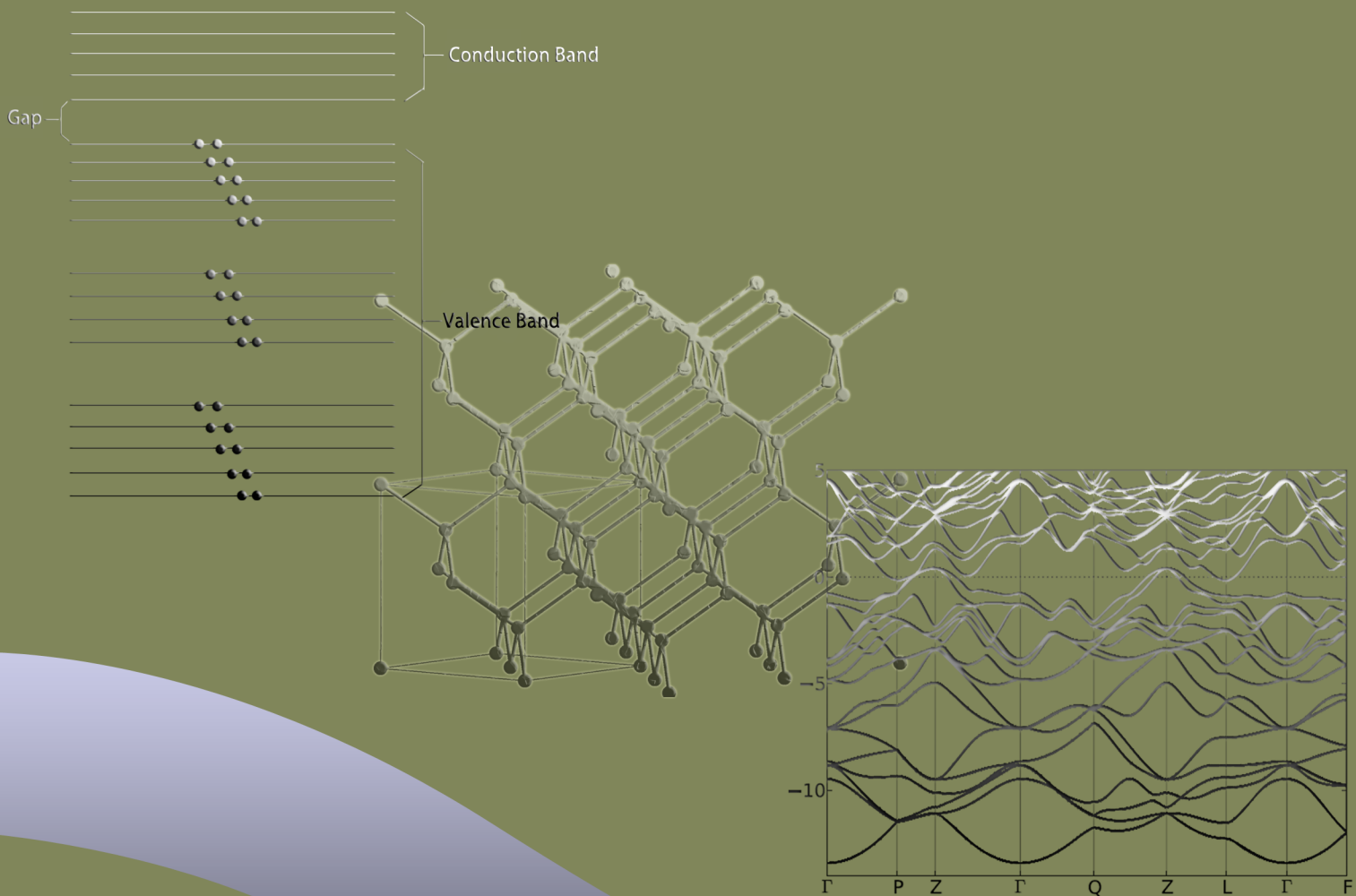


# Queen Mary, University of London

## School of Physics and Astronomy

### XAFS Calculation Related Problems



### About FDMNES Calculation

The MOST IMPORTANT thing to keep in mind before doing the calculation is that the fdmnes calculation uses FRACTIONAL coordinates but NOT Cartesian coordinates!

### About Prot2005

The Prot2005 program will list all atoms with the same group number together when inputting into 'fdmnes.inp' file. So the original atom order is then changed, hence the number specified under the keyword 'Absorber' (in 'fdmnes.inp' file) should be changed accordingly. Moreover, it is better to compare the calculation result for original structure before and after executing Prot2005 (when inputting 'in' for the first argument, Prot2005 will not change anything but the atom order according to group numbering).

