Feff

John Rehr, jjr@phys.washington.edu version 6.01, 15 August, 1995

Ab initio Multiple-Scattering X-ray Absorption Fine Structure and X-ray Absorption Near Edge Structure Code
Copyright 1992,1993, FEFF Project, Department of Physics, FM-15 University of Washington, Seattle, WA 98195

Contents

1 Chapter 1: Synopsis – PLEASE READ THIS ENTIRE SECTION **3**

2 Conceptual description **4**

2.1 Scattering potential and phase shift code, POTPH 4

2.2 Path enumeration, PATHS 4

2.3 Scattering amplitude and other XAFS parameter calculation, GENFMT 5

2.4 XAFS spectrum calculation, FF2CHI 5

2.5 Calculation Strategy 5

3 Program and file structure **6**

3.1 Table 3.1 6

3.2 Scattering potential and phase shift code, POTPH 6

3.3 Path enumeration, PATHS 7

3.4 Scattering amplitude and other XAFS parameter calculation, GENFMT 7

3.5 XAFS spectrum calculation, FF2CHI 7

4 Chapter 4: Input file and FEFF commands **7**

4.1 File format 7

4.2 Complete list of feff.inp keywords 8

4.2.1 Basic data types 8

4.2.2 Useful Options 11

4.2.3 Sometimes useful options for expert users 14

4.2.4 Obscure Options 16

5 Chapter 5: Run Control Options **16**

5.1 Paths.dat 17

5.2 Files.dat . . . 17

6 Chapter 6: Output files 17

6.1 phase.bin . . . 17

6.2 paths.dat . . . 17

6.3 crit.dat . . . 18

6.4 geom.dat . . . 18

6.5 files.dat . . . 18

6.6 chi.dat . . . 18

6.7 feffNNNN.dat . . . 18

6.8 Diagnostic files from POTPH . . . 18

6.9 xsect.bin and xmu.dat files . . . 19

6.10 Variables in output files and in calculation of XAFS . . . 19

7 Appendix A. Copyright Information, Restrictions and License 20

7.1 Restrictions and License Information . . . 20

7.2 ADDENDUM: Government Copyrights (DO NOT DELETE) . . . 20

8 Appendix B. Installation Instructions 21

8.1 PC users: . . . 21

8.2 MAC users: . . . 22

9 Appendix C. References 22

10 Appendix D. Sample input files 22

10.1 SF6 Molecule . . . 22

11 SF6 Molecule 23

11.1 Cu fcc metal, 4 shells . . . 23

11.2 YBCO High-Tc superconductor . . . 25

12 Appendix E. Code Variables, Dimensions and Descriptions 26

12.1 dim.h: . . . 26

12.2 arrays.h: . . . 26

12.3 lambda.h: . . . 27

12.4 pdata.h: 28

13 Appendix F. Changes from FEFF 5.04 **29**

14 Appendix G. Trouble-shooting FEFF problems **30**

14.1 FEFF Bug and Error Report 30

1 Chapter 1: Synopsis – PLEASE READ THIS ENTIRE SECTION

FEFF6 calculates wave phase shifts, effective scattering amplitudes and single and multiple scattering curved XAFS and XANES spectra, including polarization dependence, for clusters of atoms ($Z < 95$).

The calculation is automated and done sequentially in four steps:

1. The scattering potentials, phase shifts, dipole matrix elements and absolute energies are calculated (module **POTPH**, potentials and phases)
2. The scattering paths for the cluster are enumerated. (module **PATHS**)
3. The effective scattering amplitudes and other XAFS parameters are calculated for each scattering path. (module **GENFMT**, general-path F-matrix calculation)
4. The XAFS parameters from one or more paths are combined to calculate a total XAFS spectrum. (module **FF2CHI**, scattering amplitude to chi, feff is the effective curved wave scattering amplitude)

For details on use of the code, examples and suggestions on calculation strategies, see sections 2, 3 and 4. For details about the algorithms used see the discussion on the appropriate module in section 2 and, for more detail, the published references in Appendix C.

FEFF is written in ANSI FORTRAN 77, with the non-standard extension of double precision complex (complex*16) variables. It requires about 3 megabytes of available memory to run. See Appendix B for complete installation instructions.

FEFF6 is copyright protected software and users must obtain a license from the University of Washington Office of Technology Transfer for its use. See Appendix A for complete copyright notice.

Please cite FEFF if the code or its results are used in published work. See Appendix C for a list of appropriate citations.

The main authors of FEFF6 are:

S. I. Zabinsky

University of Washington, zabinsky@phys.washington.edu - (206) 543-0435

J. J. Rehr

University of Washington, jjr@phys.washington.edu - (206) 543-8593

A. Ankudinov

University of Washington, alex@phys.washington.edu - (206) 543-0435

R. C. Albers

Los Alamos National Laboratory, rca@lanl.gov - (505) 665-0417

Please contact us concerning any problems with the code; see Appendix G for trouble-shooting hints and problem/bug reports.

Part of the original FEFF3 code was developed in collaboration with J. Mustre de Leon and is still part of FEFF5 and FEFF6. The authors thank the users of the experimental version of this code and especially M. Newville and B. Ravel for comments and suggestions.

2 Conceptual description

This section describes the various modules and outlines the use of the code.

2.1 Scattering potential and phase shift code, POTPH

The potentials of each atomic type are calculated as if the atoms were isolated in space using a relativistic Dirac-Fock-Slater Desclaux atom code with the von Barth-Hedin ground state LDA exchange-correlation potential. The scattering potentials are calculated by overlapping the free atom densities within the muffin tin approximation (Mattheiss prescription), and including the Hedin-Lundqvist/Quinn self energy for excited states. Muffin-tin radii are determined automatically from calculated Norman radii and scattering phase shifts are determined by matching at the muffin tin radius. XAFS spectra are referenced to the threshold Fermi level, which is estimated by electron gas theory at the mean interstitial density. Atomic configurations and core-hole lifetimes are built in, and mean free paths are determined from the imaginary part of the average interstitial potential. Dipole matrix elements are calculated using atomic core and normalized continuum wavefunctions. An absolute energy scale is obtained by a Desclaux-atom calculation of the total energy of the system with and without the core-hole. Polarization dependence is optionally incorporated in the dipole-operator.

