
FEFF10

The FEFF Project
Department of Physics
University of Washington

User's Guide, FEFF version 10
updated October 5, 2020

Abstract

FEFF is an *ab initio* multiple-scattering code for calculations of excitation spectra and electronic structure. It is based on a real space Green's function approach that includes a screened core-hole, inelastic losses and self-energy shifts, and Debye-Waller factors. Optionally FEFF includes local fields (TDLDA) and an approximate treatment of multi-electron excitations. The spectra include extended x-ray absorption fine structure (EXAFS), 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), Compton scattering, Non-Resonant Inelastic X-ray Scattering (NRIXS), and the x-ray scattering amplitude (including Thomson and anomalous parts). In addition the code can treat relativistic electron energy loss spectroscopy (EELS). FEFF10 is written in Fortran 90.

This software was developed with support in part from the U.S. DOE, Office of Basic Energy Sciences, DMSE grants DE-FG03-97ER45623 (Dept. of Physics, U. of Washington) and DE-AC02-76SF00515 (Theory Institute for Materials and Energy Spectroscopies (TIMES) at SLAC).

The FEFF Project is copyright ©2020 University of Washington and SLAC National Accelerator Laboratory. All rights reserved.

This document is copyright ©2020 by J.J. Rehr, J.J. Kas, K. Jorissen, A. Ankudinov, and B. Ravel.

Permission is granted to reproduce and distribute this document in any form so long as the copyright notice and this statement of permission are preserved on all copies.

This document can also be found on the [feff project website](#).

Contents

1	Synopsis	1
1.1	Acknowledgements and history	2
1.2	The FEFF Users Guide	4
1.3	Examples	5
2	Calculating material properties with FEFF	6
2.1	General Comments	6
2.2	Running FEFF	7
2.2.1	Preparing the input	7
2.2.2	Conventions for the input file	8
2.2.3	Starting the calculation	8
2.2.4	Studying the results	11
2.3	Calculating the potential and phase shifts.	11
2.4	Calculating the Density of States	13
2.5	Calculating the Multiple Scattering Green's function.	14
2.6	Calculating the spectrum	15
2.7	EXAFS Calculation	16
2.7.1	SF ₆ Molecule	16
2.7.2	Solids	17
2.7.3	Estimate of S_0^2	18
2.7.4	Configuration Averaging Over Absorbers	19
2.7.5	Adding Self-consistency	19
2.8	XANES Calculations	19
2.8.1	Need for SCF and Additional Difficulties for XANES	19
2.8.2	GeCl ₄ Molecule	20
2.8.3	Solid: XANES and ℓ DOS	21

2.8.4	Cross-section in absolute units	21
2.9	Spin-dependent Calculations	22
2.9.1	General Description	22
2.9.2	XMCD for the Gd L1 edge	22
2.9.3	SPXAS	23
2.10	Elastic Scattering Amplitudes	25
2.11	X-ray Emission Spectra XES	25
2.12	Calculation of EELS	25
2.12.1	Changing EELS parameters	26
2.12.2	Example input.	26
2.13	NRIXS	26
2.13.1	Some example input	27
2.13.2	Limitations and practicalities.	28
2.14	Local Field Effects and Core-hole Effects (PMBSE, TDDFT)	29
2.15	Using a .cif file	29
2.15.1	Formal requirements for cif files	30
2.16	K-space FEFF	31
2.16.1	Core hole calculations without the supercell	31
2.16.2	Using the file ‘reciprocal.inp’	32
2.16.3	Converging the k-mesh	32
2.16.4	EXAFS and EXELFS	32
2.16.5	Using crystal symmetry	33
2.16.6	Speed	33
2.16.7	Example input files	33
2.17	Charge Transfer and Charge Counts	36
2.18	Ab initio Debye-Waller factors	38
2.18.1	EM and RM methods	41
2.18.2	DM method	44
2.19	Parallel Calculations	50
2.19.1	MPI calculations on the command line	51
2.20	Using the Many-Pole Self-Energy	53
2.20.1	Examples	53
2.21	Compton scattering	55
2.22	The Mixed Dynamic Form Factor	56

3	feff10 Control Cards	57
3.1	The master input file ‘feff.inp’	57
3.2	Complete List of FEFF10 Control Cards	57
3.3	Structural Information Cards	60
3.4	Spectrum Information Cards	67
3.5	FEFF Program Control Cards	74
3.5.1	General Cards	74
3.5.2	ATOMIC, POT, SCREEN : Scattering Potentials	76
3.5.3	XSPH: Cross-section and Phase Shifts	88
3.5.4	FMS and MKGTR : Full Multiple Scattering	90
3.5.5	PATH: Path Enumeration	92
3.5.6	GENFMT: XAFS Parameters	95
3.5.7	FF2X: XAFS Spectrum	96
3.5.8	SFCONV: Spectral Function Convolution	99
3.5.9	COMPTON: Compton scattering	99
3.5.10	EELS: Electron Energy Loss Spectrum	100
4	Input and Output Files	101
4.1	Data flow	102
4.2	File Reference	107
4.2.1	Main Output Data	107
4.2.2	EELS Files	107
4.2.3	Intermediate Files	108
4.2.4	Diagnostic Files	109
4.2.5	Variables in the EXAFS and XANES Formulae	110
4.3	Program Control Using Intermediate Output Files	111
4.3.1	Using ‘paths.dat’	111
4.3.2	Using ‘list.dat’	111
4.3.3	Using ‘geom.dat’	111
A	Copyright Information, Restrictions and License	112
A.1	Restrictions and License Information	112
A.2	Government Copyrights	112
A.3	FEFF10 LICENSE	113
B	Installation Instructions	114

B.1	Compiling your own code	114
C	References	116
D	Code Variables and Dimensions	119
E	Spin.f program for XMCD and SPXAFS	122
F	Technical notes for dynamical matrix based Debye-Waller factors	124
F.1	Installing DMDW	124
F.1.1	Installing as a FEFF module	124
F.1.2	Creating and installing the standalone version	124
F.2	Using DMDW to calculate Debye-Waller factors	125
F.2.1	Using within a XANES or EXAFS calculation in FEFF	125
F.2.2	Calculating DW factors using the standalone version	125
F.3	The "dym" dynamical matrix file format	127
F.4	Examples	128
F.5	Troubleshooting dmdw	128
F.5.1	Lanczos problems	128
G	Trouble-Shooting feff Problems and Bug Reports	129

Chapter 1

Synopsis

FEFF10 calculates spectroscopic properties, including 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), Compton scattering, nonresonant x-ray emission (XES), and Non-Resonant Inelastic X-ray Spectroscopy (NRIXS). In addition the code calculates electronic structure including local densities of states (LDOS), and the x-ray elastic scattering amplitude $f = f_0 + f' + if''$ including Thomson and anomalous parts, and relativistic electron energy loss spectroscopy (EELS).

FEFF uses an *ab initio* self-consistent real space multiple scattering (RSMS) approach, including polarization dependence, core-hole effects, and local field corrections, based on self-consistent, spherical muffin-tin scattering potentials. Calculations are based on an all-electron, real space relativistic Green's function formalism with no symmetry requirements. The code builds in inelastic losses in terms of a GW self-energy, and includes vibrational effects in terms of Debye-Waller factors. For periodic structures reciprocal space calculations based on periodic boundary conditions are also available. FEFF can use both full multiple scattering based on LU or Lanczos algorithms and a high-order path expansion based on the Rehr–Albers multiple scattering formalism.

The name FEFF is derived from f_{eff} , the effective curved wave scattering amplitude in the modern EXAFS equation. This was the first application of the FEFF code, and is the basis for the multiple-scattering path-expansion in the code.

For details on use of the code, examples and suggestions for calculation strategies, see Sections 2, 4, and 3. For details about the algorithms, see the discussion for the appropriate module in Section 3. For additional details, see the published references in Appendix C and the FEFF website.

FEFF is written in ANSI Fortran 90. It typically requires 250 MB - 2GB of available memory (RAM). More memory may be needed on MS Windows systems or for very large calculations. See Appendix B for installation instructions. Many problems run well on a laptop. For larger calculations of XANES, NRIXS, or COMPTON a small cluster may be more appropriate.

Please contact the authors concerning any problems with the code. See Appendix G for trouble-shooting hints and problem/bug reports or the FAQ on the [FEFF website](#).

1.1 Acknowledgements and history

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@uw.edu

telephone: (206) 543-8593, FAX: (206) 685-0635

The full FEFF10 code is copyright protected software. 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 website:

<http://www.feffproject.org>

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

Please cite an appropriate FEFF reference if the code or its results are used in published work. The main reference for calculations using the current version FEFF10 is:

Calculations of x-ray spectroscopies using FEFF10 and Corvus, J.J. Kas, F.D. Vila, D. Pemmaraju, Tun S. Tan, S. Vimolchalao, and J.J. Rehr, *Unpublished*, 2020.

Parameter-free calculations of x-ray spectra with FEFF9, J.J. Rehr, J.J. Kas, F.D. Vila, M.P. Prange, K. Jorissen, *Phys. Chem. Chem. Phys.*, 12, 5503-5513 (2010).

The main references for previous versions of FEFF are:

Ab initio theory and calculations of X-ray spectra, J.J. Rehr, J.J. Kas, M.P. Prange, A.P. Sorini, Y. Takimoto, F.D. Vila, *Comptes Rendus Physique* 10 (6) 548-559 (2009).

and for EXAFS theory:

Theoretical Approaches to X-ray Absorption Fine Structure, J. J. Rehr and R. C. Albers, *Rev. Mod. Phys.* **72**, 621, (2000).

See Appendix C or the feffproject website for a list of additional references.

FEFF Project Developers

The core developers team for FEFF10 consists of the following people:

- **Joshua Kas** led the development of FEFF10 with addition of the finite-temperature extension and other features.
- **Tun S. Tan** contributed to the finite-temperature extension
- **Brian Mattern** implemented COMPTON calculations;
- **Towfiq Ahmed** developed of the Hubbard U module;

- **Micah Prange** developed the FEFF/optical constant extension;
- **Yoshinari Takimoto** implemented the RPA core hole;
- **Adam Sorini** investigated inelastic mean free paths;
- **Aleksi Soininen** implemented q-dependent NRIXS calculations.

Kevin Jorissen and **Joshua Kas** were the principle developers of the FEFF9 series. **David Bitseff**, **Lukas Svec** and **Kevin Jorissen** developed the JFEFF GUI for FEFF9, which is no longer supported for FEFF10. **Kevin Jorissen** implemented relativistic EELS calculations, k-space algorithms, cloud computing, and others. **Fernando D. Vila** implemented improved Debye-Waller factors.

Alexei Ankudinov was the principal developer for the FEFF8 series which included automated self-consistent potential s, calculations of the ℓ DOS, the Fermi level, and charge transfer. He also added calculations of elastic scattering amplitude, x-ray natural circular dichroism and nonresonant x-ray emission, and quadrupolar transitions. **Bruce Ravel** developed the full multiple scattering (FMS) algorithm in FEFF8. **A. Nesvizhskii** added new routines for TDLDA and sum-rules applications. Anna Poiarkova and Patrick Konrad contributed algorithms for calculating multiple-scattering Debye-Waller factors and anharmonic contributions. **Jim Sims** (NIST) extended the code for MPI based parallel execution, in collaboration with **C. Bouldin** and **J.J. Rehr**. The high-order multiple-scattering routines, pathfinder and correlated Debye-Waller routines were developed by **Steven Zabinsky** and **J.J. Rehr** for FEFF5. The Hedin-Lundqvist self-energy and phase shift routines were developed by **Jose Mustre de Leon**, **Dan Lu**, **J. Rehr**, and **R.C. Albers**.

We gratefully acknowledge invaluable feedback from many users and in particular Drs. Matthew Newville, Bruce Ravel, Anatoly Frenkel, Simon Bare, Shelly Kelly, F. Farges, G. Hug, H. Wende and S.D. Conradson. We also thank Siri Vimolchalao for help testing the current version FEFF10.

References for previous FEFF versions and implementations mentioned above can be found in the appendix.

1.2 The FEFF Users Guide

Chapter (2) is devoted to **calculations of material properties**. Here, we discuss in detail how to calculate various spectra, accounting for spin variables, obtaining XMCD spectra, and so on. Many example input files are described. These examples are also included in your installation, so you can run them yourself on your computer. This chapter is required reading for serious readers.

Chapter (3) is a reference chapter describing all of **FEFF's input options**, which are specified through CARDS in the input file. First, we discuss cards that specify the system under consideration (molecule, cluster, crystal ...). Then we discuss cards that specify which spectroscopy or material property will be calculated. All of these cards globally affect the FEFF program. Lastly, we discuss more specialized cards that really do “belong” to a specific module of FEFF. This classification combines an intuitive classification for new users with efficiency for advanced users (who may not run the entire FEFF program flow).

Chapter (4) describes **input and output files**, illustrating the data flow of the program and helping users to find the information they need in FEFF's many output files.

Finally, the last pages of the UG comprise **installation instructions**, license information, and other technical information.

If you notice any mistakes or omissions in this Users Guide, please contact us at feff@uw.edu. We appreciate any feedback that will help us improve this document. The most recent version of the documentation is available on the **FEFF website** in pdf and wiki format.

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
<i>slanted typewriter font</i>	commands executed at a command line

1.3 Examples

Your FEFF installation includes a folder containing examples of calculations for nearly all different types of calculations FEFF can do. It's worth spending some time looking at the files and exploring which options were used. Most of the examples are also discussed in the next chapter of this Users Guide.

Note that many examples have reduced basis sets that allow you to run the test quickly, but do not produce a "converged" spectrum that can be compared to experiment or used in a publication. At least the cutoff radii of the SCF, FMS, and RPATH cards need to be converged to achieve proper results.

Each example has a self-contained directory containing the input file '`feff.inp`', a reference spectrum file '`xmu.dat`', and a compressed file '`REFERENCE.zip`' containing the entire calculation, including intermediary files.

Once you start setting up and running your own calculations, we recommend that you continue to launch each calculation in a separate directory to avoid problems and confusion. (Once files get mixed up, it can be very hard to figure out what happened, and the only solution is often to recalculate everything from the beginning.)

Chapter 2

Calculating material properties with FEFF

2.1 General Comments

A FEFF calculation starts with two basic tasks: calculation of the potentials, followed by calculation of the scattering phase shifts. Using these phase shifts, scattering paths are found and their scattering amplitudes calculated. There are two ways to do this: either using an explicit enumeration and summation, most appropriate for extended absorption spectra; or using the implicit summation of full multiple scattering, appropriate for near-edge absorption spectra. Finally, these scattering amplitudes are combined and processed to a particular type of spectrum - currently EXAFS, XANES, ELNES, EXELFS, XES, NRIXS, DANES, COMPTON, or XMCD.

Each of these basic steps corresponds to a handful of FEFF program modules controlled by a set of ‘feff.inp’ input options. In later chapters we discuss these aspects in a technical way by listing the details of program modules, input cards, and files. The current chapter takes a more user-oriented approach. We discuss each type of spectroscopy individually. We indicate which options (cards) and program modules are relevant for that type of spectrum, and give recommendations on strategies, as well as examples.

We encourage users to spend some time exploring the ‘examples’ folder that came with their FEFF distribution. The examples are in `~/feff10/examples`.) This folder contains several testcases, including all major types of spectroscopy. Each test case comes with an input file ‘feff.inp’, a reference output file ‘referencexmu.dat’ or ‘reference_eels.dat’, and an archive ‘REFERENCE.zip’ containing the entire calculation with intermediate output files. Every example that is quoted at length in this chapter of the Users Guide is available in the `~/feff10/examples` folder.

We endeavour to keep FEFF accessible to the novice user: you need only one input file ‘feff.inp’ and run a single command, ‘feff’. Nevertheless, we encourage you to gain insight in the underlying structure of FEFF. For example, many parameters can be changed without

requiring the entire calculation to be redone. If the model (i.e., the atom positions) doesn't change significantly, it is usually not necessary to recalculate the potentials. E.g., if one wants to add more paths to the path expansion, the calculation can be restarted with module **path**, saving much time. If one wants to change the beam orientation in an EELS calculation, only the **eels** module needs to run again. However, significant changes to the model require rerunning the entire calculation. Generally, **pot**, **fms**, and **genfms** take much time, while all other program modules are very fast.

The study of a material usually requires running several calculations, because there are parameters to converge, for example the cutoff radius of the full multiple scattering algorithm. There are no default values for these parameters that work for all materials. Debye-Waller factors may require much attention in the EXAFS regime but the XANES regime may be less sensitive. It takes some experimentation to find out what treatment is adequate for a given material.

The information in this chapter aims to illuminate such matters; some hands-on experience will do the rest.

2.2 Running FEFF

2.2.1 Preparing the input

The FEFF program consists of a set of program modules driven by a single input file '**feff.inp**'.

To prepare FEFF input, you must answer four questions:

1. What is the material I am simulating?
2. What spectrum am I interested in?
3. How do I want FEFF to do the simulation?
4. What parts of FEFF do I want to run?

The answer to question 1 requires knowledge of the structure. In most cases, this means obtaining a list of x,y,z Cartesian atom coordinates and atomic numbers. FEFF is happy with an explicit list of atoms, e.g. for molecules, aperiodic systems, or as an approximation to a periodic material. FEFF can also read '**.cif**' files, a widely accepted format for representing crystal data. Such files can be found for many materials in online databases and can be linked to in the '**feff.inp**' file. Finally, there are some third-party tools that can set up a '**feff.inp**' file for you based on atomic positions or crystallographic information. An example is **webatoms**, developed by B. Ravel; or the popular Athena program. More FEFF-compatible crystallographic tools are available from **ICMPE**.

Although FEFF input can certainly be constructed from scratch, the most common approach is to start from an existing input file, and modify it to what you want to do. Example input files included in your FEFF installation and discussed in this chapter are an excellent starting point. You can take such a file and replace the atomic coordinates with your own (typically, using

the ATOMS and POTENTIAL cards, or equivalently the CIF card). Then you can continue to edit the file to answer questions 2-4.

As you look at the examples in this chapter, it may be helpful to refer to Chapter 3, which contains an exhaustive reference list of all input options.

2.2.2 Conventions for the input file

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 starting with a CARD: a keyword possibly followed by some numbers or words. The sequence of keyword cards is arbitrary. If any card or optional data is omitted, default values are used. Most calculations do require the POTENTIALS card, and either the ATOMS card or the OVERLAP card. Together, the POTENTIALS and ATOMS/OVERLAP cards specify the physical structure of the molecule or solid one is interested in. FEFF uses Cartesian coordinates and Angstrom units. Hence it is easy to copy atom positions from, e.g., an `'xyz'` file or other popular file formats. It is also possible to import CIF files (`'*.cif'`). There are many optional cards to specify the material property to calculate (e.g., X-ray spectroscopy) and to control how FEFF calculates it.

All distances are in Angstrom (\AA) and energies are in electron-volts (eV).

2.2.3 Starting the calculation

Once the input is ready, the calculation is started by giving a single command `feff` from a command line terminal. This executes a script bundling about fifteen modules that together comprise the FEFF code.

Generally, not all modules are run for a given calculation. To some extent, each step can be controlled individually through cards or an intermediate input file (`'pot.inp'`, `'ldos.inp'`, ...). Note that it's useless to run any module unless the preceding modules have been run already. We recommend that novice users simply run the `feff` command until they develop a more detailed understanding of how the FEFF code works.

On the command line one can either call each of the modules by name, or use the **CONTROL** card in `'feff.inp'` to select modules.

The underlying program flow of the FEFF program is as follows:

1. The `'feff.inp'` file is read and checked for consistency. Various intermediate input files are written (in the module **rdinp**, read input).
2. Ab initio Debye-Waller factors are calculated if the dynamical matrix is provided. (**dmdw**, dynamical matrix to debye-waller.)
3. Atomic overlap (Mattheiss) potentials are calculated. (module **atomic**, atomic potentials.)
4. The scattering potentials are calculated self-consistently using an automated SCF loop. Absolute energies are estimated. (module **pot**, potentials.)

5. RPA screening is used to calculate the core-hole potential. (module **screen**.)
6. The angular momentum projected density of states is calculated. (module **ldos**.)
7. Some intermediate output is written to file. (module **opconsat**.)
8. The scattering phase shifts, dipole matrix elements, and x-ray cross-section are calculated. (module **xsph**, cross-section and phases.)
9. Full multiple scattering XANES/ELNES calculations are done to obtain the full Green's function for a specified cluster size. (module **fms**, full multiple scattering.)
10. The relevant elements of the FMS Green's function are traced and multiplied by matrix elements. (module **mkgtr**, trace Green's function.)
11. The leading multiple scattering paths for the cluster are enumerated. (module **path**, path enumeration.)
12. The effective scattering amplitudes f_{eff} and other XAFS parameters are calculated for each scattering path. (module **genfmt**, general-path F-matrix calculation.)
13. The XAFS parameters from one or more paths are combined to calculate a total XAFS or XANES spectrum. (module **ff2x**, scattering amplitude to χ .)
14. Regular XAS output files (e.g., 'xmu.dat') are convolved with a many body spectral function to include many body effects. (module **sfconv**.)
15. The Compton spectrum is calculated. (module **compton**, output in 'compton.dat')
16. The electron energy loss spectrum is calculated. (module **eels**. See Section 2.12.)

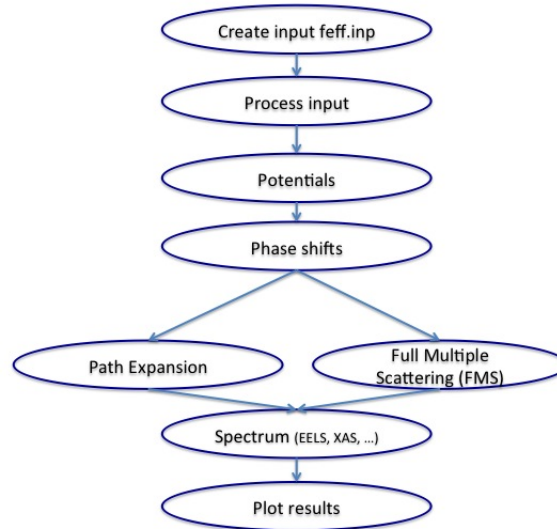


Figure 2.1: Functional flow chart of the FEFF program

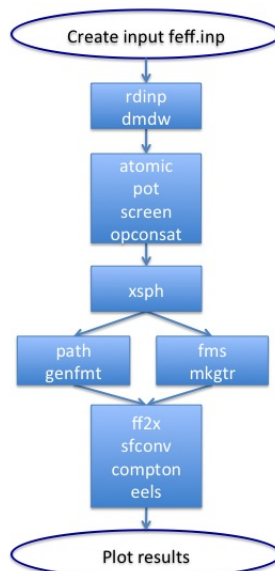


Figure 2.2: Flow chart of the FEFF program showing all submodules. The **ldos** module is not shown - it can be executed any time after the calculation of the potentials.

