file>*

Outputs the current value of all fit parameters to the text file specified by *<file>*. If this file already exists it will be overwritten! Calling `saveparams` before `startfit` will save the first guess values, whilst calling it after will save the optimised values.

4.2 Spectrum Commands

These commands define the properties of a specific spectrum. In each case, they must be prefixed by a spectrum name and the '.' delimiter according to the general rule *<spectrum>.<command> <options>*. The spectrum name, *<spectrum>*, must refer to a spectrum object previously created with the `create` command. All commands are case-sensitive and so must be entered exactly as they are written here.

4.2.1 `spectrum` *

Syntax:

`spectrum` *<file> <header> <xcol> <ycol>*

Loads an experimental DiffXAS spectrum into the *DEXA* spectrum object. The data must be in columns that are delimited by one or more white-space characters (a space or tab). The x ordinate (wavenumber) data is read from

column *<xcol>*, and the y ordinate (absorption) data from column *<ycol>*. The numbering is base-1 such that first column on a line is column 1.

The *<header>* argument must tell *DEXA* how many lines are taken up at the start of the file by its header, before the actual data is reached. There must be no additional comments after the data at the end of the file.

4.2.2 feffinp *

Syntax:

```
feffinp <file>
```

Declares the name, *<file>*, of the *FEFF* `feff.inp` file that is to be loaded into the spectrum. In reality, only the coordinates of the atoms in the *FEFF* cluster are needed by *DEXA* , so all *FEFF* flags are ignored except for POTENTIALS and ATOMS. This command must be called before setting any spectrum properties that relate to specific scattering paths.

4.2.3 pathsdat *

Syntax:

```
pathsdat <file>
```

Declares the name, *<file>*, of the *FEFF* `paths.dat` file that is to be loaded into the spectrum. This will define all the possible scattering paths that can be considered in the *DEXA* fit. However, only those paths for which a `feffNNNN.dat` file is later declared (with `addpath`) will actually be included in the analysis. This command must be called before any other commands relating to specific scattering paths.

4.2.4 addpath *

Syntax:

```
addpath <file> <header>
```

Adds a new scattering path to the spectrum, which will subsequently be considered during the analysis. At least one scattering path must be defined. The filename *<file>* must point to the `feffNNNN.dat` file produced for the scattering path that is to be added. In turn, this file must come from the same *FEFF* calculation as the `paths.dat` file specified with the `pathsdat` command.

The *<header>* argument declares the number of lines to discard in the header of the `feffNNNN.dat` file. It should be set to 8 for files produced with *FEFF* 6, and 10 for those from *FEFF* 8.

4.2.5 updatepath

Syntax:

```
updatepath <path> <property> <value>
```

Modifies a given property of the named scattering path. *<path>* must match the filename of a `feffNNNN.dat` file that has previously been loaded with the `addpath` command. *<property>* defines which path property should be modified, and must be one of `S02`, `Sig2`, or `dR` to set the EXAFS factors S_0^2, σ^2, or ΔR respectively. Each will typically be obtained from the results of a fit to the sample's conventional EXAFS spectrum. If *FEFF* 's DEBYE flag was used, the *<value>* accompanying `Sig2` should be the fitted value minus the portion calculated by *FEFF* and given in the `feffNNNN.dat`. The ΔR is the difference between the half path length found in *FEFF* 's `paths.dat` and that fitted to experiment.

The default value for each path property will be those loaded from the *FEFF* `paths.dat` and `feffNNNN.dat` files.

4.2.6 setparameter

Syntax:

```
setparameter <parameter> <value> [<path>]
```

Sets a property of the sample or DiffXAS spectrum as a whole. Unlike `updatepath`, `setparameter` is used to set properties which affect all scattering paths. The *<parameter>* must be one of those listed in Table 4.2.7. The optional *<path>* parameter is only specified for parameters that affect a single scattering path. It should be an integer, with the first path loaded by `addpath` assigned the number 0. For parameters that apply to the whole crystal, it is omitted.

The parameter list is the same as that used by `fitparameter`. It is therefore possible to use `setparameter` before starting the fitting process in order to define first guesses to the optimised paramter values. This will then improve the chance of the fitting algorithm finding the correct minimum.

<parameter>	Factor	Notes
lambdaA0	$\lambda^{\alpha,0}$	Volume magnetostriction
lambdaG2	$\lambda^{\gamma,2}$	Equivalent to $(3/2)\lambda_{100}$ in cubic symmetry
lambdaE2	$\lambda^{\epsilon,2}$	Equivalent to $(3/2)\lambda_{111}$ in cubic symmetry
lambdaD2	$\lambda^{\delta,2}$	
lambda1A2	$\lambda_1^{\alpha,2}$	
lambda1G2	$\lambda_1^{\gamma,2}$	
lambda2G2	$\lambda_2^{\gamma,2}$	
lambda3G2	$\lambda_3^{\gamma,2}$	
lambda4G2	$\lambda_4^{\gamma,2}$	
dE	ΔE	The offset between the experimental edge energy and that determined by *FEFF* in eV. This should be set from the results of a normal EXAFS fit.
dSig2j	$\Delta\sigma_j^2$	
prefX		
prefY		
prefZ		
prefDeg		

Table 4.1: The list of *<parameter>*s that may be used by the `setparameter` and `fitparameter` commands.