### About $S_0^2$

$S_0^2$  is the factor introduced in EXAFS expression accounting for amplitude reduction. It can be obtained through the fitting of experimental spectra based on the theoretical expression for EXAFS signal. The physical meaning of  $S_0^2$  actually accounts for the core-hole effect - when core electron is excited to continuum state, the original final state should definitely be influenced by the core-hole level. The actual effect is reducing the absorption somehow. Theoretically in FEFF code, it can be estimated by calculating atomic overlap integrals with and without core hole. With the core hole, the corresponding system of course contributes to absorption; in contrast, there is no contribution to absorption from the system without core hole. Thus the overlapping of core orbital obtained with and without core hole can be estimated as the effective contribution to absorption, which is, of course, smaller than 1.

### Muffin-Tin Radius in FEFF

Initially, free atom potential is calculated using Dirac-Fock atom code. Then non-overlapping muffin-tin radius is calculated based on Norman criteria, which requires the charge inside each independent muffin-tin spherically symmetric area is equal to that of corresponding atom. The calculated radius is called Norman radius. (PS: this process is just like dividing the total charges between neighbouring atoms, i.e.  $Total\ Charge \rightarrow (Charge - A | Charge - B) \rightarrow (Atom-A | Atom-B)_{MT-A, MT-B}$ .)

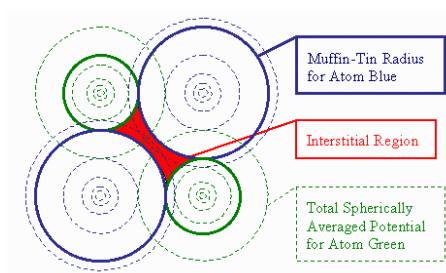


Figure 1 The illustration for Muffin-Tin radius.

Generally, the muffin-tin spheres touch, but not overlap. However, FEFF by default overlaps the muffin-tin sphere a bit, which turns out to improve the calculation accuracy in most cases. Thus for each single potential calculation (when 'Non-Overlap' is NOT specified), we should have three radius: rnm, rmt and folp (these names are used by FEFF 'log1.dat' file). Here 'rnm' refers to the original non-overlapping Norman radius, 'rmt' refers to the real muffin-tin radius considering the overlapping, and 'folp' specifies the overlapping. Fig.1 depicts clearly the



overlapping idea, where the solid blue and green line corresponds to non-overlapping Norman radius ('rnm') and the outer blue and green dotted line refers to the overlapping muffin-tin radius ('rmt') used for FEFF calculation (again, only if 'Non-Overlap' is NOT specified).

Furthermore, the above calculated scattering potential based on muffin-tin approximation can be used as initial guess to carry out self-consistent potential calculation by introducing SCF card within specified SCF cluster size.

### Many-body effect in FEFF

In muffin-tin approximation, the building of scattering potential only considers two body interaction, which means the potential at any given point only considers the influence of two nuclei. The many-body problem is just the influence from third, fourth or even more nuclei, which is difficult to deal with within the framework of muffin-tin approximation. In FEFF code, the many-body effect is solved by introducing a many-body spectral function (module `sfcov`, refer to [FEFF9.6 Manual](#) P16-item14). For further theoretical information concerning two-body approximation, refer to [Spherically Averaged Coulomb Potential Webpage](#).

### About FDMNES Compilation & Run

For simple serial calculation, FDMNES provides executable file for both Windows and Linux (not for any Mac, however). For Windows, the executable file is 'fdmnes.exe' in the home directory of FDMNES package, and 'fdmnes\_linux64' for 64 byte Linux. However for parallel calculation when calculation cluster is large for FDM mode calculation, we need to build up executable binary file by ourselves. The source code of FDMNES is provided in the directory: '\fdmneshome\prog', and there is given the example make-file for the parallel compilation. Corresponding make-file content is given as below:

```
pgf90 -c -fast -O3 -Mvect=sse main.f90 clemf0.f90 coabs.f90 convolution.f90 dirac.f90
fdm.f90 fprime.f90 general.f90

pgf90 -c -fast -O3 -Mvect=sse lecture.f90 mat.f90 metric.f90 minim.f90 not_mpi.f90
potential.f90 optic.f90 /home/ai-peng/Downloads/Light emission from Ge quan-
tum dots in silicon microcavities.pdf /home/ai-peng/Downloads/Optical Prop-
erty and Electronic Structure of Ge Quantum Dots - Comparison Between
Experimental and Computational Results.pdf /home/ai-peng/Downloads/Ge-
Si self-assembled quantum dots grown on Si-001 in an industrial high-pressure
chemical vapor deposition reactor.pdf pgf90 -c -fast -O3 -Mvect=sse scf.f90 se-
lec.f90 sphere.f90 spgroup.f90 tab_data.f90 tddft.f90 tensor.f90

pgf90 -c -fast -O3 -Mvect=sse sub_util.f

pgf90 -o fdmnes main.o clemf0.o coabs.o convolution.o dirac.o fdm.o fprime.o gen-
eral.o lecture.o mat.o metric.o minim.o not_mpi.o potential.o scf.o selec.o spgroup.o
sphere.o sub_util.o tab_data.o tddft.o tensor.o optic.o
```