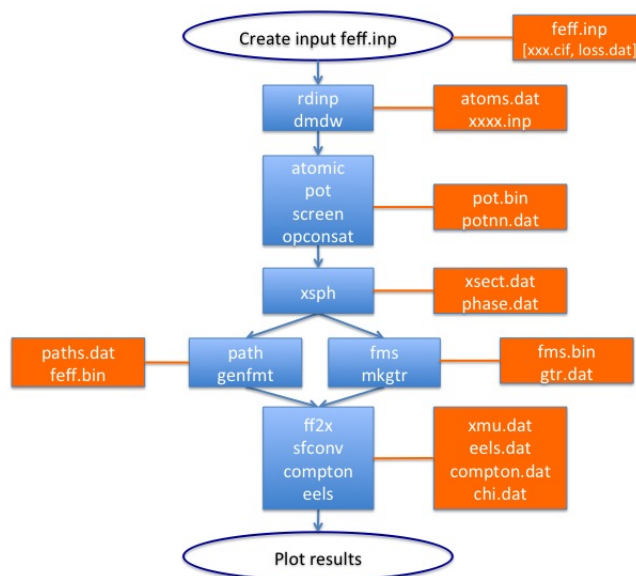


Figure 2.3: Flow chart of the FEFF program showing all submodules as well as some of the most important input and output files. Many other files are produced by the FEFF program depending on spectrum type and selected options. File descriptions are in Chapter 5.

2.2.4 Studying the results

First the quality of the calculation must be examined. We list a tentative set of questions one can ask:

- Making sure that the Self-Consistent Field (SCF) calculation of the potentials has converged (more below) - see `'log1.dat'`
- Check that the Density of States and Fermi level are reasonable (`'ldos01.dat'`, `'log1.dat'`)
- Check that the calculations are converged with respect to SCF cutoff radius
- Check that the calculations are converged with respect to FMS cutoff radius
- Check that the calculations are converged with respect to RPATH cutoff radius

In a next step, one plots the calculated spectrum (e.g. the XANES spectrum in file `'xmu.dat'`) together with a measured spectrum from an experiment. If the agreement is not satisfactory, one can check many factors in the calculation:

- Is the model of the structure correct?
- Does the calculation need more broadening applied?
- Does the calculation require a shift of the Fermi level to shift more states into the unoccupied band? (FEFF often puts the Fermi energy too high by a small amount.)
- Does a different choice of core hole treatment improve the calculation?
- Is it important to add Debye-Waller factors to the calculation to account for thermal effects?
- Will the ManyPole Self-Energy improve the calculation? (Typically yes, if peaks slightly beyond the edge need to move to slightly higher energies and require more broadening.)
- If the system contains f-states, did they converge properly in the SCF calculation?
- If the system has strongly correlated electrons, will a Hubbard (GW+U) calculation improve things? (Note: this functionality will become available in the next release of FEFF.)
- Could the spherical potential approximation be problematic for this material?
- ...

Most of these techniques are discussed in greater detail throughout this chapter. The paper by Moreno and Jorissen refers to an older version of FEFF but still provides a good case study of all the steps to check when doing a FEFF study.

2.3 Calculating the potential and phase shifts.

Potentials and phase shifts are calculated by executing the first modules of FEFF : **atomic**, **pot**, **screen**, **xsph**. This sequence corresponds to the first two fields in the CONTROL card, i.e.,

CONTROL 1 1 0 0 0 0

calculates potentials and phase shifts (and nothing more).

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. **FEFF10** 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. This behavior is activated by the SCF card in ‘feff.inp’. It is strongly recommended for calculation of near-edge properties. In this case, the radius of that small cluster is a parameter that must be converged for good results. Extended spectra, such as EXAFS or EXELFS, can typically be calculated without self-consistent potentials.

The SCF radius typically falls between 20 - 100 atoms, with 50 atoms being a reasonable first attempt; however, it must always be verified carefully if an accurate calculation is desired.

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 accurate but faster) can be estimated from the electron gas result at the mean interstitial density in the Mattheiss prescription. 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.

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. One should give consideration to adding potentials for sufficiently different atoms of the same atomic number. **FEFF10** only calculates the spherical part of the potential (“muffin tin potential”), so if atoms have environments identical up to a unitary transformation (e.g. a space group symmetry operation in a crystal), they can certainly be considered equivalent in **FEFF**. In strongly anisotropic environments, the muffin tin approximation may lead to inaccuracies in the potentials and density of states (DOS).

The potentials are written to file if the PRINT card is set high enough:

PRINT 2 0 0 0 0 0

results in files ‘pot00.dat’, ‘pot01.dat’ and so on. These contain single and overlapped potentials and densities and can be plotted, e.g. in gnuplot.

The progression of a self-consistent calculation can be checked in the ‘.scfconvergence-feff’.

Note that FEFF has historically accumulated options for setting the edge and core hole. We highly recommend that users use only the EDGE card for specifying the edge, and the COREHOLE card for selecting the core hole treatment. The HOLE and NOHOLE cards are deprecated and supported for backward compatibility only.

How best to treat the core-hole in XAS is an interesting theoretical question. The default choice in FEFF has always been a fully screened core-hole (meaning that the hole is self-consistently screened during the SCF loop), which is consistent with the “final state rule” (“COREHOLE fsr”). But this approximation sometimes breaks down for the L-shell metals. Often a better approximation is to use RPA screening. This screening approximation is similar to that in the Bethe-Salpeter equation (BSE) and is available in FEFF9 (“COREHOLE RPA”). However, missing is the exchange term in the particle-hole interaction, which tends to cancel the core-hole. Interestingly the cancellation is often strong for the L-shell materials, which is why “COREHOLE none” sometimes works better. Recently we’ve developed a BSE code to better study these phenomena. See also “Final-state rule vs the Bethe-Salpeter equation for deep-core x-ray absorption spectra,” J.J. Rehr, J. A. Soininen, and E. L. Shirley, *Physica Scripta* **T115**, 207 (2005); as well as a forthcoming paper, “BSE calculations of transition metal L-shell spectra, J. Vinson and J.J. Rehr”.

FIX add paragraph about screening.

xsph writes its main output to ‘xsect.dat’: this file contains the matrix elements and the atomic background as a function of energy. If you set the PRINT card to 1 or higher, the phases will be written to a file ‘phase.dat’, though this file is not usable for plotting. Setting the PRINT card even higher makes FEFF calculate hole counts and write them to ‘log2.dat’. Careful - this is quite slow.

```
PRINT 0 3 0 0 0 0 *hole counts - slow calculation
```

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.

2.4 Calculating the Density of States

Once the potentials and phase shifts are known, one can calculate the Density of States. This requires the LDOS card. If you are not running all of feff, you can simply run the **ldos** module. It will write files ‘ldos00.dat’, ‘ldos01.dat’ and so on, giving the l -resolved DOS for each potential type. There is currently no way to obtain m -resolved DOS.

2.5. CALCULATING THE MULTIPLE SCATTERING GREEN'S FUNCTION

The **ldos** module uses a fixed energy mesh of 81 points which sometimes frustrates users wanting the DOS over a larger energy range. The simplest solution is to simply run **ldos** several times with a different energy range and combine the output.

```
LDOS -20. 20. 0.05 * first run, -20->20 eV
*LDOS 20. 60. 0.05 * second run, 20->60eV
```

2.5 Calculating the Multiple Scattering Green's function.

Once we have the potentials and phase shifts, we can calculate the multiple scattering (MS) Green's function. This corresponds to the sequence of modules **fms**, **mkgtr**, **path** and **genfmt**. Equivalently, fields 3-5 in the CONTROL card:

```
CONTROL 0 0 1 1 1 0 * calculate G and nothing else
```

There are two ways to calculate the Green's function. The first method is suitable for near-edge properties and is executed by modules **fms** and **mkgtr**. These modules carry out a full multiple scattering XANES/ELNES calculation. In real space calculations this is done for a cluster centered on the absorbing atom. In k-space it is done for a matrix containing only the atoms of the unit cell. Thus all multiple-scattering paths within this system are summed to infinite order. This is useful for XANES and ELNES calculations, but usually cannot be used for EXAFS analysis. FMS loses accuracy beyond $k = (l_{\max} + 1)/r_{\text{mt}}$, which is typically about 4 \AA^{-1} since the muffin-tin radius r_{mt} is typically about 1 \AA . In real-space calculations, the FMS cluster radius must be converged to ensure adequate accuracy. FMS is typically the most time-consuming part of the calculation and slows down significantly with cluster or unit cell size.

The FMS Green's function is written to file '**gg.dat**'. Its trace - which yields the spectrum - can be found in '**gtr.dat**'.

For energies high above the Fermi level or edge threshold, the Path Expansion (PE) is more appropriate. It is executed by modules **path** and **genfmt**. Here, instead of summing implicitly over an infinite number of paths, we sum explicitly over a select range of scattering paths. 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 \AA) 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.

In the event of a k-space calculation, the **rdinp** module generates a large real-space list of atom coordinates from the unit cell information in **feff.inp**. This list will be used for the **path** and **genfmt** modules, which always work in real space, as this is the optimal space for treating the extended spectrum.

The list of all paths can be found in file **paths.dat**. You can manually edit this list.

For each path the code calculates the effective scattering amplitude and the total scattering phase shift along with other XAFS parameters using the scattering matrix algorithm of Rehr and Albers. This requires only the scattering phase shifts (module **xsph**) and the paths (module **path**) as input.

It is possible to combine the two strategies. E.g. the following input

```
FMS 4.0
RPATH 10.0
XANES
```

tells FEFF to calculate a XANES spectrum by doing Full Multiple Scattering up to a radius of 4. Å around the absorber (summing all paths to infinite order within this radius), and adding Path Expansion contributions from paths extending beyond this radius but no longer than 10. Å. Although this is possible, we do not recommend it. We recommend the combination FMS + XANES for near-edge spectra, and EXAFS + RPATH for extended spectra.

2.6 Calculating the spectrum

Finally, we can calculate the spectrum from the Green's function using the sequence **ff2x**, **sfconv**, **compton** and **eels**, corresponding to the last field of the CONTROL card:

```
CONTROL 0 0 0 0 0 1 * calculate the spectrum from G
```

The modules **ff2x**, **sfconv**, **compton** and **eels** construct the XAS spectrum $\chi(k)$ or μ using the XAFS parameters described in Section 4.2.5 from one or more paths, and adding the FMS contributions. Single and multiple scattering Debye–Waller factors are calculated using, for example, the correlated Debye model. The spectrum can be found in file **xmu.dat** for most types of spectroscopy. Numerous options for filtering, Debye–Waller factors, and other corrections are available. The details depend on the type of spectroscopy. For more details and examples, we refer to the section below that is appropriate for your type of calculation.

??FIX Add a paragraph or two about sfconv either here or in a section of itself??

2.7 EXAFS Calculation

Self-consistent or overlapped atom potentials are necessary for the calculation of the scattering phase shifts. Self-consistent calculations (using the SCF card) take more time. 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.

The calculation of potentials and phase shifts can be 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**.

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.

2.7.1 SF₆ Molecule

SF₆ Molecule. This is the simplest example of running FEFF to obtain results for EXAFS. Only two input cards, POTENTIALS and ATOMS, are strictly necessary. However it is good practice to list the cards EXAFS and RPATH. The output χ can be found in the file ‘chi.dat’.

```
TITLE Molecular SF6
```

```
POTENTIALS
```

```
* absorbing atom must be unique pot 0
```

```
*   ipot    z    tag
      0     16    S
      1      9    F
```

```
EXAFS 20
```

```
RPATH 10
```

```
EDGE K
```

```

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

```

2.7.2 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 4 and the **PRINT** card on page 75.

```

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
EDGE K

```

```

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 07      Cu2 core hole
EDGE K
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  0
      2    39  Y
      3    29 Cu1
      4    56 Ba

```

ATOMS

atoms list generated by the following atoms.inp file

```

-----
title YBCO: Y Ba2 Cu3 07  (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      01
  O          0      0      0.158    02
  O          0      0.5    0.379    03
  O          0.5    0      0.377    04
-----

```

2.7.3 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 $S_0^2 < 0.1$ by the one of two possible ways shown below.

```

EDGE  L3      0.0
S02    0.0

```


The result for S_0^2 is given in 'chi.dat' or 'xmu.dat'. 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.

2.7.4 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 3.3. This type of calculation is currently curtailed by the limited functionality of the CFAVERAGE card, which should be used with caution. Please contact the authors if problems occur.

FIX we need an example here

2.7.5 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 3.5.2 and 3.5.4.

2.8 XANES Calculations

2.8.1 Need for SCF and Additional Difficulties for XANES

XANES calculations usually take more time and require more experience than EXAFS calculations. They require 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). **FEFF10** thus automatically accounts for charge transfer. The **ION** card should be used only to specify the total charge of

a cluster. Overlapping of muffin tins (AFOLP) leads to better results for XANES and is done by default.

The high order MS path expansion can lead to unreliable XANES calculations when the MS series converges poorly (as is often the case near the Fermi level). Thus using Full Multiple Scattering (FMS) instead of the path expansion is essential for calculations of ℓ DOS and electronic densities and is usually an improvement for XANES. We suggest to use FMS exclusively and uses path expansion for testing its convergence. The FMS calculations 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 100-300 atoms. Calculation time scales as a third power of the number of atoms in a cluster and quickly becomes expensive. However, the description of the FMS card lists options for iterative matrix solvers that improve the scaling somewhat. Here are sample input files for XANES calculation.

2.8.2 GeCl_4 Molecule

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

```

TITLE    GeCl_4  r=2.09 /AA

COREHOLE none
EDGE K    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
    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

2.8.3 Solid: XANES and LDOS

BN crystal has a zinc sulfide structure. The multiple scattering path 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
EDGE K  1.0      *  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
-----
```

2.8.4 Cross-section in absolute units

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

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

Since our distances are in Å, we report the XAS cross section also in Å². If you multiply the 4-th or 5-th column by this normalization value you will obtain the cross section in Å². Literature often reports the absolute cross section in barns (1 Å² = 100Mbarn).

Note that this normalization can be switched off using the ABSOLUTE card. This is set by default for NRIXS, ELNES and EXELFS calculation. The normalization factor then shows as "1.0".

As of FEFF9.1, EELS spectra in 'eels.dat' are always given in units of a_0^2/eV . Multiply by 28.0010^{-18} to get units of cm^{-2}/eV . Or multiply by 28 to get units of $Mbarn/eV$.

2.9 Spin-dependent Calculations

2.9.1 General Description

This section contains information on extracting the XMCD signal and on the SPXAS technique, including example input files.

Spin-dependent calculations are automated. All spin-dependent calculations require that the SPIN card be present in 'feff.inp' (see the SPIN card in Section 3.5.2). The method depends on the value of the parameter `nspx` in the header file 'feff10/src/Common/m.dimsmod.f90'. Please contact the authors if you need help modifying the source code.

In order for the final result to be contained in 'xmu.dat', FEFF must be compiled with `nspx = 2` to combine 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.

With `nspx = 1`, one can run the code twice, once for spin-up and once for spin-down. Spin-flip terms are not calculated, but by making the proper combination of spin-up and spin-down spectrum, one obtains XMCD or SPXAS (Section 2.9.3). A simple program 'spin.f', printed in the appendix and also on the FEFF (website), can do this. 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.

2.9.2 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
```

```

XMCD
EDGE L1  1.0    *  s0^2=1.0
SPIN      1
EXCHANGE  2
RGRID .01

CONTROL  1      1      1      1      1      1
RPATH    7.29
CRITERIA 0.0  0.0  curved  plane
DEBYE    150   176  temp    debye-temp
XANES

POTENTIALS
*  ipot  z  label
    0   64  Gd
    1   64  Gd

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

```

2.9.3 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 (Wyckoff)

EDGE K  1.0    *  s0^2=1.0
SPIN    2

```

```

CONTROL  1      1      1      1      1      1
RPATH  10.0
XANES
PCRITERIA      0.8      40.0
CRITERIA      0.0      0.0
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

```

2.10 Elastic Scattering Amplitudes

All necessary components to obtain the elastic scattering amplitude can be calculated. 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.

An example can be found in the `~/feff10/examples` directory of your FEFF installation.

2.11 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. Please report any problems with this card to the authors.

An example can be found in the `~/feff10/examples` directory of your FEFF installation.

2.12 Calculation of EELS

Electron-energy loss spectroscopy (EELS) measures the spectrum of energy losses of a beam of high-energy electrons passing through a sample in an electron microscope. We include relativistic effects in the cross section. FEFF is optimized for EELS calculations and includes various instrumental parameters

To calculate EELS, add either the **ELNES** card or the **EXELFS** card to the ‘`feff.inp`’ input file. For the near-edge region, full multiple scattering is appropriate:

```
FMS 7.0
ELNES ...
```

While the extended region requires the path expansion:

```
RPATH 8.0
EXELFS ...
```

The EELS spectrum is written to the ‘`eels.dat`’ file in absolute units. See the file header for more details. Old versions of FEFF used arbitrary units and may have a wildly different order of magnitude.

The EELS engine calculates the Σ tensor, containing 'basis spectra' for all 9 polarization components. EELS calculations cannot be combined with the **POLARIZATION** or **ELLIPTICITY** cards. The corresponding variables (the polarization vector resp. the beam direction) are set internally based on the options of the **ELNES** card or the **EXELFS** card.

2.12.1 Changing EELS parameters

If you change something in the calculation of the material properties - such as atom positions, or the FMS radius - you need to rerun all or most of FEFF. If you only change the experimental setup (i.e., beam direction, beam energy, detector aperture or position, or beam convergence angle ; or toggle relativistic cross-section on/off) but leave the sample unchanged, then you need only do the following :

- Edit '**feff.inp**'
- Run module **rdinp** to update '**eels.inp**' (alternatively, you can edit '**eels.inp**' directly—but beware, this file is somewhat format-sensitive)
- Run module **eels** to calculate a new EELS spectrum.

The **eels** module is very fast and can be looped over to fit some of the experimental parameters mentioned.

2.12.2 Example input.

See the **example file** in the k-space chapter.

2.13 NRIXS

Non-Resonant Inelastic X-ray Scattering (NRIXS) is determined by the momentum transfer dependent dynamic structure factor $S(\vec{q}, E)$ (DFF). FEFF NRIXS calculations output the DFF for a given \vec{q} over a range of E . From this quantity, the NRIXS signal is obtained by simply adding a multiplicative factor containing beam energies and polarization. (See references.) NRIXS calculations are controlled by the following cards : NRIXS, LJMAX, and LDEC. In addition, one must use either the XANES and FMS cards for the near-edge region ; or the EXAFS and RPATH cards for the extended region. Note that for NRIXS, if $xkmax < 0$ in the XANES or EXAFS card, then an energy grid of approximately constant energy step is used (instead of the usual constant k-step).

Although the code has been tested for wide range of momentum transfers the code is most stable for the medium region of momentum transfer values. This typically means $0.1 < q < 14a.u.$ for light elements or shallow edges, i.e. binding energy less than 400 eV. Higher momentum transfers are possible for more tightly bound electrons (see examples below). The small momentum transfers can cause numerical instabilities at near edge region and large

momentum transfers also in the extended energy range. There is no default value for the momentum transfer.

NRIXS produces two main output files. ‘xmu.dat’ contains the total spectrum in the usual 6-column format. ‘xmul.dat’ contains:

- col 1: the excitation energy (in eV).
- col 2: the value of k (in Angstrom) at this energy.
- col 3: the atomic background value $S_0(\vec{q}, E)$
- col 4 - 4+ld: the next $ld + 1$ columns give the contribution to the atomic background $S_l^0(\vec{q}, E)$ for the final state electron angular momentum values $l = 0, \dots, ld$:

$$S_l^0(\vec{q}, E) = (2l + 1) |M_l(\vec{q}, E)|^2 \rho_l^0(E). \quad (2.1)$$

- col 4+ld+1 - end: the next $(ld+1)(ld+1)$ columns give the decomposition of fine structure i.e.

$$\chi_{\vec{q}}^{ll'}(k) = \frac{1}{S_0(\vec{q}, E)} \sum_{mm'} M_L(-\vec{q}, E) \rho_{LL'}^{sc}(E) M_{L'}(\vec{q}, E) \quad (2.2)$$

for $l \leq ld$ and $l' \leq ld$.

To plot the contribution of only s-type ($l = 0$) final states one would do in gnuplot ($ld = 2$):

```
> plot xmul.dat u 1:($4+$3*$7)
```

and for only p-type:

```
> plot xmul.dat u 1:($5+$3*$11).
```

To get the s-p non-diagonal contribution to $S_0(\vec{q}, E)\chi_{\vec{q}}(k)$:

```
> plot xmul.dat u 1:($3*($10+$8)).
```

2.13.1 Some example input

Here is the standard FEFF example for the GeCl4 molecule.

```
TITLE GeCl_4 r=2.09 /AA
COREHOLE none
EDGE K 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
  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

* Choose one of these two options :
* 1/ calculation with 24 a.u. momentum transfer along the z-axis
NRIXS 1 0.0 0.0 24.0
LJMAX 10
* 2/ Since this is a molecule (gas phase or liquid) we really should average
* over the direction
NRIXS -1 24.0
LJMAX 10
LDEC 2

END

```

2.13.2 Limitations and practicalities.

The NRIXS code requires substantial stack space. We recommend changing the available stacksize to unlimited using shell commands (in tcsh)

```
> limit stacksize unlimited
```

or (in bash)

```
> ulimit -s unlimited.
```

The following input cards cannot currently be used for NRIXS: CFAVERAGE, SPIN, ELIPTICITY, MULTIPOLE, POLARIZATION, TDLDA, RPHASES, XES, XMCD.

