FEFF8

The FEFF Project
Department of Physics
University of Washington

User's Guide

FEFF version 8.40
updated August 8, 2006

Abstract

FEFF is *ab initio* self-consistent real space multiple-scattering code for simultaneous calculations of x-ray-absorption spectra and electronic structure.

The approach builds in core-hole effects and can include local fields (TDLDA). Output includes extended x-ray-absorption fine structure (EXAFS), full multiple scattering calculations of various x-ray absorption spectra (XAS) and projected local densities of states (LDOS). The spectra include x-ray absorption near edge structure (XANES), x-ray natural and magnetic circular dichroism (XNCD and XMCD), spin polarized x-ray absorption spectra (SPXAS and SPEXAFS), non-resonant x-ray emission spectra (XES), and the x-ray scattering amplitude (Thomson and anomalous parts). Calculation of electron energy loss spectroscopy (EELS) is also possible, but is not automated in this release.

This work has been supported in part by grants from the U.S. Department of Energy and by the University of Washington Office of Technology Transfer.

This document can also be found on the internet in the FEFF Project Wiki:
http://leonardo.phys.washington.edu/feff/wiki

Contents

Chapter 1

Synopsis

FEFF8 calculates extended x-ray-absorption fine structure (EXAFS), x-ray-absorption near-edge structure (XANES), x-ray natural circular dichroism (XNCD), spin-dependent calculations of x-ray magnetic dichroism (XMCD) and spin polarized x-ray absorption (SPXAS and SPEXAFS), nonresonant x-ray emission (XES), electronic structure including local densities of states (LDOS), and x-ray elastic scattering amplitude $f = f_0 + f' + if''$. Electron energy loss (EELS) can also be calculated, but it is not an automated feature in this release.

FEFF uses an *ab initio* self-consistent real space multiple scattering (RSMS) approach for clusters of atoms ($Z < 99$), including polarization dependence, core-hole effects, and local field corrections. Calculations are based on an all-electron, real space relativistic Green's function formalism with no symmetry requirements. The method combines both full multiple scattering based on LU or Lanczos algorithms and a high-order path expansion based on the Rehr–Albers multiple scattering formalism.

For a quick start or self-guided tutorial we suggest that new users try a few of the examples in Section 4. For details on use of the code, examples and suggestions for calculation strategies, see Sections 2, 3, and 4. For details about the algorithms used, see the discussion for the appropriate module in Section 2. For additional details, see the published references listed in Appendix C.

The principal investigator of the FEFF project is:

John J. Rehr, Principal Investigator
Department of Physics, BOX 351560 University of Washington, Seattle, WA 98195
email: jjr@phys.washington.edu
telephone: (206) 543-8593, FAX: (206) 685-0635

FEFF is written in ANSI Fortran 77, with the non-standard extension of double precision complex (`complex*16`) variables. It requires at least 250 megabytes (MB) of available memory (RAM) to run. For XANES calculations, one generally needs more memory (about 500 MB of RAM for a cluster of 100 atoms, about 750 MB for a cluster of 200 atoms, and so on). See Appendix B for installation instructions.

Please contact the authors concerning any problems with the code. See Appendix F for trouble-shooting hints and problem/bug reports or the FAQ on the FEFF WWW pages (see below).

The full FEFF8 code is copyright protected software and users must obtain a license from the University of Washington Office of Technology Transfer for its use. See Appendix A for complete copyright notice and other details. Documentation and information on how to obtain the code can be found at the FEFF Project WWW URL:

http://leonardo.phys.washington.edu/feff/

or by e-mail to the FEFF Project Coordinator at feff@phys.washington.edu.

Please cite FEFF and an appropriate FEFF reference if the code or its results are used in published work. See Appendix C for a list of appropriate citations. The main published reference for FEFF8 is:

Real Space Multiple Scattering Calculation and Interpretation of X-ray Absorption Near Edge Structure, A.L. Ankudinov, B. Ravel, J.J. Rehr, and S.D. Conradson Phys. Rev. B**58**, 7565 (1998).

A manuscript dealing with the improvements added in FEFF8.20 is

Parallel calculation of electron multiple scattering using Lanczos algorthms, A.L. Ankudinov, C.E. Bouldin, J.J. Rehr, J. Sims, and H. Hung, Phys. Rev. B**65**, 104107 (2002).

FEFF Project Developers – Several developers contributed to the FEFF8.00 code, and its advanced versions FEFF8.10, FEFF8.20, and FEFF8.40.

The following developers have contributed to FEFF8.40 and other recent project developments: Fernando D. Vila, response functions and Debye-Waller factors; Micah Prange, FEFF/Optical; Joshua Kas, Bayesian XAS analysis; Yoshinari Takimoto, response functions; Adam Sorini, inelastic losses; and Kevin Jorissen, electron-energy loss spectroscopy (EELS).

Alexei Ankudinov (ALA) was the principal developer for previous versions of the FEFF8 series. ALA implemented the automated self-consistent potential algorithm, and added calculations of the ℓDOS, the Fermi level, and charge transfer. Bruce Ravel was the principal developer of the full multiple scattering (FMS) algorithm in FEFF8, which uses LU decomposition and is necessary for the SCF loop, DOS and for FMS/LU XANES calculations. In version FEFF8.10, ALA added calculations of elastic scattering amplitude, x-ray natural circular dichroism and nonresonant x-ray emission. ALA was largely responsible for FEFF8.20, which introduced an improved input/output structure to facilitate interaction with other codes. He also added quadrupolar transitions and implemented fast Lanczos methods for the iterative FMS algorithm.

New routines for sum-rules applications were added in collaboration with A. Nesvizhskii. Anna Poiarkova and Patrick Konrad contributed new codes for calculating multiple-scattering Debye–Waller factors and anharmonic contributions. Matthew Newville added an improved padded ascii output structure to facilitate the interface to XAFS analysis codes and portability between different machines. Jim Sims (NIST) modified the code for MPI based parallel execution, in collaboration with C. Bouldin and JJR.

The high-order multiple-scattering routines, pathfinder, and input/output routines were largely developed by Steven Zabinsky and JJR for FEFF5 and 6, and are still in use. The

Hedin–Lundqvist self-energy and phase shift routines were developed for early versions of FEFF in collaboration with Jose Mustre de Leon, Dan Lu and R.C. Albers and are still part of FEFF8.

The authors thank many users of experimental versions of the FEFF8 code for feedback, suggestions and bug reports. The authors also thank R.C. Albers, K. Baberschke, C. Bouldin, C. Brouder, G. Brown, S.D. Conradson, F. Farges, G. Hug, M. Jaouen, J. Sims, and E. Stern for useful comments.

Table 1.1: Typographic conventions in this document

font	denotes
SMALL CAPS	names of programs
`typewriter font`	contents of files
‘`quoted typewriter font`’	file names
ROMAN CAPITALS	names of cards in the ‘`feff.inp`’ file
`slanted typewriter font`	commands executed at a command line

Chapter 2

Input File Control Cards

The main program FEFF reads a single file ('`feff.inp`') created by the user and executes the various program modules described below. An auxiliary FORTRAN program (ATOMS), developed by Bruce Ravel, is provided to generate the '`feff.inp`' file for crystals from crystallographic input parameters. A perl-based graphical user interface to ATOMS is also available. Information about ATOMS can be found on the WWW at:

http://cars9.uchicago.edu/~ravel/software/aboutatoms.html

This section describes '`feff.inp`' and the commands that control FEFF. It may be helpful to look at one or more of the sample input files in Section 4 while reading this section.

The input for FEFF8 is similar to that for FEFF versions 5 through 7, except that several additional options have been added to permit self-consistent potential generation, full multiple scattering XANES calculations, polarization dependence, and others listed in Section 2.1. FEFF8 is backwards compatible and supports input files earlier style input files (from FEFF versions 5 through 7).

The input file '`feff.inp`' is a loosely formatted, line-oriented text file. Each type of input read by the program is on a line which starts with a *card* or *keyword* and, in some cases, is followed by alpha-numeric data. All card arguments listed below inside square brackets are optional. The sequence of keyword cards is arbitrary. If any card or optional data is omitted, default values are used; an exception is the POTENTIALS card, which is always required. (Either the ATOMS card or the OVERLAP card is also required for FEFF to run.)

Alpha-numeric values are listed in free format, separated by blanks. Tab characters are *not allowed* (due to Fortran 77 portability constraints) and may cause confusing error messages. Any characters appearing after the card and its required or optional data on a given line are ignored by FEFF and can be used as end-of-line comments. Spaces between lines (empty lines) are ignored. Any line beginning with an asterisk (`*`) is regarded as a comment and is also ignored.

All distances are in angstroms (Å) and energies in electron-volts (eV).

After reading the '`feff.inp`' file (in the "zeroth" module **RDINP**, read input), the calculations of various spectroscopies are done sequentially in six steps:

1. The scattering potentials are calculated using atomic overlap (Mattheiss) prescription or self-consistently using an automated SCF loop. Absolute energies are estimated. (module **pot**, potentials.)
2. The scattering phase shifts, dipole matrix elements, and x-ray cross-section are calculated. (module **xsph**, cross-section and phases.)
3. Full multiple scattering XANES calculations are done for a specified cluster size. (module **fms**, full multiple scattering.)
4. The leading multiple scattering paths for the cluster are enumerated. (module **path**, path enumeration.)
5. The effective scattering amplitudes f_{eff} and other XAFS parameters are calculated for each scattering path. (module **genfmt**, general-path F-matrix calculation.)
6. The XAFS parameters from one or more paths are combined to calculate a total XAFS or XANES spectrum. (module **ff2x**, scattering amplitude to χ.)

There is also one optional step:

1. If the LDOS card is present in '`feff.inp`', the angular momentum projected density of states is calculated. In previous versions of FEFF8 this calculation was part of **xsph**, but it now runs independently, between modules **pot** and **xsph**. However, it will still be described with the **xsph** cards for simplicity. (module **ldos**, ℓ-projected density of states.)

This section describes how to control each module using the '`feff.inp`' file.

2.1 Complete List of FEFF8 Control Cards

The '`feff.inp`' options fall into three categories: *standard* options frequently and easily used, *useful* options that are often used, and *advanced* options that are seldom necessary, but may be helpful in some cases. Every card in '`feff.inp`' will influence the calculations, beginning in some module. Thus, for better understanding of how each module can be affected by the input cards, we list them separately for each module.

Module 0 .... **rdinp**

Purpose of Module: Read input data

Standard cards: ATOMS, CONTROL, END, PRINT, and TITLE

Useful Cards: RMULTIPLIER

Advanced Cards: CFAVERAGE and OVERLAP

Module 1 .... **pot**

Purpose of Module: Calculate (self-consistent) scattering potentials and Fermi energy

Standard cards: AFOLP, EDGE, HOLE, POTENTIALS, and S02

Useful Cards: EXCHANGE, FOLP, NOHOLE, RGRID, SCF, and UNFREEZEF

Advanced Cards: INTERSTITIAL, ION, and SPIN

Module 2 . . . . **xsph**

Purpose of Module: Calculate cross-section and phase shifts and ℓDOS

Standard cards: EXAFS and XANES

Useful Cards: ELLIPTICITY, LDOS, MULTIPOLE, and POLARIZATION

Advanced Cards: DANES, FPRIME, RPHASES, RSIGMA, TDLDA, XES, and XNCD (or XMCD)

Module 3 . . . . **fms**

Purpose of Module: Calculate full multiple scattering for XAS

Standard cards: FMS

Useful Cards: DEBYE

Advanced Cards:

Module 4 . . . . **path**

Purpose of Module: Path enumeration

Standard cards: RPATH

Useful Cards: NLEG

Advanced Cards: PCRITERIA and SS

Module 5 . . . . **genfmt**

Purpose of Module: Calculate scattering amplitudes and other XAFS parameters

Standard cards:

Useful Cards: CRITERIA

Advanced Cards: IORDER and NSTAR

Module 6 . . . . **ff2x**

Purpose of Module: Calculate final output.

Standard cards: DEBYE

Useful Cards: CORRECTIONS and SIG2

Advanced Cards:

These data types are listed below in the same order as in the table above. Each description is of this form:

♦ **CARD** `required arguments [optional arguments]` *type*
The type is one of *Standard*, *Useful*, or *Advanced*. The argument list is a brief statement of the valid arguments to the card. Arguments in square brackets are optional. The text description will explain the arguments and their uses more fully. Example uses of the card look like this:

```
* brief description of the example
CARD  arguments
```

2.2 Main Control Cards

The cards in this section either control module **rdinp** or are not associated with any particular module, and are used throughout the FEFF calculations. In either case, they begin influencing the calculations in RDINP. The ATOMS card is used to specify the absorbing atom and its environment (if atomic coordinates are not known, then the OVERLAP card can be used to construct approximate potentials). Without this structural information, no calculations can be done. The CONTROL card is used to selectively run parts of FEFF. The PRINT card controls which output files are written by the modules.

♦ **ATOMS** *Standard*
Cartesian coordinates (in Ångstroms) and unique potential indices of each atom in the cluster are entered following the ATOMS card, one per line. See the discussion of the POTENTIALS card. An auxiliary code, ATOMS, written by Bruce Ravel, is supplied with FEFF to generate the ATOMS list for crystals from crystallographic data. See the document for ATOMS for more information.

```
ATOMS
* x      y      z     ipot     SF6 molecule
  0.0    0.0    0.0     0      S K-shell hole
  1.56   0.00   0.00    1      F 1st shell atoms
  0.00   1.56   0.00    1
  0.00   0.00   1.56    1
 -1.56   0.00   0.00    1
  0.00  -1.56   0.00    1
  0.00   0.00  -1.56    1
```

♦ **CONTROL** `ipot ixsph ifms ipaths igenfmt iff2chi` *Standard*
The CONTROL card lets you run one or more of the modules separately. There is a switch for each module: 0 means not to run that module, 1 means to run it. You can do the whole run in sequence, one module at a time, but you *must* run all modules sequentially. *Do not* skip

modules: `CONTROL 1 1 1 0 0 1` is incorrect. The default is `CONTROL 1 1 1 1 1 1`, i.e. run all 6 modules.

```
* example 1
* calculate self consistent potentials, phase shifts and fms only
CONTROL  1 1 1 0 0 0   ipot  ixsph  ifms   ipaths  igenfmt  iff2chi

* example 2
* run paths, genfmt and ff2x; do not run pot, xsph, fms
* this run assumes previous modules have already been run and
* adds MS paths between rfms  and rpath to the MS expansion
CONTROL  0 0 0 1 1 1    ipot  ixsph  ifms   ipaths  igenfmt  iff2chi
```

♦ **END** *Standard*

The END card marks the end of the portion of the '`feff.inp`' file that FEFF will read. All data following the END card is ignored. Without an END card, the entire input file is read.

```
* ignore any lines in feff.inp that follow this card
END
```

♦ **PRINT** `ppot pxsph pfms ppaths pgenfmt pff2chi` *Standard*

The PRINT card determines which output files are printed by each module. See Section 3 for details about the contents of these files. The default is print level 0 for each module.

```
* add crit.dat and feffNNNN.dat files to minimum output
PRINT  0  0  0  1  0  3
```

The print levels for each module are summarized in Table 2.1 on page 10.

♦ **TITLE** `title_line` *Standard*

User supplied title lines. You may have up to 10 of these. Titles may have up to 75 characters. Leading blanks in the titles will be removed.

```
TITLE  Andradite  (Novak and Gibbs, Am.Mineral 56,791 1971)
TITLE  K-shell 300K
```

♦ **RMULTIPLIER** `rmult` *Useful*

With RMULTIPLIER all atomic coordinates are multiplied by the supplied value. This is useful to adjust lattice spacing, for example, when fractional unit cell coordinates are used. By default, `rmult`=1.