The free atomic potentials 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 for 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.

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

2.2 Path enumeration, 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, based on a list of atomic coordinates. An efficient

degeneracy checker is used to identify equivalent paths (based on similar geometry, path reversal symmetry, and space inversion symmetry).

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 created. 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, and accurate curved wave filter is applied to all remaining paths.

2.3 Scattering amplitude and other XAFS parameter calculation, GENFMT

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

2.4 XAFS spectrum calculation, FF2CHI

The code ff2chi constructs the XAFS spectrum chi(k) using the XAFS parameters from one or more paths. Single and multiple scattering Debye-Waller factors are calculated using a correlated Debye model. Output from this module is the total XAFS spectrum and optionally, the contribution to the XAFS from each path individually.

2.5 Calculation Strategy

The scattering phase shifts for each unique potential are necessary for both PATHS and GENFMT. They are used for the importance filters in PATHS and are the basis of the XAFS parameters calculation in GENFMT. This part of the calculation can be slow, so it is usually best to run it only once and use the results while studying the paths and XAFS.

To study the necessary paths, the pathfinder module PATHS will need the atomic positions of any atoms from which scattering is expected. If the structure is completely unknown, single-scattering paths can be created explicitly. Because the number of possible paths increases exponentially with total path length, start with a short total path length, study 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 FF2CHI to assemble the results into a chi curve. Here, the slow part is GENFMT and FF2CHI 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 FF2CHI 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)

will have to be recalculated. Any time the phase shifts change, the XAFS parameters will have to be recalculated. If the path filters have been used, the path list will also have to be recomputed.

3 Program and file structure

This section describes the files and data used by the various modules. The main program FEFF reads one file created directly by the user, the file feff.inp; it is line oriented with a KEYWORD in CAPITAL LETTERS followed by data in free format. Tab characters are NOT permitted (due to portability constraints). Comments may be added at the end of any line after any required data, and spaces between lines are ignored. Any line beginning with an asterisk (*) is a comment and is ignored. An auxilliary program is provided which generates the feff.inp file from crystallographic input parameters.

Other files required by the various modules are created by FEFF from feff.inp. Some of these other files may be edited by the user as a way to modify the input data to the modules, see section 5. See the PRINT card in section 4 to obtain various diagnostic files. Table 3.1 summarizes this structure, the rest of this section describes the structure in more detail.

3.1 Table 3.1

module	input files	output files	other useful output
POTPH	feff.inp	phase.bin	diagnostic files
	potph.inp	xsect.bin	(see PRINT card)
PATHS	feff.inp	paths.dat	crit.dat
	phase.bin		
	geom.dat		
GENFMT	feff.inp	files.dat	
	phase.bin	feffNNNN.dat	
	paths.dat		
FF2CHI	feff.inp	chi.dat	chipNNNN.dat
	files.dat	xmu.dat	
	feffNNNN.dat		
	xsect.bin		

3.2 Scattering potential and phase shift code, POTPH

Reads potph.inp, which is created by FEFF's input routine from feff.inp. potph.inp is deleted by FEFF after use. POTPH writes the binary file phase.bin, which contains the scattering phase shifts and other information needed by PATHS and GENFMT. If the XANES card is used, POTPH will print the cross-section data in xsect.bin. Optionally, POTPH will write other diagnostic files with information about the potentials and phase shift calculations.

3.3 Path enumeration, PATHS

Reads geom.dat, which is created by FEFF from the information in feff.inp, and phase.bin. geom.dat is deleted after use unless explicitly saved (see PRINT card). PATHS writes paths.dat for use by GENFMT and as a complete description of each path for use of the user. PATHS 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 save geom.dat, not writing paths.dat.

3.4 Scattering amplitude and other XAFS parameter calculation, GENFMT

GENFMT reads phase.bin and paths.dat, and writes a file feffNNNN.dat for each path, where NNNN is the 4-digit path number taken from paths.dat. The feff.dat files contain the XAFS parameters for each path and can be used by FF2CHI, or can be used directly by other programs. These files are the main output of FEFF. The file files.dat is a list of these filenames with the importance of each path.

3.5 XAFS spectrum calculation, FF2CHI

FF2CHI reads files.dat, reads the files listed and files.dat and writes chi.dat with the total XAFS from these files. Additional instructions are passed to FF2CHI from feff.inp, so you can change S02, Debye temperature and some other parameters without re-doing the whole calculation. The file files.dat can be edited by hand to change the files being considered, and individual chipNNNN.dat files with chi from each path are optionally written. If the XANES card is specified and xsect.bin exists, FF2CHI will write the XANES data in xmu.dat.

4 Chapter 4: Input file and FEFF commands

The main program FEFF reads one file created directly by the user, the file feff.inp. This section describes feff.inp and the commands that tell FEFF what to do. It may be helpful to look at one or more of the sample input files in Appendix D while reading this section. An auxilliary program ATOMS can generate the feff.inp file. See the ATOMS card and the document for ATOMS for additional details. The input file for FEFF6 is identical to FEFF5, except that additional options have been added to permit XANES calculations and add polarization dependence.

4.1 File format

feff.inp is a free format line oriented file. Blank lines and lines beginning with an asterisk (comment lines) are ignored wherever the occur. End of line comments are also ignored. Each type of input read by the program must contain a card with a "keyword" in CAPITAL LETTERS, and in some cases, followed by data cards. The sequence of keyword cards is arbitrary. If any card is omitted, default values are used; an exception is the POTENTIALS card, which is required. Numeric values are listed in free format, separated by blanks. Tab characters are not allowed (due to portability constraints) and may cause confusing error messages. Characters appearing after the expected data on a given line are ignored by feff and can be used as end-of-line comments. All distances are in Angstroms and energies in eV.

4.2 Complete list of feff.inp keywords

The list of feff.inp options fall into four categories, standard options frequently and easily used, useful options that are often used, special options that are seldom necessary or helpful, and obscure options used during development that are included for completeness.