The NRIXS code cannot run in k-space.

The NRIXS code can simultaneously calculate a list of q-vectors if run from the command line.

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

These are features of FEFF that are still under development. They are not reliable in this release of FEFF10.

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 FEFF10.

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)

2.15 Using a .cif file

The ‘*.cif’ file format is a standard way of specifying structural data. Such files are available in online databases for a very large number of materials. FEFF can read such files directly, removing the need for the user to worry about, e.g., details of crystallography and space group notations to generate a list of cluster coordinates for ‘feff.inp’.

The ‘*.cif’ file import feature is currently only implemented for calculations of crystals. We intend to enable it for molecules also; please contact us for collaboration if you have an interest in this. ‘cif’ files can be used for real-space or k-space calculations, for any type of spectrum.

FEFF uses the CIFTBX library to read ‘.cif’ files. This should allow you to use any valid ‘.cif’ file. ”Warning - library not found” warnings at runtime can usually be ignored. The [Open Crystallography Database](#) is an excellent resource for ‘.cif’ files that can be accessed from anywhere. The [Inorganic Crystal Structure Database](#) is another good resource for those with an academic subscription. We recommend saving the ‘.cif’ file to the working directory containing also ‘feff.inp’ and renaming it with a useful filename.

Use the CIF CARD as in the **Cr2GeC example** below. You cannot use an ATOMS card in the same ‘feff.inp’ file. It is no longer necessary to use a POTENTIALS card. FEFF automatically assigns a potential type to each crystallographically inequivalent atom in the unit

cell. These potentials are given the default angular cutoffs (l_{max}) based on atomic number. Inspecting ‘`pot.inp`’ or ‘`log1.dat`’ shows the list of assigned potentials. Note that the current scheme could be problematic for large unit cells, where there might be many crystallographically inequivalent atoms of the same atomic number. Calculating a large number of potentials will make the calculation slow and potentially unstable (the SCF algorithm may have a hard time reaching convergence). Furthermore, in our experience respecting strict crystallographic equivalence does not always improve the accuracy of a calculation. For such large systems, it may be preferable to make all atoms of the same atomic number equivalent, or to determine equivalence based on first-shell coordination only. The EQUIVALENCE card achieves that end.

If both a CIF card and a POTENTIALS card are present in ‘`feff.inp`’, the program does the following. First, it determines if the list of potentials in the POTENTIALS card matches exactly that generated from the CIF file. If it does not match, the POTENTIALS card is ignored and a warning is printed to the screen. If the list of potentials does match, the program takes the options for the potentials (i.e., `lmax1 lmax2 xnatph spinph`) and uses these values instead of the defaults generated from the CIF file. This way, the user can control the angular momentum cutoff and set the spin-related variables.

Finally, a word of warning on using the TARGET card to indicate the corehole atom. Combined with the CIF card, this counts an atom in a list of the crystallographically inequivalent atoms in the unit cell (e.g., 2 C atoms for graphite) in the order given in the CIF file. However, combined with LATTICE/ATOMS cards, it counts an atom in a list of all atoms in the unit cells (e.g., 4 C atoms for graphite, 2 of which are equivalent by symmetry to the other 2) in the order given in the ATOMS card. Hence, the value of the TARGET card must be reevaluated if you switch from one representation to the other. Note that you must look at the file yourself to find out in what order the atoms are listed and identify which one you want to place the core hole on. If it is the third atom listed in the CIF file, use “TARGET 3” in ‘`feff.inp`’.

2.15.1 Formal requirements for cif files

Although CIF is a mature standard, there is a wild proliferation of options and input fields for cif files. We’ve come across some files that seem to contain endless “junk” entries, or that enter data in sufficiently nonstandard ways to confuse FEFF.

With most cif files there is no problem, although we sometimes delete fields we don’t need just for the sake of simplicity. When there is a problem, it is almost always due to one of two things: * The H-M space group is specified in a nonstandard way, often with extra “:1” or “ 1” symbols: the solution is to delete these extra characters; * The atomic symbols are not specified: in this case, one simply adds them.

The Cr2GeC example shows a good cif file. The following fields are required by the FEFF parser:

```
_cell_length_a, _cell_length_b, _cell_length_c
_cell_angle_alpha, _cell_angle_beta, _cell_angle_gamma
_atom_site_type_symbol, _atom_site_label, _atom_site_fract_x, _atom_site_fract_y, _atom_site_fract_z
```

```
_symmetry_space_group_name_H-M or _symmetry_Int_Tables_number
_symmetry_equiv_pos_as_xyz
```

Most cif files contain much, much more information than this.

2.16 K-space FEFF

Although FEFF is traditionally a real-space code, FEFF10 is capable of calculating infinite periodic systems (crystals) in reciprocal space. The reciprocal space engine is based on impurity KKR. This functionality is implemented through a handful of new input CARDS. The main difference from the real-space code is that the system is specified in terms of a unit cell with lattice vectors and a basis of atoms in the unit cell, instead of a real space cluster. We describe how to do this below.

First, the RECIPROCAL card switches FEFF to work in k-space.

Next the crystal structure must be specified. This can be done in two ways. The recommended way (new as of FEFF-9.5.1) is to use the CIF card to specify the name of a ‘*.cif’ file containing the crystal structure. The second way, which was the default in prior FEFF versions but is less user-friendly and more error-prone, is to use the following CARDS in ‘feff.inp’: the LATTICE card gives the basis vectors spanning the unit cell; and the ATOMS card lists all atoms in the unit cell.

In both cases, the TARGET card places the absorber on one of the sites (either the list of sites in the ATOMS card, or the list of sites in the ‘.cif’ file); and the KMESH card sets the size of the mesh used to sample Brillouin Zone integrals. There are also a few optional cards, notably COORDINATES (sets the units for the ATOMS card if one uses the LATTICE/ATOMS approach), and SGROUP (only for the LATTICE/ATOMS approach). STRFAC is an advanced card that adds internal broadening for convergence of lattice sums and is not usually needed.

If a core hole is needed, we recommend using COREHOLE RPA. Note that NRIXS cannot currently be run in k-space.

2.16.1 Core hole calculations without the supercell

It is typical to calculate spectra of crystals using band-structure codes, which calculate crystals very efficiently in k-space using periodic boundary conditions (PBC). But the introduction of a core hole is in essence identical to an impurity calculation and breaks the periodicity of the ground state of the crystal. The typical solution is to construct a supercell large enough to separate the core hole atom from its twin in the neighboring unit (super)cell so that unphysical interactions are avoided. This approach is inefficient and can be computationally expensive. It is complicated by the absence of a universally appropriate size for the supercell, requiring a convergence study.

The FEFF code operates similarly to impurity KKR calculations and is able to combine the best of both worlds. The Green’s function of the ground state crystal is first calculated in k-space and then transformed back to r-space. The core hole impurity is added to the r-space

Green's function through simple matrix algebra. This is inexpensive and avoids the need for a supercell altogether. Note that this requires using the "COREHOLE RPA" setting. This is the recommended way of calculating a core hole spectrum of a crystal in FEFF. (The default "COREHOLE FSR" setting puts the core hole explicitly in the unit cell and would correspond to the bandstructure code calculation.) No further user effort is required.

2.16.2 Using the file 'reciprocal.inp'

This intermediate file written by **rdinp** contains all the parameters for k-space calculations. The first parameter is the main switch : if it's set to 0, the calculation will be done in real space and all the parameters following it are ignored. If it's set to 1, the calculation is done in reciprocal space, and all the parameters following it are read and used. This allows the advanced user some leeway ; one could eg. calculate potentials in reciprocal space and FMS in real space by toggling this variable (perhaps for NRIXS).

2.16.3 Converging the k-mesh

The k-mesh is constructed using the tetrahedron of Bloechl et al., Phys. Rev. B, 1990. It is written to file early on. The number of k-points in the k-mesh is an unphysical parameter that simply needs to be converged. Although it is impossible to give a general guideline, starting with 1000 k-points is a good idea for small unit cells. Generally, the number of points needed scales inversely with the volume of the unit cell. Some systems require more points than others. One always needs to check. The more broadened the property of interest (e.g. ELNES as opposed to DOS), the fewer points are necessary. Also, the near edge structure requires more points, whereas more extended structure (e.g., 50-70 eV above threshold) is often converged with just a few k-points.

Generally speaking, the calculations of the potentials requires less accuracy than the calculation of FMS. Just like one usually uses a smaller SCF-radius than the FMS-radius for real-space calculations, it makes sense to use a smaller k-mesh for SCF than for FMS. Therefore, it can be a good strategy to, e.g., set the k-mesh to 200 points, run the potentials calculation, then raise the number of k-points to, e.g., 1000, and then run the FMS calculation. This can save much calculation time, but requires more skill from the user.

Some time-saving schemes are implemented through the KMESH card and are described there.

2.16.4 EXAFS and EXELFS

For extended loss structure (i.e., upwards from 50-100 eV), the real space Path Expansion method is so efficient and robust that we do not see the point in trying to reformulate it in reciprocal space. Therefore, it is always done in real space. If the RECIPROCAL card is active, the **rdinp** module generates a real-space cluster based on the crystal structure and the value of the RPATH value. This cluster is written to the atoms.dat file. The spectrum is then calculated by the **path** and **genfnt** modules, which always work in real space. It is good

practice to look at this atoms.dat file when one is still learning to use k-space FEFF - it's a good check to make sure no mistakes have been entered.

2.16.5 Using crystal symmetry

Symmetry could be used in two ways :

- reduce the k-mesh
- reduce the Green's function L, L' matrix

The first of these is currently implemented. However! The extent to which symmetry can be used depends on what one is calculating. E.g., to calculate the diagonal parts of $G_{LL'}$, one can reduce the k-mesh and just sum all the contributions from inequivalent k-vectors. But for the offdiagonal components (which are needed in order to add the core hole), symmetry is more subtle and one needs to add all the equivalent ones, which can be somewhat shortcut by reconstructing them from the inequivalent ones through unitary transformations dictated by symmetry elements of the crystal. Whereas the first strategy would yield a speedup $\times 48$ for diamond, the gain for the latter strategy is much more modest ($\times 5$?) and depends on the size of the matrices, i.e. the number of atoms in the unit cell.

So, while a number of symmetry strategies are implemented in the code, it is currently recommended to test these cautiously on a smaller k-mesh before relying on them. Testing against real-space results is another safety check. We have not tested all possible lattice types and errors may occur. Contact the authors in case of doubt. Note that many symmetry options are disabled in the code for safety reasons and would require recompilation.

The second idea - LL' symmetry within the G matrix - has not been implemented in any way.

2.16.6 Speed

Generally, **fms** matrix inversion takes most of the computation time. In real space, for every energy point FEFF does one matrix inversion of order $nclus*(lmax+1)**2*nspace$. In reciprocal space, for every energy point FEFF does nkp matrix inversions of order $nu*(lmax+1)**2*nspace$. Here, nu is the number of atoms in the unit cell, nkp the number of k-vectors in the mesh, $nclus$ the number of atoms in the real space cluster, $lmax$ the angular momentum cutoff, $nspace$ the number of spins (1 or 2). So, the relative speed is something like $(nu/nclus)^a * nkp$, where a is the scaling of matrix inversion. In general, unit cells with more atoms (nu) are larger and therefore require less k-vectors (nkp). For small systems (1-10 atoms in the unit cell), RECIPROCAL is faster. For larger systems, it depends.

2.16.7 Example input files

The Ge L3 edge of Cr2GeC using a Cr2GeC.cif file. This is the only kind of FEFF calculation that doesn't require a POTENTIALS card and an ATOMS or OVERLAP card.

```

TITLE Cr2GeC      (a=2.94 c=12.11)
* Ge L3 edge energy = 1217.0 eV
EDGE L3
S02 1.0
COREHOLE None
XANES 20.0 0.07 0.0

CONTROL 1 1 1 1 1 1
PRINT 5 1 1 1 1 1

SCF 4.0
FMS 6.0
LDOS -30 15 0.01

* Options for a k-space calculation :
RECIPROCAL
* Use 200 k-points:
KMESH 200
* Spectrum of 3rd atom type in the cif file (Ge) :
TARGET 3
* This advanced option is not usually necessary:
STRFAC 1.0 0.0 0.0

* Read crystal structure from cif file:
CIF Cr2GeC.cif
END

```

The 'Cr2GeC.cif' file :

```

data_Cr2GeC
_cell_length_a          2.9400(0)
_cell_length_b          2.9400(0)
_cell_length_c          12.1100(0)
_cell_angle_alpha       90.0000(0)
_cell_angle_beta        90.0000(0)
_cell_angle_gamma       120.0000(0)

_symmetry_space_group_name_H-M    'P 63/m m c'
_symmetry_Int_Tables_number       194
_symmetry_cell_setting            hexagonal
loop_
_symmetry_equiv_pos_as_xyz
'+x,+y,+z'
'-y,+x-y,+z'

```



```

'-x+y,-x,+z'
'-x,-y,1/2+z'
'+y,-x+y,1/2+z'
'+x-y,+x,1/2+z'
'-y,-x,+z'
'-x+y,+y,+z'
'+x,+x-y,+z'
'+y,+x,1/2+z'
'+x-y,-y,1/2+z'
'-x,-x+y,1/2+z'
'-x,-y,-z'
'+y,-x+y,-z'
'+x-y,+x,-z'
'+x,+y,1/2-z'
'-y,+x-y,1/2-z'
'-x+y,-x,1/2-z'
'+y,+x,-z'
'+x-y,-y,-z'
'-x,-x+y,-z'
'-y,-x,1/2-z'
'-x+y,+y,1/2-z'
'+x,+x-y,1/2-z'

```

```

loop_
_atom_site_type_symbol
_atom_site_label
_atom_site_fract_x
_atom_site_fract_y
_atom_site_fract_z
  C      C      0.0000    0.0000    0.0000
  Cr     Cr     0.6667    0.3333    0.0833
  Ge     Ge     0.6667    0.3333    0.7500

```

(Most 'cif' files contain many additional fields. We have removed all the information that is ignored by FEFF.)

The graphite C K edge using LATTICE and ATOMS cards.

```

TITLE graphite
* C K edge energy = 284.20 eV
EDGE      K      0.0
COREHOLE RPA
CONTROL   1  1  1  1  1  1

```

```

ELNES 5.0 0.05 0.05
300 # beam energy in keV
0 0 1 # beam direction in the crystal frame
1.10 0.3 # collection semiangle, convergence semiangle (in mrad)
100 1 # q-integration mesh : radial size, angular size
0.0 0.0 # position of the detector (x,y angle in mrad)

MAGIC 40 # create plot that shows magic angle. Evaluate at 40 eV above threshold.

FMS      6.0
SCF      4.5

POTENTIALS
*   ipot   Z   element          l_scm  l_fms  stoichiometry
      0     6   C              3      2      0.01
      1     6   C              3      2      2
      2     6   C              3      2      2

RECIPROCAL
KMESH 1000
TARGET 1
LATTICE P 2.456
      0.86603      -0.50000      0.00000
      0.00000      1.00000      0.00000
      0.00000      0.00000      2.72638

ATOMS
*   x          y          z          ipot  tag
      0.00000      0.00000      0.68160  1  C1
      0.00000      0.00000      2.04479  1  C1
      0.57735      0.00000      0.68160  2  C2
      0.28868      0.50000      2.04479  2  C2

END

```

2.17 Charge Transfer and Charge Counts

The charge transfer in each 'ldosNN.dat' file and in 'log1.dat' is related to the amount of charge transferred into the Norman radius during the SCF routine. Before the SCF routine, the Norman radius is the radius of the neutral sphere, so the charge transfer can be related to the electronegativity or the oxidation state, although the actual numbers are usually much smaller than the formal oxidation state. The electron counts give the number of valence electrons in each of the angular momentum channels after the SCF has completed. The valence electrons are assumed to be those binding binding energy above and energy "ecv", which is usually at

-40 eV, but can sometimes change if the atoms in the problem have binding energies close to -40eV. The difference between these numbers and the number of valence electrons in these channels in the atomic system should be equal to the charge transfer into each channel. If these are summed over angular momentum channels, you should get the total charge transfer. The only case where this is different is in the absorbing atom, which starts with an extra electron in the valence. As an example, we can look at NaF which gives

Electronic configuration

iph	il	N_el
0	0	0.521
0	1	6.535
0	2	0.358
0	3	0.000
1	0	1.958
1	1	5.285
1	2	0.014
1	3	0.000
2	0	0.221
2	1	6.292
2	2	0.236
2	3	0.000

Here iph = 0 is the absorbing Na atom, iph = 1 is the F atom, and iph = 2 is the other Na atom. These atoms start with the configuration

Electronic configuration

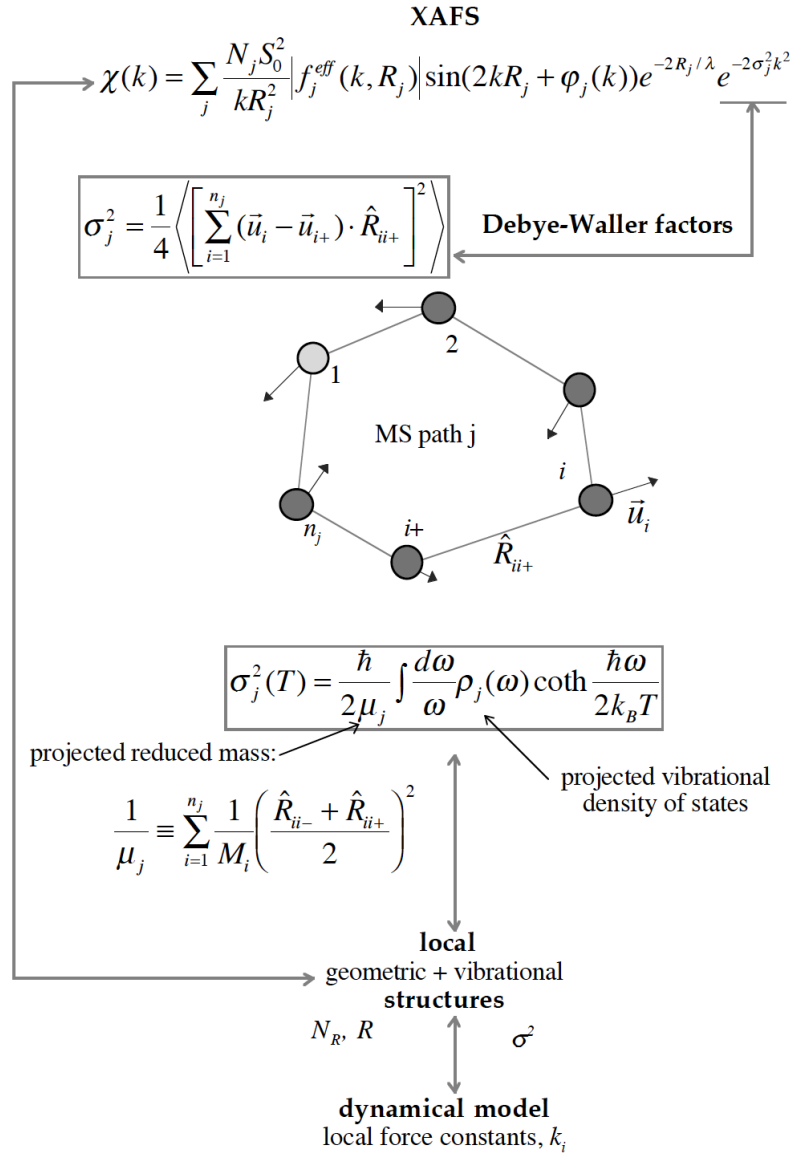
iph	il	N_el
0	0	2.0 - extra screening electron
0	1	6.0
0	2	0.0
0	3	0.0
1	0	2.0
1	1	5.0
1	2	0.0
1	3	0.0
2	0	1.0
2	1	6.0
2	2	0.0
2	3	0.0

Looking at the difference between these two, you can see that in the solid state, some electrons are transferred from the sodium s to the sodium p and d states, and some to the F p states. There is also a small transfer from the F s states, and into the F d states, but minor. Note that the final configuration allows more than 6 electrons in the Na p states. This is because the p-states are not limited to the 2p states, but include the 3p as well.

2.18 Ab initio Debye-Waller factors

XAFS analysis can provide structural information, including average near-neighbor distances R , their mean square fluctuations σ_R^2 , and coordination numbers N_R . The quantities σ_R^2 that appear in the XAFS Debye Waller(DW) factor are crucial to the success of the modern theory of XAFS and its applications. The DW factor accounts for thermal and structural disorder and generally governs the "melting" of the XAFS oscillations with respect to increasing temperature and their decay with respect to increasing photoelectron energy. In practice, the DW factors of the many multiple-scattering terms in the XAFS signal can significantly complicate the analysis. To overcome these difficulties, FEFF offers several ways to calculate the Debye-Waller factors and account for the effects of thermal disorder in the *ab initio* XAFS calculations. These are described below.

Fig. 2.18 illustrates the way DW factors enter the XAFS problem.



If one uses the multiple scattering path expansion (PE), the DW factors are added to each path individually in module **ff2x**. If one uses Full Multiple Scattering (FMS), the effect of finite temperature is approximated by multiplying each free propagator by $e^{-\sigma^2 k^2}$. This is only exact for single scattering paths, but since the effect of thermal disorder is reduced in the near-edge region anyway, it's probably adequate.

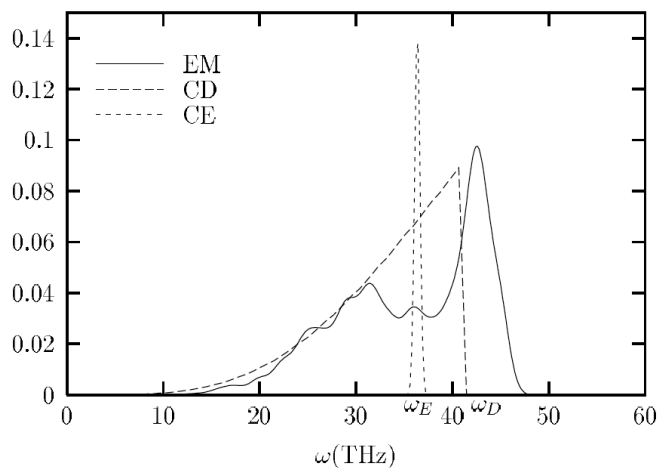
There are three ways to modify the Debye–Waller factor. The **DEBYE** card calculates a Debye–Waller factor for each path in PE or FMS. The **SIG2** and **SIG3** and **SIGGK** card add various constant or near-constant Debye–Waller factors to all paths in PE. Finally, you

can edit ‘`list.dat`’ to add a Debye–Waller factor to a particular path in PE. In PE, these three Debye–Waller factors are summed, so if the DEBYE and SIG2/SIG3/SIGGK cards are present, and if you have added a Debye–Waller factor to a particular path in ‘`list.dat`’, the Debye–Waller factor used will be the sum of all three.