However, the compilation for parallel calculation indeed depends on system. Hence before compiling, one need to make sure what parallel fortran compiler they have on the system. Actually, there are different kinds of blocks (technically called *module*) embedding the parallel fortran compiler. To check the existing MPI related module on the system, use the command 'module available' to get result like this:

```
[apw247@frontend1.apocrita]$ module avail
----- /etc/modulefiles -----
openmpi-x86_64
```



Now knowing what MPI related module we have on the system, we can then use the command 'module load openmpi-x86\_64' to load corresponding module (or whatever you get under the line '/etc/modulefiles' as shown above). Here it should be pointed the directory '/etc/modulefiles' may sometimes not available especially when one is working on remote server. To solve this problem, one may check (work for me, but not sure for other servers) the following lines when running 'module available' command:

```
[apw247@frontend1.apocrita]$ module avail
----- /usr/share/Modules/modulefiles -----
openmpi/1.6.5/gcc openmpi/1.6.5/intel
```

Loading module command then should be changed to "module load openmpi/1.6.5/intel" (or other corresponding command). After figuring out what compiler we have, we can follow similar step as shown by the above make-file example. The Makefile I use is as following:

```
MPIF90 = mpif90
FFLAGS = -O3
OBJS = main.o clemf0.o coabs.o convolution.o dirac.o fdm.o fprime.o general.o \
      lecture.o mat.o metric.o minim.o potential.o scf.o selec.o spgroup.o \
      sphere.o sub_util.o tab_data.o tddft.o tensor.o optic.o
fdmnes: $(OBJS)
      $(MPIF90) $(FFLAGS) -o $$ $^
%.o: %.f90
      $(MPIF90) -I. $(FFLAGS) -c -o $$ $<
.PHONY: clean
clean:
      rm -rf *.o fdmnes
```

However as we already know, the compilation depends on different system, to solve the platform dependent problem, there is a online available 'Makefile' which is suitable for most platforms FDMNES PMI compiling. This script can be found following the link: [FDMNES Parallel Compilation Makefile \(Universally Applicable\)](#).

At last, script for interaction with queueing system (e.g. qsub) to submit jobs is given as following, the first one is for simple serial calculation and the second one is for parallel calculation:

*For serial job:*

```
#!/bin/sh
#$ -cwd                                # Set the working directory for the job to the
                                       current directory
#$ -V
#$ -l h_rt=240:0:0                    # Request 240 hour runtime
#$ -l h_vmem=4G                        # Request 4GB RAM
./fdmnes_linux64                      #Execute 'fdmnes_linux64' binary file
                                       in current directory
```



*For array job:*

```
#!/bin/sh
#$ -cwd
#$ -V
#$ -l h_rt=240:0:0
#$ -l h_vmem=4G
cd cluster_$SGE_TASK_ID                # This line should be changed
                                         according to the problem in
                                         case.

./fdmnes_linux64
```

The command to submit the array job is like: 'qsub -t STARTING\_NUM-STOPPING\_NUM SCRIPT\_NAME.sh'.

*For parallel MPI job:*

```
#!/bin/bash
#$ -M zyroc1990@gmail.com
#$ -m be
#$ -cwd                                # Set the working directory for the job to the
                                         current directory

#$ -V
#$ -pe openmpi 1                       # Request a single openmpi slot
                                         (whole node)

#$ -l h_rt=240:0:0                     # Request 240 hour runtime
#$ -l h_vmem=28G                       # Request 28GB RAM
module load openmpi/1.6.5/intel
mpirun -np 7 ./fdmnes_mpi              # 'fdmnes_mpi' in current directory
                                         is your executable file obtained
                                         from the previous compilation step.
```

Before finishing, there are two other things to keep in mind for MPI compilation. **Firstly, one need to either delete or rename the two files in '`\fdmneshome\prog`' directory: '`mpif.h`' and '`not_mpi.f90`', since they are only for non-parallel calculation!** If not, the resulted executable binary file will not work. The other thing is that the running of compiled FDMNES binary may sometime fails to start without any error message, and this is quite possibly due to memory allocated for the calculation is not enough. For example in the above shown MPI submitting script, if we change the line '`#$ -l h_vmem=28G`' to '`#$ -l h_vmem=21G`', the calculation will be stacked sometime when we are doing calculation for large clusters.