The standard data types are

ATOMS, CONTROL, DEBYE, HOLE, POTENTIALS, PRINT, SIG2, TITLE

Useful options include

CORRECTIONS, CRITERIA, ELLIPTICITY, END, NEMAX, NLEG, NOGEOM, PCRITERIA, POLARIZATION, RMAX, RMULTIPLIER, SS, XANES,

Special data types are recommended only for advanced users, awkward situations or diagnostic purposes

AFOLP, FOLP, EXCHANGE, ION, OVERLAP,

And some (e.g. IORDER) are so obscure that we

considered not mentioning them.

These data types are listed below, alphabetically within each category. Each DATATYPE is followed by a brief explanation and an example.

4.2.1 Basic data types

ATOMS Cartesian coordinates and unique potential indices of each atom (in Angstroms) in the cluster are entered following the ATOMS card, one per line. Please see the discussion of unique potentials in section 4.2.1.5. An auxilliary code ATOMS written by B. Ravel (U. Washington) is supplied with FEFF 6 to generate the ATOMS list from given crystallographic data. See the document file to ATOMS for more information.

```
ATOMS
* x       y       z      ipot     SF6 molecule
  0.0     0.0     0.0      0      S K-shell hole

  3.61    0.00    0.00     1      F 1st shell atoms
  0.00    3.61    0.00     1
  0.00    0.00    3.61     1
 -3.61    0.00    0.00     1
  0.00   -3.61    0.00     1
  0.00    0.00   -3.61     1
```

CONTROL The CONTROL card lets you run one or more of the modules separately. There is a switch for each module, 0 meaning do not run that module, 1 meaning do run it. The default is CONTROL 1 1 1 1, run all modules.

The syntax is: CONTROL potph paths genfmt ff2chi

```
* run genfmt and ff2chi, do not run potph or paths
CONTROL  0 0 1 1    potph  paths  genfmt  ff2chi
```

DEBYE The Debye card is used to calculate Debye-Waller factors for each path using the correlated Debye Model. The model is best suited for homogeneous systems, where it is quite accurate. CAUTION: in heterogeneous systems the model only gives approximate values which can easily be off by factors of two or more. Temperatures are in K. Default: Use Debye-Waller factors from files.dat. If this card is present, explicit Debye-Waller factors in files.dat are ignored and the correlated Debye model is used.

The syntax is: DEBYE temperature Debye-temperature

```
DEBYE  190 315   Calculate Debye-Waller factors for Cu at 190K
```

HOLE The HOLE card includes the hole-code ihole and the amplitude reduction factor S0**2. An entry of 0.0 for S0**2 gives the default value s02=1. To match experiment, values of s02 are typically between 0.8 and 1.0. Defaults if HOLE card is omitted: ihole=1 (K shell), S02=1. Hole-codes presently available are as follows (if your favorite is missing, complain to the authors): 0 = no hole, 1 = K-shell, 2 = L1-shell, 3 = L2-shell, 4 = L3-shell

The syntax is: HOLE ihole S0**2

```
HOLE  1   1.0     K-shell core hole, S02 = 1
```

POTENTIALS The data following the POTENTIALS card 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 keyword POTENTIALS begins a unique potential list. Each following line (until the next keyword card) is a unique potential index, the atomic number and optional tag (max 6 characters) for that unique potential. The absorbing atom must be unique potential 0. These unique potential indices are simply labels, so the order is not important, except that the absorbing atom is ipot 0, and you may not have missing indices (ie, if you use ipot 3, you must also have defined unique potentials 1 and 2).

To save time the code calculates the overlapped atom potential for each unique potential only once, using as a sample geometry the first atom in the atom list with a given unique potential index. Thus it is essential that the neighborhood of that sample atom be representative. Failure to do so may cause the code to perform poorly (See Appendix G).

Because the phase shift calculation is the most time-consuming part of FEFF, it may be useful to assume that the potential for a given shell of atoms is the same as that of a previously calculated shell. For example, in Cu it is a good approximation to determine potentials only for the central atom and the first shell and to use the first shell potential (ipot=1) for all higher shells. Such approximations should be checked in each case.

```
* molecular SF6
```

```
POTENTIALS
*   potential-index    Z   tag
          0           16   S         Sulfur K hole, absorbing atom
          1            9   F
```

PRINT The PRINT card determines how much output is printed from each of the four modules. Default is print level 0 for each module. See section 6 for details of contents of these files.

The syntax is: PRINT potph paths genfmt ff2chi

`potph`

0 = phase.bin xsect.bin only; 1 = add misc.dat; 2 = add pot.dat, phase.dat; 5 = add atom.dat

`paths`

0 = paths.dat only; 1 = add crit.dat; 2 = add geom.dat; 3 = add f(beta) files (plane wave |f(beta)| approximations); 5 = Write only crit.dat and save geom.dat. This is very useful when exploring importance of paths for large runs. Does not write paths.dat.

`genfmt`

0 = files.dat, all feffNNNN.dat files with importance greater than or equal to two thirds of the curved wave importance criterion; 1 = keep all feffNNNN.dat files

`ff2chi`

0 = chi.dat; 1 = add sig2.dat with Debye-Waller factors; 2 = add chiNNNN.dat (chi from each path individually) This can quickly fill up your disk if you're doing a large run.

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

SIG2 Global Debye-Waller factor (sig2) written to files.dat The module ff2chi, which does the final calculation of chi, reads the Debye-Waller factors from files.dat unless over-ridden by the correlated Debye model (see the DEBYE card above.). This global Debye-Waller factor is written to files.dat. You can edit files.dat before running module ff2chi to change one or more of the Debye-Waller factors.

The syntax is: SIG2 sig2

TITLE Keyword TITLE followed by a title line. You may have up to 10 of these. Titles may have up to 75 characters, leading blanks in the titles will be removed.

The syntax is: TITLE title line...