After changing the DW factors, FEFF must be rerun starting with module **ff2x** for PE and starting with module **fms** for FMS.

The **DEBYE** card offers a choice between 5 different models for the DW factors:

- 0 Correlated-Debye method (default) (CD)
- 1 Equations of Motion method (EM)
- 2 Recursion method (RM)
- 3 Classical Correlated-Debye method (CCD)
- 4 Read from “sig2.dat” file
- 5 Dynamical-Matrix method (DM)
- ¡0 Do not calculate DW factors



Only method CD and CCD can run without additional input. These Correlated Debye models are isotropic and can be very inaccurate for anisotropic materials. Methods 1, 2 and 5 require that the force constants or the dynamical matrix be provided. We will now describe these requirements from a computational point of view. For more information on the physics behind these calculations, we refer to [Anna Poiarkova’s thesis](#) and our [paper on the DM method](#).

2.18.1 EM and RM methods

The Equation of Motion (EM) and Recursion Method (RM) require additional input to be given in the file ‘spring.inp’. The file defines the force fields for FEFF. First, FEFF searches for all similar bonds and angles in the material, creates complete lists of all bond stretches and angle bends, and then, based on the force field and geometry of the structure, calculates the cartesian force field matrix and scales it with the atomic masses, thus obtaining the dynamical matrix. Finally, the DW factors are calculated from the dynamical matrix.

The file ‘feff.inp’ has the same CARD-based structure as the master input file ‘feff.inp’. For example, the ‘spring.inp’ file for zinc tetraimidazole looks like this.

```
* 13-atom model of zinc tetraimidazole
*
      res      wmax      dosfit      acut
VDOS      0.02      1      1.2      3
PRINT 5
STRETCHES * i      j      k_ij      dR_ij [%]
           0      2      110.      2.
           1      2      626.      5.
ANGLES * i      j      k      ktheta      dtheta [%]
        2      0      5      37.      10.
        1      2      3      2590.      10.
```

The corresponding ‘feff.inp’ file contains the DEBYE card:

```
*      T      T_Debye      "2"=Equation of Motion Method
DEBYE      300.      0.      2
```

The CARDS for the ‘spring.inp’ file are:

◆ **VDOS** *res wmax dosfit [acut]*

Standard

This card is needed only for EM runs and is optional - if it is omitted, default values are used. The card is ignored for RM runs. The keywords in the VDOS card define the integration parameters used in the VDOS calculation. Here **res** is the VDOS spectral resolution width (default **res** = 0.05, i.e. 5% of the bandwidth). The smaller this number, the more fine structure is present in the spectrum and the longer the computation time. Finer resolution is usually helpful for mode analysis in small molecules. The next keyword, **res**, is a multiplication factor used to increase the maximum frequency to which the VDOS is calculated. **dosfit** is a real positive number governing how much of the low frequency part of the VDOS is to be fitted to Debye-like behavior, $A\omega^2$. If it is equal to 0 then no fitting will be applied. The higher the number, the more of the VDOS will be fitted. The default value is **dosfit** = 1 (about 10% of the total width). This parameter is useful for elimination of low frequency "noise" and zero-frequency modes. Finally, **acut** is the time integration cutoff parameter. It rarely needs to be changed (usually in cases of very small open molecular structures). The higher this number, the longer the computation time. The **acut** keyword is optional; the default value is **acut** = 3.

*	res	wmax	dosfit	acut
VDOS	0.02	1	1.2	3

◆ PRINT [iprdos]

Standard

If using the EM method, the PRINT card makes FEFF write files 'prdenNNNN.dat' containing projected VDOS for selected scattering paths. Here `iprdos` specifies that such files will be written for the first `iprdos` paths in the paths list. If using the RM method, the PRINT card makes FEFF write a file 's2_rm1.dat' containing first tier results. The keyword `iprdos` is ignored in this case.

```
PRINT 5 * print files prden0001.dat - prden0005.dat
```

◆ STRETCHES [i j k_ij dR_ij]

Standard

Required for EM and RM runs. It is followed by the list of bond stretching force constants. Here i and j are atomic indices (as in the file 'geom.dat'; the absorber has index 0), and k_{ij} is a single central force constant characterizing the interaction between atoms i and j in units of $10^2 \text{mdyn}/\text{\AA}^2$ or N/m . One should include as many distinct bonds in the list as possible and then the code will search for the similar ones and assign them the same force constant. The last parameter in each row, dR_{ij} , is the tolerance in the bond length when searching for similar bonds and is measured in percentage points. For example, if $dR_{ij} = 5$ then all bonds between pairs of atoms with the same potentials as i and j and with bond length within 5% of R_{ij} will be assigned the same stretching force constant k_{ij} .

STRETCHES	*	i	j	k_ij	dR_ij [%]
		0	2	110.	2.
		1	2	626.	5.

◆ ANGLES [i j k ktheta dtheta]

Standard

Similar to STRETCHES, but optional in most cases. This card allows one to include θ_{ijk} angle bending force constants k_{θ}^{ijk} in the calculation. The force constants are in units of $10^2 \text{mdyn}\text{\AA}/\text{rad}^2$. Here $dtheta$ is tolerance in the angle value when searching for similar angles. Sometimes it is useful to include this card in order to avoid zero-frequency modes.

ANGLES	*	i	j	k	ktheta	dtheta [%]
		2	0	5	37.	10.
		1	2	3	2590.	10.

Output files

All σ_j^2 values in the output files are given in units of \AA^2 , all frequencies are in units of THz, and all reduced masses are in atomic units. The output files from a EM calculation include:

- ‘s2_em.dat’ Contains σ_j^2 for each scattering path in ‘paths.dat’.
- ‘prdenNNN.dat’ which contains projected VDOS for selected scattering paths (as indicated by the PRINT card).

The output files from a RM calculation include:

- ‘s2_rm2.dat’ Contains σ_j^2 for each scattering path in ‘paths.dat’ calculated using the second tier approximation.
- ‘s2_rm1.dat’ Contains σ_j^2 for each scattering path in ‘paths.dat’ calculated using the first tier approximation.

Further details can be found on pp. 98 of [Anna Poiarkova’s thesis](#)

Examples

These examples can also be found in the ‘examples’ folder of the feff10 distribution.

First we present a ‘spring.inp’ example for a 177-atom cluster of a Cu crystal. Here only a single central force constant between the first nearest neighbors is taken into account. Similar input files can be constructed for other fcc structures, e.g. Pt, Al, etc.

```
* Cu crystal, single central force constant
VDOS  0.03  0.5  1
PRINT  3
STRETCHES
0  1  27.9  2.
```

As a second example we show the ‘spring.inp’ file for a 147-atom cluster of c-Ge crystal. The force constants used here were fitted to phonon dispersion curves. Similar output files can be constructed for other diamond-type lattices, e.g. Si, C, etc.

```
* c-Ge crystal
VDOS  0.02  0.7  0.  3.
PRINT  6
STRETCHES
0  1  103.58  2.
0  5   5.81  2.
0  20  -1.08  2.
0  30  -0.30  2.
ANGLES
1  0  2  31.45  2.
```

The output files for these examples can be found in the ‘examples’ folder of the feff10 distribution. They are also discussed in pp. 100-104 of [Anna Poiarkova’s thesis](#).

2.18.2 DM method

Preparation

First, the user needs to calculate the dynamical matrix. FEFF cannot do this, but many other codes can, e.g. ABINIT or Gaussian. Using such a code, the user needs to produce a file that contains the dynamical matrix, and which we will refer to as the ‘dymfile’.

Before explaining how to include dynamical matrix DW factors in EXAFS and XANES FEFF calculations, it is important that the user becomes familiar with some constraints and limitations that are present in the current implementation. These limitations stem from the fact that both the dynamical matrix file and the FEFF input file contain structural information. These structures must match for the DW values to be correct. Since FEFF internally sorts the atoms according to distance to the absorber, special care must be taken to avoid mismatching the structural information coming from the FEFF input with that from the ‘dymfile’.

Therefore, it is not recommended to start from an existing ‘feff.inp’ file, add a ‘dymfile’, and run the calculation. Instead, we recommend to start from the dymfile, and generate correctly matched ‘feff.inp’ and ‘feff.dym’ files from the dymfile. Then we can edit the newly generated ‘feff.inp’ file to add CARDS, before running the FEFF calculation.

dym2feffinp is a utility that helps in the generation of FEFF input files with structures that match those in the dynamical matrix files. The usage of dym2feffinp is as follows:

```
dym2feffinp [Options] dymfile
```

where dymfile is the name of the file containing the dynamical matrix. This command creates two files, ‘feff.dym’ and ‘feff.inp’, which contain correctly matched structures. (Note of warning: If dymfile is named “feff.dym”, then dym2feffinp will APPEND the updated dym information to it. Please don’t call the dymfile ‘feff.dym’. Future version will check that dymfile doesn’t use the “feff.dym” name.)

The default behavior of dym2feffinp can be modified with the following options:

- `-c iAbs` Use atom iAbs as absorber
- `-f fname` Write feff input to file fname
- `-d dname` Write adjusted dym file to file dname

The `-c` options allows the user to choose different absorbers. The usual approach of editing a FEFF input file and changing the potential type of a certain atom to 0 will result in mismatched FEFF and dym structures. This should be avoided. It is recommended that different input files be generated using the `-c` option.

The `-f` and `-d` change the default output filenames from “feff.inp” and “feff.dym” to fname and dname, respectively.

Feff.inp syntax

To calculate Debye-Waller factors from a dynamical matrix (or matrix of force constants or Hessian matrix) using the Lanczos recursive algorithm, the following syntax is required in ‘feff.inp’ :

```
DEBYE Temp Debye_Temp [DW_Opt [dymFile DMDW_Order DMDW_Type DMDW_Route]]
```

where:

- Temp Temperature at which the DW factors are calculate
- Debye_Temp Debye Temperature of the material
- DW_Opt The model used to calculate the DW factors - here, $DW_opt = 5$
- dymFile Name of the dynamical matrix information file. The default value is ‘feff.dym’
- DMDW_Order Lanczos recursion order to be used in the calculation. The default value is 2. Well converged results are usually obtained for $DMDW_Order = 6 - 10$. For small size systems, these values might be too large. As a rule of thumb, DMDWOrder should be less than $3 * (Numberofatoms) - 6$. Some paths, within systems with high symmetry, might require a lower DMDW_Order. The user should always check convergence.
- DMDW_Type Type of DW calculation. The possible values are: 0 Parallel s^2 (default)
- DMDW_Route Which paths to use in the dmdw module. These paths do not affect the path selection in the XAS calculations, they are used for the generation of an input file for the independent dmdw module. The possible values are:
 - 0 Skip **dmdw** module (default)
 - 1 All SS paths from absorber
 - 2 Same as 1 + all DS paths from absorber
 - 3 Same as 2 + all TS paths from absorber
 - 11 All SS paths
 - 12 Same as 1 + all DS paths
 - 13 Same as 2 + all TS paths

Example 1: Creating a FEFF input file from a dym file using dym2feff.inp

Here we demonstrate how to convert a dym file, in this case created from a Gaussian fchk file, into matched pairs of ‘feff.inp’ and ‘feff.dym’ files, for different absorbing centers.

The dym file for a CO2 molecule ‘C02.dym’, converted from the formatted checkpoint file, looks like:

```
#####
1
3
8
6
```

```
8
15.99491460
12.00000000
15.99491460
0.00000000 0.00000000 2.20979482
0.00000000 0.00000000 0.00000000
0.00000000 0.00000000 -2.20979482
1 1
3.501599e-02 -7.311989e-13 -8.941376e-12
-7.311989e-13 3.501599e-02 3.206256e-11
-8.941376e-12 3.206256e-11 1.042343e+00
1 2
-7.001817e-02 -5.485621e-12 2.501278e-11
-3.211930e-12 -7.001817e-02 -2.341738e-11
-3.124223e-11 -3.615378e-11 -9.594793e-01
1 3
3.500217e-02 -8.681691e-13 -1.846564e-11
8.681691e-13 3.500217e-02 -1.245331e-11
-1.846599e-11 1.245327e-11 -8.286417e-02
2 1
-7.001817e-02 -3.211930e-12 -3.124223e-11
-5.485621e-12 -7.001817e-02 -3.615378e-11
2.501278e-11 -2.341738e-11 -9.594793e-01
2 2
1.400363e-01 1.883207e-12 2.058909e-11
1.883207e-12 1.400363e-01 4.005329e-11
2.058909e-11 4.005329e-11 1.918959e+00
2 3
-7.001817e-02 2.586772e-12 5.500370e-12
4.227571e-12 -7.001817e-02 3.602353e-12
3.495310e-11 1.591451e-11 -9.594793e-01
3 1
3.500217e-02 8.681691e-13 -1.846599e-11
-8.681691e-13 3.500217e-02 1.245327e-11
-1.846564e-11 -1.245331e-11 -8.286417e-02
3 2
-7.001817e-02 4.227571e-12 3.495310e-11
2.586772e-12 -7.001817e-02 1.591451e-11
5.500370e-12 3.602353e-12 -9.594793e-01
3 3
3.501599e-02 7.311989e-13 -8.940090e-12
7.311989e-13 3.501599e-02 -3.206153e-11
-8.940090e-12 -3.206153e-11 1.042343e+00
#####
```

The dym files are not required to be sorted in any particular order, they retain the atom order of the program that generated the dynamical matrix. In this case, the order is O, C and O, with the C atom at the origin. Since FEFF internally sorts the atoms according to their distance to the absorber, if this dym file is used as is to compute the EXAFS/XANES for the C atom, the results would be incorrect. We can generate the appropriate FEFF input file and associated dym file with the following command:

```
dym2feffinp --c 2 --f C02-C-feff.inp --d C02-C-feff.dym C02.dym
```

This creates new files 'C02-C-feff.inp' and 'C02-C-feff.dym'. The C02-C-feff.dym file:

```
#####
1
3
6
8
8
12.000000
15.994915
15.994915
0.00000000 0.00000000 0.00000000
0.00000000 0.00000000 2.20979482
0.00000000 0.00000000 -2.20979482
1 1
1.400363E-01 1.883207E-12 2.058909E-11
1.883207E-12 1.400363E-01 4.005329E-11
2.058909E-11 4.005329E-11 1.918959E+00
1 2
-7.001817E-02 -3.211930E-12 -3.124223E-11
-5.485621E-12 -7.001817E-02 -3.615378E-11
2.501278E-11 -2.341738E-11 -9.594793E-01
1 3
-7.001817E-02 2.586772E-12 5.500370E-12
4.227571E-12 -7.001817E-02 3.602353E-12
3.495310E-11 1.591451E-11 -9.594793E-01
2 1
-7.001817E-02 -5.485621E-12 2.501278E-11
-3.211930E-12 -7.001817E-02 -2.341738E-11
-3.124223E-11 -3.615378E-11 -9.594793E-01
2 2
3.501599E-02 -7.311989E-13 -8.941376E-12
-7.311989E-13 3.501599E-02 3.206256E-11
-8.941376E-12 3.206256E-11 1.042343E+00
2 3
```

```

3.500217E-02 -8.681691E-13 -1.846564E-11
8.681691E-13 3.500217E-02 -1.245331E-11
-1.846599E-11 1.245327E-11 -8.286417E-02
3 1
-7.001817E-02 4.227571E-12 3.495310E-11
2.586772E-12 -7.001817E-02 1.591451E-11
5.500370E-12 3.602353E-12 -9.594793E-01
3 2
3.500217E-02 8.681691E-13 -1.846599E-11
-8.681691E-13 3.500217E-02 1.245327E-11
-1.846564E-11 -1.245331E-11 -8.286417E-02
3 3
3.501599E-02 7.311989E-13 -8.940090E-12
7.311989E-13 3.501599E-02 -3.206153E-11
-8.940090E-12 -3.206153E-11 1.042343E+00
#####

```

This 'feff.dym' file is now centered on the C atom and the atoms are sorted correctly. The associated CO2-C-feff.inp FEFF input file has the following structure section:

```

#####
POTENTIALS
0 6 C
1 8 0

ATOMS
0.00000 0.00000 0.00000 0 C 0.00000 0
0.00000 0.00000 1.16937 1 0 1.16937 1
0.00000 0.00000 -1.16937 1 0 1.16937 2
END
#####

```

which is correctly centered and has the same structure as the dym file.

Example 2: XANES and EXAFS calculation

A typical FEFF input file that uses ab initio DW factors in a XANES calculation looks as follows:

```

* This feff9 input file was generated by dym2feffinp
EDGE      K      1.0
CONTROL    1      1      1      1      1      1
SCF        4.000
XANES      4.000

```

```

FMS                6.000

DEBYE      500.0  1073.0  5 feff.dym  6  0  1

POTENTIALS
  0      8      0
  1      1      H

ATOMS
  0.00000      0.00000      0.00000      0      0  0.00000      0
  0.96141     -0.12674      0.00000      1      H  0.96972      1
 -0.12674      0.96141      0.00000      1      H  0.96972      2

END

```

This input calculates the XANES O K edge spectrum of a single water molecule. If an EXAFS calculation is required, the same DEBYE card parameters apply, but the XANES card should be substituted by the EXAFS card. It uses *ab initio* DW factors at 500K and a dynamical matrix stored in the file 'feff.dym'. The number of Lanczos recursion iterations is set to 6, the type of DW calculation is to calculate parallel σ^2 , and it asks that all single scattering paths from the absorber be calculated independently in the **dmdw** module. When FEFF reads this 'feff.inp' file, it will create an intermediate input file 'dmdw.inp':

```

6
1    500.000    500.000
0
feff.dym
1
2    1    0          3.20

```

The first line gives the number of Lanczos iterations. The second defines the temperature, in this case a grid with a single point. The third defines the type of DW calculation and the fourth the name of the dynamical matrix file. The fifth line declares that the input contains a single path descriptor, which is included in the next line. This descriptor defines all single scattering paths from the absorber that are less than 3.2 Bohr long.

The dynamical matrix file 'feff.dym' for this calculation looks like this:

```

1
3
8
1
1
15.994915
1.007825
1.007825

```

```

0.00000000    0.00000000    0.00000000
1.81679640   -0.23950080    0.00000240
-0.23950080    1.81679640    0.00000240
1      1
5.398996E-01 -1.171079E-01  5.031484E-07
-1.171079E-01  5.399060E-01  9.690730E-07
5.031484E-07  9.690730E-07 -1.841479E-03
1      2 # next 3 lines omitted here for brevity
1      3 # and again ...
2      1
2      2
2      3
3      1
3      2
3      3
5.501607E-02 -7.712230E-02 -1.185055E-07
-7.712230E-02  5.034998E-01  8.548785E-07
-1.185055E-07  8.548785E-07 -8.715942E-04

```

The user must obtain this file using another program, for example, Gaussian or ABINIT. Given these files ‘feff.inp’ and ‘feff.dym’, FEFF calculates a XANES spectrum including the Debye-Waller factors derived from the given dynamical matrix.

Additional technical information about the calculation of ab initio Debye-Waller factors is provided in Appendix [F](#).

2.19 Parallel Calculations

Although FEFF is an efficient code and many spectra can be calculated on a laptop computer, some calculations require more calculation time, e.g. those using large FMS clusters. To meet the memory and cpu needs of such calculations, FEFF can be run in parallel. Running parallel calculations is not trivial. It requires:

- Suitable hardware, e.g. a multicore computer, or a computing cluster
- An MPI environment. There are different MPI implementations, and configuring MPI on your system can be non-trivial. If you will run parallel calculations across several computers (e.g., the nodes of a cluster) then these computers must also share disk access and have passwordless access to each other. Your system administrator can help set up MPI or can tell you the specifications for your computer. Unfortunately, it’s very hard for us to provide support as we don’t know your setup.
- FEFF binaries for your configuration. Although some versions of FEFF come with precompiled parallel binaries, it is unlikely that they will work on your system. You’ll probably have to compile FEFF-mpi yourself. That requires a suitable FORTRAN90 compiler.

FEFF-MPI scales well even on slow networks because the parallel threads communicate very little. We simply distribute a grid of energy points over the MPI processor grid and calculate the Green's function for each energy point. The limitation of this approach is that we cannot use more cores than there are energy points in the grid. Typically, FEFF maxes out at $N = 64$ or $N = 128$. Speedup will be slightly lower than linear because of sequential tasks and the efficiency of distributing the energy points.

Several parallel threads may write identical output to the screen or log file. This may cause the standard output to be less readable.