Table 2.1: Print levels controlling output files from the modules.

module	print levels
pot	0 write 'pot.bin' only
	1 add 'misc.dat'
	2 add 'potNN.dat'
	3 add 'atomNN.dat'
xsph	0 write 'phase.bin' and 'xsect.bin' only; with the LDOS card add 'ldosNN.dat' and 'logdos.dat'
	1 add 'axafs.dat'
	2 add 'phaseNN.dat' and 'phminNN.dat'
	3 add 'ratio.dat' (for XMCD normalization) and 'emesh.dat'
fms	0 write 'fms.bin'
path	0 write 'paths.dat' only
	1 add 'crit.dat'
	3 add 'fbeta' files (plane wave $\|f(\beta)\|$ approximations)
	5 Write only 'crit.dat' and do not write 'paths.dat'. (This is useful when exploring the importance of paths for large runs.)
genfmt	0 write 'list.dat', and write 'feff.bin' with all paths with importance greater than or equal to two thirds of the curved wave importance criterion
	1 write all paths to 'feff.bin'
ff2x	0 write 'chi.dat' and 'xmu.dat'
	2 add 'chipNNNN.dat' ($\chi(k)$ from each path individually, this can quickly fill up your disk if you're doing a large run)
	3 add 'feffNNNN.dat' (input files for Matt Newville's FEFFIT program) and 'files.dat', and do not add 'chipNNNN.dat' files

```
*increase distances by 1%
RMULTIPLIER 1.01
```

♦ **CFAVERAGE** `iphabs nabs rclabs` *Advanced*

A "configuration" average over the spectra of multiple absorbing atoms is done if the CFAVERAGE card is used. CFAVERAGE currently assumes phase transferability, which is usually good for EXAFS calculations, but may not be accurate for XANES. Note that the CFAVERAGE card is currently unreliable in general, and in particular is incompatible with the DEBYE card for options other than the correlated Debye model (`idwopt` > 0).

`iphabs`
: potential index for the type of absorbing atoms over which to make the configuration average (any potential index is allowed).

`nabs`
: the configuration average is made over the first **`nabs`** absorbers in the '`feff.inp`' file of type **`iphabs`**. You do not need to have potential of index 0 in your input file when using the CFAVERAGE card, but you must have the same type of potential for iph=0 and iph=iphabs. The configurational average is done over ALL atoms of type **`iphabs`**, if **`nabs`** is less than or equal to zero.

`rclabs`
: radius to make a small atom list from a bigger one allowed in '`feff.inp`'. Currently the parameter controlling the maximum size of the list, **`natxx`**, is set to 100,000, but this can be increased. The pathfinder will choke on too big an atoms list. You must choose `rclabs` to have fewer than 1,000 atoms in the small atom list. If your cluster is less 1,000 atoms simply use `rclabs`=0 or negative always to include all atoms.

Default values are `iphabs`=0, `nabs`=1, `rclabs`=0 (where `rclabs` = 0 means to consider an infinite cluster size).

```
*average over all atoms with iph=2 in feff.inp with less than 1000 atoms
CFAVERAGE 2  0  0
```

♦ **OVERLAP** `iph` *Advanced*

The OVERLAP card can be used to construct approximate overlapped atom potentials when atomic coordinates are not known or specified. If the atomic positions are listed following the ATOMS card, the OVERLAP card is not needed. FEFF8 will stop if both the ATOMS and OVERLAP cards are used. The OVERLAP card contains the potential index of the atom being overlapped and is followed by a list specifying the potential index, number of atoms of a given type to be overlapped and their distance to the atom being overlapped. The examples below demonstrate the use of an OVERLAP list. This option can be useful for initial single scattering XAFS calculations in complex materials where very little is known about the structure.

You should verify that the coordination chemistry built in using the OVERLAP cards is realistic. It is particularly important to specify all the nearest neighbors of a typical atom in the shell to be overlapped. The most important factor in determining the scattering amplitudes is the atomic number of the scatterer, but the coordination chemistry should be approximately correct to ensure good scattering potentials. Thus it is important to specify as accurately as possible the coordination environment of the scatterer. Note: If you use the OVERLAP card, then you cannot use the FMS or SCF cards. Also the pathfinder won't be called and you must explicitly specify single scattering paths using the SS card, which is described in Section 2.6.

```
* Example 1. Simple usage
* Determine approximate overlap for central and 1st nearest neighbor in Cu
OVERLAP 0         determine overlap for central atom of Cu
  *iphovr   novr   rovr      * ipot, number in shell, distance
   1        12     2.55266
OVERLAP 1         determine approximate overlap for 1st shell atoms
  *iphovr   novr   rovr      * ipot, number in shell, distance
   0        12     2.55266

* Example 2. More precise usage
* Determine approximate overlap for 3rd shell atoms of Cu
OVERLAP 3
  0  1 2.55266    ipot, number in shell, distance
  1  4 2.55266
  2  7 2.55266
  2  6 3.61000
  2 24 4.42133
```

2.3 POT: Scattering Potentials

Initially the free atom potentials of each atomic type are calculated using a relativistic Dirac–Fock atom code, treating the atoms as if they were isolated in space. Scattering potentials are calculated by overlapping the free atom densities in the muffin tin approximation (Mattheiss prescription), and then including the Hedin–Lundqvist/Quinn self energy for excited states. Non-overlapping muffin-tin radii are determined automatically from the calculated Norman radii. Automatic overlapping of muffin tin spheres (see the AFOLP card) is done by default, since it typically leads to better results than non overlapping muffin-tin spheres. FEFF8 can also calculate self-consistent potentials by successively calculating the electron density of states, electron density and Fermi level at each stage within a small cluster and then iterating, using the Mattheiss prescription for the initial iteration.

XAFS spectra are referenced to the threshold Fermi level. This quantity is best determined with the self-consistent field procedure (typically to within a fraction of an eV), or (less accurately but faster) can be estimated from the electron gas result at the mean interstitial density in the Mattheiss prescription, as in FEFF7. An absolute energy scale is obtained by an atomic calculation of the total energy of the system with and without the core-hole. Atomic configurations and core-hole lifetimes are built in, and mean free paths are determined from the imaginary part of the average interstitial potential, including self-energy and lifetime contributions.

The potential calculations need as input only the atomic number of the atoms, and, for the absorbing atom, the type of the core hole being considered. To do the overlapping of the unique potentials, the neighboring atoms must be identified, either by position (from a list of the Cartesian coordinates of each atom) or by explicit overlapping instructions using the OVERLAP card described in the previous section.

To save time the code calculates the overlapped atom potential for each unique potential only once, using as a sample geometry the atom with a given unique potential index that is closest to the absorbing atom. Thus it is essential that the neighborhood of each sample atom be appropriate.

♦ **AFOLP** `folpx` *Standard*

This automatically overlaps all muffin-tins to a specified maximum value (default `folpx`=1.15) to reduce the effects of potential discontinuities at the muffin-tins. Automatic overlapping is done by default and is useful in highly inhomogeneous materials. Typical values of the overlapping fraction should be between 1.0 and 1.3. See FOLP for a non-automated version. Automatic overlapping is done by default; to switch overlapping off, use 1.0 as the argument for AFOLP.

```
* touching muffin-tins; do not use automatic overlapping
AFOLP  1.0
```

♦ **EDGE** `label s02` *Standard*

The EDGE card is equivalent to the HOLE card, but you don't have to look up the appropriate integer index. Simply use the hole label: `K` means K-shell, `L1` means L_I, and so on. NO means no hole, but FEFF will not accept it and one must use the NOHOLE card instead. Calculations with very shallow edges, e.g. M-shells and higher, are not well tested; please complain to the authors if you encounter problems. As with the HOLE card, you may also use the integer index instead of the label. All comments for the HOLE card are valid for the EDGE card—see the description below. If the entry for the amplitude reduction factor S_0^2 is less than 0.1, S_0^2 will be estimated from atomic overlap integrals.

```
* L1-shell core hole, S02 = 1
EDGE  L1   1.0
```

♦ **HOLE** `ihole s02` *Standard*

The HOLE card is equivalent to the EDGE card, but the shell is specified by a hole-code index. It includes the amplitude reduction factor S_0^2 just as the EDGE card does. If the entry for S_0^2

is less than 0.1, then S_0^2 will be estimated from atomic overlap integrals. Experimental values of S_0^2 are typically between 0.8 and 1.0. The defaults if the HOLE card is omitted are `ihole`=1 for the K shell and S_0^2=1. The hole codes are shown in Table 2.2, however, note that FEFF will not accept ihole=0 and one must use the NOHOLE card instead to calculate without the core-hole.

For `ihole` > 4, the core-hole lifetime parameter ($\gamma_{\rm ch}$) is not tabulated in FEFF and is set equal to 0.1 eV, since the final state losses are then dominated by the self-energy. Use the EXCHANGE card to make adjustments ($\gamma_{\rm ch} = 0.1 + 2 \cdot \texttt{vi0}$).

```
* K-shell core hole, S02 estimated by overlap integrals
HOLE  1   0.0
```

index	edge	index	edge	index	edge	index	edge
0	NO	7	M3	14	N5	21	O5
1	K	8	M4	15	N6	22	O6
2	L1	9	M5	16	N7	23	O7
3	L2	10	N1	17	O1	24	P1
4	L3	11	N2	18	O2	25	P2
5	M1	12	N3	19	O3	26	P3
6	M2	13	N4	20	O4		

Table 2.2: Available hole codes. The entries in the columns marked edge are written as they are recognized by the EDGE card. Index 0, `NO`, is the no hole option described in the NOHOLE card.

♦ **POTENTIALS** `ipot Z [tag lmax1 lmax2 xnatph spinph]` *Standard*

The POTENTIALS card is followed by a list which assigns a unique potential index to each distinguishable atom. The potential index `ipot` is the index of the potential to be used for the phase shift calculation.

The required list entries are the unique potential index `ipot` and the atomic number `Z`. The `tag` is at most 6 characters and is used to identify the unique potential in the '`paths.dat`' output file.

The optional list entries `lmax1` and `lmax2` are used to limit the angular momentum bases of the self-consistent potentials (SCF) and full multiple scattering calculations (FMS). If a negative number (e.g., `lmax1` $= -1$) is specified for either `lmax1` or `lmax2`, FEFF will automatically use a default based upon the atomic number of the species normal lmax(atomic).

The next optional entry, `xnatph`, can be used to specify the stoichiometric number of each unique potential in the unit cell of a crystalline material. This helps in the calculation of the Fermi level. In the case of an infinite solid, `xnatph` $= 0.01$ (default value) is a suitable value for the absorbing atom. For materials finite in extent, see the example below.

The last optional entry `spinph` is used to specify the spin amplitude and relative spin alignment for spin-dependent calculations (see the SPIN card in this section for more information on spin-dependent calculations).

The absorbing atom must be given unique potential index 0 (there is an exception to this rule when the CFAVERAGE card is used). These unique potential indices are simply labels, so the order is not important, except that the absorbing atom is index 0, and you may not have missing indices (i.e., if you use index 3, you must also have defined unique potentials 1 and 2).

To save time, the code calculates the overlapped atom potential for each unique potential only once, using the atom with a given unique potential index. Thus it is essential that the neighborhood of that sample atom be representative. If it is not, the code may generate inaccurate potentials and phase shifts, as well as poor XAS results.

It is often useful to assume that the potential for a given shell of atoms is the same as that of a previously calculated shell in order to save calculation time. For example, in Cu it is a good approximation to determine potentials only for the central atom and the first shell and to use the first shell potential (`ipot` = 1) for all higher shells. Such approximations should be checked in each case, however.

```
* molecular SF6  Sulfur K edge, lamx1=default, lmax2=3 (spdf basis)
POTENTIALS
*   ipot     Z  tag  lmax1 lmax2  xnatph
     0      16   S    -1     3     1
     1       9   F    -1     3     6
```

An example of spin-dependent usage can be found in the SPIN card description in the next section.

♦ **S02** `s02` *Standard*

The S02 card specifies the amplitude reduction factor S_0^2. If the entry for S_0^2 is less than 0.1, then the value of S_0^2 is estimated from atomic overlap integrals. Experimental values of S_0^2 are typically between 0.8 and 1.0.

Alternatively, you can specify the value of S_0^2 in the HOLE or EDGE card; however, the meaning of the parameters in the '`feff.inp`' file is more clear if you use the S02 card.

```
* let FEFF calculate S02
S02    0.0
```

♦ **EXCHANGE** `ixc vr0 vi0 [ixc0]` *Useful*

The EXCHANGE card specifies the energy dependent exchange correlation potential to be used for the fine structure and for the atomic background.

`ixc` is an index specifying the potential model to use for the fine structure and the optional `ixc0` is the index of the model to use for the background function.

The calculated potential can be corrected by adding a constant shift to the Fermi level given by `vr0` and to a pure imaginary "optical" potential (i.e., uniform decay) given by `vi0`. Typical errors in FEFF's self-consistent Fermi level estimate are about 1 eV. (The CORRECTIONS card in Section 2.8 is similar but allows the user to make small changes in `vi0` and `vr0` *after* the rest of the calculation is completed, for example in a fitting process.)

The Hedin–Lundqvist self-energy is used by default and appears to be the best choice for most applications we have tested in detail. The partially nonlocal model (ixc=5) gives slightly better results in some cases, but has not been tested extensively.

Another useful exchange model is the Dirac-Hara exchange correlation potential with a specified imaginary potential vi0. This may be useful to correct the typical error in non-self-consistent estimates of the Fermi level of about +3 eV and to add final state and instrumental broadening.

Defaults if the EXCHANGE card is omitted are: `ixc`=0 (Hedin–Lundquist), vr0=0.0, vi0=0.0. For XANES, the ground state potential (`ixc0`=2) is used for the background function and for EXAFS the Hedin–Lundqvist (`ixc0`=0) is used.

Indices for the available exchange models:

- `0` Hedin–Lundqvist + a constant imaginary part
- `1` Dirac–Hara + a constant imaginary part
- `2` ground state + a constant imaginary part
- `3` Dirac–Hara + HL imag part + a constant imaginary part
- `5` Partially nonlocal: Dirac–Fock for core + HL for valence electrons + a constant imaginary part

```
*Hedin-Lundqvist -2eV edge shift and 1eV expt broadening
EXCHANGE 0 2. 1.

*Dirac-Hara exchange -3 eV edge shift and 5 eV optical potential
EXCHANGE 1 3. 5.
```

♦ NOHOLE *Useful*

This card roughly simulates the effect of complete core-hole screening. It is useful to test the final state rule for calculated XAS, and to compare with other calculations (such as band structure or other codes) that do not have a core hole. The code will use the final states specified by the HOLE or EDGE card for the matrix element calculation — the NOHOLE card will cause FEFF to calculate potentials and phase shifts as if there is no core-hole. For dDOS and L_{II} or L_{III} absorption calculations, for example, NOHOLE often gives better agreement for white line intensities. Conversely NOHOLE tends to give poor XANES intensities for K-shell absorption in insulators.

♦ RGRID `delta` *Useful*

The radial grid used for the potential and phase shift calculation is

$$r(i) = \exp(-8.8 + (i-1) \cdot \delta)$$

where $\delta \equiv$ `delta`. $\delta = 0.05$ by default, which is sufficient for most cases. However, occasionally there are convergence problems in the atomic background at very high energies (the background curves upward) and in the phase shifts for very large atoms. If such convergence problems are encountered we suggest reducing δ to 0.03 or even 0.01. This will solve these problems at the cost of longer computation times (the time is proportional to $1/\delta$). This option is also useful for testing and improving convergence of atomic background calculations.

```
* reduce grid for more accurate background at high energy
RGRID 0.03
```

♦ **SCF** `rfms1 [lfms1 nscmt ca nmix]` *Useful*

This card controls FEFF's automated self-consistent potential calculations. Thus all fields except rfms1 are optional. If this card is not specified, then all calculations are done with the non-self-consistent (overlapped atomic) potential. By default `lfms1`=0, `nscmt`=30, `ca`=0.2, and `nmix`=1.