## About FEFF Compilation

For most calculation, FEFF does not need parallel execution since it is efficient enough even for a laptop. However for large cluster especially for those calculation using self-consistent calculation or doing Full Multiple Scattering (FMS) with large size (size for FMS), one may need parallel calculation as stated in the FEFF9.6 manual. Basically, we can find everything for FEFF package (including the source code, a binary file named 'feff' suitable only for installed system, etc.) in 'JFEFF' directory and sub-directories. For example if we use FEFF9.6, we can find the already compiled executable binary files in 'JFEFF/feff90/YOURSYSTEM/'. Here 'YOURSYSTEM' is different, depending on either FEFF is installed on Windows, Mac or Unix. These binary files may or may not work when copied from one machine to another, which means one need to reinstall when transferring to another calculation platform.

Again taking FEFF9.6 for example, all the original fortran modules can be found in 'JFEFF/feff90/mod', and there is also a compilation file within this directory specifically for compiling all fortran modules for current system (named 'Compile\_mac' or whatever). Then at the command line, one can easily type 'sh Compile\_mac' when you are already in 'JFEFF/feff90/mod' directory. Once you read the 'Compile\_mac' script, you will find this script does compile the modules into 'JFEFF/feff90/YOURSYSTEM/' directory which we have talked about in above context. For the MPI compilation, it is quite similar to that for FDMNES. First of all, one needs to make sure what MPI compiler they have got on the system, followed by the command 'module load YOUR-MPI-LIBRARY' (shown in detail for [FDMNES MPI compilation](#) in last part). Then we can 'cd' to the directory 'JFEFF/feff90/mod/MPI', followed by executing (using 'csh' but 'sh', since this script in a 'csh' script) the compiling script 'Compile.unix' and it should be ready for parallel calculation.

Here specifically for Unix based system, there will be another problem when installing JFEFF package. When the '.run' file for JFEFF installation is 32-bit version, then it is impossible to run it under 64-bit system. Thus we need to configure the source code to run FEFF from command line. By decompressing the linux package we can obtain JFEFF folder containing all the source code and related packages. In the directory 'feff90', there is a folder named 'unix', which contains the executable modules for FEFF calculation. We should first give execution privilege to all files under 'JFEFF/feff90/unix' directory, and then we go back to the JFEFF folder and edit 'feff' file using any editor, and could find the first line containing the full directory for all executable files under 'unix' directory, if it is not correct specifically for our own system, just change it according to where we put the JFEFF folder. Then what we need to do is to add the JFEFF directory to the system environment for the convenience of executing FEFF calculation universally.

## Using DOS script to work with FitIt

The using of FitIt program for the structural fitting does require lots of preparations such as the geometry transformation and corresponding fdmnes calculations. Since it is quite tedious to handle all necessary input files by hand, series of batch files (find them following [DOS script](#), the main batch file is 'proc.trans.bat') were written for the convenience of preparing FitIt input files. And this part describes how the batch file works, which is accompanied by some key points that people should pay attention to when they are using FitIt together with FDMNES for the structural fitting.

Before introducing the batch files, there are a few things to keep in mind. Firstly, it is strongly recommended by FitIt authors that we should create an independent folders for each set of calculations and structural fitting (either for different materials, or for different way of structural





tuning for the same material), which is actually a common sense. Located in the independent directory that we created for specific calculation and fitting, we should necessarily have the following files in current directory:

Exp.txt	—	Experimental spectrum for which we want to fit the structure
fdmfile.txt	—	The file needed to locate the input file for fdmnes calculation
Prot2005.exe	—	The executable program for geometry transformation.
xsect.dat	—	The Info file necessary for fdmnes calculation, which can be copied to current directory from the installation home directory of fdmnes program.