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

4.2.2 Useful Options

CORRECTIONS The real energy shift will shift E0 in the final chi and the imaginary energy shift adds broadening to the result. The energy shift is useful to correct for the error in FEFF's fermi level, which is typically too high by about 3 eV, and the the broadening 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 FF2CHI, which combines the results in all of the feff.dat files. It is useful because you can simply make these energy corrections and see the results without redoing the entire XAFS parameter calculation. Both energies are in eV. *SeealsotheEXCHANGEcard.*

The syntax is: CORRECTIONS real-energy-shift imaginary-energy-shift

```
* Reduce E0 by 3.0 eV and add 1 eV of broadening
* This will only affect module 4, ff2chi
CORRECTIONS   3.0   1.0       real shift, imag shift
```

CRITERIA 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 FEFF6 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)|*dk.

critcw is the result of the full curved wave calculation. A calculation of critcw requires a complete spherical wave calculation, which takes about 10 seconds on a SUN4. The default value of critcw is 4%, meaning that any path with mean amplitude exceeding 4% of largest path will be used in calculation of chi. The criterion critcw is used by GENFMT; since the XAFS parameter calculation is already done, the savings is not in computer time, but in disk space and ease of analysis. The values of critcw for each path are written in the file files.dat written by module GENFMT.

critpw is a plane-wave approximation to chi. 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 2.5% of 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 PATHS) tells you critpw for each path that passes the criterion.

The method of calculation of these importance factors has been improved for the current release, so don't worry if the values for some paths has changed slightly from previous versions. $Defaultvaluescritcw = 4.\%critpw = 2.5\%$

The syntax is: CRITERIA critcw critpw

```
CRITERIA  6.0  3.0   critcw 6%, critpw 3%

CRITERIA  0  0       use all paths (cw and pw criteria turned off)
```

ELLIPTICITY Ellipticity is the ratio of amplitudes of electric field in two orthogonal directions for elliptically polarized light. Only the absolute value of the ratio is important for nonmagnetic materials. The present code cannot distinguish left and right circular polarization. A zero value of the ellipticity corresponds to linear polarization, and unity to circular polarization. Default value is zero.

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

The syntax is: ELLIPTICITY ellipticity x y z

```
ELLIPTICITY  1.0  0.0 0.0 -2.0   for circular polarization about z-axis
```

END The END card marks the end of reading the feff.inp file; all data following the END card is ignored. This is optional, but useful when making short runs with only part of an input file.

```
END    ignore any cards in feff.inp that follow this
```

NEMAX This limits the number of energy points. NEMAX is normally 49, which will get you to k=20 invA. Smaller values will avoid the high energy parts of the calculation and may save you some time. This parameter is used when the scattering phase shifts are calculated, so once you have limited the number of energy points, that's what you will get for the rest of the calculations you do (based on the phase shifts). To change the number of energy points, re-run the phase shift calculation.

The syntax is: NEMAX nemax

```
* use only first 40 energy grid points
NEMAX  40
```

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

The syntax is: NLEG nleg

```
NLEG 2      find only single scattering paths (ie, 2 legged paths)
```

NOGEOM When this card is present, the file geom.dat will not be produced. Use this option when you want to take advantage of the symmetry in a cluster to speed the path calculation by using the geom.dat file produced by the auxilliary code ATOM rather than letting feff produce geom.dat from the atom list in feff.inp.

PCRITERIA 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 (pcrit's) 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 above. 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 looks at the amplitude of chi (in the plane wave approx) 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 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 we can eliminate a path 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 rmax doesn't help), study the results in crit.dat from runs with shorter rmax and experiment with the heap criterion accordingly. In the future, we hope to improve this filter.

Before using these criterion, study the output in the file crit.dat (use print option 1 for paths, see PRINT card), which has the values of critpw, keep factor and heap factor 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.

The syntax is: PCRITERIA keep-criterion heap-criterion

```
* 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
```

POLARIZATION This card specifies the direction of electric field in the incident beam. (Main axis of the ellipse in the case of elliptical polarization). x, y, z - coordinates of any nonzero vector. Only this card is necessary in the case of linear polarization. If the POLARIZATION card is omitted, polarization 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 yield faster results.

The syntax is: POLARIZATION x y z

```
POLARIZATION  1.0  2.5  0.0
```

RMAX The RMAX card determines the maximum effective distance, rmax, of a given path. Note that rmax 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 RMAX = twice the near neighbor distance.

The syntax is: RMAX rmax

```
RMAX  5.10   only include paths with effective length up to 5.10 Ang
```

RMULTIPLIER The use of the RMULTIPLIER card multiples all atomic coordinates by a constant factor, rmult. This is useful, for example, if unit cell coordinates are used. (Default value rmult=1.)

The syntax is: RMULTIPLIER rmult

```
RMULTIPLIER 1.01   (increase distances by 1%)
```

SS The SS card is used when the structure of a given shell is unknown. No multiple scattering is produced but a single scattering path of degeneracy deg is produced. Overlap cards may be used to construct the potential for such a path. The parameters are a shell index, which is a label used for feffNNNN.dat file name, a unique potential index ipot, identifying the unique potential of atom from which to scatter, the degeneracy, i.e., the multiplicity of this single scattering path, and the distance to central atom, rss. This could be used to add single scattering beyond the limits of a cluster.

The syntax is: SS index ipot deg rss

```
*  index  ipot   deg  rss
SS   29     1     48  5.98       parameters for 19th shell of Cu
```

XANES The XANES card is used when a calculation of the near edge structure including the atomic background and absolute energies are desired.

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 vacuum but above the muffin-tin zero show up as resonances. The absolute energies are based on atomic total energy calculations using the Desclaux code; the accuracy of this approximation varies from a few eV to a few hundred eV for very large Z. No parameters are needed. Default: XANES not calculated.

4.2.3 Sometimes useful options for expert users

AFOLP This automatically overlaps all muffin-tins to reduce the effects of potential discontinuities at the muffin-tins. It is useful in highly inhomogeneous materials. It works fairly well, but may fail in some cases. See FOLP for non-automatic version.

```
AFOLP     use automatic maximum overlapping
```

FOLP The FOLP card sets a parameter which determines by what factor muffin-tin radii are overlapped. We recommend that the AFOLP card be used In cases with severe anisotropy, and FOLP only used for diagnostic purposes.