2.19.1 MPI calculations on the command line

Although the precise syntax varies with software and hardware environment, a command like

```
prompt> mpirun -NP 40 --hosts n1,n2,n3,n4,n5 --hostfile /home/me/stuff/my.hostfile
/home/me/feff10/bin/MPI/fms
```

will execute **fms** in parallel using 40 threads. Your FEFF installation includes a simple 'feffmpi' script that contains the above instruction for all consecutive FEFF modules. Or you can copy it here:

```
#!/bin/bash
# Adjust this line: it must point to a directory containing the MPI FEFF modules
FeffPath=/home/jorissen/feff10/bin/MPI
# Adjust this line to meet your system configuration: "
MPICommand="mpirun -n 12 --host n20,n20,n20,n20,n17,n17,n17,n17,n18,n18,n18,n18 "
# In this example, we are using 12 parallel threads on 3 cluster nodes
# (n17, n18, and n20) with 4 (or more) cores each.
# Other common mpirun options are: --hostfile ; --nolocal ; etc.
# The calculation will be (a little less than) 12 times faster than a non-parallel
# calculation on the same computer.

# There should be no need to edit the following lines:
$MPICommand $FeffPath/rdinp
$MPICommand $FeffPath/dmdw
$MPICommand $FeffPath/atomic
$MPICommand $FeffPath/pot
$MPICommand $FeffPath/screen
$MPICommand $FeffPath/opconsat
$MPICommand $FeffPath/xsph
$MPICommand $FeffPath/fms
$MPICommand $FeffPath/mkgtr
$MPICommand $FeffPath/path
$MPICommand $FeffPath/genfmt
$MPICommand $FeffPath/ff2x
```

```
$MPICommand $FeffPath/sfconv
$MPICommand $FeffPath/compton
$MPICommand $FeffPath/eels
$MPICommand $FeffPath/ldos
```

On some computers you can omit the hosts and hostfile options (e.g. when running on a multicore desktop). On other computers that are shared among many users you may have to interact with a queuing system. This may involve writing a small script that determines node availability on the fly and sets the hosts list. A simpler example is shown here:

```
#!/bin/csh

# This is an example for submitting a parallel FEFF job to a PBS queue.
# Details may depend on your configuration.

# Typically, you save this file as "feff.pbs.sh" in the pwd containing feff.inp .
# Then submit as "qsub feff.pbs.sh".
# You can monitor the job using "qstat".
# But find out the specifics of your configuration before submitting.
# E.g. oftentimes you have to specify maximum runtime.

# Configuring the PBS run:
# The lines starting with '#PBS' are NOT comments - don't delete them!
# Here, we are using 4 nodes with 16 cores each, or 64 cores total.
# The job name is "feff9-Cu" (look for this in the queue monitor).
# Stdout/Stderr will go to the files specified below.
# (You may want to purge these files if you run FEFF several times.)

#PBS -l nodes=4:ppn=16
#PBS -N feff9-Cu
#PBS -o feff.out
#PBS -e feff.err
#PBS -q batch
#PBS -V

# Adjust this number to = nodes x ppn, e.g. 4 x 16 here:
set NP=64
# Adjust next line to point to the location of the FEFF MPI executables:
set FEFF_HOME=/home/me/feff10/bin/MPI

# No need to edit below:
cd $PBS_O_WORKDIR
mpirun -np $NP --hostfile $PBS_NODEFILE $FEFF_HOME/rdinp
mpirun -np $NP --hostfile $PBS_NODEFILE $FEFF_HOME/atomic
```

```

mpirun -np $NP --hostfile $PBS_NODEFILE $FEFF_HOME/dmdw
mpirun -np $NP --hostfile $PBS_NODEFILE $FEFF_HOME/pot
mpirun -np $NP --hostfile $PBS_NODEFILE $FEFF_HOME/ldos
mpirun -np $NP --hostfile $PBS_NODEFILE $FEFF_HOME/screen
mpirun -np $NP --hostfile $PBS_NODEFILE $FEFF_HOME/opconsat
mpirun -np $NP --hostfile $PBS_NODEFILE $FEFF_HOME/xsph
mpirun -np $NP --hostfile $PBS_NODEFILE $FEFF_HOME/fms
mpirun -np $NP --hostfile $PBS_NODEFILE $FEFF_HOME/mkgtr
mpirun -np $NP --hostfile $PBS_NODEFILE $FEFF_HOME/path
mpirun -np $NP --hostfile $PBS_NODEFILE $FEFF_HOME/genfmt
mpirun -np $NP --hostfile $PBS_NODEFILE $FEFF_HOME/ff2x
mpirun -np $NP --hostfile $PBS_NODEFILE $FEFF_HOME/sfconv
mpirun -np $NP --hostfile $PBS_NODEFILE $FEFF_HOME/compton
mpirun -np $NP --hostfile $PBS_NODEFILE $FEFF_HOME/eels

```

More information can be found in Sec. B. If you are unable to figure out the particularities of your local setup, we ask that you consult your system administrator or system documentation for help first.

2.20 Using the Many-Pole Self-Energy

FIX give definition of SE.

To use the many-pole model self-energy within FEFF, you must obtain an estimate of the loss function $L(\omega) = \text{Im}[\epsilon^{-1}(\omega)]$. The loss function can either be calculated, or measured experimentally. A rough estimate can be calculated very quickly with the feff code using the **OPCONS** card, along with (optionally) the **NUMDENS** card. For more accurate calculations of the loss function, one can use codes based on the Beth-Salpeter equation or time-dependent density functional theory. Several codes can perform these calculations:

- AI2NBSE
- YAMBO
- WIEN2K
- QUANTUM ESPRESSO
- ...

2.20.1 Examples

An input file that uses OPCONS to calculate the loss function and MPSE/SFCONV to calculate the resulting many-pole self-energy and many-body spectral function for a Cu K-edge XANES calculation follows:

```
TITLE Cu crystal
```

```

* COREHOLE treatment RPA or FSR or NONE
COREHOLE RPA

*          pot  xsph fms path genfmt ff2x
CONTROL  1   1   1   1   1   1

* Use the atomic database to form the loss function
OPCONS

* Use many-pole self-energy with density dependence
MPSE 2

* Spectral function convolution
SFCONV

* This specifies HL exchange for both fine structure
* and background, but in this case, the many-pole model
* will be used, since the MPSE card is present.
EXCHANGE      0      0      0      0

* Full multiple scattering with a cluster of 8.0 angstroms.
FMS 9.5

* Shift the fermi-level back 1.5 eV and add 0.3 eV of experimental broadening.
CORRECTIONS 1.5 0.3

* Self consistent potentials with cluster size of 5.0 angstroms.
SCF      7.0

* Calculate XANES (changes energy grid used)
XANES 5.

* Calculate Debye-Waller factors for a temp of 10 K and Debye temp of
* 315 K using correlated Debye model.
DEBYE 10 315 0

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

POTENTIALS
  0 29 Cu  3 3 0.01
  1 29 Cu  3 3 1.0

ATOMS
  0.0000      0.0000      0.0000      0  Cu_00      0.0000

```

0.0000	-1.8050	1.8050	1	Cu_01	2.5527
1.8050	0.0000	-1.8050	1	Cu_01	2.5527
-1.8050	0.0000	-1.8050	1	Cu_01	2.5527
1.8050	-1.8050	0.0000	1	Cu_01	2.5527
-1.8050	1.8050	0.0000	1	Cu_01	2.5527
.					
.					
.					

This input file specifies that we want to calculate the Cu K-edge XANES of bulk Cu using the many-pole model dielectric function to calculate self-energy effects (MPSE) as well as multi-electron excitation effects (SFCONV). The file 'loss.dat' is required input for the many-pole dielectric function and in this case is obtained by specifying the OPCONS card. Note that the OPCONS card gives only a rough estimate of the loss function.

For a more accurate calculation, comment out the "OPCONS" card and provide your own 'loss.dat' file, e.g. from an experimental measurement or an ab initio calculation. An example of a 'loss.dat' file can be found in [~/feff10/examplesMPSE/Cu/](#). You can compare this to the 'loss.dat' file that will be generated from the OPCONS card in the example above or at [~/feff10/examples/MPSE/Cu_OPCONS/](#). You can also compare the resulting spectra. (Note that the 2 examples provided have many other differences. You should compare 2 calculations differing only in the use of the OPCONS card to study its effects.)

2.21 Compton scattering

As of version 9.5.1, FEFF can calculate Compton scattering. This is activated by the **COMPTON** card. Optionally one can use the **CGRID** card to set the grid, and **RHOZZP** card to calculate a z, z' slice of ρ . An example is given below. Note that calculations of Compton scattering tend to be fairly slow but can be parallellized. They consume rather a lot of disk space by FEFF standards (of the order of 100MB) and results in the creation of hundreds of individual files, which can be cumbersome.

First, $J(z, z')$ is calculated. This is the Fourier transform of $J(p_q)$. Since this is computationally expensive, it is saved to 'jzzp.dat'. The final Compton profile is saved in 'compton.dat'.

```
TITLE Cu crystal fcc, lattice parameter a=3.61 (Kittel)
CONTROL 1 1 1 1 1 1
SCF      4.0
COREHOLE None
LDOS -30 0 0.1
FMS 4.0
* Calculate the Compton profile:
COMPTON
* Calculate a slice rho(z,z')
```

```
RHOZZP
* The grid for calculating rho(r,r')
CGRID 10 32 32 32 120

POTENTIALS
0 29 Cu 3 3 0.01
1 29 Cu 3 3 1.0

ATOMS
  0.00000    0.00000    0.00000    0 Cu_00    0.0000
  0.00000   -1.80500    1.80500    1 Cu_01    2.5527
  1.80500    0.00000   -1.80500    1 Cu_01    2.5527
 -1.80500    0.00000   -1.80500    1 Cu_01    2.5527
  1.80500   -1.80500    0.00000    1 Cu_01    2.5527
* [Truncated]
END
```

2.22 The Mixed Dynamic Form Factor

Currently available to Developers only.

Chapter 3

feff10 Control Cards

3.1 The master input file ‘feff.inp’

The FEFF program consists of a set of program modules driven by a single input file ‘feff.inp’.

This section describes the input. It may be helpful to look at the sample input files in Section 2 while reading this section. The current chapter is meant as a reference, while Section 2 makes for friendlier, tutorial-style reading.

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, which is a keyword possibly followed by alpha-numeric data. The sequence of keyword cards is arbitrary. If any card or optional data is omitted, default values are used. An exception is that the user must always specify the crystal structure, which requires one or more of the POTENTIALS, ATOMS, CIF, LATTICE, and OVERLAP cards.

Alpha-numeric values are listed in free format, separated by blanks. Tab characters are not allowed 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. 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 are in electron-volts (eV).

An auxiliary program (ATOMS), developed by Bruce Ravel, can generate the ‘feff.inp’ file for crystals from crystallographic input parameters. A GUI to ATOMS is available at <http://cars9.uchicago.edu/~ravel/software/aboutatoms.html>

Additionally, FEFF10 itself can treat crystals specified by their unit cell.

3.2 Complete List of FEFF10 Control Cards

We divide the ‘feff.inp’ options into three categories:

- structural information describing the molecule or solid
- spectrum information specifying what type of spectroscopy to calculate
- options that determine how FEFF calculates the required spectrum for the given structure

Additionally, we can classify CARDS as *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.

Structural information

Purpose: Specify the structure

Standard Cards: **ATOMS**, **POTENTIALS**, **RECIPROCAL**, **REAL**, **CIF**, **LATTICE**, **TARGET**, and **TITLE**

Useful Cards: **COORDINATES**, **RMULTIPLIER** and **SGROUP**

Advanced Cards: **CFAVERAGE**, **OVERLAP** and **EQUIVALENCE**

Spectrum information

Purpose: Specify the spectrum or material property to be calculated

Standard Cards: **EXAFS**, **ELNES**, **EXELFS**, **LDOS**, and **XANES**

Useful Cards: **ELLIPTICITY**, **MULTIPOLE**, and **POLARIZATION**

Advanced Cards: **COMPTON**, **DANES**, **FPRIME**, **MDFF**, **NRIXS**, **XES**, and **XNCD**
(or **XMCD**)

FEFF program control

Purpose: Control the way FEFF works

general options

Purpose of cards: general settings

Standard cards: **CONTROL**, **END**, **KMESH**, and **PRINT**

Advanced Cards: **DIMS** and **EGRID**

atomic, pot, screen, opconsat

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

Standard cards: **AFOLP**, **COREHOLE**, **EDGE**, **SCF**, and **S02**

Useful Cards: **CHBROAD**, **CONFIG**, **EXCHANGE**, **FOLP**, **HOLE**, **NOHOLE**, **RGRID**, and **UNFREEZE**

Advanced Cards: **CHSHIFT**, **CHWIDTH**, **CORVAL**, **EGAP**, **EPS0**, **EXTPOT**, **INTERSTITIAL**, **ION**, **JUMPRM**, **NUMDENS**, **OPCONS**, **PREP**, **RESTART**, **SCREEN**, **SETE**, and **SPIN**

xsph

	Purpose of Module: Calculate cross-section and phase shifts Advanced Cards: LJMAX, LDEC, MPSE, PLASMON, PMBSE, RPHASES, RSIGMA, and TDLDA
fms and mkgtr	
	Purpose of Module: Calculate full multiple scattering for XANES and ELNES Standard cards: FMS Useful Cards: DEBYE Advanced Cards: BANDSTRUCTURE and STRFACTORS
path	
	Purpose of Module: Path enumeration Standard cards: RPATH Useful Cards: NLEG Advanced Cards: PCRITERIA, SS, and SYMMETRY
genfnt	
	Purpose of Module: Calculate scattering amplitudes and other XAFS parameters Useful Cards: CRITERIA Advanced Cards: IORDER and NSTAR
ff2x	
	Purpose of Module: Calculate X-ray spectra. Standard cards: DEBYE Useful Cards: ABSOLUTE, CORRECTIONS, SIG2, SIG3 and SIGGK Advanced Cards: MBCONV
sfconv	
	Purpose of Module: Convolve output with spectral function. Useful Cards: SFCONV Advanced Cards: RCONV, SELF and SFSE
compton	
	Purpose of Module: Calculate Compton scattering. Useful Cards: CGRID and RHOZZP
eels	
	Purpose of Module: Calculate EELS spectra. Useful Cards: MAGIC

These CARDS are listed below in the same order as in the table above. Each CARD 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 explains the arguments and their uses more fully. Example uses of the card look like this:

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

3.3 Structural Information Cards

The ATOMS card is used to specify the absorbing atom and its environment. Alternatively, if atomic coordinates are not known, the OVERLAP card can be used to construct approximate potentials. Without the structural information of either ATOMS or OVERLAP card, no calculations can be done.

◆ **ATOMS** *Standard*
 ATOMS specifies the Cartesian coordinates (in Ångstroms) and unique potential indices of each atom in the cluster, one atom per line. If the LATTICE card is used to define the unit cell for reciprocal space calculations, the ATOMS card lists the atoms in the unit cell. See the discussion of the **POTENTIALS** card and the **COORDINATES** card for more info on how to specify atom types and coordinates.

An auxiliary code, **ATOMS**, can generate a real-space ATOMS list for crystals from crystallographic data.

```
* A real-space example : the SF6 molecule
* Coordinates in Angstrom
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

* A reciprocal-space example : the hexagonal (wurtzite) GaN crystal
* See the LATTICE card for the corresponding lattice vectors
* Coordinates are expressed as fractions of the lattice vectors
* Note that CIF files are now the preferred way to input crystals by their unit cell.
ATOMS                                     * this unit cell contains 4 atoms
*      x                                     y                                     z
  0.288675130000      0.500000000000      0.000000000000  1      Ga
```

0.577350270000	0.000000000000	0.811850000000	1	Ga
0.288675130000	0.500000000000	0.609700000000	2	N
0.577350270000	0.000000000000	1.414242700000	2	N

◆ CIF cif_file

Standard

This card specifies the name of a file containing the structural information in CIF format. CIF is incompatible with ATOMS, LATTICE, SGROUP, and COORDINATES. POTENTIALS is not required with CIF, as the potential types will be generated automatically. However, POTENTIALS can be used to set the details (angular momentum cutoffs and spin variables) of the potentials. See (K-space FEFF) for more information. While both relative and absolute paths are supported, we recommend copying the '.cif' file to the working directory to avoid mistakes and confusion. CIF currently requires that RECIPROCAL, TARGET, and KMESH are also set. Note that TARGET in conjunction with CIF refers to the list of atoms as given in the '.cif' file (i.e., a list of the crystallographically inequivalent atom positions in the unit cell).

```
* the file GaN.cif defines the unit cell of hexagonal (wurtzite) GaN :
CIF  GaN.cif
```

◆ LATTICE type scale

Standard

This card specifies the lattice. First, its type must be specified using a single letter : P for primitive, F for face centered cubic, I for body centered cubic, H for hexagonal. The following three lines give the three basis vectors in Cartesian Angstrom coordinates. They are multiplied by scale (e.g., 0.529177 to convert from bohr to Angstrom).

```
* the unit cell of hexagonal (wurtzite) GaN :
LATTICE P    3.18800
    0.86603    -0.50000    0.00000    * ax ay az
    0.00000     1.00000    0.00000    * bx by bz
    0.00000     0.00000    1.62359    * cx cy cz
```

◆ 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.

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 on atomic number.

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. 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). Exception: If the material is defined by its unit cell the absorbing atom is not defined through the POTENTIALS card but in the **TARGET** instead - in this case, there should not be a unique potential index 0 in the POTENTIALS card. There is another exception to the rule when the **CFAVERAGE** card is used.

To save time, the code calculates the overlapped atom potential for each unique potential only once, using the first atom with a given unique potential index. 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 always be checked. If the neighborhood of the sample atom is not representative, the code will generate inaccurate potentials and phase shifts, as well as poor XAS results. Atoms of the same Z may have different potentials if their local environment is substantially different. FEFF leaves such judgments to the user. One can always make two atoms inequivalent, and compare their potentials, electronic configurations, and IDOS in the FEFFoutput.

```
* 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.

◆ REAL

Standard

This card tells FEFF to work in real space. This is the default mode, so it is never strictly necessary to use this card. This card may be used for clarity in input files that mix real-space and k-space calculations. See also the RECIPROCAL card.

```
* do a real-space calculation
REAL
```

◆ RECIPROCAL

Standard

This card tells FEFF to work in reciprocal space. It affects modules pot, xsph, fms and ldos. This card requires the use of CIF or ATOMS and LATTICE; TARGET; and KMESH.

```
* do a k-space calculation of a crystal
RECIPROCAL
```

◆ TARGET *ic*

Standard

Specifies the location of the absorber atom for reciprocal space calculations. It is entry *ic* of the ATOMS card if an ATOMS card and LATTICE card are used. In conjunction with the CIF card it is entry *ic* the list of atoms as given in the '.cif' file (i.e., a list of the crystallographically inequivalent atom positions in the unit cell). The target needs to be specified also for NOHOLE calculations. Note that this cannot be specified in the POTENTIALS list because periodic boundary conditions would then produce an infinite number of core holes.

```
* calculate a spectrum for the second atom in the ATOMS list or CIF file.
TARGET 2
```

◆ TITLE *any_descriptive_text*

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
```

◆ COORDINATES *i*

Useful

i must be an integer from 1 through 6. It specifies the units of the atoms of the unit cell given in the ATOMS card for reciprocal space calculations. If the card is omitted, the default value *icoord* = 3 is assumed. FIX check this

1. Cartesian coordinates, Angstrom units. Like FEFF - you can copy from a real-space feff.inp file if your lattice vectors coincide with atoms in that feff.inp file.
2. Cartesian coordinates, fractional units (i.e., fractions of the lattice vectors ; should be numbers between 0 and 1). Similar to FEFF.
3. Cartesian coordinates, units are fractional with respect to FIRST lattice vector. Like SPRKKR. (default)
4. Given in lattice coordinates, in fractional units. Like WIEN2k (but beware of some 'funny' lattice types, e.g. rhombohedral, in WIEN2k case.struct if you're copy-pasting)
5. Given in lattice coordinates, units are fractional with respect to FIRST lattice vector.
6. Given in lattice coordinates, Angstrom units.

```

* Say that a diamond lattice has been defined as :
LATTICE P 6.0
0.0 0.5 0.5
0.5 0.0 0.5
0.5 0.5 0.0
* Now the atoms can be entered as :
ATOMS
0.0 0.0 0.0
1.5 1.5 1.5
COORDINATES 1      * identical to 6 for this example
* Or another way is :
ATOMS
0.0 0.0 0.0
0.25 0.25 0.25
COORDINATES 2      * identical to 3, 4, and 5 for this example

```

◆ **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.

```

*increase distances by 1%
RMULTIPLIER 1.01

```

◆ **SGROUP** *igroup*

Useful

This card specifies the space group of the crystal (number from 1 through 230). Currently not used and informative only.

```

* simple primitive cell
SGROUP 1

```

◆ **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 has fewer than 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
CFAVERAGE 2 0 0
```

FIX check that there's a more substantial example of this somewhere and put a link to it

◆ 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. FEFF10 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, 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 3.5.5 .

```
* 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

```

◆ EQUIVALENCE `ieq`

Advanced

This optional card is only active in combination with the **CIF** card. It tells FEFF how to generate potential types from the list of atom positions in the ‘cif’ file.

If `ieq = 1`, the crystallographic equivalence as expressed in the ‘cif’ file is respected; that is, every separate line containing a generating atom position will lead to a separate potential type. This means that, e.g., in HOPG graphite, the two generating positions will give rise to two independent C potentials. This is also the default behavior if the EQUIVALENCE card is not specified.

If `ieq = 2`, unique potentials are assigned based on atomic number Z only. That is, all C atoms will share a C potential and so on. This is how most FEFF calculations are run. Whether it is sensible or not to do this depends on the system and on the property one wishes to calculate. Keep in mind that FEFF is a muffin tin code, and may therefore be indifferent to certain differences between crystallographically inequivalent sites. On the other hand, if an element occurs in the crystal with different oxidation states, it may be necessary to assign separate potentials to these different types in order to describe the crystal properly and get accurate spectra.

If `ieq = 3`, unique potentials are assigned based on atomic number Z and the first shell. This can be useful e.g. to treat larger systems with crystal defects, where only first neighbors of the defect need to be treated differently from all more distant atoms of a certain Z . (To be implemented.)

If `ieq = 4`, a hybrid of methods 1 and 2 is used. That is, if the number of unique crystallographic positions does not exceed a hard-coded limit (`nphx=9` in the current version), they are treated with the correct crystallographic equivalence. If the number of unique crystallographically inequivalent sites is larger, they get combined by atomic number Z . This ad hoc approach is a practical way of simply limiting the number of unique potentials. This makes sense because, first of all, there are certain hardcoded limits that would require recompilation of the code, requiring more RAM memory and more work than a user may want to do. Secondly, our

SCF algorithm tends to have a harder time reaching convergence as the number of potentials increases, leading to substantially longer calculation times or even convergence failure if the number of potentials becomes very large.

If $ieq = 5$, unique potentials are assigned based on a label in the 'cif' file. That is, the user can edit the 'cif' file in a text editor and mark different sites with labels such as "Ti1" and "Ti2". FEFF will assign the same unique potential to all sites with the same label. This gives the user complete control over potential assignment. (To be implemented.)

If you require one of the solutions marked as "To be implemented", please contact us for assistance.