The second thing is about the using of Prot2005 program for geometry transformation. Here we are not going into details about how to use Prot2005 (detailed introduction can be found in FitIt user guide P7-8). Basically, to use Prot2005, we need to have corresponding '\*.pdb' file in which all the atoms are classified into different groups. Each group of atoms was then rotated or shifted as a rigid. Hence before the structural tuning using Prot2005, we should make sure which group(s) we are going to deal with and how we are going to deal with them (rotate? or shift? or rotate while shifting?). Keeping this in mind, then we could understand why the following batch file that we are going to mainly talk about later requires us to input names for two sets of directories. The first directory is actually for all stuff related to our previously determined which group(s) that we are going to deal with. That is to say, after we decide to deal with, let's say, 4 groups of atoms, then the calculation output files and all the other files related to these 4 groups of atoms should go into the first directory that we create when executing the batch file. The second set of directories is for each way of tuning these 4 groups of atoms, which is to say that for each way of structural tuning we should have its corresponding directory (and the second set of directories is automatically put into the first created directory).

The last but not least, it should be pointed again that FDMENS calculation uses fractional atomic coordinates. The disadvantage of using fractional coordinates is that it is not that easy for us to visually see the corresponding structure. Our solution for this problem is that we create another '\*.pdb' file which is all the same with the one we use for FDMNES calculation except the coordinate is real Cartesian but not fractional. Then we change the geometry for the real '\*.pdb' file (which can be used for visual 3-D modeling) following exactly the same way with that for '\*.pdb' with fractional coordinates (for FDMENS calculation) file using Prot2005.

Keeping in mind all stuff talked about in above section, we then move on to introduce our batch file. As said above, we need to create two sets of directories. When we execute the batch file, it will ask for the name for corresponding directories and you can pick up any name that is easy to recognize. However before that, we need a temporary directory which contains all directories that we will create, the temporary directory is embedded in the batch file (in our case, it is "c:\Users\yuanpeng\Temp\"), one can replace this with their own temporary directory in the batch file. In the temporary directory, we need three files: 'fdmnes.inp' file and two '\*.pdb' files. The 'fdmnes.inp' is the FDMENS input file, which has all keywords except atomic coordinates. Here it should be pointed out that before doing the structural fitting using FitIt, we should have already made sure that the FDMNES calculation for initial structure consists well with experimental spectrum for standard reference sample. If not, it is meaningless to fit the structure because we start from the wrong point in that case. Therefore, all the keywords in 'fdmnes.inp' here should be the same with those used for initial structure



calculation. As for the two '\*.pdb' files, we have already talked about them in details in the above section.

Having all stuff configured correctly, we could then execute the batch file from FitIt working directory (the one containing 'Exp.txt', 'fdmfile.txt', etc.). And there shouldn't be any problems to input what is required in DOS terminal. If there is any problems, the notes within the batch file could be helpful for understanding what the program is doing in each step. Here there are several points to notice: the first thing is that when Prot2005 program is first called by the batch file, it is doing geometry transformation on '\*.pdb' file with real Cartesian coordinates (for visual modeling use), and when Prot2005 asks for file name input for this transformation it is better to input 'tr0' for every single way of geometry transformation (as said above, all stuff concerning each way of geometry transformation will go to different folders, so there is no overwriting problem here). Then second time calling for Prot2005 is for the real calculation, and the geometry transformation result will go into 'fdmnes.inp' file. Again, Prot2005 will ask for the name for this set of geometry transformation, then we can use names that are easy for us to recognize in future. After the geometry transformation, the batch file will ask for transformation name again, which is for the convenience of moving files between folders because after geometry transformation the batch file will move all the output files to the independent directory we created in last step. Then what we need to do is to input the same name as we input in the second call of Prot2005. After each geometry transformation, all the resulted output files including 'trd.\*' (containing all our input for corresponding geometry transformation), '\*.pdb' (the new one after geometry transformation), 'fdmnes.inp' (the new FDMNES input file containing all fractional coordinates after geometry transformation), etc. will be moved to corresponding directory (the one we created in last step for each way of geometry transformation). Then we can go into each independent directory and execute the correspondingly created 'fdmnes.bat' file to calculate the theoretical XANES spectra for the structure after geometry transformation, and then the calculated spectrum will be transformed to the format required by FitIt automatically. After all the ab initio calculation, we then could execute the 'get\_data.bat' file (one of the batch files that can be found in the link at the very beginning) from the current working directory of FitIt to copy all necessary 'trd.\*' and 'fpsecs.\*' files there. Then it should be ready to run FitIt main program and do the structural fitting without problems.

