OpenMX tutorials

Magnons in Fe using OpenMX

Computational Materials Physics Bootcamp

University of Delaware

19-22 August, 2024

By Ali Kefayati

UNIVERSITY OF DELAWARE

Fe using OpenMX:

Ground-State, Band Structure, DOS

Fe using OpenMX: Ground-State, Band Structure, DOS

1. Create a subdirectory inside your main directory using ‘mkdir’ command:

```
mkdir    OMX-Fe-GS
```

2. Copy all data from ../alikefay/OpenMX-Fe directory to your own subdirectory using cp command:

```
cp    /home/cmpb/OpenMX/Fe-GS/*    ./OMX-Fe-GS
```

3. Go to your subdirectory using ‘cd’ command:

```
cd    ./OMX-Fe-GS
```

4. Open the Fe.in file and try to understand the keywords.
5. Now, if you feel ready for the run, it’s time to submit the job. We need first to activate the module using:

```
module    unload    openmpi/4.1.4-intel-2023.0.0
module    unload    intel/2023.0.0
module    load    intel/2020.3
module    load    openmx
```

Fe using OpenMX: Ground-State, Band Structure, DOS

Fe.in

```
# File Name
System.CurrrentDirectory          ./                                    # default=./
System.Name                       Fe
DATA.PATH                         /home/opt/pkg/openmx/3.9.9-intel-2020.3/DFT_DATA19
level.of.stdout                   1                                     # default=1 (1-3)
level.of.fileout                  0                                     # default=1 (0-2)

# Definition of Atomic Species
Species.Number       1
<Definition.of.Atomic.Species
  Fe   Fe6.0S-s3p2d1    Fe_PBE19H
Definition.of.Atomic.Species>

# Atoms
Atoms.Number                      1
Atoms.SpeciesAndCoordinates.Unit  Ang                                   # Ang|AU
<Atoms.SpeciesAndCoordinates
 1   Fe   0.0000     0.0000     0.0000   9.0   7.0
Atoms.SpeciesAndCoordinates>

Atoms.UnitVectors.Unit            Ang                                   # Ang|AU
<Atoms.UnitVectors
  -1.43300   1.43300   1.43300
   1.43300  -1.43300   1.43300
   1.43300   1.43300  -1.43300
Atoms.UnitVectors>
```

Fe using OpenMX: Ground-State, Band Structure, DOS

Fe.in

```
# SCF or Electronic System
scf.XcType                      GGA-PBE                 # LDA|LSDA-CA|LSDA-PW|GGA-PBE
scf.SpinPolarization            On                      # On|Off|NC
scf.SpinOrbit.Coupling          Off                     # On|Off, default=off
HS.fileout                      Off                     # on|off, default=off
scf.ElectronicTemperature       300.0                   # default=300 (K)
scf.energycutoff                600.0                   # default=150 (Ry)
scf.maxIter                     800                     # default=40
scf.EigenvalueSolver            band                    # Recursion|Cluster|Band
scf.Kgrid                       9 9 9                   # means nk1xnk2xnk3
scf.Mixing.Type                 rmm-diisk               # Simple|Rmm-Diis|Gr-Pulay
scf.Init.Mixing.Weight          0.300                   # default=0.30
scf.Min.Mixing.Weight           0.001                   # default=0.001
scf.Max.Mixing.Weight           0.400                   # default=0.40
scf.Mixing.History              5                       # default=5
scf.Mixing.StartPulay           6                       # default=6
scf.criterion                   1.0e-7                  # default=1.0e-6 (Hartree)
Dos.fileout                     on                      # DOS calculation
Dos.Erange                      -25.0 25.0              # DOS energy range (0=chemical potential)
Dos.Kgrid                       27 27 27                # DOS k-point discritization

Band.dispersion                 on                      # on|off, default=off
Band.Nkpath                     5
<Band.kpath
    40  0.0000 0.0000 0.0000   0.5000 -0.500 0.5000   G H
    50  0.5000 -0.500 0.5000   0.0000 0.0000 0.5000   H N
    30  0.0000 0.0000 0.5000   0.0000 0.0000 0.0000   N G
    30  0.0000 0.0000 0.0000   0.2500 0.2500 0.2500   G P
    40  0.2500 0.2500 0.2500   0.5000 -0.500 0.5000   P H
Band.kpath>

# MD or Geometry Optimization
MD.Type                         NOMD                    # NOMD|OptC1|OptC2|OptC3|OptC4|OptC5|OptC6|OptC7|RFC5|RFC6|RFC7
MD.maxIter                      1                       # default=1
MD.Opt.criterion                3.0e-4                  # default=0.0003 (Hartree/Bohr)
```