```
* Example : Do a traditional FEFF calculation where all atoms with the same Z
*             have the same potential
CIF graphite.cif
EQUIVALENCE 2
* This would be equivalent to a file using LATTICE and ATOMS card, and
* POTENTIALS
**          ipot      z      label  lmax1  lmax2
*           0         6        C      -1     -1      * for the core hole atom
*           1         6        C      -1     -1      * for all other C atoms

* Example 2 : Do a calculation with true crystallographic equivalence,
*             as most bandstructure codes do:
CIF graphite.cif
EQUIVALENCE 1 * This is the default and could be omitted for the same results
* This would be equivalent to a file using LATTICE and ATOMS card, and
* POTENTIALS
**          ipot      z      label  lmax1  lmax2
*           0         6        C      -1     -1      * for the core hole atom
*           1         6        C      -1     -1      * for half of the C atoms
*           2         6        C      -1     -1      * for the other half of the C atoms
```

3.4 Spectrum Information Cards

These cards tell FEFF which material properties to calculate. In general, one can choose only one spectroscopy card (EXAFS, XANES, DANES, XMCD, ELNES, EXELFS, FPRIME, NRIXS, XES). To calculate a second type of spectrum, a new FEFF calculation is generally required, although part of the previous calculation may be reused. The LDOS can be combined with any spectroscopy. The NRIXS card must be combined with either EXAFS or XANES - the output will be a NRIXS spectrum but the other card tells FEFF how to calculate it. The cards ELLIPTICITY, POLARIZATION and MULTIPOLE may be combined with certain spectroscopy cards.

◆ EXAFS [xkmax]

Standard

EXAFS is the default type of spectroscopy. As such, the card may in principle be omitted, though it is good practice to always explicitly set the spectroscopy type being calculated. The EXAFS card sets the maximum value of k for EXAFS calculations. k is set by **xkmax**, and the default value is 20 \AA^{-1} . The code can calculate to even higher values, however, the user may be prompted to increase compilation time dimension settings. For high k calculations it might be necessary to make smaller steps using the **RGRID** card.

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

◆ ELNES [xkmax xkstep vixan]

Standard

```
E [aver [cross [relat]]]
```

```
kx ky kz
```

```
 $\beta$   $\alpha$ 
```

```
nr na
```

```
dx dy
```

xkmax

The maximum k-value for the calculation. **xkmax**, **xkstep** and **vixan** are exactly the same parameters as those used for the **XANES** card.

xkstep

The step size of the upper part of the k-mesh

vixan

The step size of the lower part of the k-mesh

E

energy of the electron beam in keV (typical values are 100-400 keV);

aver

1: calculate orientation averaged spectrum (e.g., a polycrystalline sample, working at the magic angle) ; 0 : use specific sample to beam orientation (default);

cross

1: use cross terms for the cross section (e.g., xy or yx ; default); 0: use only direct terms (eg., atom coordinates entered in symmetric coordinate frame ; assumed as default if **aver** is set to 1);

relat

1 (default): to use relativistic formula for the cross-section (default, always recommended) ; 0 to use nonrelativistic formula;

kx,ky,kz

wave vector of the incoming electron in the crystal frame (i.e., the Cartesian coordinate

system in which the atom positions of the **ATOMS** card are given). In arbitrary units (only the direction, not the size of \mathbf{k} is used). This line must be present for orientation-sensitive calculations, and absent for averaged calculations.

β

the collection semiangle of the EELS detector in mrad (typical values are of the order of 1 mrad);

α

The convergence semiangle of the incoming beam in mrad (typical values are of the order of 1 mrad);

nr, na

The cross section is integrated over the values of impulse transfer q allowed by α and β . The integration grid consists of **nr** concentric circles sampling a disc of radius $\alpha + \beta$. Circle i contains **na** * (2*i* - 1) points, making for **nr** * **nr** * **na** points total. These are nonphysical parameters and should be converged. Typical would be 50, 1; the integration is quite fast. Only for small values of **nr** is it necessary to increase **na** above 1.

dx, dy

The position of the detector in the scattering plane, specified by angles in mrad along the x and y axes (the same as used in the **ATOMS** card) (typical values are 0.0, 0.0)

The line giving beam orientation must be present if and only if an oriented spectrum is calculated.

The following example simulates an experiment with a 300 keV beam hitting the sample along the y-axis. The detector is set in the forward direction and has a 2.4 mrad (semi-)opening; the width of the incoming beam is 0 mrad. To do the integration over the detector aperture, 5*5*3 = 75 points are used. The calculation is relativistic and takes sample to beam orientation into account. Default settings are used for the energy/ \mathbf{k} -mesh.

```
ELNES # calculate elnes.
300 # beam energy in keV
0 1 0 # beam direction in the crystal frame
2.4 0.0 # collection semiangle, convergence semiangle (in mrad)
5 3 # q-integration mesh : radial size, angular size
0.0 0.0 # position of the detector (x,y angle in mrad)
```

◆ **EXELFS** **xkmax**

Standard

E [aver [cross [relat]]]

kx ky kz

β α

nr na

dx dy

See the **ELNES** card for a description of these parameters. Note that **xkmax**, the maximum k value, is the only parameter immediately following the EXELFS card.

◆ **LDOS** `emin emax eimag`

Standard

To obtain the ℓ DOS you need only run the **rdinp**, **atomic**, **pot** modules first to produce the file 'pot.bin'. LDOS is calculated in a separate **ldos** module, which runs if the LDOS card appears in 'feff.inp'. It uses the cluster cutoff radius *rfms2* specified by the FMS card.

The angular momentum projected density of states is placed by default on a standard grid of 101 points. **emin** and **emax** are the minimum and maximum energies of the ℓ DOS calculation and **eimag** is the imaginary part of potential used in the calculations. This is equivalent to Lorentzian broadening of the ℓ DOS 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 101 points are not enough, you can divide the energy range by 2 and run the **ldos** module 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. If one calculates the ℓ DOS of a crystal in real space, it will always be broadened due to the effect of finite cluster size.

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

◆ **XANES** [`xkmax xkstep vixan`]

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. The XANES card is normally accompanied by the **FMS** card for accurate results.

The XANES calculation is 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.

vixan

This argument specifies the energy step of the grid at the edge.

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

Cannot be used with ELNES, EXELFS, or NRIXS.

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

◆ **MULTIPOLE** `le2 [121p]`

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). This card cannot be used with NRIXS and is not supported with EXELFS and ELNES.

The additional field `121p` can be used to calculate individual dipolar contributions coming from $L \rightarrow L + 1$ (`121p` = 1) and from $L \rightarrow L - 1$ (`121p` = -1). Notice that in polarization dependent data there is also a cross term, which is calculated only when `121p` = 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.

Cannot be used with ELNES, EXELFS, or NRIXS.

```
POLARIZATION  1.0  2.5  0.0
```

◆ COMPTON [pqmax npq force-jzzp]*Advanced*

To calculate the Compton scattering $J(p_q)$. pqmax is the upper limit on p_q . npq is the number of p_q points. force-jzzp forces recalculation of intermediate $J(z, z')$. Since $J(p_q)$ is symmetric, the lower limit on p_q is hardcoded to be 0. First, $J(z, z')$ is calculated. This is the Fourier transform of $J(p_q)$. Since this is computationally expensive, it is saved to 'jzzp.dat'. The final Compton profile is saved in 'compton.dat'.

COMPTON

◆ DANES [xkmax xkstep vixan]*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 the 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. 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

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

$f_0(Q)$ 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 form, and in the more accurate Kissel-Pratt form. See the references for more details.

◆ MDFF imdff [qqmdff cosmdff]*Advanced*

Experimental feature. Calculate the Mixed Dynamic Form Factor. Currently available to Developers only.

- imdff = 3 : EELS type MDFF calculation selected - summed over all q,q' pairs
- imdff = 2 : NRIXS type MDFF calculation selected - for a single q,q' pair only
- imdff = 1 : NRIXS type MDFF calculation selected - summed over all q,q' pairs
- imdff ; 1 : MDFF calculation disabled

If imdff=1 or =2 and qqmdff and cosmdff are not specified, calculate MDFF(q,q') using q and q' vectors from the NRIXS list of q-vectors (needs to contain at least two vectors). If imdff=1 or =2 and qqmdff and cosmdff are specified, calculate MDFF(q,q') using q vectors from the

NRIXS list of q-vectors and generating vector q' as having length q_{mdff} and making an angle cosmdff with the vector q . If $\text{imdff}=3$, calculate $\text{MDFF}(q, q')$ using the parameters of the ELNES or EXELFS card. (Output on a grid; or sum if appropriate Bloch wave coefficients are provided. Not implemented.) Note that $\text{imdff}=3$ calculates a dipole-selected MDFF, but does it really rapidly and using relativistic corrections appropriate for EELS. ‘**feff.inp**’ must also contain the ELNES or EXELFS card. Using $\text{imdff}=1$ or $\text{imdff}=2$ calculates the MDFF without a selection rule (or determined using the LJMAX or LDECMX cards), but is slower and does not have relativistic corrections. Note that the NRIXS routines provide for weights, meaning that Bloch wave type (complex) coefficients can be added to simulate dynamical diffraction, if an external code is used to provide these coefficients. ‘**feff.inp**’ must also contain the NRIXS card. This card is probably too complex and may be butchered in the future. If you want to use any of this functionality, you’d better be in touch for collaboration, or really like Fortran :-P.

◆ **NRIXS** $nq \ qx \ qy \ qz$

Advanced

Calculate the NRIXS spectrum for given momentum transfer \vec{q} . Currently, nq must be set either to any negative value to calculate a spherical average over q-vectors of fixed length qx (qy and qz ignored) ; or set to 1 to calculate for the q-vector $qx \ qy \ qz$. $nq > 1$ is implemented experimentally; contact the authors for assistance if needed. Further options are available through the **LDEC** and **LJMAX** cards.

```
NRIXS -1 0.5 0.0 0.0      * spherical average
```

```
NRIXS  1 0.2 0.2 0.1      * orientation-sensitive
```

◆ **XES** $\text{emin} \ \text{emax} \ \text{estep}$

Advanced

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

◆ **XMCD or XNCD** $[\text{xkmax} \ \text{xkstep} \ \text{estep}]$

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 the magnetic field. The XMCD card has to be present in ‘**feff.inp**’ for these calculations.

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 2.9 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 2.9, where you can also find more details on the signal extraction process.

3.5 FEFF Program Control Cards

3.5.1 General Cards

The cards in this section generally affect the entire FEFF calculation. The CONTROL card is used to selectively run parts of FEFF. The PRINT card controls which output files are written.

◆ **CONTROL** `ipot ixsph ifms ipaths igenfmt iff2x` *Standard*

The CONTROL card lets you run one or more of the FEFF program modules separately. There is a switch for each of six parts of FEFF : 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. *ipot* controls modules atomic, pot and screen ; *ifms* controls modules fms and mkgtr ; and *iff2x* controls modules ff2x, sfconv, and eels. The ldos module is not affected by the CONTROL card ; it is activated using the corresponding LDOS card.

```
* example 1
* calculate self consistent potentials, phase shifts and fms only
CONTROL 1 1 1 0 0 0  ipot ixsph ifms  ipaths igenfmt iff2x

* 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 iff2x
```

◆ **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
```

◆ **KMESH** `nkpx nkpy nkpz [ktype [usesym]]]` *Standard*

This card specifies the mesh of k-vectors used to sample the full Brillouin Zone for the evaluation of Brillouin Zone integrals. Nkp is the number of points used in the full zone. It can be specified either as "nkpx nkpy nkpz", "nkp", or "nkp 0 0". If usesym = 1, the zone is reduced to its irreducible wedge using the symmetry options specified in file symfile, which must be present in the working directory. The k-mesh is constructed using the tetrahedron method of Bloechl et al., Phys. Rev. B, 1990. The parameter ktype is meant for time-saving only and means:

- ktype=1 : regular mesh of nkp points for all modules
- ktype=2 : use nkp points for ldos/fms and nkp/5 points for pot (significant time savings)
- ktype=3 : use nkp points for ldos/fms and nkp/5 points for pot (near edge) ; reduce nkp for all modules as we get away from near-edge (somewhat experimental)

```
* use a k-mesh of 1000 points in the full BZ for a k-space calculation of a crystal
KMESH 1000
```

```
* use a k-mesh of 10x5x3 points for a large, irregular cell
KMESH 10 5 3
```

```
* use a k-mesh of 1000 points and try to save time:
KMESH 1000 0 0 3
```

◆ **PRINT** `ppot pxsph pfms ppaths pgenfmt pff2x` *Standard*

The PRINT card determines which output files are printed by each module. See Section 4 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 4.1 on page 103.

◆ **DIMS** `nmax lmax` *Advanced*

This card limits the size of arrays so as not to exceed available memory. nmax is the maximum number of atoms in the cluster for FMS matrix inversion. Lmax is the maximal l-value for the potentials and Green's function. FEFF10 first determines the number of atoms and maximal l-value from user input, i.e. number of atoms given in the ATOMS card, the FMS-radius given in the FMS and SCF card, and l-values given in the POTENTIALS card. In a second step, it will truncate these values using the values of the DIMS card if present. If the DIMS card is not present, default cutoff values will be loaded from 'feff10/src/COMMON/m_dimsmod.f90' - these are fixed at compilation time.

```
* Limit l-values to 2 or lower.  Leave number of atoms alone
* (i.e. use compilation time limit for number of atoms).
* (negative numbers are ignored).
DIMS -1 2
```

◆ EGRID

Advanced

This card can be used to customize the energy grid. The EGRID card is followed by lines specifying the type of grid, minimum and maximum values for the grid, and the grid step, i.e.

```
grid_type grid_min grid_max grid_step
```

The `grid_type` parameter is a string that can take the values *e_grid*, *k_grid*, or *exp_grid*. When using the *e_grid* or *k_grid* grid types, `grid_min`, `grid_max`, and `grid_step` are given in eV or Å⁻¹ respectively. For the *exp_grid* type, `grid_min` and `grid_max` are the minimum and maximum grid values in eV, and `grid_step` is the exponential, i.e. $E_i = E_{Min} + \exp(\text{grid_step} * i) - 1.0$. A fourth grid-type *user_grid* is also available for FEFF. *user_grid* is followed by an arbitrary number of lines, each specifying an energy point in eV, i.e.,

```
user_grid
0.1
1.5
3.45
6.0
.
.
.
```

Note that the energies are all defined relative to the Fermi energy, i.e. the edge.

Up to ten grids can be specified for FEFF. The grids can also overlap. If one is using multiple grid types, a useful parameter *last* can be used in place of `grid_min`, i.e.

```
e_grid -10 10 0.1
k_grid last 10 0.1
```

The *last* parameter will use the last point of the previous grid as `grid_min`.

3.5.2 ATOMIC, POT, SCREEN : Scattering Potentials

◆ 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
```

◆ COREHOLE type

Standard

While the HOLE or EDGE card specifies which edge to calculate, the COREHOLE card determines how the core state is treated. There are three options : *none*, equivalent to the old NOHOLE card, meaning there is no core hole ; *RPA*, meaning the screen module calculates an RPA-screened core hole ; or a simple Final State Rule core hole (default). It is recommended to use the RPA core hole for k-space calculations. See 2.3 for more comments on choosing a core hole.

```
* To use the RPA screened core hole :
COREHOLE RPA
* To calculate without a core hole :
COREHOLE none
* To use a final state rule (non-screened) core hole :
COREHOLE FSR * or omit the card
```

◆ EDGE label s02

Standard

The EDGE card sets the edge. Simply use the hole label: K means *K*-shell, L1 means L_I , and so on. 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. 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
```

◆ SCF rfms1 [lfms1 nscmt ca nmix]

Standard

This card controls FEFF's automated self-consistent potential calculations. All fields except rfms1 are optional. If this card is not specified, then all calculations are done with non-self-consistent (overlapped atomic) potentials. 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. Will be ignored completely in k-space calculations. Must be converged in real-space calculations.

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 ℓ DOS. 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 UNFREEZE we sometimes had to use
SCF  3.7 0 30 0.05 10
```

◆ 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
```

◆ CONFIG input [configuration]

Useful

This card modifies the electron configuration. It allows the user to specify the ground-state occupation numbers of the orbitals of a species of atoms, either by atomic number Z or by potential type iph . It is possible to assign atoms with the same Z to different potential types with different configurations. The parameter *input* can take the following values:

- *feff7* signifying the same occupation numbers as were used in this old FEFF version;
- *file* indicating there's a file called 'config.inp' containing the relevant information;
- *card* stating that the information is passed inside the card itself in the optional parameter *configuration*.

Note that several instances of CONFIG card can appear in 'feff.inp' and all will be taken into account. There is a syntax to be used for specifying configurations; it follows the usual noble gas notations closely. The resulting configuration is written to file 'config.dat' for user inspection. The general format is *iph name [NobleGas] istate iocc- [iocc+] istate ...* where *iph* is a potential index of 'feff.inp' and *name* a 2-letter element name (C, Au, ...). If *iph* is negative, the card applies to all atoms of type *name*. It can even be used when there are no such atoms specified in 'feff.inp'. (Due to the presence of a core hole or when the ION card is used, FEFF may use this information.) The configuration can optionally be specified from a noble gas; acceptable values are He, Ne, Ar, Kr, Xe, Hg, or Rn. All remaining fields must specify states. *istate* = 1s, 2p, 4f, ... For s-states, only one occupation number *iocc* follows; for higher states, a $l-1/2$ and a $l+1/2$ occupation number is required. Occupation numbers can be fractional (e.g. 1.5). A positive number indicates a 'valence state'; a negative number indicates a 'core state'. This distinction is only used when calculating the exchange-correlation potential and has no implications for the rest of the calculation.

```
* For all (iph=-1) C atoms (Z=6) use 1s^2 2s^2 2p^2 (note there's 2p_1/2 and 2p_3/2)
* and consider the 2s and 2p valence electrons and 1s core electrons :
CONFIG card 1 * <-- "1" counts the following lines
-1 C 1s -2 2s 2 2p 1 1
```

A second example:

```
* Start from FEFF7 type configurations, but for the I atoms of potential type iph=2,
* start from Kr configuration and add 4d^10 5s^1 5p^6 (all valence)
```

```

CONFIG feff7
CONFIG card 1
2 I Kr 4d 4 6 5s 1 5p 2 4

```

Finally, it is also possible to set the ispn variable:

```

* For all Cr atoms of potential type iph=4:
CONFIG card 1
4 Cr Ar 3d 4 0 4s 1 4p 1 s 1 0 s 0

```

IMPORTANT: This card is new and experimental in FEFF10. Please double-check ‘`config.dat`’ to make sure your input has been parsed correctly, and contact the authors in case of doubt (or bugs). Also be aware that non-standard configurations may lead FEFF to fail, or may produce nonsense.

◆ 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 3.5.7 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–Lundqvist), `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.
```

◆ HOLE ihole s02

Useful

Deprecated. Supported for compatibility only. 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 3.1, 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 (γ_{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 ($\gamma_{\text{ch}} = 0.1 + 2 \cdot \text{vi0}$).

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

index	edge	index	edge	index	edge	index	edge
-	--	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 3.1: Available hole codes. The entries in the column *edge* are recognized by the EDGE card.

◆ NOHOLE

Useful

Deprecated. Supported for compatibility only. 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 — but will calculate potentials and phase shifts as if there is no core-hole. For $d\text{DOS}$ 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.

It is now recommended to use the COREHOLE card instead. However, old ‘feff.inp’ files with NOHOLE still work.

◆ 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 $\delta \equiv \text{delta}$. $\delta = 0.05$ 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
```

◆ UNFREEZEF

Useful

In some applications of f -electron systems, we found that it is preferable to freeze the f -electron density at the atomic value in order to achieve well converged SCF potentials. However, this may sometimes lead to less accurate results, and it doesn’t improve convergence in all cases. Freezing f -electrons is the default in FEFF8.4. If one wants to attempt to calculate the f -DOS self-consistently, as in FEFF8.0 and 8.1, the UNFREEZEF card should be used.

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

◆ CHSHIFT `Advanced`

csh

calculate initial core-state energy level using the final self-consistent potential. This card will give more accurate relative shifts of the edge, however, the absolute edge energy is not accurate with this method.

◆ CHBROADENING `igammach`

Advanced

If $iGammaCH = 1$, the core state lifetime effects are taken into account by convolving the final spectrum with a Lorentzian of width Γ_{ch} , as is done with broadening given by the CORRECTIONS card. If $iGammaCH = 0$ (default), the Green’s function is calculated for complex energy $E + i\Gamma_{ch}/2.0$, as is done for imaginary energy given in the EXCHANGE card. These should be equivalent for small Γ_{ch} . This card is useful for providing quick (rough) results for multiple edges. For example, one might like to calculate both the L2 and L3 edges of Fe without re-running the SCF and FMS modules. In this case the results should be quite good since

the core $P_{1/2}$ and $P_{3/2}$ core-hole potentials are essentially equivalent. In order to do this the following steps should be performed:

1. Put the CHBROADENING card in the feff.inp file with igammach set to 1, i.e.

CHBROADENING 1

2. Run feff through completely for the L2 edge, i.e., set the EDGE L2 card in the feff.inp file.
3. Copy the output files of interest (xmu.dat, eels.dat, chi.dat, ...) to another name or another directory.
4. Change the edge to L3, i.e., put EDGE L3 in the feff.inp file
5. Run the following executables in the following order
 - rdinp
 - atomic
 - mkgtr
 - path
 - genfmt
 - ff2x
 - eels

You should now have output files (xmu.dat, eels.dat, chi.dat, ...) for the L3 edge. Note: A similar procedure can be used for quickly obtaining correct results of different polarizations of a single edge.

◆ CHWIDTH gamma

Advanced

Set the core hole lifetime manually, instead of using FEFF's internal table. Gamma is in eV.

◆ CORVAL emin

Advanced

The core-valence separation energy is found by scanning the DOS within an energy window. The `emin` parameter sets the lower bound (in eV) of this energy window. It is 70 eV by default. For some materials it is necessary to lower this bound, e.g. to 100 eV. For example, when SCF convergence is elusive because occupation numbers for one or more l -values are changing drastically between SCF iterations due to states moving above and below a poor estimate of the core-valence separation energy. We plan to replace the current mechanism by a more robust and automated algorithm, but in the meantime users can use the CORVAL card to handle some of these difficult cases.