Here there are still several things to point out: the first one is that we should know that FitIt already links up with 'fdmnes.exe' and 'jolyspec.exe' when we install it, which means we should have a compiled environment within FitIt for us to execute all the stuff we talked about. Then why we write the batch script? It is mainly because we feel it is not convenient to do the calculation and structural fitting in systematic way within FitIt. When we use the scripts to work for us, we could then do things like this: calculate initial spectrum → change the structure → calculate spectrum after geometry transformation → copy all necessary files back to working directory → do the fitting → clear all stuff → backup all stuff. And more importantly, all these steps are compiled into one batch file, hence one click all done. For clearing, another simple batch file named 'remove\_files.bat' was written for the double-click tidying up. The last thing to mention is about 'fdmnes.inp' file. Sometimes there may already exists 'fdmnes.inp' file in our FitIt working directory, hence we need to clear all the existing coordinates in that file. The way we do this is that we initially put a original 'fdmnes.inp' file in our temporary directory and then copy it to cover the existing one in current FitIt working directory.

### About hole count in FEFF

In FEFF calculation, if the print level is set up to 3 or even higher, FEFF will do the calculation





of hole counting, which is quite slow (see FEFF9.6 manual P21). Then the question is: what is hole counting? For X-ray absorption problem, when the core electron is excited to continuum state, corresponding core-hole will be created. Traditionally, the core-hole was treated using full screen method - self-consistently screening the core-hole during SCF loop, which is consistent with the 'final state rule'. In this case, since the core-hole is self-consistently screened, thus there is no 'holes' in final state. However, when using other method to account for core-hole effect (such as the RPA method introduced in FEFF9.6 manual P21), there could exist holes in the final state, thus for these kinds of final state, the count of holes can be calculated once we obtain the final state, which is just what the 'hole count' means here.

### About convolution in FDMNES

Sometimes for large cluster calculation, it is faster to separate the energy grid to several parts in series and run each calculation, independently. Or sometimes the calculation may be interrupted for some reason, then we need to continue the calculation from where the program stops. In both case, after the calculation, we need to compose the calculation result for different energy grid into the overall result. To do this, we need to first compose the raw calculation results (those before convolution), followed by the convolution process for the composed raw calculation result. If convolution was done for each independent energy grid, there will be huge jump between different energy grid.

### About energy grid in FDMNES

In FDMNES, the energy grid for the calculation is defined in the following format:

*Start\_Energy Interval Intermediate\_Energy\_1 Interval Intermediate\_Energy\_2  
Interval ... Intermediate\_Energy\_n Interval Stop\_Energy*

For the 'Start\_Energy', 'Stop\_Energy' and all the intermediate energy points, the value has to be integer, e.g. it can be given as '-10' or '-10.0', but CANNOT be '-10.3', etc.

Moreover, the energy value between each individual calculation should better not overlap with each other, since otherwise there will be jump between different energy grid when combining the calculation result after convolution for each.

### About multiple sites calculation

When there are more than one inequivalent atoms in the unit cell, FDMNES will calculate the raw absorption coefficient for each site and then combine them all for convolution. In this case, if the 'Crystal' card was used for the calculation, then FDMNES will take care of all the procedures by itself. However if the 'Molecule' card was used and each site was calculated manually, and independently, we then need to convolute all the raw output to obtain the final absorption coefficient. To do this, an input file containing only the convolution related cards should be created. For convoluting multiple sites calculation, we then need to introduce multiple raw outputs into one single convolution process, which means all the multiple sites calculation raw output files should be specified in the input file for convolution purpose. The following is an example:

*Calculation*

*xanout/test\_stand/Ge\_ST12\_MS\_r7\_1.txt*

*1.0 0.023*

*xanout/test\_stand/Ge\_ST12\_MS\_r7\_2.txt*

*1.0 0.000*



*Conv\_out*

*xanout/test\_stand/Ge\_ST12\_MS\_r7\_Efermi\_0\_conv.txt*

*Convolution*

There are two numbers following each raw calculation output file, and the first one is the weight for each output. For the above example, the final convolution output will be:

$$0.5 \times \text{Output}_1_{\text{Conv}} + 0.5 \times \text{Output}_2_{\text{Conv}} \quad (0-1)$$

The second number is the energy grid shift for each raw output. First of all, the reason for such shift is obvious since the local environment for each site should be a bit different from each other and there may be shift of the band structure calculated (thus influence the DOS and the corresponding absorption coefficient). Then the problem is how to obtain the value for the shift of each site. Taking the above input file as an example, The two raw output files are 'Ge\_ST12\_MS\_r7\_1.txt' and 'Ge\_ST12\_MS\_r7\_2.txt', which is given under the card 'Calculation'. In each raw output file, it can be found corresponding 'Epsii' value in the top line of the file. Then if the input file for convolution is exactly like it is in the above example, the Fermi level will be determined independently for each site. Then to obtain the energy shift, we can first calculate the average of the two 'Epsii' values and determine the relative shift between 'Epsii' value of the single site and the average - use the average to minus the 'Epsii' value for each site and keep the sign. Attention should be paid that three decimal places should be kept for the average 'Epsii' based on the rounding up principle.