The syntax is: FOLP ipot folp

```
FOLP 1  1.1  (10% overlap of muffin tin of unique potential 1)
```

EXCHANGE The EXCHANGE card specifies the energy dependent exchange correlation potential to be used. (See also the CORRECTIONS card which is similar but allows the user to refine values of vr0 and vi0 after a calculation is completed.) The EXCHANGE card contains the index of the xc pot (ixc) and the constant imaginary part of the self-energy (vi0). The Hedin-Lundqvist self-energy (default) appears to be the best choice for all applications we have tested in detail. Optionally one may use the Dirac-Hara exchange correlation potential and an appropriate imaginary potential vi0. This may be useful to correct FEFF's typical error of 2 eV (high) in the location of the Fermi level and to add instrumental broadening. Defaults if EXCHANGE card is ommitted: ixc=0 (Hedin-Lundquist), vr0=0.0 vi0=0.0.

The syntax is: EXCHANGE ixc vr0 vi0

ixc values

ixc=0: Hedin-Lundqvist + const imag part; ixc=1: Dirac-Hara + const imag part; ixc=2: ground state + const imag part; ixc=3: Dirac-Hara + HL imag part + const imag part

vi values

vi0 is (the negative of the) imaginary const shift

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

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

ION The ION card ionizes all atoms with atom type ipot. Negative values and non-integers are not permitted (however, we may allow these in future versions of FEFF). The ION card is used to ionize all atoms of a particular potential index. For example, for diatomics like Br2, the fully relaxed configuration has ionization=1 on the scattering atom. The ionization card should be used with caution; because of charge transfer, the actual degree of ionization is not directly related to the chemical valency. The default (non-ionized) scattering potentials are often superior to those empirically ionized, and the results should be checked both ways. Defaults if ION cards are ommitted: atoms are not ionized.

The syntax is: ION ipot ionization

```
ION  1  1   ipot, ionization
```

OVERLAP The OVERLAP card contains information needed to construct the overlapped atom potentials when atomic coordinates are not known or specified. If the atomic positions are listed following the ATOMS cards, the OVERLAP cards are probably not needed. The OVERLAP card contains the potential index of the atom being overlapped and must be followed by cards specifying the potential index, number of atoms of a given type to be overlapped and and their distance to the atom being overlapped. This option can be useful in crystals, especially for distant shells – see sample input files for an example. It can also be useful for calculating single scattering XAFS in very 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 important to specify all the close 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 must be approximately correct to ensure good scattering potentials. Thus it is important to specify as accurately as possible the coordination environment of the scatterer.

The syntax is: OVERLAP iph

```
OVERLAP 4         determine overlap for 3rd shell of Cu
  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
```

4.2.4 Obscure Options

IORDER Order of the approximation to use in module GENFMT. We use order 2, which is correct to terms of order (1/pR)**2, and corresponds to 6x6 matrices. However, we do single scattering exactly. This 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). 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 subroutine SETLAM. Iord is passed to setlam for processing. You may need to change the code parameter lamtot if you want to do higher order calculations. This is another of the arcane cards that we have used for testing. For details of the algorithm used by GENFMT, see the paper by J.J.Rehr and R.C.Albers (see Appendix C, references).

The syntax is: IORDER iord

5 Chapter 5: Run Control Options

In addition to the CONTROL card and the FEFF 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 change explicit Debye-Waller factors in the final result by editing files.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 – these files are read by a computer program that is fussy about what it receives.

The files that can be used in this way are paths.dat and files.dat.

5.1 Paths.dat

You can modify a path, or even invent new ones. 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; i.e., for the choice, index = 845, the exafs parameters will appear in feff845.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 wrong results).

5.2 Files.dat

This is the list of files that ff2chi uses to calculate chi. It includes the filenames of the feffNNNN.dat files 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. FF2CHI will use the Debye-Waller factors in this file unless you have asked to use the correlated Debye model. You may also delete file names from this file if you want to combine some particular set of paths. (Save the original, or you'll have to re-run GENFMT!)

You may re-name the feffNNNN.dat files as long as you remember to change the filenames in files.dat when you want to calculate the XAFS using FF2CHI.

6 Chapter 6: Output files

6.1 phase.bin

This is a binary file with the scattering phase shifts for each unique potential. It is used by GENFMT and the pathfinder.

6.2 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 by hand, see section 4.

6.3 crit.dat

Values of the quantities tested against the various criteria in the pathfinder.

6.4 geom.dat

Written by feff for use with the pathfinder, and deleted after use. It is an atom list created from the user-input list in feff.inp. For more information, see the NOGEOM card in section 4.

6.5 files.dat

List of files to use to create chi.dat. Written by GENFMT when the xafs parameters are calculated and used by FF2CHI. This contains the curved wave importance ratios, which you may wish to study. For details on editing this by hand, see section 5.

6.6 chi.dat

Standard xafs data k, chi(k), |chi(k)| relative to threshold (k=0): The header also contains enough information to specify what model was used to create this file.

6.7 feffNNNN.dat

Effective scattering amplitude and phase shift data, with k referenced to threshold for shell nn: k, phi_c, |feff|, ph(feff), redfac, lambda, Re(p)

6.8 Diagnostic files from POTPH

misc.dat

Header file for quick reference.

phaseNN.dat

Complex phase shifts for each shell.

potNN.dat

Detailed atomic potentials and densities.

atomNN.dat

Diagnostic information on Desclaux free atom NN.

6.9 xsect.bin and xmu.dat files

Dipole matrix elements and absolute energies for XANES calculations. Note: the file xsect.bin does not contain the fermi-function cutoff at the edge, while xmu.dat does. The file xmu.dat contains both XANES and XAFS data mu, mu_0, and chi as functions of absolute energy E, relative energy E-e_f and wave number k.