`rfms1`
: This specifies the radius of the cluster for full multiple scattering during the self-consistency loop. Typically one needs about 30 atoms within the sphere specified by `rfms1`. Usually this value is smaller than the value `rfms` used in the FMS card, but it should be larger than the radius of the second coordination shell.

`lfms1`
: The default value 0 is appropriate for solids; in this case the sphere defined by `rfms1` is located on the atom for which the density of states is calculated. The value 1 is appropriate for molecular calculations and will probably save computation time, but may lead to inaccurate potentials for solids. When `lfms1` = 1 the center of the sphere is located on the absorbing atom.

`nscmt`
: This is the maximum number of iterations the potential will be recalculated. A value of 0 leads to non-self-consistent potentials and Fermi energy estimates. A value of 1 also yields non-self-consistent potentials but the Fermi energy is estimated more reliably from calculations of the ℓDOS. Otherwise, the value of `nscmt` sets an upper bound on the number of iterations in the self-consistency loop. Usually self-consistency is reached in about 10 iterations.

`ca`
: The convergence accelerator factor. This is needed only for the first iteration, since FEFF uses the Broyden algorithm to reach self-consistency. A typical value is 0.2; however, you may want to try smaller values if there are problems with convergence. After a new density is calculated from the new Fermi level, the density after the first iteration is

$$\rho_{\text{next}} = ca * \rho_{\text{new}} + (1 - ca) * \rho_{\text{old}}$$

`ca` = 1.0 is extremely unstable and should not be used.

`nmix`

This specifies how many iterations to do with the mixing algorithm before starting the Broyden algorithm. The SCF calculations in materials containing f-elements may not converge. We encountered such a problem for Pu. However, the SCF procedure converged if we started the Broyden algorithm after 10 iterations with the mixing algorithm with `ca` = 0.05. `nmix` must be between 1 and 30; a value outside of this range will be ignored, and replaced with an acceptable value.

```
* Automated FMS SCF potentials for a molecule of radius 3.1 Angstroms
SCF  3.1 1

* To reach SCF for f-elements and UNFREEZEF we sometimes had to use
SCF  3.7 0  30  0.05  10
```

♦ UNFREEZEF *Useful*

In many applications of f-electron systems, we found that it is usually preferable to freeze the f-electron density at the atomic value in order to achieve well converged SCF potentials. This is the default in FEFF8.4. If one still wants to attempt to calculate the f-DOS self-consistently, as in FEFF8.00 and 8.10, the UNFREEZEF card should be used.

```
* To include f-electrons in SCF calculations use
UNFREEZEF
```

♦ FOLP `ipot folp` *Advanced*

The FOLP card sets a parameter which determines by what factor the muffin-tin radii are overlapped for the specified potential index. We recommend that the AFOLP card be used (default overlap = 1.15) in cases with severe anisotropy. FOLP should be used with caution, for example, for hydrogen or for fitting AXAFS. Typically only values larger than 1 and less than 1.3 should be used, except for hydrogen atoms, where we recommend the value 0.8. The AFOLP card is ignored when FOLP is used for a particular potential type.

```
*  +20% overlap of muffin tin with unique potential 1
*  -20% overlap of muffin tin with unique potential 2
FOLP 1  1.2    * adjust overlap to fit AXAFS
FOLP 2  0.8    * use 0.8 for hydrogen
```

♦ INTERSTITIAL `inters vtot` *Advanced*

The construction of the interstitial potential and density may be changed by using this card. `inters` = `ipot`+2∗`irav`+6∗`irmt`. `ipot`=1 might be useful when only the surroundings of the absorbing atom are specified in '`feff.inp`'. `irmt` and `irav` are described only for completeness, and use of nonzero values is strongly discouraged.

`ipot`

potential index. `ipot` defines how to find the interstitial potential. `ipot`=0 (default): the interstitial potential is found by averaging over the entire extended cluster in '`feff.inp`'. `ipot`=1 : the interstitial potential is found locally around the absorbing atom.

`irav`

also changes how the interstitial potential is found. `irav`=0 (default): the equation for V_int is constructed at `rav`=`rnrm`. `irav`=1 : at `rav`=(`rmt` +`rnrm`)/2. `irav`=2 : at rav=rmt, where `rmt` is the muffin-tin radius and rnrm is the Norman radius. irmt apparently does not exist in the code

`irmt`

`irmt`=0 (default): Norman prescription for mt radii. `irmt`=1 : Matching point prescription for mt radii (do not use).

`vtot`

the volume per atom normalized by `ratmin`^{3} (`vtot`=(volume per atom)/`ratmin`^{3}), where `ratmin` is the shortest bond for the absorbing atom. This quantity defines the total volume (needed to calculate the interstitial density) of the extended cluster specified in '`feff.inp`'. If `vtot` ≤ 0 then the total volume is calculated as a sum of Norman sphere volumes. Otherwise, *total volume* = `nat` $*$ (`vtot` $*$ `ratmin`^{3}), where `nat` is the number of atoms in an extended cluster. Thus `vtot`=1.0 is appropriate for cubic structures, such as NaCl. The INTERSTITIAL card may be useful for open systems (e.g. those which have ZnS structure).

```
* improve interstitial density for ZnS structures.
* vtot = (unit_cell_volume/number_of_atoms_in_unit_cell)/ratmin**3)=1.54
INTERSTITIAL  0 1.54
```

♦ **ION** `ipot ionization` *Advanced*

The ION card ionizes all atoms with unique potential index `ipot`. Negative values and non-integers are permitted, however ionicities larger than 2 and less than -1 often yield unphysical results. Our experience with charge transfers using the SCF card suggests values for `ionization` about 5–10 times smaller than the formal oxidation state. The ION card is probably not needed if the potential is self-consistent. However, it can be used to put some total charge on a cluster. In this case we suggest using the same ionicity for all atoms in the cluster (i.e. total ionization divided by number of atoms). For example, for diatomics like Br2, the fully relaxed configuration has a formal ionization of 1 on the scattering atom. Because of charge transfer, the actual degree of ionization is much smaller. In non-self-consistent calculations the default (non-ionized) scattering potentials are often superior to those empirically ionized, and the results should be checked both ways. If ION is omitted, the atoms are not ionized by default.

```
* Simulates effective ionization for formal valence state +1
```

```
* ipot, ionization
ION  1  0.2
```

♦ **SPIN** `ispin [x y z]` *Advanced*

This card is used to specify the type of spin-dependent calculation for spin along the (x, y, z) direction, along the z-axis by default. The SPIN card is required for the calculation of all spin-dependent effects, including XMCD and SPXAS (see Section 4.4.3).

Whenever spin-dependent calculations are made, the spin amplitude and relative spin orientation should be specified in the `spinph` argument of the POTENTIALS card.

If FEFF has been compiled with the parameter `nspx`=1 (default), the values of the index `ispin` correspond to the following calculations:

`ispin`	meaning
2	spin-up SPXAS and ℓDOS
-2	spin-down SPXAS and ℓDOS
1	spin-up portion of XMCD calculations
-1	spin-down portion of XMCD calculations

Table 2.3: Allowed values of the `ispin` argument of the SPIN card.

The default `ispin` = 0 is used for spin-independent calculations.

If FEFF is compiled with `nspx` = 2, one simply uses `ispin` = 1 or `ispin` = 2 as needed, and the up and down spins are taken care of automatically. See Section 4.4 for more information on the strategies and options available for spin-dependent calculations.

The spin-dependent potentials are calculated from the spin-dependent densities, using von Barth-Hedin results for the uniform electron gas. We use this approximation to construct the spin-dependent muffin-tin potential. This should be fine for EXAFS, where small details of the potential are irrelevant, but may not be good enough in the XANES region, where the self-consistent spin-dependent muffin-tin potential can lead to better results.

```
* Make spin-up XMCD calculation (XMCD card must also be included)
SPIN 1
XMCD
```

POTENTIALS card for spin-dependent calculations:

```
* GdFeO, spin-dependent
* Note that SPIN card must also be present in feff.inp, as well as
* additional cards specific to desired calculations
POTENTIALS
*  ipot  z  tag  lmax1  lmax2  xnatph  spinph
```

```
0   64   Gd   3   3   0.1    7.0   (c sublattice)   up
1   26   Fe   2   3   2      4.0   (a sublattice)   up
2   26   Fe   2   3   3     -4.0   (d sublattice)   down
3    8   O    2   3   12.0   0.0   (h sublattice)
4   64   Gd   3   3   2.9    7.0   (c sublattice)   up
```

2.4 XSPH: Cross-section and Phase Shifts

Relativistic dipole matrix elements (alpha form) are calculated using atomic core and normalized continuum wave functions. Polarization dependence is optionally incorporated in the dipole-operator. Scattering phase shifts are determined by matching at the muffin-tin radius.

FEFF is designed to calculate absorption from completely filled shells. You can try to simulate absorption from valence electrons with FEFF, but you may get unreliable results. If you encounter difficulties and need valence shell absorption, please contact the authors.

The ℓ-projected density of states used to be calculated in this module. This is now done in a separate part of the code, but the LDOS card is still described in this section for the sake of simplicity. This calculation is of limited quality due to finite cluster calculations and neglect of nonspherical corrections.

♦ **EXAFS** `[xkmax]` *Standard*

The EXAFS card is used to change the maximum value of k for EXAFS calculations. k is set by `xkmax`, and the default value is 20 Å^{-1}. The code can calculate to even higher values, however, the user may be prompted to increase dimensions in the file '`dim.h`' and recompile the code. For high k calculations it might be necessary to make smaller steps using the RGRID card.

```
*make EXAFS calculation through k=25 Angstroms^-1
EXAFS 25
```

♦ **XANES** `[xkmax xkstep estep]` *Standard*

The XANES card is used when a calculation of the near edge structure including the atomic background and absolute energies is desired. All parameters are optional, and are used to change the output energy mesh for the XANES calculation. XANES is not calculated unless the XANES card is present.

The XANES calculation is currently limited to the (extended) continuum spectrum beyond the Fermi level. Thus bound states are not generally included; however, in molecules weakly bound states that are below the vacuum but above the muffin-tin zero will show up as resonances. The absolute energies are based on atomic total energy calculations using the Dirac-Fock-Desclaux atom code. The accuracy of this approximation varies from a few eV at low Z to a few hundred eV for very large Z.

`xkmax`

The maximum k value of the XANES calculation. If FMS calculations are being made, note that these are not accurate beyond about $k = 6$; for larger values of k, e.g. $k = 20$ with the path expansion, FMS must be turned off.

`xkstep`
: This argument specifies the size of the output k grid far from the edge.

`estep`
: This argument specifies the energy step of the grid at the edge.

The default values are `xkmax` = 8, `xkstep` = 0.07, and `estep` = 0.0.

```
* finer grid for XANES calculation
XANES  6 0.05 0.3
```

♦ **ELLIPTICITY** `elpty x y z` *Useful*

This card is used with the POLARIZATION card (see below). The ellipticity `elpty` is the ratio of amplitudes of electric field in the two orthogonal directions of elliptically polarized light. Only the absolute value of the ratio is important for nonmagnetic materials. The present code can distinguish left- and right-circular polarization only with the XMCD or XNCD cards. A zero value of the ellipticity corresponds to linear polarization, and unity to circular polarization. The default value is zero.

`x`, `y`, `z` are coordinates of any nonzero vector in the direction of the incident beam. This vector should be approximately normal to the polarization vector.

```
* Average over linear polarization in the xy-plane
ELLIPTICITY  1.0  0.0  0.0  -2.0
```

♦ **LDOS** `emin emax eimag` *Useful*

To obtain the ℓDOS you need only run the **pot** module first to produce the file '`pot.bin`' (this can be accomplished using the CONTROL card from Section 2.2). The calculations are made in a separate program, which will run as long as the LDOS card appears in '`feff.inp`'.

The angular momentum projected density of states is placed by default on a standard grid currently fixed at 84 points. `emin` and `emax` are the minimum and maximum energies of the ℓDOS calculation and `eimag` is the imaginary part of potential used in the calculations. This is equivalent to Lorentzian broadening of the ℓDOS with half-width = `eimag`. If `eimag` is negative, the code automatically sets it to be 1/3 of the energy step. The output will be written in '`ldosNN.dat`' files. If 84 points are not enough, you can divide the energy range by 2 and run the code twice. The LDOS card is very useful when examining densities of states for interpreting XANES or when the self-consistency loop fails or gives very strange results. For crystals our ℓDOS will always be broadened due to the effect of finite cluster size.

```
*     emin emax eimag
LDOS  -20  20   0.2
```

♦ **MULTIPOLE** `le2 [l2lp]` *Useful*

Specifies which multipole transitions to include in the calculations. The options are: only dipole (`le2` $= 0$, default), dipole and magnetic dipole (`le2` $= 1$), dipole and quadrupole (`le2` $= 2$).

The additional field `l2lp` can be used to calculate individual dipolar contributions coming from $L \to L + 1$ (`l2lp` $= 1$) and from $L \to L - 1$ (`l2lp` $= -1$). Notice that in polarization dependent data there is also a cross term, which is calculated only when `l2lp` $= 0$.

```
MULTIPOLE  2   0  *combine dipole and quadrupole transitions.
MULTIPOLE  0  -1  *calculate dipolar L -> L-1 transitions
```

♦ **POLARIZATION** `x y z` *Useful*

This card specifies the direction of the electric field of the incident beam, or the main axis of the ellipse, in the case of elliptical polarization. `x`, `y`, `z` are the coordinates of the nonzero polarization vector. The ELLIPTICITY card is not needed for linear polarization. If the POLARIZATION card is omitted, spherically averaged XAFS will be calculated.

Note that polarization reduces the degeneracy of the paths, increasing the calculation time. Choosing polarization in the directions of symmetry axes will result in a faster calculation.

```
POLARIZATION  1.0  2.5  0.0
```

♦ **DANES** `[xkmax xkstep estep]` *Advanced*

To calculate the x-ray scattering amplitude f' instead of absorption f'', including solid state effects. Calculates the contribution from the specified edge and grid, which is specified as in the XANES card.

♦ **FPRIME** `emin emax estep` *Advanced*

To calculate ehe x-ray scattering factor f' far from the edge (only the atomic part). The energy grid is regular in energy with `estep` between `emin` and `emax`. This is typically needed to find the contributions from edges other than those calculated with the DANES card. Later it may be automated. The total scattering amplitude is

$$f'(Q, E) = f_0(Q) + f'(E) + if''(E)$$

In the dipole approximation f' and f'' do not depend on Q, but this is not true with quadrupole transitions added. This is currently neglected and

$$f'(E) = \text{DANES(edge)} + \text{FPRIME(all other edges)} + \text{(total energy term in fpf0.dat)}$$

$f_0(Q)$ is also tabulated in '`fpf0.dat`'; f'' is printed out by FPRIME and can be used to obtain the total f'. The total energy correction to f' is given in the first line of '`fpf0.dat`' in

Cromer-Liberman, and the more accurate Kissel-Pratt form. See our paper on elastic scattering amplitude for references and details.

♦ **RPHASES** *Advanced*

If this card is present, real phase shifts rather than complex phase shifts will be used. The results of the calculation will not be accurate. This option is intended to allow users to obtain real scattering phase shifts for use with other programs, or for diagnostic purposes. The phase shifts can be written to output files '`phaseNN.dat`' using the PRINT card. If the RPHASES card is present, these will contain the real phase shifts.