In most cases for the convolution, the 'Efermi' parameter should be specified, which then means the Fermi level was the same for each site BEFORE the shift. In this case, the determination of the shift value is a bit different from that used for the previous case (when no 'Efermi' parameter given in the input file). Again we use the previous example but giving the 'Efermi' card:

*Calculation*

*xanout/test\_stand/Ge\_ST12\_MS\_r7\_1.txt*

*1.0 0.023*

*xanout/test\_stand/Ge\_ST12\_MS\_r7\_2.txt*

*1.0 0.000*

*Conv\_out*

*xanout/test\_stand/Ge\_ST12\_MS\_r7\_Efermi\_0\_conv.txt*

*Efermi*

*0.0*

*Convolution*

In this case, the 'Epsii' value from the last site raw output (for the example here, it is 'Ge\_ST12\_MS\_r7\_2.txt') will be used as a baseline, which means the shift of 'Ge\_ST12\_MS\_r7\_2.txt' will be set to 0.000. Then the shift for other sites is determined by subtracting the 'Epsii' value for the last site from that of the corresponding site in problem. And again, three decimal places are necessary.

At last, it should be mentioned that the way to determine the energy shift is what is used as a default option by FDMNES, thus it is better to keep consistence when we do it by ourselves.



### About energy shift in FDMNES

In FDMNES manual (P34, May-2013), it says 'If one has got a reference for the initial orbital, it is possible to give it under the keyword *epsii*'. This means we can manually shift the energy grid to a already known reference position. However here the 'initial orbital' should come from other external electronic structure calculation sources. In real case, there is another way to align the energy grid, which is through the comparison between the calculation and experimental spectrum. Following the second way, the typical features in the exoerimental XANES spectrum can be taken as the reference position, e.g. for diamond-type Ge, the first main feature just above the white line can be regarded as such a *reference* feature.

The most convenient and reliable way to do this is to introduce a shift when doing the convolution process independently after the raw XANES calculation. For those calculations with multiple inequivalent positions, the corresponding method has already been introduced in the above term **About multiple sites calculation**. However when we try to do the same trick for single site calculation (e.g. for diamond-type Ge), FDMNES won't allow us to do such shift. Here it is possible to introduce a fake input (for convolution) and then carry out the same steps as shown in the above term **About multiple sites calculation**. Here is given the example for such trick:

*Calculation*

*xanout/test\_stand/Ge\_Dia\_MS-r7.txt*

*1.0 0.020*

*xanout/test\_stand/Ge\_Dia\_MS-r7.txt*

*0.0 0.000*

*Conv\_out*

*xanout/test\_stand/Ge\_Dia\_MS-r7-Efermi\_0\_conv.txt*

*Efermi*

*0.0*

*Convolution*

where the same raw output result was used as the input(s) for convolution and the second one was given the weight of 0, by which we mean 'fake' input.

### Fermi level and 'V0imp' in FDMNES

By default, a calculation cycle is performed over the occupied states to determine the Fermi energy in FDMNES (see the FDMNES User Guide – Page-24). However, the calculated Fermi level is actually not self-consistent. The reason is, when the core-electron is excited to the continuum, the energy level will definitely be influenced due to the 'rearrangement' of all the occupied energy levels. In another word, the Fermi level calculated (by default) corresponds to the 'Ground state' only. Therefore, FDMNES provides the alternative to calculate the Fermi level self-consistently, which means the core-hole effect is then taken into consideration based on the sol-called Final State Rule (FSR).

For Multiple-scattering calculation, the muffin-tin ground potential (which is also the reference for the photoelectron kinetic energy) is determined automatically as the average of the potential between the absorbing atom and the first crown. (FDMNES User Guide, P29) Thus when the potential field is determined, the muffin-tin ground is then directly determined based on the rule given in the manual. However, it is still possible to impose the muffin-tin ground at the



Fermi level which is determined as discussed in the first paragraph. To do this, the parameter 'V0imp' is then needed, and the value given is relative to the Fermi level. This does not only manually set the muffin-tin ground potential energy, but also determine the shape of the muffin-tin potential used for the calculation, since the muffin-tin radius is determined by the value it gets when the atomic potential is equal to the interstitial potential (FDMNES User Guide, P43).