6.10 Variables in output files and in calculation of XAFS

k

wave number (ang**-1); k=sqrt(e-e_f) where e is energy and e_fermi is the fermi level computed from electron gas theory at the avg interstitial charge density

chi(k)

s0^2 * redfac sum_shells $natsh(|feff|/kR**2)**exp(-2R/lambda)sin(2kR + ph(feff) + phi_c))exp(-2k**2*sigma**2)$

phi_c

2*delta_l_c -l*pi (total central atom phase)

feff

effective curved-wave backscattering amplitude feff(k) in xafs formula for each shell

ph(feff)

phase of feff(k) for each shell

redfac

exp(-2 imag (delta_c)), total central atom loss factor

R

distance to central atom for each shell

natsh

mean number atoms in each shell

sigma2**

mean square variation of R for each shell

lambda

mean free path (angstroms) =-1./imag (p)

kf

fermi momentum at avg interstitial charge density

p=p(r)

local momentum (p(r)**2=k**2+kf(r)**2+sigma-sigma_f)

sigma

self energy at energy e, sigma_f at energy e_f

mu

total absorption cross-section

mu_0

atomic background absorption

7 Appendix A. Copyright Information, Restrictions and License

7.1 Restrictions and License Information

Feff is copyrighted software and a license from the University of Washington Office of Technology Transfer must be obtained for its use. This is necessary to protect the interests of users and the University of Washington. Both academic/non-profit and commercial licenses are available - see the addendum to this document for details. New users should request the latest version of this code from the authors. The license may be obtained by writing to FEFF Project, c/o Todd Case, Dept. of Physics, FM-15, Univ. of Washington, Seattle, WA 98195. Tel: (206) 685-4091, E-mail: case@phys.washington.edu

7.2 ADDENDUM: Government Copyrights (DO NOT DELETE)

This work was supported in part by DOE Grant DE-FG06-90ER45416. 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.

8 Appendix B. Installation Instructions

The program FEFF6 is provided as a single source file, which contains a main program and all necessary subroutines. Simply compile and link on your system using a Fortran 77 compiler and your usual linker. The executable file produced is all you will need. The code requires about 3 megabytes to run on a VAX. Since most of this memory is data space, we expect it to require about this amount of memory on most machines. PC users, please see section below about DOS machines.

We have endeavored to make FEFF portable to most machines without any modification. If your machine does not reproduce the test output files chi.dat and feffNNNN.dat to high accuracy, some changes may be needed. Please report any problems or warning messages to the authors, as this will help us achieve full portability. If changes are needed, keep in mind the following points:

Floating point calculations must be done to precision 64 bits. Thus the code uses double precision variables throughout, i.e., real*8 for real numbers and complex*16 for complex numbers. If your machine uses 8 bytes for single precision floating point numbers (for example, Crays and some CDC machines), use the compiler option to ignore double precision statements in the code. This option is available (in various forms) on all Cray and CDC machines that we have seen. On the CRAY YMP at Ohio State, the compiler option to ignore double precision in the source code is

```
cf77 -c -Wf"-dp"
```

Code is written in ANSI FORTRAN 77, except that COMPLEX*16 variables are required. Since data type COMPLEX*16 is not part of the ANSI standard, minor compiler-dependent modification may be necessary. We have used the VAX extensions to ANSI FORTRAN 77 since they seem to be the most portable. The non-standard statements and intrinsic functions used are: COMPLEX*16 variables and arrays; DIMAG(arg) returns a double precision imaginary part of arg; DCMPLX(arg) returns a complex*16 version of arg; DBLE(arg) returns a double precision real part of arg; SQRT, EXP, ABS and other generic math functions are assumed to accept complex*16 arguments and return double precision or complex*16 results.

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

8.1 PC users:

Because of the awkwardness of DOS, many users do not have FORTRAN compilers and many of those compilers are difficult to use with large codes. An executable version of FEFF for PCs is available. It has been compiled using Watcom's FORTRAN77 compiler which works with a DOS extender (DOS4GW) to use extended memory. You will need a PC with either a 386 and 387 math coprocessor or a 486 with the math coprocessor embedded in it (SX does not have this, DX does). You will also need about 3 Megabytes of RAM available, IN ADDITION TO whatever memory is used by DOS and memory-resident utilities. Please contact us if you need the PC executable. Special installation instructions for the PC version are in the READ.ME file on the PC distribution disk.

If you prefer your own compiler, or are using an operating system other than DOS, simply compile the source code using your FORTRAN compiler and linker as you would for any other machine.

8.2 MAC users:

Diskettes with an executable version of FEFF for MAC's are not yet available, but could be given sufficient demand. Please contact us if you need a MAC version please contact us.

9 Appendix C. References

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

1) Multiple scattering XAFS

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

2) X-ray Absorption Near Edge Structure (XANES)

J.J. Rehr, **Jpn. J. Appl. Phys. 32**, 8 (1993).

3) FEFF standards

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).

4) Technical reference #1

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. B44**, 4146 (1991).

5) Technical reference #2

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. B41**, 8139 (1990).

10 Appendix D. Sample input files

10.1 SF6 Molecule

```
TITLE Molecular SF6

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

ATOMS
```

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

11 SF6 Molecule

```
TITLE Molecular SF6

POTENTIALS
*    ipot    z   tag
       0    16   S        absorbing atom must be unique pot 0
       1     9   F
RMULTIPLIER 1.56
ATOMS
*  x      y      z     ipot
   0      0      0       0          S absorber
   1      0      0       1          6 F backscatters
   0      1      0       1
   0      0      1       1
  -1      0      0       1
   0     -1      0       1
   0      0     -1       1
```