♦ **RSIGMA** *Advanced*

If this card is present, the imaginary part of the self-energy will be neglected. It might be useful for calculations in the XANES region, since the imaginary part of the Hedin-Lundqvist self-energy tends to overestimate losses in this region.

♦ **TDLDA** `ixfc` *Advanced*

Uses time-dependent local density approximation (TDLDA) theory to account for screening of the x-ray field and of the photoelectron–core-hole interaction. The parameter `ixfc` determines whether static or dynamic screening is used. `ixfc` = 0 (static screening) accounts for screening of the x-ray field (but not the field of the core-hole), blue-shifting the spectrum. Thus `TDLDA 0` works well at high energies. `ixfc` = 1 (dynamic screening) accounts for screening of the x-ray and core-hole fields, blue-shifting the spectrum less than `ixfc` = 0. The TDLDA card affects only module 2.

TDLDA theory takes into account polarizationtype many body effects (i.e., polarization of the electronic charge) which screen the local x-ray field. These effects are most important for xrays with energies less than 1 keV, hence `TDLDA 0` works well at high energies. The screened interaction is calculated partially based on the Bethe-Salpeter equation, in the basis of local atomic states. This approximation yields efficient calculations of the spectra in terms of screened transition matrix elements. Note that TDLDA does not account for core-hole relaxation effects.

L-shell x-ray absorption in 3d transition metals is sensitive to dynamic screening effects. For rare-gas solids, dynamic screening accounts for deviations of the L_3/L_2 intensity branching ratio from the 2:1 value of independent-electron theory.

See the paper on dynamic screening by Ankudinov, Nesvizhskii, and Rehr (see the refereces in Appendix C) for further details on the implementation of TDLDA in the FEFF calculations. See also A.I. Nesvizhskii's thesis, Ch. 8 (pp. 82-105, available on the FEFF webpage, see http://feff.phys.washington.edu/) for a brief description of the TDLDA theory.

```
* use static screening. this will only impact module 2, XSPH
TDLDA 0
```

♦ **XES** `emin emax estep` *Advanced*

To calculate nonresonant x-ray emission spectra (XES) for a specified grid. XES may be compared to the occupied DOS.

♦ **XMCD or XNCD** `[xkmax xkstep estep]` *Advanced*
Use either of the cards to calculate x-ray circular dichroism (the output will contain both magnetic and natural). The code calculates XMCD and XNCD from specified edge and grid, specified by auxiliary fields exactly as in the XANES card.

For nonmagnetic systems only XNCD will be present, while for magnetic materials with high symmetry only XMCD is present. Both will be present for magnetic materials with low symmetry, and x-ray direction (ELLIPTICITY card) must be used to disentangle the two contributions. The EXAFS region can also be used to determine the position of spins relative to magnetic field. The XMCD card has to be present in '`feff.inp`' for these calculations with FEFF8.40.

The XNCD originates from cross dipole-quadrupole contributions for certain nonmagnetic materials, such as special types of crystals. It will change sign for opposite direction of propagation (use the ELLIPTICITY card to do this). We performed calculations for $LiIO_3$ and found results very similar to previous multiple scattering XNCD calculations. XNCD requires that the XANES card also be used.

The XMCD (dipolar and quadrupolar) does not change sign under the change of direction of x-ray propagation, and is zero for nonmagnetic systems. The origin of the effect is that due to spin-orbit coupling, the right circular polarized light will produce more electrons with spin along or opposite to the direction of x-ray propagation. Thus it is important to use spin-dependent calculations for XMCD calculations. See the SPIN card and Section 4.4 for more details on spin-dependent calculations and an example for XMCD.

Note that the XMCD signal will only be contained in the output if the FEFF code has been compiled with `nspx` = `2`. For the (default) value `nspx` = `1`, you have to combine data from two '`xmu.dat`' files. A simple program to do this, '`spin.f`' is available on the FEFF web site, and printed in Section 4.4, where you can also find more details on the signal extraction process.

2.5 FMS: Full Multiple Scattering

This module carries out a full multiple scattering XANES calculation for a cluster centered on the absorbing atom. Thus all multiple-scattering paths within this cluster are summed to infinite order. This is useful for XANES calculations, but usually cannot be used for EXAFS analysis. FMS loses accuracy beyond $k = (l_{\rm max} + 1)/r_{\rm mt}$, which is typically about 4 $\mathrm{\AA}^{-1}$ since the muffin-tin radius $r_{\rm mt}$ is typically about 1 Å.

♦ **FMS** `rfms [lfms2 minv toler1 toler2 rdirec]` *Standard*
Compute full multiple scattering within a sphere of radius `rfms` centered on the absorbing atom. If the FMS card is not present in '`feff.inp`', the multiple scattering path expansion is used.

`rfms`

This is the cluster radius used in all modules but **pot**. Specifically, it is used for the ℓDOS and FMS, and as the lower limit of pathfinder calculations. The FMS module sums all MS paths within the specified cluster. Typically, a converged XANES calculation requires about 50-150 atoms in a cluster, but sometimes more are needed. The number of atoms in this cluster is limited to 175 by default, but one can manually change the dimension parameter `nclusx` in the ancillary FEFF8 source code file '`dim.h`' to alter the maximum cluster size. If there are more than `nclusx` atoms within the specified cluster size, FEFF will reduce the cluster size and issue a warning.

For EXAFS analysis, one typically calculates to $k = 20$, but FMS results are not accurate at high energies. Thus if you are running FEFF8 for EXAFS, you should not use the FMS card.

If the value of RPATH as described in Section 2.6 is greater than `rfms`, the pathfinder will look for paths which extend beyond the cluster used for the FMS and add them to the FMS calculation of the ℓDOS and XANES:

$$G_{\text{tot}} = G_{\texttt{fms}} + G_0 t_i G_0 + G_0 t_i G_0 t_j G_0 + \cdots$$

where at least one atoms i in the path is outside the FMS cluster and the value of RPATH is the maximum half path length for the LDOS, FMS and pathfinder modules. Note: this approximation may not be accurate and should be used with caution.

The MS expansion sometimes does not converge well in the XANES energy region. Thus one should avoid adding paths for LDOS and XANES, and RPATH should be less than `rfms`. Adding single scattering paths only (NLEG 2) usually works well to check the convergence of FMS. But adding double scattering (NLEG 3) often leads to very bad results in XANES. Thus RPATH is useful for EXAFS or for XANES only when the path expansion is stable.

`lfms2`

Optional argument. This is a logical flag that defines how the FMS is done, similar to the flag `lfms1` in the SCF card. With the default value of 0 (appropriate for solids), the FMS is calculated for a cluster of size `rfms` around each representative unique potential. With `lfms`=1 (appropriate for molecules), FMS is done only once for a cluster of size `rfms` around the absorbing atom only. The proper use of this flag can lead to a considerable time savings.

For example, if you calculate FMS for a molecule smaller than 40 atoms, there is no need to invert **nph** + 1 matrices, and `lfms1` = 1 will reduce time for calculations by a factor (**nph** + 1), where **nph** is the number of unique potentials listed in the POTENTIALS card).

Typically the FMS card will be used with `lfms2` = 0, for example:

```
* for XANES and LDOS need about 100 atom cluster
FMS  6.0
```

For molecules of less than 30 atoms of radius 4.0 Å we suggest using lfms2 = lfms1 = 1, as in:

```
SCF 5.0 1
FMS  5.0 1
RPATH  -1
```

minv

Optional. This is an index that defines the FMS algorithm used in the calculations. By default, (minv = 0) the FMS matrix inversion is performed using LU decomposition. However, several alternatives have been designed for the FMS algorithm that start to work faster than LU decomposition for clusters of more than 100 atoms. (See the FEFF8.2 reference). We strongly recommend the Lanczos recursion method (minv = 2) which is very robust and speeds the calculations by a factor of 3 or more. The Broyden algorithm (minv = 3) is faster, but less reliable, and may fail to converge if the FMS matrix has large eigenvalues.

toler1

Optional. This defines the tolerance to stop recursion and Broyden algorithm. The default value of 0.001 gives results in agreement with LU decomposition to within a linewidth.

toler2

Optional. Sets the matrix element of the Gt matrix to zero if its value is less than toler2 (default 0.001).

rdirec

Optional. Sets the matrix element of the Gt matrix to zero if the distance between atoms is larger than rdirec.

The last two variables can make the matrix Gt very sparse so both recursion and Broyden algorithms work faster. For example for large Si calculations with the Lanczos algorithm, we used:

```
FMS  29.4 0  2  0.001 0.001 40.0
```

♦ **DEBYE** temp thetad [idwopt] *Useful*

See the full description in Section 2.8 for details. The effect of temperature on FMS is approximated by multiplying each free propagator by $\exp(-\sigma^2 k^2)$, which gives correct DW factors for single scattering. The DW factors for multiple scattering are not exact, but their contribution is reduced both by thermal factors and by the mean free path. Also if you are running the **fms** module, then you can only obtain XANES, where this approximate treatment of thermal effects is probably adequate.

2.6 PATHS: Path Enumeration

The code uses a constructive algorithm with several path importance filters to explore all significant multiple-scattering paths in order of increasing path length. The paths are determined from the list of atomic coordinates in '`feff.inp`'. An efficient degeneracy checker is used to identify equivalent paths (based on similar geometry, path reversal symmetry, and space inversion symmetry). To avoid roundoff errors, the degeneracy checker is conservative and occasionally treats two degenerate paths as not degenerate. These errors occur in the third or fourth decimal place (less than 0.001 Ang) but are fail-safe; that is, no paths will be lost. All paths which are completely inside the FMS cluster are automatically excluded from the paths list, if specified by the FMS card.

The criteria used in filtering are based on increasingly accurate estimates of each path's amplitude. The earliest filters, the pathfinder heap and keep filters, are applied as the paths are being searched for. A plane wave filter, based on the plane wave approximation (plus a curved wave correction for multiple-scattering paths) and accurate to about 30%, is applied after the paths have been enumerated and sorted. Finally, an accurate curved wave filter is applied to all remaining paths.

♦ **RPATH** `rpath` *Standard*

The RPATH card determines the maximum effective (half-path) distance, **`rpath`**, of a given path. RPATH is equivalent to the RMAX card in the FEFF7 code. We changed the name because it provides a clearer distinction between the max distance in the MS path expansion and that for FMS calculations. Typically **`rpath`** is needed for EXAFS calculations only to set limits on the number of calculated paths. Note that **`rpath`** is one-half of the total path length in multiple-scattering paths. Setting this too large can cause the heap in the pathfinder to fill up. Default is **`rpath`** = 2.2 times the nearest neighbor distance. Since the multiple scattering expansion is unstable close to the absorption edge, the path (MS) expansion should be used only for EXAFS calculations or for diagnosing the XANES or ℓDOS calculations. If you use FMS for XANES calculations, better results are obtained without the MS contribution. For EXAFS analysis this card is extremely useful, since it cuts off long paths which contribute only at high R values in R-space.

```
* include MS paths with effective length up to 5.10 Ang
RPATH    5.10
```

♦ **NLEG** `nleg` *Useful*

The NLEG card limits the number of legs of each scattering path to **`nleg`**. If **`nleg`** is set to 2, only single scattering paths are found. The default is nleg = 8.

```
* only single scattering paths (i.e. 2 legged paths)
NLEG 2
```

♦ **PCRITERIA** `pcritk pcrith` *Advanced*

These criteria, like those described in the CRITERIA card, also limit the number of paths. However, they are applied in the pathfinder and eliminate unimportant paths while the pathfinder is doing its search. The pathfinder criteria do not know the degeneracy of a path and are therefore much less reliable than the curved wave and plane wave criteria in the CRITERIA card below. These pathfinder criteria (keep and heap) are turned off by default, and we recommend that they be used only with very large runs, and then with caution.

The keep-criterion `pcritk` looks at the amplitude of χ (in the plane wave approximation) for the current path and compares it to a single scattering path of the same effective length. To set this value, consider the maximum degeneracy you expect and divide your plane wave criterion by this number. For example, in fcc Cu, typical degeneracies are 196 for paths with large r, and the minimum degeneracy is 6. So a keep criterion of 0.08% is appropriate for a pw criteria of 2.5%.

The heap-criterion `pcrith` filters paths as the pathfinder puts all paths into a heap (a partially ordered data structure), then removes them in order of increasing total path length. Each path that is removed from the heap is modified and then considered again as part of the search algorithm. The heap filter is used to decide if a path has enough amplitude in it to be worth further consideration. If a path can be eliminated at this point, entire trees of derivative paths can be neglected, leading to enormous time savings. This test does not come into play until paths with at least 4 legs are being considered, so single scattering and triangular (2 and 3 legged) paths will always pass this test. Because only a small part of a path is used for this criterion, it is difficult to predict what appropriate values will be. To use this (it is only necessary if your heap is filling up, and if limiting rpath doesn't help), study the results in '`crit.dat`' from runs with shorter rpath and experiment with the heap criterion accordingly. In the future, we hope to improve this filter.

Before using these criteria, study the output in the file '`crit.dat`' (use print option 1 for paths, see Table 2.1), which has the values of `critpw`, keep criterion, and heap criterion for all paths which pass the `critpw` filter.

Default: If this card is omitted, the keep and heap criteria are set to zero, that is, no filtering will be done at this step in the calculation.

```
* fcc Cu had degeneracies from 6 to 196, so correct for this by
* dividing pw-crit of 2.5% by 30 to get 0.08 for keep crit. Check this
* empirically by running with pcrits turned off and studying crit.dat.
* After studying crit.dat, choose 0.5 for heap crit.
PCRITERIA   0.08  0.5
```

♦ **SS** `index ipot deg rss` *Advanced*

The SS card can *only* be used with the OVERLAP card when the atomic structure is unknown, but the distance and coordination numbers are known, and one wants to generate an approximate EXAFS contribution. Thus the pathfinder cannot be used in this case. Instead, the user has to specify explicitly the single scattering paths and their degeneracy. The OVERLAP card must be used to construct the potentials for use with the SS card. The parameters are:

`index`
: The shell index and label used for the '`feffNNNN.dat`' file name.

`ipot`
: The unique potential index identifying the unique potential of the scattering atom.

`deg`
: The degeneracy (or multiplicity) of the single scattering path.

`rss`
: The distance to the central atom.

This information is used to write the file '`paths.dat`' and is not needed when the ATOMS card is used. To generate single scattering paths with ATOMS, use NLEG 2.

```
 *  index  ipot   deg  rss    generate single scattering results
 SS   19     1     48  5.98       parameters for 19th shell of Cu
```

2.7 GENFMT: XAFS Parameters

For each path the code calculates, the effective scattering amplitude (F_{eff}, from which FEFF gets its name, see Section 3.2.4) and the total scattering phase shift along with other XAFS parameters using the scattering matrix algorithm of Rehr and Albers. Once the scattering phase shifts and the paths are determined, no other input is necessary for this calculation.

♦ **CRITERIA** `critcw critpw` *Useful*

Since the number of multiple scattering paths gets large very quickly, it is necessary to eliminate as many paths as possible. Fortunately, we have found that most multiple scattering paths have small amplitudes and can be neglected. Various cutoff criteria are used in FEFF8 to limit the number of paths to consider. These criteria are based on the importance of the path, defined as the integral over the full energy range of $\chi(k) \cdot$ `dk`. Very close to the edge these cutoff criteria should be examined with care and in some cases reduced from the values used for EXAFS.

`critcw`
: This is the cutoff for a full curved wave calculation. A typical curved wave calculation requires a complete spherical wave calculation, which typically takes seconds of CPU time per path. The default value of `critcw` is 4%, meaning that any path with mean amplitude exceeding 4% of the largest path will be used in the calculation of χ. The criterion `critcw` is used by **genfmt**. Since the XAFS parameter calculation is already done, the savings is not in computation time, but in disk space and ease of analysis.