For the 'FDM' mode calculation, the meaning of the 'V0imp' parameter is a bit different. Of course it also defines the reference of the kinetic energy, but it refers to the kinetic energy of the photoelectron outside the calculation sphere (the specified calculation cluster with the 'radius' parameter). And the default way that FDMNES follows to determine this reference (for 'FDM' mode) is to set it to be the value of the potential energy corresponding to the frontier of outside sphere (the position at the boundary between the calculation cluster and the outside). This part of information can also be found in the manual of FDMNES in the 'V0imp' introduction part.

### About 'Epsii' in FDMNES

In the FDMNES manual, it says that the 'Epsii' parameter refers to the energy of the initial orbital. To get an idea about what the initial orbital means, we need to get back to the electron configuration. Taking Ge as an example, there are four valent electrons in each Ge atom, therefore if we are given the total number of Ge atoms in the calculation cluster, we can then have the total number of valent electrons in the system. Then following the DFT calculation, we may have all the orbitals with energy from low to high. Here it should be noticed that the orbitals talked about here takes only the valent electrons into consideration. And among all the orbitals, the one with the lowest energy is of course the initial orbital. And the value obtained by the FDMNES program can then be compared to the result from other calculation package, which is for the purpose of shifting the energy to its correct position. And the corresponding method to do the energy shifting is given in the FDMNES manual (under the 'Convolution' card) and also is given in [current document](#).

### About Muffin-tin Potential

The details about muffin-tin potential will not be given here, instead, here is given the point that muffin-tin potential is actually a serious approximation. As is pointed out in Rehr's early report (see [DOI for the paper](#)), there may be accuracy problem of the muffin-tin potential for certain system (e.g., small molecules), especially at low or near-edge photoelectron energies. And indeed there are uncertainties in the accuracy of muffin-tin potential working together with the multiple paths expansion. The reason is that for XAFS calculation, the various lifetime (of the excited state) effects tend to smear out the probably existing inaccuracy of the muffin-tin potential. Again it should be pointed out that the discussion given here is all from Rehr's paper as given above.

### About size effect in XANES and EXAFS calculation

First of all, the contribution to the X-ray absorption signal can be divided to two basic parts. The first part is the single scattering portion and the second one should then be the contribution from multiple scattering. And it is also clear that the contribution from multiple scattering is stronger in the near-edge region than that for the extended region where the energy of X-ray is much higher. The main reason for this should be that for the high-energy X-ray, it is less possible to be scattered or in another word, the influence of the scattering potential has less effect on the incoming high-energy X-ray. The dependence of the scattering effect (due to



the scattering potential) on the incoming X-ray energy can be deduced based on the standard scattering theory in Sakurai's book about modern quantum mechanics (see Chapter 7).

The other thing is the photoelectron mean free path is relatively longer (due to the lower kinetic energy) corresponding to the near-edge region as compared to that corresponding to the extended region. This was confirmed by the examination of the electron mean free path as the function of incoming X-ray energy in previous report (see the backup papers collection). However, as aforementioned, the multiple scattering effect of the low-energy photoelectron is much stronger than that for the high-energy photoelectron, therefore the photoelectron tends to 'wonder around' the local coordination environment instead of going a long way to 'feel' the further coordination shells. And as a matter of fact, this is one of the aspects that make the XANES spectra more sensitive to the local environment as compared to EXAFS.

The above mentioned theoretical consideration for the different influence of multiple scattering on XANES or EXAFS should then naturally lead to the expectation that the size of the cluster size for the real simulation should have more effect in EXAFS than for XANES. For XANES, as is already discussed, the multiple scattering effect is strong, therefore for a large cluster size, if considering the contribution from coordination shell far from the absorption atom, the contribution from the multiple scattering should then lead to rather long scattering path. And if borrowing the idea from that standard EXAFS equation, we could see that for large scattering path length, the contribution to the signal should be quite weak since the signal intensity is inverse proportional to the scattering path length. Here it should be mentioned that the standard EXAFS equation actually cannot describe the exact absorption near the edge, again, partly due to the contribution from multiple scattering effects. Even though it still makes sense to just 'borrow' the idea to understand the 'smearing' effect (therefore little contribution to the absorption signal) of the long-length scattering path in the near-edge region as is discussed above.