11.1 Cu fcc metal, 4 shells

```
TITLE Cu crystal, 4 shells

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

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

POTENTIALS
  0  29  Cu0
  1  29  Cu

ATOMS
     .0000     .0000     .0000  0    shell    0     .0000
   -1.8050   -1.8050     .0000  1    shell    1    2.5527
   -1.8050    1.8050     .0000  1
    1.8050   -1.8050     .0000  1
    1.8050    1.8050     .0000  1
```

```
-1.8050     .0000   -1.8050  1
-1.8050     .0000    1.8050  1
 1.8050     .0000   -1.8050  1
 1.8050     .0000    1.8050  1
  .0000   -1.8050   -1.8050  1
  .0000   -1.8050    1.8050  1
  .0000    1.8050   -1.8050  1
  .0000    1.8050    1.8050  1
-3.6100     .0000     .0000  1    shell    2    3.6100
 3.6100     .0000     .0000  1
  .0000   -3.6100     .0000  1
  .0000    3.6100     .0000  1
  .0000     .0000   -3.6100  1
  .0000     .0000    3.6100  1
-3.6100   -1.8050   -1.8050  1    shell    3    4.4213
-3.6100   -1.8050    1.8050  1
-3.6100    1.8050   -1.8050  1
-3.6100    1.8050    1.8050  1
 3.6100   -1.8050   -1.8050  1
 3.6100   -1.8050    1.8050  1
 3.6100    1.8050   -1.8050  1
 3.6100    1.8050    1.8050  1
-1.8050   -3.6100   -1.8050  1
-1.8050   -3.6100    1.8050  1
-1.8050    3.6100   -1.8050  1
-1.8050    3.6100    1.8050  1
 1.8050   -3.6100   -1.8050  1
 1.8050   -3.6100    1.8050  1
 1.8050    3.6100   -1.8050  1
 1.8050    3.6100    1.8050  1
-1.8050   -1.8050   -3.6100  1
-1.8050   -1.8050    3.6100  1
-1.8050    1.8050   -3.6100  1
-1.8050    1.8050    3.6100  1
 1.8050   -1.8050   -3.6100  1
 1.8050   -1.8050    3.6100  1
 1.8050    1.8050   -3.6100  1
 1.8050    1.8050    3.6100  1
-3.6100   -3.6100     .0000  1    shell    4    5.1053
-3.6100    3.6100     .0000  1
 3.6100   -3.6100     .0000  1
 3.6100    3.6100     .0000  1
-3.6100     .0000   -3.6100  1
-3.6100     .0000    3.6100  1
 3.6100     .0000   -3.6100  1
 3.6100     .0000    3.6100  1
  .0000   -3.6100   -3.6100  1
  .0000   -3.6100    3.6100  1
```

```
     .0000    3.6100   -3.6100  1
     .0000    3.6100    3.6100  1
```

11.2 YBCO High-Tc superconductor

```
TITLE  YBCO: Y BA2 CU3 O7      Cu2, core hole

RMAX   4.5

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

ATOMS
 0.0000    0.0000    0.0000    0        CU2  0.0000

 1.9115    0.0000    0.2453    1        O    1.9272
-1.9115    0.0000    0.2453    1        O    1.9272
 0.0000    1.9430    0.2687    1        O    1.9615
 0.0000   -1.9430    0.2687    1        O    1.9615
 0.0000    0.0000   -2.3128    1        O    2.3128

 1.9115   -1.9430    1.6821    2        Y    3.2029
-1.9115    1.9430    1.6821    2        Y    3.2029
-1.9115   -1.9430    1.6821    2        Y    3.2029
 1.9115    1.9430    1.6821    2        Y    3.2029
 0.0000    0.0000    3.3641    3        CU1  3.3641
 1.9115   -1.9430   -2.0091    4        BA   3.3861
-1.9115    1.9430   -2.0091    4        BA   3.3861
-1.9115   -1.9430   -2.0091    4        BA   3.3861
 1.9115    1.9430   -2.0091    4        BA   3.3861
 0.0000    1.9430    3.0955    1        O    3.6547
 0.0000   -1.9430    3.0955    1        O    3.6547
-1.9115    0.0000    3.1188    1        O    3.6580
 1.9115    0.0000    3.1188    1        O    3.6580
 3.8230    0.0000    0.0000    3        CU1  3.8230
-3.8230    0.0000    0.0000    3        CU1  3.8230
 0.0000    3.8860    0.0000    3        CU1  3.8860
 0.0000   -3.8860    0.0000    3        CU1  3.8860
 0.0000    0.0000   -4.1584    3        CU1  4.1584
-3.8230   -1.9430    0.2687    1        O    4.2968
 3.8230    1.9430    0.2687    1        O    4.2968
 3.8230   -1.9430    0.2687    1        O    4.2968
-3.8230    1.9430    0.2687    1        O    4.2968
```

```
 1.9115    3.8860    0.2453    1        0    4.3376
 1.9115   -3.8860    0.2453    1        0    4.3376
-1.9115    3.8860    0.2453    1        0    4.3376
-1.9115   -3.8860    0.2453    1        0    4.3376
```

12 Appendix E. Code Variables, Dimensions and Descriptions

The end of line comments were removed from the code because they are not standard Fortran. The array names are a bit cryptic due to the six character limit in standard FORTRAN – the comments included here explain what the names mean. If you need to run larger problems than the dimension statements in the code allow, simply change the dimensions in all the relevant parameter statements and recompile. However, dimensions associated with the atom and phase shift codes, e.g. the number of r-mesh points and the max number of angular momentum levels should NOT be changed.

12.1 dim.h:

```
c     These names have been checked with all current feff code.
c     Changing the common names in the future may be a good idea.
c     June 12, 1992

      parameter (nphx = 7)      !max number of unique potentials (potph)
      parameter (npotx = nphx)  !max number of unique pots (genfmt, paths)
      parameter (nfrx = nphx)   !max number of free atom types
      parameter (novrx = 8)     !max number of overlap shells
      parameter (natx = 250)    !max number of atoms in problem
      parameter (ltot = 24)     !max number of ang mom (arrays 1:ltot+1)
      parameter (nrptx = 250)   !Loucks r grid used through overlap
      parameter (nex = 100)     !Number of energy points genfmt, etc.
      parameter (lamtot=15)     !Max number of distinct lambda's for genfmt
                                !15 handles iord 2 and exact ss
      parameter (mtot=2,ntot=2) !vary mmax and nmax independently
      parameter (legtot=7)      !max number of legs, used in genfmt,
                                !NOT in path finder
      parameter (npatx = 8)     !max number of path atoms, used in path
                                !finder, NOT in genfmt
```