The values of `critcw` for each path are written in the file '`list.dat`' written by module **genfmt**.

`critpw`
: This is a plane-wave approximation to χ. This is extremely fast to calculate, and is used in the pathfinder. The default value of critpw is 2.5, meaning that any path with mean amplitude exceeding 2.5% of the largest path, including degeneracy factors, (in plane wave approximation) will be kept. Any path that does not meet this criterion will not be written to '`paths.dat`', and there is no need to calculate the XAFS parameters for this path. The default for `critpw` is less than that for `critcw` since some paths are more important when the full curved wave calculation is done than they appear in the plane wave approximation. Since the plane wave estimate is extremely fast, use this to filter out as many paths as you can. The file '`crit.dat`' (written by the module **path**) tells you `critpw` for each path that passes the criterion.

The method of calculation of these importance factors has been improved for FEFF8, so don't worry if the values for some paths have changed slightly from previous versions. (Default values critcw=4% critpw=2.5%)

```
CRITERIA  6.0  3.0   * critcw 6%, critpw 3%
CRITERIA  0    0     * use all paths (cw and pw criteria turned off)
```

♦ IORDER `iord` *Advanced*

Order of the approximation used in module **genfmt**. FEFF uses order 2 by default, which is correct to terms of order $1/(pR)^2$, and corresponds to 6x6 scattering matrices in the Rehr–Albers formalism. Single scattering is calculated exactly to this order. The 6x6 approximation is accurate to within a few percent in every case we have tried (that is, higher order doesn't change the result more than a few percent). However M_{IV} shells and higher shells may require increased iorder for coupling the matrix elements. Changing the default values requires some familiarity with the Rehr–Albers paper and the structure of the module **genfmt**. To do so, follow the instructions in the FEFF source code in subroutine `setlam`. The key `iord` is passed to `setlam` for processing. You may need to change the code parameter `lamtot` if you want to do higher order calculations. For details of the algorithm used by **genfmt**, see the paper by J.J. Rehr and R.C. Albers (see the references in Appendix C). For the M_{IV} and higher edges, you may receive an error message like: `Lambda array overfilled`. In that case the calculations should be repeated with IORDER -70202 (10x10 matrices).

```
* change iorder for M4 calculations
IORDER -70202
```

♦ NSTAR *Advanced*

When this card is present, **genfmt** will write the file '`nstar.dat`' with the effective coordination number $N^{\star}$ which is the coordination number weighted by $\cos^2(\theta)$ to correct for polarization dependence in SEXAFS calculations.

2.8 FF2X: XAFS Spectrum

The module **ff2x** constructs the XAS spectrum $\chi(k)$ or μ using the XAFS parameters described in Section 3.2.4 from one or more paths, including any FMS contributions. Single and multiple scattering Debye–Waller factors are calculated using, for example, the correlated Debye model. Output from this module is the total XAFS spectrum and optionally, the contribution to the XAFS from each path individually. Numerous options for filtering, Debye–Waller factors, and other corrections are available.

♦ **DEBYE** `temperature Debye-temperature [idwopt]` *Standard*
The DEBYE card is used to calculate Debye–Waller factors for each path using the correlated Debye Model. The model is best suited for homogeneous systems, where it is quite accurate. CAUTION: in heterogeneous systems the model only gives approximate values which can easily be off by factors of two or more. If this card is present, the correlated Debye model Debye–Waller factors will be summed with the DW factors from the SIG2 card and from the '`list.dat`' file, if present. Note that the DEBYE card is currently incompatible with the CFAVERAGE card for options other than the correlated Debye model (`idwopt` > 0). Temperatures are specified in kelvin.

```
*Debye-Waller factors for Cu at 190K with correlated Debye Model
DEBYE  190 315
```

By default, `idwopt` $= 0$ specifies that the correlated Debye model is used to calculate EXAFS Debye–Waller factors. Two additional models for calculating DW factors are available in FEFF8 based on the information about the harmonic force constants in the material. `idwopt` $= 1$ means the equation of motion (EM) method is used to get Debye–Waller factors and `idwopt` $= 2$ means the recursion method (RM) which is an improved correlated Einstein model. Both methods are faster than molecular dynamics simulations, and the recursion method is much faster than the equation of motion method. However, the equation of motion method leads to somewhat more accurate results than the recursion method. These additional methods seem to be superior to the correlated Debye model in cases with tetrahedral coordination, such as solid Ge and many biological materials. Both EM and RM methods need additional input (the force constants) and a complete description of both is given in Anna Poiarkova's thesis (see the FEFF project web site, http://feff.phys.washington.edu/feff/) and in the associated documentation.

```
* Calculate Debye-Waller factors for Cu at 190K with equation of motion
DEBYE  190 0 1
```

♦ **CORRECTIONS** `vrcorr vicorr` *Useful*
The real energy shift `vrcorr` moves E_0 in the final $\chi(k)$ and the imaginary energy shift `vicorr` adds broadening to the result. The real energy shift is useful to correct the error in FEFF's

Fermi level estimate and the imaginary part can be used to correct for experimental resolution or errors in the core-hole lifetime. This error in the Fermi level is typically about 1 eV with self-consistent calculations and about 3 eV with overlapped atom potentials. The imaginary energy is typically used to correct for instrument broadening or as a correction to the mean free path calculated by FEFF. This affects only the module **ff2x**, which combines the results in all of the '`feffNNNN.dat`' files. This card is useful in fitting loops because you can simply make such energy corrections and see the results without redoing the entire XAFS parameter calculation. CAUTION: the results are not as accurate as those obtained with the EXCHANGE card. Both energies are in eV. (See also the EXCHANGE card in Section 2.3).

```
* Reduce E0 by 3.0 eV and add 1 eV of broadening (full width)
* This will only affect module ff2x
CORRECTIONS   3.0   1.0       real shift, imag shift
```

♦ **SIG2** `sig2` *Useful*

Specify a global Debye–Waller factor to be used or added to Debye–Waller calculations (see the DEBYE card) for all paths. This value will be summed with the correlated Debye model value (if the DEBYE card is present) and any value added to '`list.dat`'. Units are Å^2. This card can be used, for example to add Debye–Waller factors from structural disorder.

```
SIG2 0.001    add 0.001 globally to all DW factors
```

Chapter 3

Input and Output Files

Other files required by the various modules are created by FEFF from '`feff.inp`'. Some of these other files may be edited by the user as a way to modify the input data to the modules, see Section 3.3. See the PRINT card in Section 2.2 to obtain various diagnostic files. Section 3.1 summarizes this structure, the rest of this section describes the structure in more detail.

3.1 Module Input and Output Files

Module 0 .... **rdinp**

Purpose of Module: Read input data

Input files: '`feff.inp`'

Output files: '`geom.dat`', '`global.dat`', and '`modN.inp`' (N=1-6)

Other output: '`paths.dat`' (only if the SS card is used)

Description: Reads the '`feff.inp`' file, makes appropriate operations on the data, and writes the resulting information into several output files, which contain formatted data needed for all modules.

Module 1 .... **pot**

Purpose of Module: Calculate embedded atomic potentials for the photoelectron

Input files: '`mod1.inp`' and '`geom.dat`'

Output files: '`pot.bin`'

Other output: diagnostic files (see Table 2.1 on page 10)

Description: Reads '`mod1.inp`' and calculates potentials for the photoelectron, which are written into '`pot.bin`'. Optionally, **pot** will write other diagnostic files with information about the potentials.

Module 2 .... **xsph**

Purpose of Module: Calculate cross-section and phase shifts

Input files: '`mod2.inp`', '`geom.dat`', '`global.dat`' and '`pot.bin`'

Output files: '`phase.bin`', and '`xsect.bin`',

Other output: diagnostic files (see Table 2.1 on page 10), '`axafs.dat`', and '`ldosNN.dat`' (ℓDOS)

Description: **xsph** writes the binary file '`phase.bin`', which contains the scattering phase shifts and other information needed by **path** and **genfmt**. The atomic cross-section data is written in '`xsect.bin`' and used in the final module (**FF2**) for overall normalization. Optionally, **xsph** will write other diagnostic files with information about the phase shift calculations.

Module 3 .... fms

Purpose of Module: Calculate full multiple scattering for XANES and ℓDOS

Input files: '`mod3.inp`', '`global.dat`', '`geom.dat`', and '`phase.bin`',

Output files: '`fms.bin`'

Other output:

Description: Performs the full multiple scattering algorithm. Writes output into '`fms.bin`' for the **ff2x** module, which contains the $\chi(k)$ from FMS.

Module 4 .... path

Purpose of Module: Path enumeration

Input files: '`mod4.inp`', '`geom.dat`', '`global.dat`' and '`phase.bin`'

Output files: '`paths.dat`'

Other output: '`crit.dat`'

Description: **path** writes '`paths.dat`' for use by **genfmt** and as a complete description of each path for use of the user. **path** will optionally write other diagnostic files. The file '`crit.dat`' is particularly useful when studying large numbers of paths. When studying large numbers of paths, this module will optionally write only '`crit.dat`' and not '`paths.dat`'.

Module 5 .... genfmt

Purpose of Module: Calculate scattering amplitudes and other XAFS parameters

Input files: '`mod5.inp`', '`global.dat`', '`phase.bin`', and '`paths.dat`'

Output files: '`feff.bin`', and '`list.dat`'

Other output:

Description: **genfmt** reads input files, and writes a file '`feff.bin`', which contains all the EXAFS information for the paths, and '`list.dat`', which contains some basic information about them. These files are the main output of FEFF for EXAFS analysis. To read '`feff.bin`' into your own program, use the subroutine feffdt as an example.

Module 6 .... **ff2x**

Purpose of Module: Calculate specified x-ray spectrum

Input files: '`mod6.inp`', '`global.dat`', '`list.dat`', '`feff.bin`', '`fms.bin`', '`xsect.bin`'

Output files: '`chi.dat`' and '`xmu.dat`'

Other output: '`chipNNNN.dat`' and '`feffNNNN.dat`'

Description: **ff2x** reads '`list.dat`', '`fms.bin`', '`feff.bin`', and writes '`chi.dat`' with the total XAFS from the paths specified in '`list.dat`'. Additional instructions are passed to **ff2x** from '`feff.bin`', so you can change S02, the Debye temperature and some other parameters without re-doing the whole calculation. The file '`list.dat`' can be edited by hand to change the paths being considered, and individual '`chipNNNN.dat`' files with $\chi(k)$ from each path are optionally written. If any of the XANES, DANES, FPRIME or XNCD cards are specified, **ff2x** will write the corresponding calculated data in '`xmu.dat`'. Various corrections are possible at this point in the calculations—see the input cards above.

There is an internal limit on the number of paths (set to 1200) that will be read from '`feff.bin`'. This limit was chosen to handle any reasonable problem without using an excessive amount of memory. If you must use more paths, change the parameter **`npx`** in the FEFF source in subroutine `ff2chi` to whatever you need. This will require more memory. We have not had a case where the filter criteria were not able to solve the problem with fewer than 1200 paths.

3.2 Descriptions of Output Files

3.2.1 Intermediate Output Files

'`modN.inp`' and '`ldos.inp`'

These ASCII files contain basic information from '`feff.inp`' for a particular module. They can still be edited, for example to take advantage of symmetries.

'`global.dat`'

This ASCII file contains global information about x-ray polarization and about configurational averaging.

'`geom.dat`'

This ASCII file contains Cartesian coordinates of all atoms and first-bounce information for the degeneracy reduction in the pathfinder.

'`pot.bin`'

Charge density and potential (SCF or not) for all types of atoms. This file is used by the **xsph** module.

'`phase.bin`'

This is a binary file with the scattering phase shifts for each unique potential and with relativistic dipole matrix elements, normalized to total cross section in '`xsect.bin`'. It is used by the **fms**, **path** and **genfmt** modules.

'`xsect.bin`'
: Total atomic cross section for x-ray absorption. This is an ASCII file, but it is highly sensitive to format. The information it contains can be viewed, but editing this file is *not* recommended.

'`ldosNN.dat`'
: ℓ-projected density of states for the NN$^{\mathrm{th}}$ potential index (see the LDOS card)

'`fms.bin`'
: contains the results of FMS calculations. Used by **ff2x** to get the total XAFS or XANES.

'`paths.dat`'
: Written by the pathfinder, this is a description of all the paths that fit the criteria used by the pathfinder. It is used by **genfmt**. The path descriptions include Cartesian coordinates of atoms in the path, scattering angles, leg lengths and degeneracy. For details on editing this file by hand, see Section 3.3. ('`pathNN.dat`' files are also created during the ℓDOS calculations for each type of potential, but they are deleted after use.)

'`crit.dat`'
: Values of the quantities tested against the various criteria in the pathfinder.

'`list.dat`'
: List of paths to use for the final calculations. Written by **genfmt** when the XAFS parameters are calculated and used by **ff2x**. It contains the curved wave importance ratios, which you may wish to study. For details on editing this file by hand, see Section 3.3.

The curved wave importance ratios are the importance of a particular path relative to the shortest single scattering path.

3.2.2 Diagnostic Files

'`misc.dat`'
: Header file for quick reference.

'`phaseNN.dat`'
: Complex phase shifts for each shell.

'`phminNN.dat`'
: Real part of phase shifts for ℓ=0,1,2 only. They are smaller versions of corresponding '`phaseNN.dat`'.

'`potNN.dat`'
: Detailed atomic potentials and densities.

'`atomNN.dat`'
: Diagnostic information on Desclaux free atom NN.

3.2.3 Main Output Data

'chi.dat'
Standard XAFS data containing k, $\chi(k)$, $|\chi(k)|$ relative to threshold ($k = 0$). The header also contains enough information to specify which model was used to create this file.

'xmu.dat'
The file '`xmu.dat`' contains both XANES and XAFS data μ, μ_0, and $\tilde{\chi} = \chi \frac{\mu_0}{\mu_0(\text{edge+50eV})}$ as functions of absolute energy E, relative energy $E - E_f$ and wave number k.

'feff.bin'
A binary file that contains all the information about the XAFS from all of the paths. This replaces the old '`feffNNNN.dat`' files (which you can make using the PRINT card). If you want to use this file with your own analysis package, use the code in subroutine feffdt as an example of how to read it.

'feffNNNN.dat'
You have to use the PRINT option to obtain these files. Effective scattering amplitude and phase shift data, with k referenced to threshold for shell nn: k, ϕ_c, $|F_{\text{eff}}|$, ϕ_{eff}, the reduction factor, λ, $\text{Re}(p)$.

If you need these, use the PRINT option for **ff2x** greater than or equal to 3, which will read '`feff.bin`' and write the '`feffNNNN.dat`' files in exactly the form you're used to.

'fpf0.dat'
Thomson scattering amplitude $f_0(Q)$ and constant contribution to f' from total energy term.

'ratio.dat'
Ratio $\mu_0(E)$, $\rho_0(E)$ and their ratio versus energy, for XMCD sum rules normalization.

3.2.4 Variables in the EXAFS and XANES Formulae

k — The wave number in units of Å^{-1}. $k = \sqrt{E - E_f}$ where E is energy and E_f is the Fermi level computed from electron gas theory at the average interstitial charge density.

$\chi(k)$

$$\chi(k) = S_0^2 \mathcal{R} \sum_{\text{shells}} \frac{N F_{\text{eff}}}{k R^2} \exp(-2r/\lambda) \sin(2kR + \phi_{\text{eff}} + \phi_c) \exp(-2k^2\sigma^2)$$

ϕ_c — The total central atom phase shift, $\phi_c = 2\delta_{\ell,c} - \ell\pi$

F_{eff} — The effective curved-wave backscattering amplitude in the EXAFS formula for each shell.