◆ **CORVAL** Emin*Advanced*

FEFF distinguishes between core and valence states. It searches for a so-called "core-valence separation energy" within an energy window that works well for most cases. Sometimes, the value found within the standard window is problematic, when states that are just above or below it shift across it during the SCF procedure. This can lead to convergence difficulties or invalid results. Using the CORVAL card, one can change the default lower bound of the energy window (usually in order to lower it). Input in eV. Default is $E_{min} = -70\text{eV}$.

◆ **EGAP***Advanced*

This card will only change results if run with the **MPSE** card. For a material with a band gap, the Fermi energy is usually set approximately at the top of the gap. The self-energy should however be referenced to the middle of the gap. If the band gap is known, the EGAP card can be used to tell FEFF its value.

◆ **EPS0** eps0*Advanced*

This card will only change results if run with the **MPSE** card. This card will renormalize the poles which represent the loss function so that it is consistent with the dielectric function specified by **eps0**. The expected effect is that the broadening in the spectrum caused by the self-energy will increase as a function of **eps0**, while the stretch (blue shifting of peaks relative to the edge) will decrease. Users should be careful with this card since the renormalization of the loss function shouldn't change its shape too much. A drastic change of shape in the loss function indicates that the original loss function is probably not a very good representation of the true loss function, or the dielectric constant is not correct. To check the shape of the new loss function, compare a plot of columns 1 and 4 in exc.dat with a plot of columns 1 and 2 in loss.dat.

◆ **EXTPOT***Advanced*

Use external potentials calculated by another code. Documentation currently unavailable. Experimental feature.

◆ **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=1` might be useful when only the surroundings of the absorbing atom are specified in 'feff.inp'.

inters

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

vtot

the volume per atom normalized by `ratmin`³ ($vtot = (\text{volume per atom}) / \text{ratmin}^3$), 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 $vtot \leq 0$ 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

This card can be unstable and should be used with caution. 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
```

◆ JUMPRM

Advanced

Remove potential jumps at muffin tin radii. Documentation currently unavailable. FIX

◆ NUMDENS `ipot numdens`*Advanced*

This card will only change results if used with the **OPCONS** card. The NUMDENS card sets the number density n_i of the atoms of type i specified by potential `ipot`. The number density is used to calculate the loss function for use with the many-pole model self-energy. By default the number density is estimated using the cluster of atoms provided in the `feff.inp` file as well as the stoichiometry. If the stoichiometry is not provided, it will be estimated as well based on the cluster provided. In some cases these estimates are inaccurate and a correct number density should be provided. The number is given by $n_i = N_i/V$, where N_i is the total number of atoms in a unit cell, and V is the unit cell volume.

◆ OPCONS

Standard

This card provides a fast and easy way to compute a rough estimate of the loss function using a weighted average of atomic loss functions. The output is contained in the file `loss.dat` and will be used in calculations of the self-energy if the **MPSE** card is also specified. The **NUMDENS** is useful if the OPCONS card is specified.

◆ PREPS

Advanced

If PREPS is specified along with the **OPCONS** card, the atomic dielectric function used for calculating the loss function will be printed in `epsilon.dat`.

◆ RESTART

Advanced

If RESTART is specified, FEFF will start the SCF calculation of the potentials from an existing `pot.bin` file, rather than starting from overlapped atomic potentials. This way one can continue an earlier SCF calculation (e.g. after it ran out of iterations), or one can use an existing set of potentials as a good starting point for a similar structure (e.g., after making small changes to the atom coordinates). Note that this strategy is not guaranteed to lead to good results, or to even converge. This is a new feature and has not been tested in all possible situations. If you need assistance, please contact us.

◆ SCREEN `parameter value`*scr*

The **screen** module, which calculates the RPA core hole potential, is a 'silent' module: it has no obvious input but instead runs entirely on default values. Using the SCREEN card you can change these default values. They will be written to an optional `screen.inp` file (which you can also edit manually). The SCREEN card can occur more than once in `feff.inp`; all entries will be applied to the calculation. **parameter** must be one of : `ner` (40), `nei` (20), `maxl` (4), `irrh` (1), `iend` (0), `lfxc` (0), `emin` (-40 eV), `emax` (0 eV), `eimax` (2 eV), `ermin` (0.001 eV), `rfms` (4.0), `nrptx0` (251). For most calculations the default values given (between brackets) are fine. Occasionally we've changed `rfms`, `maxl`, or `emin`. Note that the **screen** is only active with COREHOLE RPA.

```
* Set the cluster radius for the RPA potential calculation higher than the default of 4.0
COREHOLE RPA
SCREEN rfms 5.0
```

◆ **SETEDGE***Advanced*

This will set the edge energies to those found in the Elam/Mcmasters tables for the element and edge in question. By default the estimated edge energy is based on Hartree-Fock atomic calculations. This card is meant to be used with an external program for obtaining the full optical constants including response from all occupied electrons.

◆ **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 2.9.3) .

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 **nsp_x**=1 (default), the values of the index **ispin** correspond to the following calculations:

ispin	meaning
2	spin-up SPXAS and ℓ DOS
-2	spin-down SPXAS and ℓ DOS
1	spin-up portion of XMCD calculations
-1	spin-down portion of XMCD calculations

Table 3.2: 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 **nsp_x** = 2, one simply uses **ispin** = 1 or **ispin** = 2 as needed, and the up and down spins are taken care of automatically. See Section 2.9 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.

```
* Do 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

```

3.5.3 XSPH: Cross-section and Phase Shifts

◆ LDEC 1d

Advanced

Only active for NRIXS. This option gives the number of angular momentum channels $l \leq ld$ for which the decomposition of the inelastic scattering spectrum is given. The output for this card is written into file 'xmul.dat' in **ff2x**. Changing ld does not change the total spectrum, but only the amount of partial data written to file.

```

* Output momentum channels up to l=2
LDEC 2

```

◆ LJMAX 1j

Advanced

This is an NRIXS option and will be ignored for any other type of spectroscopy. LJMAX influences all modules starting from xsph. This card specifies the number of terms included in the calculation of the excitation matrix elements

$$M_L(\vec{q}, E) = \langle R_L(E) | e^{i\vec{q} \cdot \vec{r}} | i \rangle \approx \langle R_L(E) | 4\pi \sum_{lm}^{l_j \geq l} i^l j_l(qr) Y_{lm}^*(\vec{q}) Y_{lm}(\vec{r}) | i \rangle \quad (3.1)$$

Typically the value of angular momentum channels needed is dependent on the momentum transfer and energy transfer range that one is calculating. For typical XANES calculation one does not need $lj > 10$ and most often $lj = 5$ is sufficient. For EXAFS one needs more than for XANES. However, most often this parameter has to be determined on case-by-case basis. If this parameter is too large it can occasionally cause small errors. For $lj > 25$ the code starts to become unstable. Default value is $lj = 1$, approximately corresponding to the dipole selection rule. This is almost certainly too low for any situation. There is no physical limitation on lj ; it simply needs to be converged.

```

* Use contributions to the GOS up to l=5
LJMAX 5

```

◆ **MPSE** `ipl`*Standard*

This card runs FEFF with a many-pole model for the self energy.

The values of `ipl` correspond to the following: 1: use an “average” self-energy which is applied to the whole system (default). 2: use a density dependent self-energy which is different at every point inside the muffin-tin radius.

It takes as input a file ‘`loss.dat`’ containing the loss function. This can either be measured or taken from a calculation. For a rough but very fast estimate of the loss function, see the **OPCONS** card. See the references for more information. The output file ‘`mpse.dat`’ contains the self energy and the renormalization constant (at the interstitial radius) as a function of photoelectron energy relative to the Fermi Energy in eV.

PLASMON can be used as an alias to the MPSE card.

```
* make regular FEFF calculations, print out the self-energy
MPSE 0
```

◆ **PMBSE** `ipmbse nonlocal ifxc ibasis`*Advanced*

This card is currently in development.

◆ **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** `ifxc`*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 `ifxc` determines whether static or dynamic screening is used. `ifxc = 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. `ifxc = 1` (dynamic screening) accounts for screening of the x-ray and core-hole fields, blue-shifting the spectrum less than `ifxc = 0`. The TDLDA card affects only module xsph.

TDLDA theory takes into account polarization-type 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 references in [Appendix C](#)) for further details on the implementation of TDLDA in the FEFF calculations. See also Ch. 8 (pp. 82-105) of [A.I. Nesvizhskii's thesis](#) for a brief description of the TDLDA theory.

On certain occasions, instabilities have been observed in the TDLDA routines, causing sometimes NaN output or unphysical glitches in the tails of edges. If such symptoms occur, the FEFF authors should be contacted for help.

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

3.5.4 FMS and MKGTR : Full Multiple Scattering

◆ **FMS** `rfms [lfms2 minv toler1 toler2 rdirec]` *Standard*
 Compute full multiple scattering within a sphere of radius `rfms` centered on the absorbing atom (real space) or for the unit cell of the crystal (k-space).

`rfms`

Completely ignored in k-space calculations. This is the cluster radius used in modules **ldos** and **fms** but not in **pot**. It is also 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.

For EXAFS analysis, one typically calculates to $k = 20$, but FMS results are not accurate at high energies. Thus if you are running FEFF10 for EXAFS, you should not use the FMS card.

If the value of **RPATH** as described in Section 3.5.5 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 ℓ DOS and XANES:

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

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 **path** modules. However, this is generally not recommended, as the MS expansion sometimes does not converge well in the

XANES energy region. Thus it is generally best not to add 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, but 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 description [here](#).

◆ **BANDSTRUCTURE**

Advanced

This card activates the module that calculates and prints out bandstructure. This card does not work in the current version of FEFF, and you may get an error if you try to use it. The authors really want you to have your spaghetti, but they're still in the kitchen working on it! Keep fingers crossed ...

```
* give me a bandstructure. Will likely acquire more options in the future
BANDSTRUCTURE
```

◆ **STRFAC** eta gmax rmax

Advanced

This card gives three non-physical internal parameters for the calculation of the KKR structure factors : the Ewald parameter and a multiplicative cutoff factor for sums over reciprocal (gmax) and real space (rmax) sums. Multiplicative means the code makes a 'smart' guess of a cutoff radius, but if one suspects something fishy is going on, one can here e.g. use gmax=2 to multiply this guess by 2. Eta is an absolute number. Given the stability of the Ewald algorithm, it shouldn't be necessary to use this card. Its use is not recommended. Only active in combination with the RECIPROCAL card.

```
STRFAC 0.4 2.0 2.0
```

3.5.5 PATH: Path Enumeration

◆ **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. 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 ℓ DOS 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 path finder 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 **path**, see Table 4.1), 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   29     1    48  5.98      parameters for 19th shell of Cu
```

◆ SYMMETRY isym

Advanced

FEFF uses symmetry to reduce the list of paths. This process is automated and takes into account both the properties of the system and of the spectroscopy (eg., time reversal symmetry,

...). However sometimes one may wish to control the type of symmetry directly using the SYMMETRY card with isym ranging from 1 to 7.

1. any path rotation, reflection and reversal are allowed
2. any rotation around evec, reflections in planes normal and parallel to evec, path reversal
3. reflections in 2 planes (xivec, evec) and (xivec, B field)
4. rotations around xivec, path reversal
5. rotations around spin axis
6. only 180 degrees rotation around spin axis
7. no symmetry operations

* force using no symmetry at all - warning : long and slow path list!
SYMMETRY 7

3.5.6 GENFMT: XAFS Parameters

◆ 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 FEFF10 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 FEFF10, 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` writes the file 'nstar.dat' with the effective coordination number N^* which is the coordination number weighted by $\cos^2(\theta)$ to correct for polarization dependence in SEXAFS calculations.

3.5.7 FF2X: XAFS Spectrum

◆ DEBYE temperature Debye-temperature [idwopt]

Standard

The DEBYE card is used to calculate Debye-Waller (DW) factors to account for thermal

disorder. The effect of the card depends somewhat on the situation.

When used with the MS path expansion (EXAFS and RPATH cards), a DW factor is calculated for each path using the correlated Debye Model. They will be summed with the DW factors from the **SIG2** card and from the 'list.dat' file, if present - so the total DW factor for any path is the sum of these three factors!

When used with the FMS technique (XANES and FMS cards), a single-scattering DW factor is calculated for each pair of atoms in the cluster and a factor $\exp(-\sigma^2 k^2)$ is multiplied to the free propagator. This approach is inexact beyond single scattering paths. However, it is likely adequate in the near-edge region where FMS is valid, since the shorter mean free path makes this region less sensitive to thermal effects.

By default, `idwopt = 0` specifies the correlated Debye model for Debye–Waller factors. It 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. Note that the **DEBYE** card is incompatible with the **CFAVERAGE** card for options other than the correlated Debye model. Temperatures are specified in Kelvin.

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

Several additional models for calculating DW factors are available in FEFF10 based on the information about the harmonic force constants in the material. The **DEBYE** card offers a choice between 5 different models for the DW factors:

- 0 Correlated-Debye method (default) (CD)
- 1 Equations of Motion method (EM)
- 2 Recursion method (RM)
- 3 Classical Correlated-Debye method (CCD)
- 4 Read from "sig2.dat" file
- 5 Dynamical-Matrix method (DM)

`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. `idwopt = 5` uses ab initio DW factors from a dynamical matrix using the **dmdw** module. This provides very accurate DW factors. All these methods are described in more detail in **Chapter 3**.

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

◆ ABSOLUTE

Useful

By default FEFF normalizes spectra at 40 eV above threshold. This card disables normalization. ABSOLUTE is automatically set whenever ELNES, EXELFS, or NRIXS cards are used.

```
ABSOLUTE      * don't normalize spectra in ff2x
```

◆ 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 3.5.2).

```
* 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 \AA^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
```

◆ SIG3 AlphaT ThetaE

Useful

Documentation currently unavailable. FIX

◆ MBCONV

Advanced

Many-body convolution. Documentation currently unavailable. FIX

3.5.8 SFCONV: Spectral Function Convolution

FIX

◆ SFCONV

Useful

SFCONV convolutes the single particle XAS files ('xmu.dat', 'chi.dat', 'feffNNNN.dat', etc.) from FEFF with a many body spectral function to include many body effects in the spectra. This includes an ab-initio calculation of the amplitude reduction factor S_0^2 . SFCONV uses Hedin-Lundqvist self energies, so it will probably work best with XAS files calculated using FEFF's HL self energies.

SFCONV

◆ RCONV cen cname

Advanced

Documentation currently unavailable. FIX

◆ SELF

Advanced

Print out on-shell self energy as calculated by this module.

SELF

◆ SFSE k0

Useful

Print out off-shell self energy $\Sigma(k0, E)$.

SFSE

3.5.9 COMPTON: Compton scattering

◆ CGRID [zpmax ns nphi nz nzp]

Useful

Sets the grid on which $\rho(z, z')$ is evaluated for Compton scattering. zpmax is the upper limit on z' . The other four parameters set the density of the spatial grid.

CGRID 10 32 32 32 120

◆ RHOZZP

Useful

Calculate a slice $\rho(z, z')$ with z fixed near the origin and z' moving outward along the z -axis to zpmax (set in CGRID). The result is saved in 'rhozzp.dat'. Useful as a test, or to get a quick estimate of some parameters in the other COMPTON cards. Only to be used in conjunction with the COMPTON card.

RHOZZP

3.5.10 EELS: Electron Energy Loss Spectrum

This module assembles an EELS spectrum as specified by the card ELNES or EXELFS. See Section 2.12 for a more complete description of EELS calculations.

◆ **MAGIC** `emagic`

Useful

The MAGIC card makes FEFF calculate cross sections as a function of collection angle at a particular energy loss. So, in addition to the energy-resolved output produced by the EELS cards, the MAGIC card produces angle-resolved output.

This card enables one to find the magic angle for a material very quickly : run FEFF for two different orientations, and see where the sp^2 -curves cross. This gives you the magic collection angle for the convergence angle and EELS edge chosen in ‘`feff.inp`’. Plus the entire workings of the `eels` module are based on black magic, of course.

```
* create a plot that shows the  $sp^2$  ratio at 20 eV above threshold.  
MAGIC 20
```

Chapter 4

Input and Output Files

Any FEFF calculation produces a large number of files that may be confusing to the novice. The most important files are the master input file ‘`feff.inp`’, which we discussed at length in the previous chapters ; the output file ‘`xmu.dat`’, which contains the XAS spectrum ; and the various ‘`log*.dat`’ files, which are log files written by FEFF during execution.

FEFF creates a variety of output files depending on the spectroscopy type selected by the user, and depending on verbosity settings. See the **PRINT** card in Section 3.5.1 and 4 to obtain various diagnostic files.

FEFF also writes several intermediate files. Some of these group relevant settings from ‘`feff.inp`’ and program defaults for each of FEFF’s program modules. These files, called ‘`pot.inp`’, ‘`xsph.inp`’ and so on, can be manipulated by the expert user who knows what she is doing; see Section 4.3. However we advise that novice users always work through ‘`feff.inp`’ to avoid mistakes.

Section 4.1 summarizes data flow. The rest of this section describes selected files in more detail.

Regrettably this chapter is somewhat out of date and a handful of newer files are not described. This includes NRIXS and COMPTON output.

4.1 Data flow

Module 0 rdinp

Purpose of Module: Process input data

Input files: 'feff.inp'

Output files: 'geom.dat', 'global.inp', 'pot.inp', 'xsph.inp', 'fms.inp', 'path.inp', 'genfmt.inp', 'ff2x.inp', 'ldos.inp', and 'eels.inp'

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 intermediate files, which contain formatted data needed for all modules.

Module 1 atomic

Purpose of Module: Calculate atomic potentials for the photoelectron

Input files: 'pot.inp' and 'geom.dat'

Output files: 'apot.bin'

Other output: 'fpf0.dat'

Description: Reads 'pot.inp' and calculates atomic potentials for the photoelectron, which are written into 'apot.bin'.

Module 2 pot

Purpose of Module: Calculate embedded atomic potentials for the photoelectron

Input files: 'pot.inp', 'apot.bin' and 'geom.dat'

Output files: 'pot.bin'

Other output: diagnostic files (see Table 4.1 on page 103)

Description: Reads 'pot.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 3 screen

Purpose of Module: Calculate embedded atomic potentials for the photoelectron

Input files: 'pot.bin' and, optionally, 'screen.inp'

Output files: 'wscrn.dat'

Description: Calculates screening, which is written into 'wscrn.dat'.

Module 4 opconsat

Purpose of Module: Unknown

Input files: Unknown

Table 4.1: Print levels controlling output files from the modules. Log-files are not included. For ELNES calculations, a proliferation of files may occur, eg. ‘xmu.dat’ through ‘xmu09.dat’ will be produced. NRIXS calculations not included in this table. **rdinp** not included.

module	print levels
atomic	0 write ‘apot.bin’ and ‘fpf0.dat’
pot	0 write ‘pot.bin’ only 1 add ‘misc.dat’ 2 add ‘potNN.dat’ 3 add ‘atomNN.dat’
screen	0 write ‘wscrn.dat’ if COREHOLE RPA used
ldos	0 write ‘ldosNN.dat’ if LDOS used
xsph	0 write ‘phase.bin’ and ‘xsect.bin’ only 1 add ‘axafs.dat’ and ‘phase.dat’ 2 add ‘phaseNN.dat’ and ‘phminNN.dat’ 3 add ‘ratio.dat’ (for XMCD normalization) and ‘emesh.dat’. Careful! xsph now takes a lot of time.
fms	0 write ‘gg.bin’ 1 write ‘gg.dat’
mkgtr	0 write ‘fms.bin’ and ‘gtr.dat’
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’ ($\chi(k)$ for each path individually) 3 add ‘feffNNNN.dat’ and ‘files.dat’, and do not add ‘chipNNNN.dat’ files
sfconv	0 overwrites ‘xmu.dat’
eels	0 write ‘eels.dat’; write ‘magic.dat’ if MAGIC card used
compton	0 ‘compton.dat’, ‘rhozzp.dat’, ‘izzp.dat’, ‘ipq.dat’

Output files: Unknown

Description: Unknown

Module 5 ldos

Purpose of Module: Calculate LDOS

Input files: 'ldos.inp', 'geom.dat', and 'pot.bin'.

Output files: 'ldosNN.dat' (ℓ DOS) and 'logdos.dat'

Other output:

Description: LDOS runs only if the LDOS card is present in 'feff.inp'. It outputs the angular momentum (ℓ) projected DOS into 'ldosNN.dat' files, with each value of NN corresponding to each unique potential.

Module 6 xsph

Purpose of Module: Calculate cross-section and phase shifts

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

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

Other output: diagnostic files (see Table 4.1 on page 103), and 'axafs.dat'.

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.dat' and used in the module (**ff2x**) for overall normalization. Optionally, **xsph** will write other diagnostic files with information about the phase shift calculations.

Module 7 fms

Purpose of Module: Calculate full multiple scattering for XANES, ELNES and ℓ DOS

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

Output files: 'gg.bin'

Other output: optionally 'gg.dat'

Description: Performs the full multiple scattering algorithm. Writes the Green's function matrix to 'gg.bin'.

Module 8 mkgtr

Purpose of Module: Calculate full multiple scattering for XANES, ELNES and ℓ DOS

Input files: 'fms.inp', 'global.dat', 'geom.dat', 'gg.bin' and 'phase.bin'

Output files: 'fms.bin'

Other output: optionally 'gtr.dat'

Description: Traces the Green's function matrix and adds in matrix elements. Output written to 'fms.bin'. Writes output into 'fms.bin' for the **ff2x** module, which contains the $\chi(k)$ from **fms**.

If an ELNES/EXELFS card is present, all of the requested components of the sigma tensor are written to 'fms.bin', instead of just one.

Module 9 **path**

Purpose of Module: Path enumeration

Input files: 'path.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'.

If an ELNES/EXELFS card is present, a separate 'list.dat' file is written for each polarization component (i.e., 'list.dat', 'list02.dat', etc.).

Module 10 **genfmt**

Purpose of Module: Calculate scattering amplitudes and other XAFS parameters

Input files: 'genfmt.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.

If an ELNES/EXELFS card is present, a 'listNN.dat' file is written for each polarization component, and a separate 'feff.bin' file is written (i.e., 'feff.bin', 'feff02.bin', etc.). The format of the files is unchanged.