12.2 arrays.h:

```
c     Notes:
c        nat    number of atoms in problem
c        nph    number of unique potentials
c        nfr    number of unique free atoms
c        ihole  hole code of absorbing atom
c        iph=0 for central atom
c        ifr=0 for central atom
```

```
c     Specific atom input data
      dimension iphat(natx)     !given specific atom, which unique pot?
      dimension rat(3,natx)     !cartesian coords of specific atom

c     Unique potential input data
      dimension iatph(0:nphx)   !given unique pot, which atom is model?
                                !(0 if none specified for this unique pot)
      dimension ifrph(0:nphx)   !given unique pot, which free atom?
      dimension xnatph(0:nphx)  !given unique pot, how many atoms are there
                                !of this type? (used for interstitial calc)
      character*6 potlbl(0:nphx)        !label for user convenience

      dimension folp(0:nphx)    !overlap factor for rmt calculation
      dimension novr(0:nphx)    !number of overlap shells for unique pot
      dimension iphovr(novrx,0:nphx)    !unique pot for this overlap shell
      dimension nnovr(novrx,0:nphx)     !number of atoms in overlap shell
      dimension rovr(novrx,0:nphx)      !r for overlap shell

c     Free atom data
      dimension ion(0:nfrx)     !ionicity, input
      dimension iz(0:nfrx)      !atomic number, input

c     ATOM output
c     Note that ATOM output is dimensioned 251, all other r grid
c     data is set to nrptx, currently 250
      dimension rho(251,0:nfrx)         !density*4*pi
      dimension vcoul(251,0:nfrx)       !coulomb potential

c     Overlap calculation results
      dimension edens(nrptx,0:nphx)     !overlapped density*4*pi
      dimension vclap(nrptx,0:nphx)     !overlapped coul pot
      dimension vtot (nrptx,0:nphx)     !overlapped total potential

c     Muffin tin calculation results
      dimension imt(0:nphx)     !r mesh index just inside rmt
      dimension inrm(0:nphx)    !r mesh index just inside rnorman
      dimension rmt(0:nphx)     !muffin tin radius
      dimension rnrm(0:nphx)    !norman radius

c     PHASE output
      complex*16 eref(nex)              !interstitial energy ref
      complex*16 ph(nex,ltot+1,0:nphx)  !phase shifts
      dimension lmax(0:nphx)            !number of ang mom levels
```

12.3 lambda.h:

```
      common /lambda/
```

```
     4   mlam(lamtot),  !mu for each lambda
     5   nlam(lamtot),  !nu for each lambda
     1   lamx,          !max lambda in problem
     2   laml0x,        !max lambda for vectors involving absorbing atom
     3   mmaxp1, nmax   !max mu in problem + 1, max nu in problem
```

12.4 pdata.h:

```
c     Note that leg nleg is the leg ending at the central atom, so that
c     ipot(nleg) is central atom potential, rat(nleg) position of
c     central atom.
c     Central atom has ipot=0
c     For later convenience, rat(,0) and ipot(0) refer to the central
c     atom, and are the same as rat(,nleg), ipot(nleg).

c     text and title arrays include carriage control
      character*80 text, title
      character*6  potlbl
      common /str/ text(40),    !text header from potph
     1             title(5),    !title from paths.dat
     1             potlbl(0:npotx)      ! potential labels for output

      complex*16 ph, eref
      common /pdata/
     1 ph(nex,ltot+1,0:npotx),  !complex phase shifts,
     1                                 !central atom ipot=0
     1 rat(3,0:legtot),         !position of each atom, code units (bohr)
     1 eref(nex),               !complex energy reference
     1 em(nex),         !energy mesh
     1 ri(legtot), beta(legtot), eta(legtot),   !r, beta, eta for each leg
     1 deg, rnrmav, xmu, edge,  !(output only)
     1 lmax(nex,0:npotx),       !max l with non-zero phase for each energy
     1 ipot(0:legtot),  !potential for each atom in path
     1 iz(0:npotx),     !atomic number (output only)
     1 ltext(40), ltitle(5),    !length of each string
     1 nsc, nleg,       !nscatters, nlegs (nleg = nsc+1)
     1 npot, ne,        !number of potentials, energy points
     1 ik0,             !index of energy grid corresponding to k=0 (edge)
     1 ipath,   !index of current path (output only)
     1 ihole,   !(output only)
     1 l0, il0, !lfinal and lfinal+1 (used for indices)
     1 lmaxp1,  !largest lmax in problem + 1
     1 ntext, ntitle    !number of text and title lines
```

13 Appendix F. Changes from FEFF 5.04

We have made some minor changes to the input files and code. For complete descriptions of the new cards and modifications of old ones, see section III. With the exception of POLARIZATION and XANES capabilities, and improved path finder and degeneracy checks, FEFF 6 is nearly identical to FEFF5.

Two cards are no longer recognized. These are:

- KLIMIT (replaced by NEMAX)
- REQUIRE

There are several changes and new options. These are:

CORRECTIONS

Adds real and imaginary part to energy

CRITERIA

The importance criteria are now formulated differently

CSIG

Code now uses complex momentum in Debye-Waller factors by default

NEMAX

Max number of energy grid points (replaces KLIMIT)

PCRITERIA

The path finder criteria are now formulated differently

PRINT

Several new options

POTENTIALS

Now has optional labels for unique potentials

XANES spectra, absolute energies and Atomic Background absorption can be calculated with the new option

- XANES

Polarization dependent feff can be calculated with the help of the following new cards:

POLARIZATION

Specifies polarization vector

ELLIPTICITY

Ellipticity and direction of the incident beam for the elliptical polarization.

14 Appendix G. Trouble-shooting FEFF problems

FEFF 6 has been extensively tested but occasionally new bugs show up. In an effort to maintain trouble-free codes we take all bug reports seriously. Often we receive reports by users of older versions of FEFF of bugs that are fixed in the latest releases. Other code failures can often be traced to input file errors, sometimes quite subtle. An example would be 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. Another example is an error in assigining 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. If, after you have verified that your input file is not the problem, the problem persists, please contact us at the e-mail addresses listed in Sec. 1. To address your problem, we need the following information:

14.1 FEFF Bug and Error Report

1. Brief description of problem.
2. Code and Version number (e.g., FEFF5 version 5.03)
3. Input file causing problem `feff.inp`
4. If available, attach sample output (e.g. `chi.dat`) or log file.