ϕ_{eff} — The phase shift for each shell

$\mathcal{R}$ — The total central atom loss factor, $\mathcal{R} = \exp(-2\,\text{Im}(\delta_c))$

R	The distance to the central atom for each shell
N	The mean number atoms in each shell
σ^2	The mean square fluctuation in R for each shell
λ	The mean free path in Å, $\lambda = 1/\lvert \operatorname{Im} p \rvert$
k_f	The Fermi momentum at the average interstitial charge density
$p(r)$	The local momentum, $p^2(r) = k^2 + k_f^2(r) + \Sigma - \Sigma_f$
$\Sigma(E)$	The energy dependent self energy at energy E, Σ_f is the self energy at the Fermi energy.
$\mu(E)$	The total absorption cross-section
$\mu_0(E)$	The embedded atomic background absorption

3.3 Program Control Using Intermediate Output Files

In addition to the CONTROL card and other options in '`feff.inp`', some parameters in the files read by the various modules can be changed. For example, you can create your own paths by editing '`paths.dat`' and explicitly change Debye–Waller factors in the final result by editing '`list.dat`'.

Users may edit the some files as a quick and sometimes convenient way to prepare a given run. It is easiest to use an existing file as a template, since the code that reads these files is fussy about their format.

3.3.1 Using '`paths.dat`'

You can modify a path, or even invent new ones, such as paths with more than the pathfinder maximum of 8 legs. For example, you could make a path to determine the effect of a focusing atom on a distant scatterer. Whatever index you enter for the path will be used in the filename given to the '`feffNNNN.dat`' file. For example, for the choice of index 845, the EXAFS parameters will appear in '`feff0845.dat`'. A handy way to add a single scattering path of length R is to make a 2-leg path with the central atom at (0, 0, 0) and the scatterer at (R, 0, 0).

genfmt will need the positions, unique potentials, and character tags for each atom in the path. The angles and leg lengths are printed out for your information, and you can omit them when creating your own paths by hand. The label lines in the file are required (there is code that skips them, and if they're missing, you'll get incorrect results).

3.3.2 Using 'list.dat'

This is the list of files that **ff2x** uses to calculate chi. It includes the paths written by module **genfmt**, curved wave importance factors, and user-defined Debye–Waller factors. If you want to set Debye–Waller factors for individual paths, you may edit this file to set them. **ff2x** will sum the Debye–Waller factors in this file with the correlated Debye model σ^2 and the global σ^2, if present. You may also delete paths from this file if you want to combine some particular set of paths. (CAUTION: Save the original, or you'll have to re-run **genfmt**!)

3.3.3 Using 'geom.dat'

This file can be manually edited to take advantage of the paths symmetries.

Chapter 4

Calculation Strategies and Examples

4.1 General Comments

Self-consistent or overlapped atom potentials are necessary for the calculation of the scattering phase shifts. Self-consistent calculations take more time, and are often essential for XANES, especially for cases with significant charge transfer. Although the effect of self-consistency on EXAFS is small, such calculations give an accurate determination of E_0, thus eliminating an important parameter in EXAFS distance determinations.

Scattering phase shifts for each unique potential are necessary for **fms**, **path** and **genfmt**. They are needed for the importance filters in **path** and are the basis of the XAFS parameters calculation in **genfmt**. This part of the calculation is relatively slow, so it is usually best to run it only once and use the results while studying the paths and XAFS.

To enumerate the necessary paths, the pathfinder module **path** needs the atomic positions of any atoms from which scattering is expected. If the structure is completely unknown, only single-scattering paths can be created explicitly. Because the number of possible paths increases exponentially with total path length, one should start with a short total path length, examine the few paths (representing scattering from the nearest neighbors), and gradually increase the total path length, possibly studying the path importance coefficients and using the filters to limit the number of paths. This process is not automated, and if done carelessly can yield so many paths that no analysis will be possible.

Finally, use **genfmt** to calculate the XAFS parameters, and **ff2x** to assemble the results into a chi curve. Here, the slow part is **genfmt** and **ff2x** is very fast. Therefore, to explore parameters such as Debye–Waller factors, mean free path and energy zero shifts, various combinations of paths and coordination numbers, run only module **ff2x** using the results saved from **genfmt**.

There are three ways to modify the Debye–Waller factor, all of which affect only the module **ff2x**. The DEBYE card calculates a Debye–Waller factor for each path. The SIG2 card adds a constant Debye–Waller factor to each path. And you can edit '`list.dat`' to add a particular Debye–Waller factor to a particular path. These three Debye–Waller factors are summed, so if the DEBYE and SIG2 cards are present, and if you have added a Debye–Waller factor to a particular path, the Debye–Waller factor used will be the sum of all three. See documentation below for details.

If your model changes significantly, the phase shifts (which are based in part on the structure of the material) should be recalculated. Any time the phase shifts change, the XAFS parameters will also have to be re-calculated. If the path filters have been used, the path list will also have to be recomputed.

4.2 EXAFS Calculation

4.2.1 SF_6 Molecule

SF_6 Molecule. This is the simplest example of running FEFF to obtain results for EXAFS. Only two input cards are necessary. Only the file '`chi.dat`' is produced.

```
TITLE Molecular SF6

POTENTIALS
* absorbing atom must be unique pot 0
*    ipot    z   tag
       0    16   S
       1     9   F

ATOMS
*  x      y      z     ipot
   0      0      0       0          S absorber
   1.56   0      0       1          6 F backscatters
   0      1.56   0       1
   0      0      1.56    1
  -1.56   0      0       1
   0     -1.56   0       1
   0      0     -1.56    1
```

4.2.2 Solids

Cu metal

Cu, fcc metal, 4 shells. The list of atomic coordinates (ATOMS card) for crystals can be produced by the program ATOMS. Thus instead of giving a long atoms list, we present a short '`atoms.inp`' file. For connection with EXAFS fitting programs see Section 3 and the PRINT card on page 9.

```
TITLE Cu crystal, 4 shells
```

```
* Cu is fcc, lattice parameter a=3.61 (Kittel)

*Cu at 190K, Debye temp 315K (Ashcroft & Mermin)
DEBYE  190  315 0

POTENTIALS
  0  29  Cu0
  1  29  Cu

ATOMS
atoms list generated using atoms.inp file below
--------------------------------------------
title Cu  metal  fcc a=3.6032
fcc                   ! shorthand for F M 3 M
rmax= 11.13   a=3.6032
out=feff.inp          ! index=true
geom = true
atom
!   At.type    x    y    z
Cu        0.0  0.0  0.0
--------------------------------------------
```

YBCO High-Tc superconductor

```
TITLE  YBCO: Y Ba2 Cu3 O7      Cu2 core hole

CONTROL  1  1  1  1  1  1
PRINT    0  0  0  0  0  0

RPATH   4.5

POTENTIALS
*    ipot  z  tag
      0   29  Cu2
      1    8  O
      2   39  Y
      3   29  Cu1
      4   56  Ba

ATOMS
atoms list generated by the following atoms.inp file
-----------------------------------
title YBCO: Y Ba2 Cu3 O7  (1-2-3 structure)
space P M M M
```

```
rmax=5.2          a=3.823  b=3.886  c=11.681
core = Cu1
atom
! At.type  x     y     z       tag
  Y       0.5   0.5   0.5
  Ba      0.5   0.5   0.184
  Cu      0     0     0       Cu1
  Cu      0     0     0.356   Cu2
  O       0     0.5   0       O1
  O       0     0     0.158   O2
  O       0     0.5   0.379   O3
  O       0.5   0     0.377   O4
--------------------------------------
```

4.2.3 Estimate of S_0^2

All above examples yield calculations for the K edge (default). To do calculations for other edges, use the EDGE or HOLE cards. These cards will also yield an estimate of S_0^2 from atomic calculations if you set `S02` < 0.1 by the one of three possible ways shown below.

```
EDGE   L3    0.0
HOLE    4   0.0
S02     0.0
```

The result for S02 is given in '`chi.dat`' or '`xmu.dat`' files. S_0^2 is a square of determinant of overlap integrals for core orbitals calculated with and without core hole. The core-valence separation can be changed by editing the subroutine `getorb`, but it is currently set by default to the most chemically reasonable one.

4.2.4 Configuration Averaging Over Absorbers

In amorphous materials or materials with distortions from regular crystals, the absorbing atoms (with the same number in the periodic table) may have different surroundings. Thus one may want to average the calculation over different types of sites for the same atom or even over all atoms in the '`feff.inp`' file. This can be accomplished using CFAVERAGE card of Section 2.2. The usefulness of this type of calculation is currently curtailed by the limited functionality of the CFAVERAGE card, which should be used with caution.

4.2.5 Adding Self-consistency

Self-consistency is expected to be more important for XANES calculations, but even for EXAFS one may want to have a more reliable determination of Fermi level or to account for charge transfers in order to do fits with a single energy shift E_0. Our experience shows that reliable EXAFS phase shifts are best achieved using the SCF card.

```
*calculate EXAFS with SCF potentials and paths to R=6 angstroms
EXAFS
SCF   3.8
RPATH 6.0
```

The above example works for solids or large molecules, but for molecules with less than 30 atoms, calculations can be done faster if you set `lfms1` = 1:

```
SCF   10.0  1
```

For details see the SCF and FMS cards in Sections 2.3 and 2.5.

4.3 XANES Calculations

4.3.1 Need for SCF and Additional Difficulties for XANES

XANES calculations are usually more challenging than EXAFS calculations. They usually take more time and require more experience from the user. Fortunately FEFF8 automates many steps in the procedure and includes important full multiple scattering terms and self-consistency. This improves on the high order path expansion approach to XANES in FEFF7 which only allowed the maximum number of paths of amplitude larger than a plane wave criteria in PCRITERIA card. To account for the poor electron gas estimates of the Fermi level, the CORRECTIONS card was needed. Moreover, to obtain good results for the spectra, one had also to play with AFOLP, EXCHANGE and ION cards, which alter the way the scattering potential is constructed in somewhat uncontrolled ways.

One of the main advantages of FEFF8 is that it yields self-consistent potentials using the SCF card. The use of the SCF card also gives a more reliable estimate of the Fermi level (the CORRECTIONS card can still be used, since the error in Fermi level position is only a few eV). FEFF8 thus automatically accounts for charge transfer. The ION card should be used only to specify the total charge of a cluster. AFOLP in general leads to better results for XANES and is done by default. With FEFF7 we also had to use an EXCHANGE 5 model for Pu hydrates, but with the self-consistent FEFF8 the standard EXCHANGE 0 works well. Thus the use of self-consistency leads to closer results for different exchange correlation models.

The use of the high order MS path expansion and PCRITERIA can lead to unreliable XANES calculations when the MS series converges poorly (for example, near the Fermi level). Thus the inclusion of FMS capabilities in FEFF8 is essential for calculations of ℓDOS and electronic densities and is often an improvement on path calculations of XANES. We suggest that for ℓDOS calculations one uses FMS exclusively and uses path expansion for testing its convergence. This will cost CPU time, but will lead to more reliable results. The FMS calculations for a cluster of 175 atoms typically take more time and memory than the other 5 modules. The results can be somewhat better with larger clusters, but typically one achieves convergence with about 50-200 atoms and calculation time scales as a third power of number of atoms in a cluster and quickly becomes prohibitive. Below we present several sample input files for XANES calculations.

4.3.2 $GeCl_4$ Molecule

This is historically the first molecule for which EXAFS was calculated by Hartree, Kronig and Peterson (1934), using short range order theory.

```
TITLE   GeCl_4  r=2.09 /AA

NOHOLE
HOLE 1   1.0
RSIGMA

CONTROL   1  1  1  1  1  1

SCF    3.0  1
FMS    3.0  1
RPATH  1.0
XANES  8.0 0.05

AFOLP  1.30

POTENTIALS
*   ipot   z  label
      0   32   Ge  3 3
      1   17   Cl  3 3
ATOMS
*     x          y          z     ipot atom           distance
   0.0000     0.0000     0.0000     0   Ge
   1.2100     1.2100     1.2100     1   Cl
   1.2100    -1.2100    -1.2100     1   Cl
  -1.2100     1.2100    -1.2100     1   Cl
  -1.2100    -1.2100     1.2100     1   Cl
END
```

4.3.3 Solid: XANES and ℓDOS

BN crystal has a zinc sulfide structure, and is a case in which a multiple scattering expansion does not converge near the Fermi level. Using the full multiple scattering approach leads to good agreement with experiment.

```
TITLE   BN cubic zinc sulfide structure
CONTROL   1 1 1 1 1 1
PRINT     5 0 0 0 0 0

SCF  3.1
```

```
HOLE 1   1.0     1=k edge, s0^2=1.0
EXCHANGE  0  0  1.0
LDOS -20  10  0.5
FMS  5.1
RPATH   1.0
XANES 4.0

INTERSTITIAL 0  1.54

POTENTIALS
*   ipot   z  label lmax1  lmax2
      0    5   B     2    2  0.1
      1    7   N     2    2  1
      2    5   B     2    2  1

ATOMS
list generated by ATOMS program
-------------------------------------
title BN (zincblende structure)
Space  zns
a=3.615 rmax=8.0   core=B
atom
! At.type  x        y       z      tag
   B      0.0      0.0     0.0
   N      0.25     0.25    0.25
----------------------------------------
```

4.3.4 Absolute Cross-section

The absolute cross section can be obtained from the output in '`xmu.dat`'. Look for this line:

```
xsedge+100, used to normalize mu          2.5908E-04
```

Since our distances are in Å, we report cross section also in $Å^2$. If you multiply the 4-th or 5-th column by this normalization value you will obtain the cross section in $Å^2$. Often the absolute cross section is reported in barns, which have a simple connection with our units (1 $Å^2$ = 100Mbarn)

4.4 Spin-dependent Calculations

4.4.1 General Description

This section contains information on extracting the XMCD signal and a description of the SPXAS technique, as well as example input files.

Spin-dependent calculations have been automated in FEFF8.40. All spin-dependent calculations require that the SPIN card be present in '`feff.inp`' (see the SPIN card in Section 2.3 for additional details on its use). The method one uses to extract the signal depends on the value of the parameter **nspx**, which is found in the header file '`dim.h`'.

In order for the final result to be contained in '`feff.inp`', FEFF must be compiled with **nspx** = 2 to take care of both the spin-up and spin-down calculations. This will also add the contribution from spin-flip processes (which we find typically very small), but may require up to 4 times the memory and 8 times the execution time for the XANES region. Thus, in general, **nspx** = 2 will not always be feasible.

With **nspx** = 1, the code must be run twice, once for spin-up and once for spin-down. An additional simple program '`spin.f`' is useful for taking care of the different normalizations and giving the final results in XMCD or SPXAS (Section 4.4.3). '`spin.f`' is printed below, and may also be found on the FEFF website (http://feff.phys.washington.edu/feff/).

In the monolithic source code, all instances of **nspx** = 1 must be changed to **nspx** = 2 in order to automate the calculations. In the source tree, '**dim.h**' is located in '**src/HEADERS/**', and **nspx** can be changed there. Please contact the authors if you need help modifying the source code.

The same paths should be used for spin-up and spin-down calculations, otherwise the difference between 2 calculations may be due to different paths used. Typically the paths list in '`paths.dat`' should be generated by running the usual EXAFS calculations and comparing with experiment (to make sure that all important paths are included). Then, when running with SPIN, turn off the pathfinder module using the CONTROL card.