Fe using OpenMX: Ground-State, Band Structure, DOS

6. Now, submit the job using the following command (if you get permission error first run: chmod -R 777 ./),

   nohup mpirun -np 5 openmx Fe.in > Fe.out -f openmp &

7. The Fe.out is the job log and you can see the progress of the scf loops by opening it.

8. When the job is finished open the Fe.out and find the magnetic moment of Fe atom and compare it with the experimental value of 2.22 μ_B. What is the ground state energy? How many scf loops are performed?

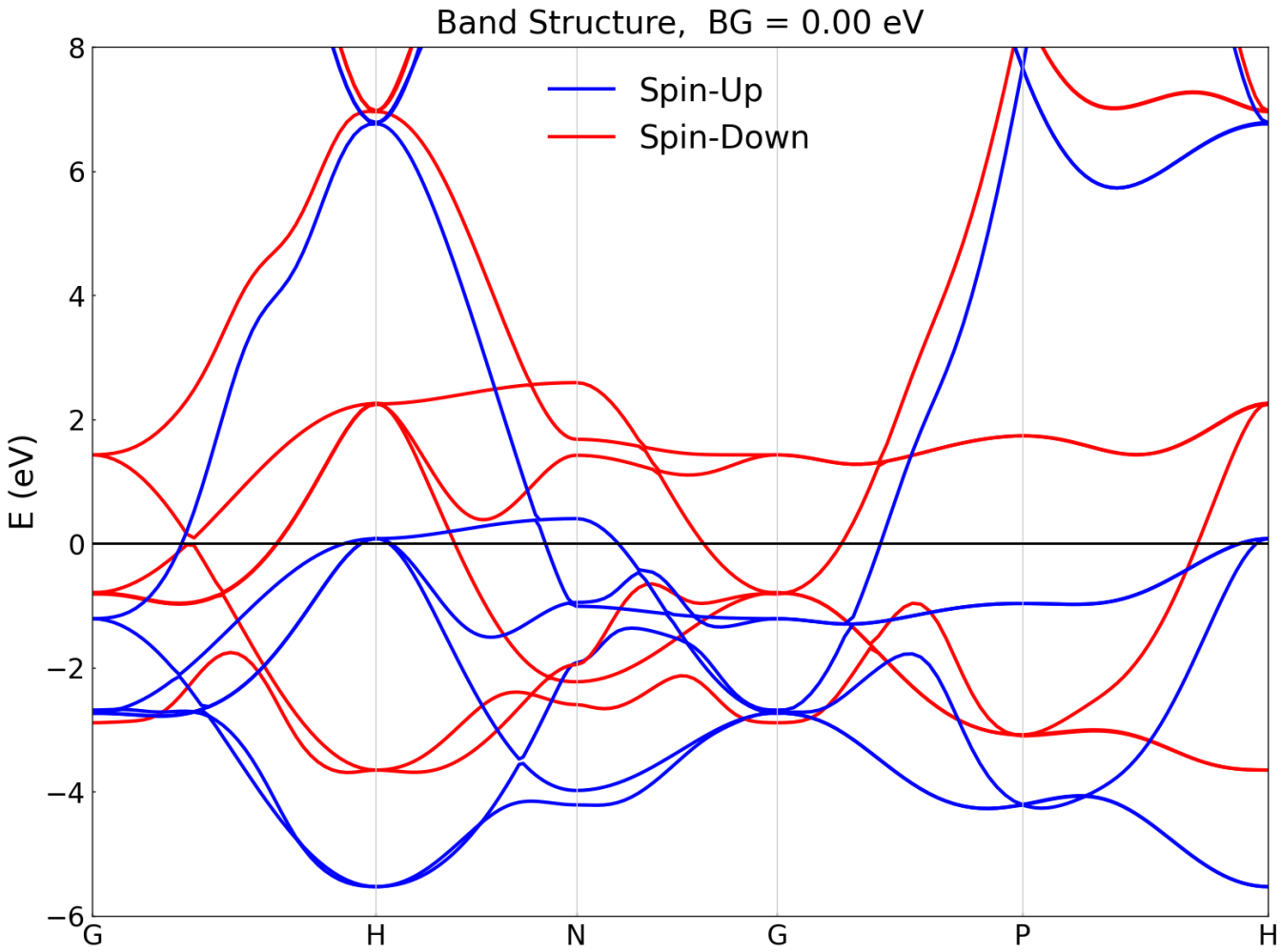
9. Use the following command to prepare the band structure data for plotting,