Module 11 **ff2x**

Purpose of Module: Calculate specified x-ray spectrum

Input files: 'ff2x.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.

If an ELNES/EXELFS card is present, this module reads the large ‘fms.bin’ and all the ‘feffNN.bin’ files, and produces a ‘xmuNN.dat’ file containing the corresponding component of the sigma tensor (‘xmu.dat’, ‘xmu02.dat’, ..., ‘xmu09.dat’). Those files have the traditional ‘xmu.dat’ format. Similarly, ‘chiNN.dat’ files are produced.

Module 12 **sfconv**

Purpose of Module: Convolve output files with the spectral function.

Input files: ‘sfconv.inp’, ‘xmu.dat’, ‘chi.dat’, ‘chipNNNN.dat’, or ‘feffNNNN.dat’ files

Output files: ‘specfunct.dat’,

Other output: The following files are overwritten with convolved spectral data: ‘xmu.dat’, ‘chi.dat’, ‘chipNNNN.dat’, and ‘feffNNNN.dat’.

Description: SFCNV convolutes the single particle XAS files with a many body spectral function to include many body effects on the spectra, including an ab-initio calculation of the amplitude reduction factor S_0^2 . This module runs after **ff2x** if the SFCNV card is present in ‘feff.inp’.

In the presence of an EELS card, the module is run for all components of the sigma tensor: each ‘xmu.dat’ file is opened and altered individually.

Module 13 **eels**

Purpose of Module: Calculate EELS

Input files: ‘eels.inp’

Output files: ‘eels.dat’ and ‘logeels.dat’

Other output: optionally, ‘magic.dat’

Description: EELS runs only in the presence of an ELNES or EXELFS card. It reads ‘eels.inp’, and sums the partial spectra from the various polarizations to assemble a physical EELS spectrum.

4.2 File Reference

4.2.1 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 XANES, EXAFS or NRIXS data depending on the situation; μ , μ_0 , and $\tilde{\chi} = \chi \frac{\mu_0}{\mu_0(\text{edge}+50\text{eV})}$ 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.

‘eels.dat’

This contains the EELS spectrum: energy loss in eV, total spectrum, contribution from each component of the cross section tensor.

‘magic.dat’

This file is only written if the **MAGIC** card is used. It contains the collection angle in rad, the pi to sigma ratio, the pi and sigma components of the spectrum, and the total spectrum.

4.2.2 EELS Files

‘eels.inp’

is read by the **eels** module and determines what FEFF will actually do. Expert users can

tweak this file directly. It contains all the options of the EELS and MAGIC cards. The very first parameter determines whether **eels** is executed (=1) or not (=0). The parameters on the next line select the components (1-9) of the sigma tensor to be calculated. These parameters are very important because most other program modules check for the presence of **'eels.inp'** and the values of these parameters to determine their course of action. People who have done EELS and then want to do something else in the same working directory may want to set the execution switch to 0 (or comment the EELS card in **'feff.inp'** and rerun **rdinp**, which amounts to the same) to make sure none of the regular modules do anything special for EELS.

'eels.dat'

contains the EELS spectrum. Its first column has energy loss in eV, the second column the total spectrum, and the next columns contain the contribution to the total spectrum from each of the nine components of the cross section tensor (xx, xy, ..., zz)

'magic.dat'

is only written if the MAGIC card is present in feff.inp. It contains the collection angle in rad, the π to σ ratio, the π and σ components of the spectrum, and the total spectrum; all as a function of collection angle, evaluated at the energy loss set by the MAGIC card.

'logeels.dat'

contains reports on the execution of the **eels** module. In particular, it contains a summary of the input options used. Most of the information in the file is also written to the screen during program execution.

4.2.3 Intermediate 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 NNth 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 4.3. (‘pathNN.dat’ files are also created during the ℓ DOS 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 4.3.

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

‘specfunct.dat’

This is a binary file containing the spectral function. If it is not present, **sfconv** will create it. Every time the material is changed, ‘specfunct.dat’ will be recomputed.

4.2.4 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 $\ell=0,1,2$ 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.

`'dimensions.dat'`

Passes array size between modules. Mess with this and all hell will break loose.

`'scfconvergence-feff'`

Tracks charge density difference and convergence throughout SCF iterations in **pot**.

4.2.5 Variables in the EXAFS and XANES Formulae

FIX move to chapter 3. add other spectroscopies.

k The wave number in units of \AA^{-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.

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

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 \AA , $\lambda = 1/|\text{Im } p|$

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

4.3 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.

4.3.1 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).

4.3.2 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**!)

4.3.3 Using `'geom.dat'`

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

Appendix A

Copyright Information, Restrictions and License

A.1 Restrictions and License Information

The FEFF10 distribution is copyrighted software according to the terms in the FEFF10 License and Government copyrights in Appendices A.2 and A.3 below.

A.2 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 FEFF10 LICENSE

Distribution: FEFF10.0 Copyright (c) 2020 FEFF Project, University of Washington and SLAC National Accelerator Laboratory. All rights reserved.

This software was prepared with support in part from the U.S. DOE, Office of Basic Energy Sciences, DMSE grants DE-FG03-97ER45623 (Dept. of Physics, U.Washington) and DE-AC02-76SF00515 (Theory Institute for Materials and Energy Spectroscopies (TIMES) at SLAC.

Redistribution and use of this Distribution in source and binary formats, with or without modification, is permitted, provided the following conditions are met:

Redistributions must retain the above notices and the following list of conditions and disclaimer;

Modified formats carry the marking "Based on or developed using Distribution: FEFF10.0 Copyright (c) 2020 FEFF Project, University of Washington and SLAC National Accelerator Laboratory. All rights reserved."

Recipient acknowledges the rights of the University of Washington and SLAC to prepare uses of this Distribution and its modifications that may be substantially similar or functionally equivalent to Recipient-prepared modifications.

Recipient and anyone obtaining access to the Distribution through recipient's actions accept all risk associated with possession and use of the Distribution.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE UNIVERSITY OF WASHINGTON OR CONTRIBUTORS TO THE DISTRIBUTION BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR REVENUE; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Appendix B

Installation Instructions

B.1 Compiling your own code

Compiling your own FEFF binaries is a good idea, as the resulting executables will generally be faster than ours. If you obtained the FEFF source tree (probably as a package `'feff10.tar.gz'`), then the code is in `'feff10/src'` - so that's where you need to go.

If you're on Linux/Unix/OS X, you're lucky. Open the `'Compiler.mk'` file and edit the statement specifying the fortran90 compiler and its compilation options. Close the file, and type *make* on the command line. If you're interested in using parallel MPI calculations for increased speed on parallel computer infrastructure, give the *make mpi* command. When switching between sequential and MPI compilation, you should ALWAYS use the *make clean* command to flush intermediate files, otherwise the executables may crash and burn. The resulting executables now ought to be in `'feff10/bin/Seq'` or `'feff10/bin/MPI'`. These folders will contain a set of 16 executables that together comprise the FEFF code.

We can make "project files" available for Mac OS X + XCode 3 + Intel Fortran Composer ; and for MS Windows + MS Visual Studio + Intel Fortran Composer . This is how we produce MS Windows executables. Alternatively, you could try to make the "Makefile" setup work on MS Windows + Cygwin.

Since the two main dimensioning parameters, *nclusx* and *lx*, are now dynamically set by the code during each run, there is less need for recompilation than in older versions of FEFF. Other dimensioning parameters, including the maximum number of potentials *nphx* and the maximum number of spin states *nsp_x*, are still static and can only be changed through recompilation. You may want to edit `'feff10/src/COMMON/m.dimsmod.f90'` to set these to the desired value before you compile.

We have endeavoured to make FEFF portable to all architectures 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.

Compilation requirements have changed from previous versions of FEFF. FEFF is now a

fortran90 code, and hence a fortran90 compiler is now required. If you do not currently have a fortran90 compiler, Intel Fortran Compiler is free for personal use on Linux platforms, and reduced pricing is available for academic licenses.

We provide precompiled binaries for different architectures, compiled using Intel Fortran Composer 12.5 on all platforms. We are satisfied with that compiler, but you can also try gfortran, g95, pgf90, or any compiler you like.

We generally recommend starting out with a conservative compilation - all optimization disabled, maximum safety compiler flags especially for maintaining floating point precision. Then recompile a faster version checking against the original results. Use a testcase containing the SCF and FMS cards to make sure the FMS routines are safely compiled.

The FEFF source code is fully self-contained, i.e. it doesn't need to be linked against external libraries.

We've noticed that on some machines it is necessary to increase stack size or available virtual memory in order to run the code successfully. Your sys admin or your system's documentation can help you do this.

Appendix C

References

Please cite FEFF and an appropriate FEFF reference if the code or its results are used in published work.

The main references for the theory of FEFF are:

Calculations of x-ray spectroscopies using FEFF10 and Corvus, J.J. Kas, F.D. Vila, D. Pemmaraju, Tun S. Tan, S. Vimolchalao, and J.J. Rehr, *Unpublished*, 2020.

Ab initio theory and calculations of X-ray spectra, J.J. Rehr, J.J. Kas, M.P. Prange, A.P. Sorini, Y. Takimoto, F.D. Vila, *Comptes Rendus Physique* 10 (6) 548-559 (2009)

and for EXAFS theory:

Theoretical Approaches to X-ray Absorption Fine Structure, J. J. Rehr and R. C. Albers, *Rev. Mod. Phys.* **72**, 621, (2000).

The main reference for calculations using the current version FEFF10 is:

Parameter-free calculations of X-ray spectra with FEFF9, J.J. Rehr, J.J. Kas, F.D. Vila, M.P. Prange, K. Jorissen, *Phys. Chem. Chem. Phys.* 12 (21) 5503-5513 (2010).

Other publications discussing specific FEFF developments are given below.

FEFF9

“Parameter-free calculations of X-ray spectra with FEFF9”, J.J. Rehr, J.J. Kas, F.D. Vila, M.P. Prange, K. Jorissen, *Phys. Chem. Chem. Phys.* 12 (21) 5503-5513 (2010).

K-space

“Calculations of electron energy loss and x-ray absorption spectra in periodic systems without a supercell,” K. Jorissen and J.J. Rehr, *Phys. Rev. B* 81, 245124 (2010).

EELS “Multiple scattering calculations of relativistic electron energy loss spectra,” K. Jorissen, J. J. Rehr, and J. Verbeeck, *Phys. Rev. B* 81, 155108 (2010).

NRIXS

“Inelastic Scattering from Core-electrons: a Multiple Scattering Approach,” J. A. Soininen, A. L. Ankudinov, and J. J. Rehr, *Phys. Rev. B* 72, 045136 (2005).

Inelastic Mean Free Path

“Ab initio calculations of mean free paths and stopping powers,” A. P. Sorini, J. J. Kas, J. J. Rehr, M. P. Prange and Z. H. Levine, *Phys. Rev. B* **74**, 165111 (2006).

Debye-Waller Factors

“Theoretical X-Ray Absorption Debye-Waller Factors,” Fernando D. Vila, J. J. Rehr, H. H. Rossner, H. J. Krappe, *Phys. Rev. B* **76**, 014301 (2007).

Many-Pole Self Energy

“Many-pole model of inelastic losses in x-ray absorption spectra,” J.J. Kas, A. P. Sorini, M. P. Prange, L. W. Cambell, and J. A. Soininen, and J. J. Rehr, *Phys. Rev. B* **76**, 195116 (2007).

FEFF8

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

Parallelization and Lanczos

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

TDLDA

“Dynamic screening effects in x-ray absorption spectra,” A.L. Ankudinov, A.I. Nesvizhskii, and J.J. Rehr, *Phys. Rev. B* **67**, 115120 (2003).

EELS “Practical aspects of electron energy-loss spectroscopy (EELS) calculations using FEFF8,” M.S. Moreno, K. Jorissen, and J.J. Rehr, *Micron*, **38**, 1 (2007).

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); 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).

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).

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).

Sum rules

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

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 atomic 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

For historical reasons FEFF is a mix of older fortran77 code and newer fortran90 code. Hence the most important array dimensions are treated dynamically, while others are set at compilation time. If you need to run larger problems than the dimension statements in the code allow, you must change the dimensions in the file ‘feff10/src/COMMON/m_dimsmod.f90’ and recompile. If you need help, please contact the authors.

The following parameters are supported for dynamical treatment :

```
c      number of atoms for FMS.
        integer nclusx
c      highest orbital momentum for FMS module.
        integer lx
```

For these parameters, the module **rdinp** will determine from user input what array size is needed, eg. to fit all atoms of the cluster and all orbital momentum values. These will be truncated to a precompiled upper limit to protect exceeding available memory (which might result in nasty errors that would be problematic especially on large computing systems where tasks have to go through a queue and crashes cannot be immediately rectified) ; optionally, the user can set these truncation values using the **DIMS** card.

For all static parameters below, all arrays will be allocated with a fixed dimension regardless of the problem at hand. These parameters are (values may be different in your release) :

```
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 =2000)
c      max number of atoms in problem for the rdinp and ffsort
        parameter (nattx =10000)
c      max number of unique potentials (potph) (nphx must be ODD to
c      avoid compilation warnings about alignment in COMMON blocks)
        parameter (nphx = 9)
```

```

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)

```

Of these parameters, a user is most likely to want to change `nphx` and `nspx`. Care should be taken in changing these parameters. Whenever the expected length of an array changes, errors, or, worse, unphysical results may occur when FEFF then reads intermediate files produced before the changes. In practice, this means that

- whenever one changes user input (`'feff.inp'`) or the hardwired truncation limits for the dynamical dimension parameters such that the length of any arrays in the calculation will change, all affected parts of the calculation must be rerun before continuing or reusing parts of it.
- whenever one changes any of the static parameters, all intermediate files containing arrays dimensioned by the affected parameter will become useless and the calculation must be redone to regenerate these files before they can be used.

Of course, it is entirely possible to have more than one compiled version of FEFF on your system, as long as you don't mix them up and always use the same version for a given calculation.

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 `ff2x` 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.

The files `‘.feff.dims’` and `‘.feff.comp.stats’` list the values of all important parameters used in a given calculation. It’d be a really bad idea to edit these files, but looking won’t hurt.

Appendix E

Spin.f program for XMCD and SPXAFS

'spin.f':

```

      program spin_sum
c    written in grandmother's fortran-77
      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  1 or -1
c    case 2) you want XMCD signal and used SPIN 2 or -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

```

Appendix F

Technical notes for dynamical matrix based Debye-Waller factors

DMDW is a set of tools developed to calculate Debye-Waller (DW) factors and other related quantities from a dynamical matrix (matrix of force constants or Hessian matrix) using the Lanczos recursive algorithm.[Refs.] This set includes a module integrated into FEFF, a standalone version that can be used independently of FEFF and a Fortran module that can be integrated into third-party programs. DMDW also includes conversion tools to generate the required input files from different ab initio programs.

F.1 Installing DMDW

F.1.1 Installing as a FEFF module

DMDW is included in the FEFF distribution and is installed as a FEFF module with all the other program modules. Please refer to the FEFF installation instructions. After the FEFF installation is completed a module called **dmdw** is located under the FEFF '**feff10/bin/Seq**' directory structure.

F.1.2 Creating and installing the standalone version

Within the '**src/DMDW**' directory in the FEFF distribution, execute the following command:

```
make standalone
```

This will create a directory '**feff10/src/DMDW/dmdw_standalone**'. Edit '**dmdw_standalone/src/Makefile**' and change the "F90" variable to your fortran 90 compiler of choice. Then execute the following commands:

```
make
```

```
make install
make examples
```

If everything worked correctly the reference results located in ‘`dmdw_standalone/examples/Reference.Results`’ should agree with those generated in ‘`dmdw_standalone/examples`’ using the newly compiled executable.

F.2 Using DMDW to calculate Debye-Waller factors

F.2.1 Using within a XANES or EXAFS calculation in FEFF

See [Chapter 5](#).

F.2.2 Calculating DW factors using the standalone version

Capabilities of DMDW beyond the FEFF DEBYE card can be accessed by means of the **dmdw** module or by compiling the standalone version. All the details described below apply to the input used by both the **dmdw** module and the standalone version. During the execution of a normal FEFF run using ab initio DW factors, an input file ‘`dmdw.inp`’ for the **dmdw** module is automatically generated based on the options used in the DEBYE card. This autogenerated input can be used “as is” with the standalone version, or further edited to access other capabilities.

‘`dmdw.inp`’ is composed of at least 5 lines of input. All the parameters must be entered and no default values are available:

```
Lanczos_Order
nT T_Min T_Max
DW_Type
Filename
nPathDesc
PathDesc1
PathDesc2
.
.
```

where:

- Line 1 - Lanczos_Order: Number of Lanczos iterations (integer).

This parameter is equivalent to the `DMDW_Order` parameter described for the DEBYE card. It corresponds to the number of Lanczos iterations to be used in the calculation. Well converged results are usually obtained for $DMDW_Order = 6 - 10$. For small size systems, these values might be too large. As a rule of thumb, this value should be less than $3 * (Numberofatoms) - 6$. Some paths in systems with high symmetry might

require a lower recursion order. The user should always check for convergence with this parameter.

- Line 2 - nT: Number of temperature values in grid (integer) T_Min, T_Max: Minimum and maximum temperature values (real, in K)

Define a grid of temperatures in which to calculate the DW factors. This option is very efficient in the generation of whole temperature curves since it performs the Lanczos procedure only once and then calculates the DW for each temperature.

- Line 3 - DMDW_Type: Type of DW calculation (integer).

This parameter is equivalent to the one described for the DEBYE card, but more options are available. The possible values of DMDW_Type are: 0 Parallel s^2 1 Perpendicular s^2 2 Crystallographic u^2 3 Vibrational free energy calculation

The parallel s^2 is the usual mean-square relative displacement (MSRD) along a path. The perpendicular s^2 is the MSRD orthogonal to a path. The crystallographic u^2 is the mean-square displacement of a given atom with respect to its stationary position. Finally, the vibrational free energy associated with that crystallographic u^2 can also be calculated.

[Options 1-3 are not fully activated in this release. Also, the meaning of this parameter might change in a future release]

- Line 4 - Filename: Name of file containing the dynamical matrix (string)

The file must be present in the same directory as the DMDW input and be in "dym" format (see below).

- Line 5 - nPathDesc: Number of path descriptors (integer)

Define the number of path descriptors to use for the generation of paths.

- Lines 6... - PathDescN: Nth path descriptor used to generate a list of paths (integer and real, see below)

A path descriptor has the following form:

```
nAt At(1)...At(nAt) Path\_Length
```

where:

nAt: Number of atoms in the path (integer) At(i): Index of atom that must be included in the path (integer)

These indices correspond to the ones used in the "dym" file. The number 0 is a wildcard representing any atom in the structure. For instance, the atom indices "1 0 2" represent a double scattering paths starting at atom 1, ending at atom 2 and passing through every other allowed atom in the system. The paths are generated in such a way that no consecutive repeated indices are allowed.

Path.Length: Effective path length cutoff (real, in Bohr)

This parameter helps fine-tune the generated path list, removing paths that are longer than necessary.

F.3 The "dym" dynamical matrix file format

A "dym" file contains the information required by the Lanczos algorithm. This includes the atomic masses, structure and force constants. Two conversion scripts are included in the 'feff10/bin/Seq' directory to convert Gaussian 03 formatted checkpoint ("fchk") files (fchk2dym) and Quantum Espresso dynamical matrix files (dynG2dym) into our "dym" dynamical matrix format. The "fchk2dym" command has been thoroughly tested, but the "dynG2dym" has not.

We have also used dynamical matrices calculated by the ABINIT program. However, we don't currently offer automated tools for this scenario. Users may contact the authors for assistance using ABINIT and **dmdw**.

The current format of the "dym" files is as follows:

- Line 1 - dym_Type: Dynamical matrix file type (integer)
This value is for future use. Set to 1 for now.
- Line 2 - nAt: Number of atoms (integer)
Number of atoms in the system.
- Lines 2..2+nAt - Atomic numbers (integer)
Atomic numbers of atoms in the system.
- Lines 2+nAt+1..2+2*nAt - Atomic masses (real, in AMU)
Atomic masses of the atoms in the system.
- Lines 2+2*nAt+1..2+3*nAt - Atomic coordinates (real, in Bohr)
Cartesian coordinates ("x y z") of the atoms in the system.
- Lines 2+3*nAt+1..End - Dynamical matrix in atom pair block format (integer and real, see below, in atomic units):

The force constants in the system are stored for each pair of atoms in the system using the following block format:

```
i j d2E/dxidxj d2E/dxidyj d2E/dxidzj d2E/dyidxj d2E/dyidyj d2E/dyidzj d2E/dzidxj
d2E/dzidyj d2E/dzidzj
```

where:

i, j: Indices defining the atomic pair d2E/daidbj: Second derivative of the energy (i.e. force constant) with respect to the a coordinate of atom i and the b coordinate of atom j, where a,b=x,y,z.

Example "dym" files can be found in the 'feff10/test/DMDW' directory.

F.4 Examples

See [Chapter 5](#).

F.5 Troubleshooting dmdw

F.5.1 Lanczos problems

- If the code warns that there are less poles than Lanczos iterations, it usually means that the iteration order is too high. Try with a smaller number.
- When the structure associated with a dynamical matrix is not sufficiently optimized, the program is likely to report that certain paths result in poles associated with imaginary frequencies. The code currently ignores these poles by setting their weight to zero. Usually this doesn't affect the results significantly, but they should be considered very carefully anyway.
- The code checks the symmetry of the dynamical matrix. If it isn't sufficiently symmetric, the results should be examined carefully.

Appendix G

Trouble-Shooting feff Problems and Bug Reports

FEFF 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 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.

- For systems with a large number of potentials (eg. adamantane, a system of C and H containing 26 different potentials) small negative numbers occur during the SCF calculation (negative density warning) and the DOS calculation (negative "noise" in the ldos). This is due to limited precision in the FMS routines with so many different potentials, and probably also more likely for very light atoms. For reasons of efficiency, the FMS routines work in single precision. One can eliminate the negative numbers by recompiling FEFF using double precision as default for real numbers (eg. using the -r8 option for the ifort compiler). However, this will make the calculation significantly slower. Usually it is enough to look at the DOS and make sure it is physical.