'`spin.f`':

```
      implicit double precision (a-h,o-z)
c     This program read two xmu.dat files for spin -up and -down,
c     calculated with Feff8.20 for the SAME paths list.
c     spin-up file is fort.1, spin-down file is fort.2
c     Both have to be edited: All lines should be deleted except
c       1) line: xsedge+100, used to normalize mu          1.3953E-04
c          leave only on this line:  1.3953E-04
c       2) 6-column data lines
c     The output will be written in fort.3 in 6 columns
c     E+shift1  E(edge)+shift2  xk cmd_total cmd_background  cmd_fs
c     where total = atomic background + fine structure

c     There are 3 possibilities
c     case 1) you want XMCD signal and used SPIN \pm 1
c     case 2) you want XMCD signal and used SPIN \pm 2, in order
c       to use non-relativistic formula for XMCD
c       factor li/2j+1 which was not convenient to do in a program
c     case 3) you want SPXAFS and used  SPIN \pm 2
```

```
c     ENTER your case here (icase is positive integer only)
      icase = 2

c     if icase=2 ENTER factor=(-1)**(L+1/2-J) * L/(2*J+1)
c     where L,J are for your edge (ex. for L3 L=1 J=3/2, for L2 L=1 J=1/2)
c     for L3
c      factor = 0.25
c     for L2
c     factor = -0.5

c     ENTER the energy shift you want for columns 1 and 2 in xmu.dat
      shift1 = 0
      shift2 = 0

c     everything below is automated further
      read (1,*,end=10) ap
      read (2,*,end=10) am
      xnorm = 0.5 *(ap+am)
c     read the data
   3     read(1,*,end=10)   x1, x2, ek, y1, y2, y3
         read(2,*,end=10)  x1, x2, ek, z1, z2, z3
         if (icase.eq.1) then
c           no xafs in this case:xfs - atomic part of XMCD
            t1 = (y1*ap + z1*am)/xnorm
            t2 = (y2*ap + z2*am)/xnorm
            t3 = (y3*ap + z3*am) /xnorm

         elseif (icase.eq.2) then
            t1 = (y1*ap - z1*am)*factor /xnorm
            t2 = (y2*ap - z2*am)*factor /xnorm
            t3 = (y3*ap - z3*am)*factor /xnorm

         elseif (icase.eq.3) then
c           factor=0.5 always for SPXAFS
            t1 = (y1*ap - z1*am)/2.0/xnorm
            t2 = (y2*ap - z2*am)/2.0/xnorm
            t3 = (y3*ap - z3*am)/2.0/xnorm
c           you may want average total XAS as output in last column
c           t3 = (y1*ap + z1*am)/2.0/xnorm
         endif
         x1 =x1 + shift1
         x2 =x2 + shift2
         write(3,5)    x1, x2, ek, t1, t2, t3
   5     format (6e13.5)
```

```
    goto 3
10  continue
    stop
    end
```

4.4.2 XMCD for the Gd L1 edge

Example input file for calculation of XMCD for the Gd L1 edge. Note the presence of both the SPIN and XMCD cards.

```
TITLE   Gd  l1  hcp

HOLE 2   1.0     2=l1 edge, s0^2=1.0
SPIN    1
EXCHANGE  2   0.0  0.0
RGRID .01

CONTROL   1      1     1     1    1    1

RPATH    7.29
PRINT   5 0 0 0 0 3

CRITERIA   0.0  0.0  curved   plane
DEBYE   150    176   temp     debye-temp
XANES
XMCD

POTENTIALS
*   ipot   z  label
      0   64   Gd
      1   64   Gd

ATOMS
the list of atoms is created by ATOMS program
----------------------------------------------
title    Gd , hcp
! Wycoff, vol.1 p.331
space  hcp
rmax = 9.0
a = 3.6354
c = 5.7817
atom
  Gd   0.33333   0.66667   0.25   center
-----------------------------------
```

4.4.3 SPXAS

For antiferromagnets, the XMCD should be zero, and a measure of the spin-up and spin-down signals can be accomplished using SPXAS. SPXAS is a technique where you measure the spin-up and spin-down signal by measuring the intensity of two spin-split K_β lines. This corresponds to measuring spin-order relative to the spin on the absorber (not relative to the external magnetic field as in XMCD). As an example, let us look at the Mn K edge of antiferromagnetic MnF_2. Our calculations agree well with experiment in the EXAFS region.

Here is an example input file for MnF_2:

```
TITLE   MnF2 (rutile) cassiterite (Wykoff)

HOLE 1   1.0     1=k edge, s0^2=1.0
SPIN   2

CONTROL   1      1     1     1     1     1
PRINT     0      0     0     2     0     4
EXCHANGE  0  0.0  0.0
CORRECTIONS  0.0   0.0

RPATH  10.0
XANES

PCRITERIA       0.8     40.0
*CRITERIA     curved   plane
CRITERIA       0.0     0.0
*DEBYE        temp     debye-temp
DEBYE        300       350
NLEG         4

POTENTIALS
*   ipot   z  label
      0   25   Mnup
      1    9   F
      2   25   Mnup
      3   25   Mndown

ATOMS
  0.0000     0.0000     0.0000    0   Mnup             0.0000
  1.4864     1.4864     0.0000    1   F                2.1021
 -1.4864    -1.4864     0.0000    1   F                2.1021
  0.9503    -0.9503     1.6550    1   F                2.1319
  0.9503    -0.9503    -1.6550    1   F                2.1319
 -0.9503     0.9503     1.6550    1   F                2.1319
```

```
 -0.9503     0.9503    -1.6550    1    F                2.1319
  0.0000     0.0000    -3.3099    3    Mndown           3.3099
  0.0000     0.0000     3.3099    3    Mndown           3.3099
 -3.3870     1.4864     0.0000    1    F                3.6988
  3.3870    -1.4864     0.0000    1    F                3.6988
  1.4864    -3.3870     0.0000    1    F                3.6988
 -1.4864     3.3870     0.0000    1    F                3.6988
  2.4367     2.4367    -1.6550    2    Mnup             3.8228
 -2.4367    -2.4367    -1.6550    2    Mnup             3.8228
  2.4367    -2.4367    -1.6550    3    Mndown           3.8228
 -2.4367    -2.4367     1.6550    3    Mndown           3.8228
 -2.4367     2.4367    -1.6550    3    Mndown           3.8228
  2.4367     2.4367     1.6550    3    Mndown           3.8228
  2.4367    -2.4367     1.6550    2    Mnup             3.8228
 -2.4367     2.4367     1.6550    2    Mnup             3.8228
...
END
```

4.5 Elastic Scattering Amplitudes

All necessary components to obtain the elastic scattering amplitude can be calculated with the FEFF8.2 code and higher. The Thomson scattering amplitudes are written in the file '`fpf0.dat`'. The elastic amplitudes near a specific edge are calculated with the DANES card, while those far from the edge are calculated with the FPRIME card, which neglects solid state effects on f'. f'' can be obtained with the XANES card. The formula connecting f'' and the absorption cross section σ is (in atomic units) $f'' = \omega c \sigma / 4/\pi$. For calculations at energies well above the absorption edge, we found that ground state potentials yield better results and that quadrupolar transitions have to be included.

4.6 X-ray Emission Spectra XES

Nonresonant x-ray emission spectra (fluorescence spectra) are treated in the same way as the x-ray absorption process for states below the Fermi level. To perform these calculations one simply replaces the XANES card with XES. Preliminary comparisons with experiment for the phosphorous K_β line show good agreement with experiment for various compounds. Further tests are in progress. Please report any problems with this card to the authors.

4.7 Calculation of EELS

EELS, or electron-energy loss spectroscopy, measures the spectrum of energy losses of a beam of high-energy electrons passing through a sample in an electron microscope. To good approximation, EELS is very similar to XAS, and one can use FEFF's XAS spectra (i.e., calculations for XANES or EXAFS) to model EELS experiments.

For more practical advice on how to use FEFF8.4 for EELS, please refer to the EELS paper in the references in Appendix C.

A future release of FEFF (FEFF8.5) will be highly optimized for EELS calculations and able to handle additional instrumental parameters.

4.8 Local Field and Core-hole Effects (PMBSE, TDDFT)

These are features of FEFF that are still under development, which are not reliable in this release of FEFF8.

Many-body effects such as local fields and the core-hole interaction can be significant in x-ray absorption spectra, even several hundred eV above an absorption edge. The treatment of these effects requires theories beyond the independent-particle approximation, e.g., the Bethe-Salpeter equation (BSE) or the time-dependent density-functional theory (TDDFT).

The projection-operator method Bethe-Salpeter equation (PMBSE) is used for core-hole absorption spectra calculations. The BSE is usually limited to low energies, while the TDDFT often ignores the nonlocality of the core-hole interaction. Time dependent density functional theory (TDDFT) is a general framework for studying non-stationary electronic processes. TDDFT is used for local-field absorption spectra calculations.

The approach being developed for FEFF is a combined approach for calculations of the x-ray spectra that include both of these effects, together with inelastic losses and self-energy shifts over a wide energy range.

Note that the TDLDA card accounts for some of the same effects, and is functional in this release of FEFF8.

See *Combined Bethe-Salpeter equations and time-dependent density-functional theory approach for x-ray absorption calculations*; A.L. Ankudinov, Y. Takimoto, and J.J. Rehr, Phys. Rev. B 71, 165110 (2005)

Appendix A

Copyright Information, Restrictions and License

A.1 Restrictions and License Information

The full FEFF distribution is copyrighted software and a license from the University of Washington Office of Technology Transfer must be obtained for its use. This is necessary to protect the interests both of users and the University of Washington. Both academic/non-profit and commercial licenses are available — see Section A.2 of this document for details. New users should request the latest version of this code. The license form may be obtained from the FEFF WWW pages,

http://leonardo.phys.washington.edu/feff/

or by writing or sending a FAX to

The FEFF Project
c/o FEFF Project Assistant
Department of Physics
BOX 351560
University of Washington
Seattle, WA 98195

E-mail: feff@phys.washington.edu
Telephone: (206) 543-4615 Fax: (206) 685-0635

A.2 ADDENDUM: Government Copyrights

This work was supported in part by Grants from DOE. In accordance with the DOE FAR rules part 600.33 “Rights in Technical Data - Modified Short Form” the following clause applies to FEFF:

(c)(1)The grantee agrees to and does hereby grant to the U.S. Government and to others acting on its behalf:

(i) A royalty-free, nonexclusive, irrevocable, world-wide license for Governmental purposes to reproduce, distribute, display, and perform all copyrightable material first produced or composed in the performance of this grant by the grantee, its employees or any individual or concern specifically employed or assigned to originate and prepare such material and to prepare derivative works based thereon,

(ii) A license as aforesaid under any and all copyrighted or copyrightable work not first produced or composed by the grantee in the performance of this grant but which is incorporated in the material furnished under the grant, provided that such license shall be only to the extent the grantee now has, or prior to completion or close-out of the grant, may acquire the right to grant such license without becoming liable to pay compensation to others solely because of such grant.

(c)(2) The grantee agrees that it will not knowingly include any material copyrighted by others in any written or copyrightable material furnished or delivered under this grant without a license as provided for in paragraph (c)(1)(ii) of this section, or without the consent of the copyright owner, unless it obtains specific written approval of the Contracting Officer for the inclusion of such copyright material.

A.3 FEFF8 LICENSE

FEFF PROGRAMS (referred to below as "the System") Copyright (c) 1986-2002, University of Washington.

END-USER LICENSE

A signed End-user License Agreement from the University of Washington Office of Technology Transfer is required to use these programs and subroutines.

See the URL: http://leonardo.phys.washington.edu/feff/

USE RESTRICTIONS:

1. The End-user agrees that neither the System, nor any of its components shall be used as the basis of a commercial product, and that the System shall not be rewritten or otherwise adapted to circumvent the need for obtaining additional license rights. Components of the System subject to other license agreements are excluded from this restriction.

2. Modification of the System is permitted, e.g., to facilitate its performance by the End-user. Use of the System or any of its components for any purpose other than that specified in this Agreement requires prior approval in writing from the University of Washington.

3. The license granted hereunder and the licensed System may not be assigned, sublicensed, or otherwise transferred by the End-user.

4. The End-user shall take reasonable precautions to ensure that neither the System nor its components are copied, or transferred out side of his/her current academic or government affiliated laboratory or disclosed to parties other than the End-user.

5. In no event shall the End-user install or provide this System on any computer system on which the End-user purchases or sells computer-related services.

6. Nothing in this agreement shall be construed as conferring rights to use in advertising, publicity, or otherwise any trademark or the names of the System or the UW. In published accounts of the use or application of FEFF the System should be referred to by this name, with an appropriate literature reference:

FEFF8: A.L. Ankudinov, B. Ravel, J.J. Rehr, and S.D. Conradson, Phys. Rev. B 58, pp. 7565-7576 (1998).

LIMITATION OF LIABILITY:

1. THE UW MAKES NO WARRANTIES , EITHER EXPRESSED OR IMPLIED, AS TO THE CONDITION OF THE SYSTEM, ITS MERCHANTABILITY, OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. THE END-USER AGREES TO ACCEPT THE SYSTEM 'AS IS' AND IT IS UNDERSTOOD THAT THE UW IS NOT OBLIGATED TO PROVIDE MAINTENANCE, IMPROVEMENTS, DEBUGGING OR SUPPORT OF ANY KIND.

2. THE UW SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES SUFFERED BY THE END-USER OR ANY OTHER PARTIES FROM THE USE OF THE SYSTEM.

3. The End-user agrees to indemnify the UW for liability resulting from the use of the System by End-user. The End-user and the UW each agree to hold the other harmless for their own negligence.

TITLE:

1. Title patent, copyright and trademark rights to the System are retained by the UW. The End-user shall take all reasonable precautions to preserve these rights.

2. The UW reserves the right to license or grant any other rights to the System to other persons or entities.

Note: According to the terms of the above End-user license, no part of the standard distributions of FEFF can be included in other codes without a license or permission from the authors. However, some subroutines in FEFF explicitly contain such a license, and all components subject to other license agreements are excluded from the restrictions of the End-user license. Moreover, we are willing to collaborate with other code developers, and our development version with all the comments (and subroutines in individual files!) can be made available, although it makes use of some features that are not standard FORTRAN. Also we cannot guarantee that any new version of FEFF will be compatible with these subroutines or with any changes you make in the codes.

Appendix B

Installation Instructions

The program FEFF8.40 is provided as a single file '`feff84.f`'. An experimental modular form consisting of eight source files '`rdinp-tot.f`', '`pot-tot.f`', '`ldos-tot.f`', '`xsph-tot.f`', '`fms-tot.f`', '`path-tot.f`', '`genfmt-tot.f`' and '`ff2x-tot.f`' is available by request. ℓDOS calculations are no longer executed in the second module, since it is now a separate program, but this has not affected compatibility with other FEFF input files. The modular code requires less memory and yields more effective compilation and execution on small machines; however, the CFAVERAGE card is disabled in the modular version. Parallel execution on MPI clusters has been tested for modular code only and is still experimental. For details, please contact the authors.

Each file contains a main program and all necessary subroutines. Simply compile (and link) every program on your system using a Fortran 77 compiler (and your usual linker), e.g., for any UNIX system the command *`f77 -o module module.f`* is usually sufficient, but may be augmented by optimization flags.

For the monolithic version, there is only a single executable to run. For the modular code, the 8 executable files must be run in the order listed: **rdinp**, **pot**, **ldos**, **xsph**, **fms**, **path**, **genfmt**, **ff2x**, so that the necessary input files for each successive module are produced. We suggest that you write a small script (named '`feff8`'), that executes all modules on your OS, if you use the modular code. Samples are available from the authors. Parallel versions of the modules and appropriate compile and run scripts are also available and require a working version of the MPI libraries. Please contact the authors for details.