   bandgnu13 Fe.Band

10. Next use the python script in your directory to plot the band structure using the command,

    Python3 Plot_band.py

Fe using OpenMX: Ground-State, Band Structure, DOS



Fe using OpenMX: Ground-State, Band Structure, DOS

11. Now, we are ready to calculate the DOS. The keyword setting in the code are:

```
Dos.fileout          on              # DOS calculation
Dos.Erange        -25.0 25.0         # DOS energy range (0=chemical potential)
Dos.Kgrid          27 27 27          # DOS k-point discritization
```

Since the keyword is already set in the input files, you just need to do the post processing. Use the following command to calculate the total DOS (PDOS)

DosMain Fe.Dos.val Fe.Dos.vec

Then, you are interactively asked from the program as follow:

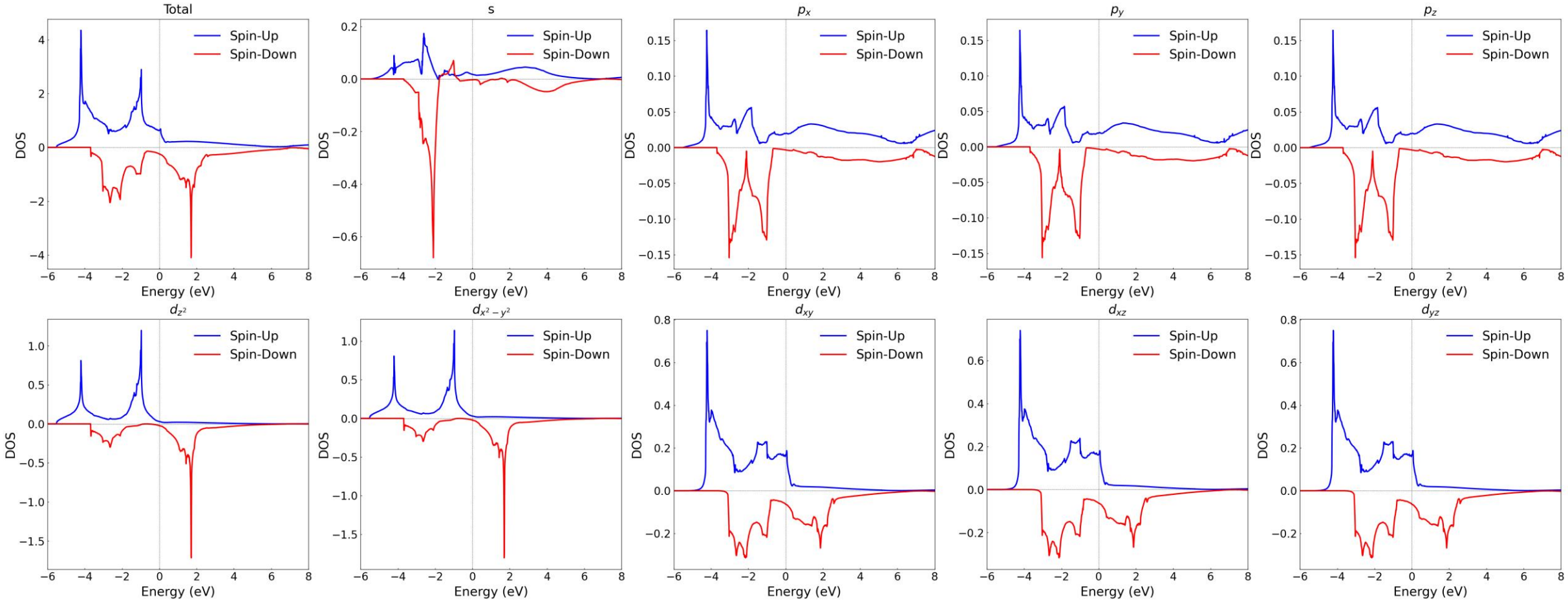
```
DosMain Fe.Dos.val Fe.Dos.vec
Max of Spe_Total_CNO = 14
1 1 1 101 102 103 101 102 103 201 202 203 204 205
<Fe.Dos.val>
<Fe>
Which method do you use?, Tetrahedron(1), Gaussian Broadening(2)
1
Do you want Dos(1) or PDos(2)?
2

Number of atoms=1
pdos_n=1
1
<Spectra_Tetrahedron> start
```