4.2.7 `fitparameter`

Syntax:

`fitparameter` *<name> <parameter> [<path>]*

Adds the specified *<parameter>* to the list of those to be fitted by *DEXA* once `startfit` is called. The It must be one of the parameters listed in Table 4.2.7. The *<path>* argument is only used for parameters that apply to a specific scattering path. It should be an integer, with the first path loaded by `addpath` assigned the number 0. For parameters that apply to the whole crystal, it is omitted.

The *<name>* may be anything you like, but is important when linking parameters. All parameters with a unique *<name>* are fitted individually, but when two or more parameters have the same *<name>*, *DEXA* will link them during the fit so that they are all assigned the same numerical value. See section **??** for more details on linking parameters.

<parameter>	Factor	Iso	Cub	Tet	Cyl	Hex
lambdaA0	$\lambda^{\alpha,0}$	X	X	X	X	
lambdaG2	$\lambda^{\gamma,2}$	X	X	X		
lambdaE2	$\lambda^{\epsilon,2}$		X	X		
lambdaD2	$\lambda^{\delta,2}$			X	X	
lambda1A2	$\lambda_1^{\alpha,2}$				X	
lambda1G2	$\lambda_1^{\gamma,2}$				X	
lambda2G2	$\lambda_2^{\gamma,2}$				X	
lambda3G2	$\lambda_3^{\gamma,2}$					
lambda4G2	$\lambda_4^{\gamma,2}$					

Table 4.2: The tensor coefficients currently handled by *DEXA* . Each coefficient is only valid for the crystal symmetries marked with an 'X'. Some symmetries have yet to be implemented and so are absent from this list.

4.2.8 preforientation

Syntax:

`preforientation` *<θ_1>* *<θ_2>* *<θ_3>* *<degree>*

Defines a preferential orientation for the crystal or crystallites in the sample during the time of the experiment.

For a single crystal, one `preforientation` should be given, with a *<degree>* of 1, to orientate the crystal correctly with respect to the beamline frame. `numavesteps` should then be set to 1 to prevent the crystal being rotated about this axis to average multiple contributions, as would be necessary for a polycrystalline sample.

By default, there no preferential orientations are defined.

4.2.9 savespectrum

Syntax:

`savespectrum` *<file>*

Saves the theory spectrum, to *<file>*, generated from the current parameter values. If *<file>* already exists, it will be overwritten. Calling `savespectrum` just before `startfit` will save the first guess, used at the start of the optimisation process, and calling it just after will save the optimised result.

4.2.10 saveexperiment

Syntax:

```
saveexperiment <file>
```

Saves the expermental spectrum to the file specified by *<file>*. If *<file>* already exists, it will be overwritten. *<file>* should therefore not be the same file name from which the raw experimental data was loaded! This command can be useful if some pre-processing has been performed on the experimental data, such as Fourier filtering it with the `filter` command.

4.2.11 symmetry

Syntax:

```
symmetry <symmetry>
```

Specifies the symmetry class of the crystal under study. This will then define which, and how many, tensor coefficients are needed in order to fully describe the crystal property observed in the DiffXAS measurement. The list of possible values for *<symmetry>* are shown in table 4.2.7, along with their associated coefficients. Any coefficients not needed within the specified crystal symmetry will be completely ignored by *DEXA* . The default symmetry is `cubic`.

4.2.12 filter

Syntax:

```
filter <kmin> <kmax> <dk> <rmin> <rmax> <dr>
```

Fourier filters the experimental spectrum using a Hann window. This command can only be called after `spectrum`. The back-transformed (q-space) spectrum will replace the original k-space spectrum in *DEXA* 's spectrum object, causing the DiffXAS fit to take place in q- rather than k-space.

<kmin>,*<kmax>*, and *<dk>* respectively define the lower and upper limits of the original k-space data to transform, and the width of the transition region of the Hann window. All data outside the region $k_{min} - \Delta k \leq k \leq k_{max} + \Delta k$ will be discarded; all data in the range $k_{min} + \Delta k < k < k_{max} - \Delta k$ preserved; and the data in the two intermediate regions, $k_{min} \pm \Delta k$ and $k_{max} \pm \Delta k$, modified according to:

$$k \tag{4.1}$$

The similar applies for *<rmin>*,*<rmax>*, and *<dr>*, when back-transforming the data from R- to filtered q-space.