Typically the monolithic FEFF8 requires about 512 MB of RAM to run for clusters of up to 175 atoms. To adjust the required memory to your computer, you should change the `nclusx` parameter *globally*. The required memory scales approximately as $\texttt{nclusx}^2$. For larger clusters, you may wish to reduce the maximum angular momentum `lx` to keep the total size reasonable, e.g., set `nclusx` $= 300$, `lx` $= 2$ globally in all subroutines.

FEFF8 will also run on older '`feff.inp`' files for FEFF7 (and also FEFF6) and will yield results for EXAFS and XANES comparable to older versions of FEFF. The first module of FEFF7 has now been split into 3 modules and therefore FEFF8 will use the first value in FEFF7 CONTROL and PRINT cards for the first 3 modules of FEFF8. But we suggest that users make use of the SCF, FMS, LDOS and other cards to take advantage of FEFF8's new capabilities.

The FEFF code is written in ANSI Fortran 77, except that `complex*16` variables are required. Since data type `complex*16` is not part of the ANSI standard, minor compiler-dependent modification may be necessary. We have used the VAX extensions to ANSI Fortran 77 since they seem to be the most portable. The non-standard statements and intrinsic functions used are: `complex*16` variables and arrays; `dimag(arg)` returns a double precision imaginary part of it argument; `dcmplx(arg)` returns a `complex*16` version of its argument; `DBLE(arg)` returns a double precision real part of its argument; `sqrt`, `exp`, `abs` and other generic math functions are assumed to accept `complex*16` arguments and return double precision or `complex*16` results.

Files are opened with the `open` statement. As the ANSI definition gives some leeway in how the `open` statement interacts with the operating system, we have chosen file names and conventions that work on UNIX, VAX/VMS, IBM PCs (and clones) with MS FORTRAN, CRAY, MAC's and CDC machines. It may be necessary to modify the open statements on other systems.

B.1 UNIX

We have endeavored to make FEFF portable to all UNIX (including HP, AIX, LINUX, Alpha, BSD and CRAY) machines without any modification. If your machine does not reproduce the test output files '`xmu.dat`' and/or '`chi.dat`' to high accuracy, please let us know. Also, please report any compiler problems or warning messages to the authors, as this will help us achieve full portability.

On SGI machines a significant increase in speed may be achieved with the following optimization flags

```
f77 -Ofast -LNO:opt=0 -IPA:INLINE=OFF module.f -o module
```

Inlining is a potentially dangerous operation which modifies the source code, so the output ('`xmu.dat`') should be checked. Switching off inlining may slow down the code, as we found for the **fms** module, but some other modules would not operate properly with inlining.

On many UNIX systems it may be necessary to increase the memory stack size to allow calculations on large clusters. For example, for 200 atom clusters, you should execute the following command before running FEFF8 for the CSH or one of its derivatives

```
limit stacksize 170000
```

or, if you use the Bourne shell or one of its derivatives, the following

```
ulimit -s 170000
```

If you use the GNU g77 compiler, for example, on LINUX, BSD, and other systems try for example:

g77 -O2 -ffast-math -m486 -Wall -g -fno-silent

or (since the compile flags often change) simply

g77 -O2

Do not use the *-pedantic* flag, although a casual reading of the g77 document would suggest it is a good idea. *-pedantic* does not allow the use of double complex which is essential in FEFF.

On some AIX machines, the intrinsic real functions need to be promoted to double precision (e.g. -qautodbl=dbl option on RS6000).

B.2 CRAY, SGI-CRAY, and CDC UNIX

For CRAY, SGI-CRAY and CDC UNIX, please keep in mind the following points: Floating point calculations in FEFF are usually done to 64 bit precision. Thus for 32 bit word machines, the code uses double precision variables throughout, i.e., `real*8` for real numbers and `complex*16` for complex numbers. If your machine uses 8 bytes (64 bits) for single precision floating point numbers and integers (for example, CRAYs and some CDC machines), you should use the CRAY FORTRAN compiler option to ignore double precision statements in the code. The compile flags depend on machine vintage (see the cf77 or f90 man pages for details) and are of the form:

f90 -c -dp or *cf77 -c -Wf"-dp"*

B.3 MS-DOS, WIN-NT,9X,ME,2K, etc

Because of the awkwardness of DOS, many users do not have FORTRAN compilers and many of those compilers are difficult to use with large codes. Thus we have made executable versions of FEFF for PCs available. These versions should run on any 32-bit Microsoft OS on an IA-32 (Pentium compatible) machine with enough memory. Please contact the authors if you encounter any difficulties, or if you require a compatible executable.

If you prefer your own compiler (e.g., Compaq Visual Fortran or the free GNU g77 for Win32) or are using an operating system other than DOS, simply compile the source code using your FORTRAN compiler and linker as you would for any other machine.

You will need at least a Pentium (or compatible) CPU, and approximately 500 MB of RAM. Machines with older processors (e.g., a 486 with math coprocessor) may be compatible, but usually support considerably less memory than this version of FEFF requires. Less than 500 MB of RAM is not recommended for running FEFF8.4, particularly for more intensive calculations (e.g., large cluster XANES calculations), but the code has been tested for small clusters on Windows machines with as little as 256 MB of physical memory installed.

B.4 Macintosh

FEFF is often difficult to compile on Macintosh machines with Mac OS 9 and below, but executable versions of FEFF for Macintosh computers including the G4 are available from the FEFF Project.

For Mac OS X UNIX use the following options with Absoft's F77 compiler

```
f77 -N113 -N11 -f -O module.f
```

-f makes F77 case insensitive. *-N113* promotes intrinsic routines from real to double precision. *-N11* allows 32 bit operations, and *-O* makes basic optimizations.

There is another option for Mac OS X UNIX. At least one user has found that f2c and the cc compiler can be made to work.

B.5 Other Machines: VMS, NEXT, etc

To compile FEFF8 on VMS (6.0 to 6.2 versions) machine we had to increase virtual memory using SYSGEN from a SYSTEM account.

```
$mc sysgen
  sysgen> use current
  sysgen>show virtualpagecnt
  sysgen>set virtualpagecnt (value + 25%)
  sysgen>write current
  sysgen>exit
$reboot
```

If the code still does not compile one should further increase virtual memory. It may be impossible to compile the full code on old VAX stations. To run FEFF8 on VMS machines, you may need to increase quotas (*pgflquo=500,000*) for the users using AUTHORIZE from a SYSTEM account.

```
$run authorize
  authorize> modify username /pgflquo=500000
  authorize>exit
$reboot
```

Use the following options with Absoft's F77 for NeXT

```
f77 -N53 -f -s -O module.f
```

-f makes F77 case insensitive. *-s* makes a code for units larger than 512K. *-N53* uses the 68030/68040 processors with 68881/2 math coprocessor. *-O* is an optimization flag.

Appendix C

References

Please cite at least one of the following articles if FEFF is used in published work.

FEFF8 *Main FEFF8 reference* A.L. Ankudinov, B. Ravel, J.J. Rehr, and S.D. Conradson, *Real Space Multiple Scattering Calculation of XANES*, Phys. Rev. B **58**, 7565 (1998).

FEFF8.1
A.L. Ankudinov, and J.J. Rehr, *Theory of solid state contributions to the x-ray elastic scattering amplitude*, Phys. Rev. B **62**, 2437 (2000).

FEFF8.2
A.L. Ankudinov, C. Bouldin, J.J. Rehr, J. Sims, H. Hung, *Parallel calculation of electron multiple scattering using Lanczos algorithms*, Phys. Rev. B **65**, 104107 (2002).

FEFF8.4
A.L. Ankudinov, A.I. Nesvizhskii, and J.J. Rehr, *Dynamic screening effects in x-ray absorption spectra*, Phys. Rev. B 67, 115120 (2003); this contains information about TDLDA calculations in FEFF. M.S. Moreno, K. Jorissen, and J.J. Rehr, *Practical aspects of electron energy-loss spectroscopy (EELS) calculations using FEFF8*, Micron, in press (2006); this paper has information on EELS calculations using FEFF8.4.

FEFF7 A.L. Ankudinov and J.J. Rehr, *Relativistic Spin-dependent X-ray Absorption Theory*, Phys. Rev. B **56**, R1712 (1997). A.L. Ankudinov, PhD Thesis, *Relativistic Spin-dependent X-ray Absorption Theory*, University of Washington, (1996); this contains a review of x-ray absorption theory, a whole chapter of information about FEFF for expert users, example applications, and the full FEFF7 program tree.

FEFF6 S.I. Zabinsky, J.J. Rehr, A. Ankudinov, R.C. Albers and M.J. Eller, *Multiple Scattering Calculations of X-ray Absorption Spectra*, Phys. Rev. B **52**, 2995 (1995).

FEFF5 J.J. Rehr, S.I. Zabinsky and R.C. Albers, *High-order multiple scattering calculations of x-ray-absorption fine structure*, Phys. Rev. Lett. **69**, 3397 (1992).

FEFF3 and FEFF4

J. Mustre de Leon, J.J. Rehr, S.I. Zabinsky, and R.C. Albers, *Ab initio curved-wave x-ray-absorption fine structure*, Phys. Rev. B **44**, 4146 (1991).

FEFF3 J.J. Rehr, J. Mustre de Leon, S.I. Zabinsky, and R.C. Albers, *Theoretical X-ray Absorption Fine Structure Standards*, J. Am. Chem. Soc. **113**, 5135 (1991).

FEFF Review

J.J. Rehr and R.C. Albers, *Modern Theory of XAFS*, Rev. Mod. Phys. **72**, 621 (2000).

Sum rule normalization procedure in FEFF8.2

A.I. Nesvizhskii, A.L. Ankudinov, and J.J. Rehr, *Normalization and convergence of x-ray absorption sum rules*, Phys. Rev. B **63**, 094412 (2001).

Multiple Scattering theory in FEFF

J.J. Rehr and R.C. Albers, *Scattering-matrix formulation of curved-wave multiple-scattering theory: Application to x-ray-absorption fine structure*, Phys. Rev. B **41**, 8139 (1990).

Dirac–Fock atom code

A.L. Ankudinov, S.I. Zabinsky and J.J. Rehr, *Single configuration Dirac-Fock atom code*, Comp. Phys. Comm. **98**, 359 (1996).

Appendix D

Code Variables and Dimensions

The array names in FEFF are a bit cryptic due to the six character limit in standard FORTRAN — the comments given in the source code explain what the names mean. If you need to run larger problems than the dimension statements in the code allow, simply change the dimensions in all the relevant parameter statements and recompile. The main parameters to change are `nclusx`, which specifies the maximum cluster size for full multiple scattering, and `lx`, which specifies the maximum angular momentum in SCF potentials. These and other user changeable parameters are listed in '`dim.h`', which is now incorporated explicitly in the code. If you need help to modify the FEFF code, please contact the authors.

```
c       maximum number of atoms for FMS. Reduce nclusx if you need
c       smaller executable.
        parameter (nclusx=175)
c       max number of spins: 1 for spin average; 2 for spin-dep
        parameter (nspx=1)
c       max number of atoms in problem for the pathfinder
        parameter (natx =1000)
c       max number of atoms in problem for the rdinp and ffsort
        parameter (nattx =10000)
c       max orbital momentum for fms module.
        parameter (lx=3)
c       max number of unique potentials (potph) (nphx must be ODD to
c       avoid compilation warnings about alignment in COMMON blocks)
        parameter (nphx = 7)
c       max number of ang mom (arrays 1:ltot+1)
        parameter (ltot = 24)
c       Loucks r grid used through overlap and in phase work arrays
        parameter (nrptx = 1251)
c       Number of energy points genfmt, etc.
        parameter (nex = 450)
```

```
c       Max number of distinct lambda's for genfmt
c       15 handles iord 2 and exact ss
        parameter (lamtot=15)
c       vary mmax and nmax independently
        parameter (mtot=4, ntot=2)
c       max number of path atoms, used in path finder, NOT in genfmt
        parameter (npatx = 8)
c       matches path finder, used in genfmt
        parameter (legtot=npatx+1)
c       max number of overlap shells (OVERLAP card)
        parameter (novrx=8)
c       max number of header lines
        parameter (nheadx=30)
```

It should also be noted here that there is an internal limit on the number of paths (set to 1200) that will be read from '`feff.bin`'. This limit was chosen to handle any reasonable problem without using an excessive amount of memory. If you must use more paths, change the parameter **npx** in the FEFF source in subroutine **ff2chi** to whatever you need. This will require more memory. We have not had a case where the filter criteria were not able to solve the problem with fewer than 1200 paths.

Appendix E

Changes From Previous Versions of FEFF

FEFF8.40 automates spin-dependent calculations, simplifying calculation of spin-polarized absorption spectra (SPXAS, SPEXAFS) and XMCD, and introduces TDLDA calculations to account for core-hole and local field screening effects. Calculations have been added using TDDFT and PMBSE to account for additional core-hole effects, but these are still under development and are not reliable in this release.

FEFF8.20 extends the calculation of quadrupolar transitions and x-ray emission spectra calculations, and also permits faster XANES calculations using iterative Lanczos FMS algorithms. Improved potentials for f-electron materials are also included. The code has been restructured to simplify future developments and is also available as separate modules which can be run on parallel machines.

FEFF8.10 fixes a few bugs (notably that for polarization dependent calculations for initial states other than s-character; i.e., L2, L3 edges,etc) and adds some new capabilities. The code has been extended to calculate elastic scattering amplitude and x-ray natural dichroism. An additional output file, '`ratio.dat`', has been added for use in our procedure for sum-rule normalization.

FEFF8 potentials can now be calculated self-consistently (SCF card) which also gives a more accurate Fermi level position and accounts for the charge transfer. Full multiple scattering capability was also added (FMS card). This is essential for SCF potentials, L-projected density of states (LDOS also is a new card) and often XANES. All these new cards are not essential for EXAFS calculations, but the SCF potential can be used to reduce number of EXAFS fitting parameters by calculating the Fermi level and non-integral charge counts on each site. The possibility of calculating multiple-scattering Debye–Waller factors from force constants and/or dynamical matrices has also been added. The possibility of configurational averages (CFAVERAGE card) of EXAFS (or XANES) over different absorbers of the same type has been added. The CONTROL structure has been changed to accommodate the new cards, but backward compatibility has been maintained. Also several cards (e.g. EXAFS, XANES and POTENTIALS) now have additional optional fields.

Appendix F

Trouble-Shooting FEFF Problems and Bug Reports

FEFF8 has been extensively tested on many different architectures, but occasionally new bugs show up. In an effort to maintain portable and trouble-free codes, we take all bug reports seriously. Please let us know if you encounter any compilation error or warning messages. Often we receive reports by users of older versions of FEFF of bugs that have been fixed in more recent releases. Other code failures can often be traced to input file errors, sometimes quite subtle, and some are compiler bugs, for which we try to find a workaround.

To report a bug, please tell us the version of the code you are using and which operating system and compiler you have. Please include a '`feff.inp`' if the problem occurs after compilation and enough detail concerning the warning or error messages or other difficulties you have so that we can attempt to reproduce the problem.

Some known and commonly encountered difficulties are:

- Non-physical, widely spaced distributions of atoms. Symptoms of this common problem are very large muffin-tin radii (see the header of any '`.dat`' file) and possibly a failure of the phase-shift program to converge. This gives error message `hard test fails in fovrg`.
- An error in assigning potential indices; the first atom with a given potential index must have the geometry representative of this potential type. This is sometimes fixed by using a somewhat larger cluster; in fact it is usually desirable to have a larger cluster for potential construction than that used in the XAFS calculation due to errors in the potentials at surfaces. Unless the atom distribution is physically possible, you can expect the code to have problems.
- Hash collision in the pathfinder. This is now rare, but can usually be corrected simply by changing distances in the fourth decimal place.
- For the M_{IV} and higher edges you may receive the error message like: `Lambda array overfilled`. The calculations should be repeated with IORDER -70202 card.