By answering these questions, you can calculate DOS or PDOS for a specific atom and orbitals.

Fe using OpenMX: Ground-State, Band Structure, DOS

Now, use the following command to plot the DOS

python3 Plot-DOS.py



Fe using OpenMX: Ground-State, Band Structure, DOS

Next, change the calculation from collinear to noncollinear DFT using the following changes:

```
# Atoms
Atoms.Number                          1
Atoms.SpeciesAndCoordinates.Unit   FRAC                                  # Ang|AU|FRAC
<Atoms.SpeciesAndCoordinates
 1   Fe   0.0000     0.0000     0.0000  9.0   7.0    0.0   0.0 0.0    0.0    0    off
Atoms.SpeciesAndCoordinates>
```

and,

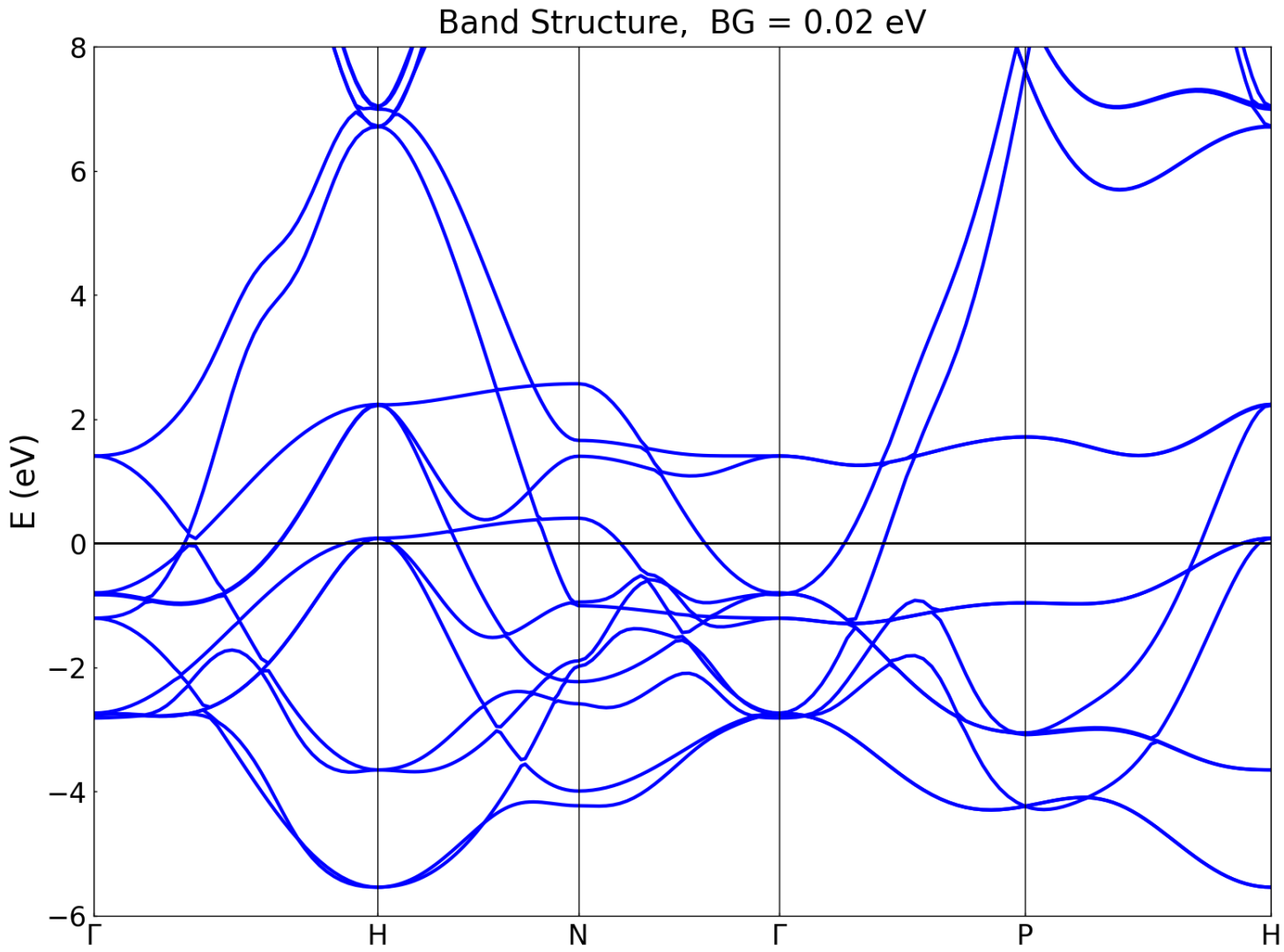
```
scf.SpinPolarization                  NC                                  # On|Off|NC
```

Repeat the calculation and plot the band structure.

- What is the difference between the band structure in collinear and noncollinear calculation?
- What is the difference between the total energy in two calculations?

Fe using OpenMX: Ground-State, Band Structure, DOS

Noncollinear band structure:



Fe using OpenMX:

Heisenberg Exchange Parameters

Fe: Adiabatic Magnon Dispersion

In this exercise, you will learn how to calculate the Heisenberg exchange parameters of ferromagnetic Fe using OpenMX. In the next step, we will calculate the magnon dispersion of Fe within spin wave theory.

1. Create a subdirectory in your directory and call it ‘Fe-Jx’

```
mkdir    ./OMX-Fe-Jx
```

2. First copy the following directory to you own directory using cp command:

```
cp   /home/cmpb/OpenMX/Fe-Jx/*   ./OMX-Fe-Jx
cd   ./OMX-Fe-Jx
```

3. ‘cd’ to the subdirectory and open the Fe.in file. Try to understand all the keywords. As you can see, we are using many k-points for this calculation. This is because we want to get good convergence and accuracy for the Heisenberg exchange parameter.
4. I addition, the basis set of Fe is chosen as Fe4.0H-s2p2d2. This is because we want very localized magnetic moments for adopting the Heisenberg model.

Fe: Adiabatic Magnon Dispersion

Fe.in

```
# File Name
System.CurrrentDirectory          ./                                    # default=./
System.Name                       Fe
DATA.PATH                         /home/opt/pkg/openmx/3.9.9-intel-2020.3/DFT_DATA19
level.of.stdout                   1                                     # default=1 (1-3)
level.of.fileout                  0                                     # default=1 (0-2)

# Definition of Atomic Species
Species.Number       1
<Definition.of.Atomic.Species
  Fe  Fe4.0H-s2p2d2   Fe_PBE19H
Definition.of.Atomic.Species>

# Atoms
Atoms.Number                      1
Atoms.SpeciesAndCoordinates.Unit  Ang                                   # Ang|AU
<Atoms.SpeciesAndCoordinates
 1  Fe  0.0000    0.0000    0.0000  9.0  7.0
Atoms.SpeciesAndCoordinates>

Atoms.UnitVectors.Unit            Ang                                   # Ang|AU
<Atoms.UnitVectors
  -1.43300  1.43300   1.43300
   1.43300 -1.43300   1.43300
   1.43300  1.43300  -1.43300
Atoms.UnitVectors>
```

Fe: Adiabatic Magnon Dispersion

```
# SCF or Electronic System
scf.XcType                      GGA-PBE                 # LDA|LSDA-CA|LSDA-PW|GGA-PBE
scf.SpinPolarization            On                      # On|Off|NC
scf.SpinOrbit.Coupling          Off                     # On|Off, default=off
HS.fileout                      On                      # on|off, default=off
scf.ElectronicTemperature       300.0                   # default=300 (K)
scf.energycutoff                600.0                   # default=150 (Ry)
scf.maxIter                     800                     # default=40
scf.EigenvalueSolver            band                    # Recursion|Cluster|Band
scf.Kgrid                       27 27 27                # means nk1xnk2xnk3
scf.Mixing.Type                 rmm-diisk               # Simple|Rmm-Diis|Gr-Pulay
scf.Init.Mixing.Weight          0.300                   # default=0.30
scf.Min.Mixing.Weight           0.001                   # default=0.001
scf.Max.Mixing.Weight           0.400                   # default=0.40
scf.Mixing.History              5                       # default=5
scf.Mixing.StartPulay           6                       # default=6
scf.criterion                   1.0e-7                  # default=1.0e-6 (Hartree)

Band.dispersion                 on                      # on|off, default=off
Band.Nkpath                     5
<Band.kpath
    40  0.0000 0.0000 0.0000   0.5000 -0.500 0.5000   G H
    50  0.5000 -0.500 0.5000   0.0000 0.0000 0.5000   H N
    30  0.0000 0.0000 0.5000   0.0000 0.0000 0.0000   N G
    30  0.0000 0.0000 0.0000   0.2500 0.2500 0.2500   G P
    40  0.2500 0.2500 0.2500   0.5000 -0.500 0.5000   P H
Band.kpath>

# MD or Geometry Optimization
MD.Type                         NOMD                    # NOMD|OptC1|OptC2|OptC3|OptC4|OptC5|OptC6|OptC7|RFC5|RFC6|RFC7
MD.maxIter                      1                       # default=1
MD.Opt.criterion                3.0e-4                  # default=0.0003 (Hartree/Bohr)
```

Fe: Adiabatic Magnon Dispersion

4. Another important keyword is 'HS.fileout On'. In fact, jx is a post processing calculation and we need first to run a ground state calculation. Please note that current version of OpenMX only support colinear calculation for jx.

5. Activate the module using:

```
module unload openmpi/4.1.4-intel-2023.0.0
module unload intel/2023.0.0
module load intel/2020.3
module load openmx
```

5. Now you can run the job. While you are waiting for the run to be finished, open the Fe.config file and try to understand the keywords.

```
nohup mpirun -np 5 openmx Fe.in > Fe.out -f openmp &
```

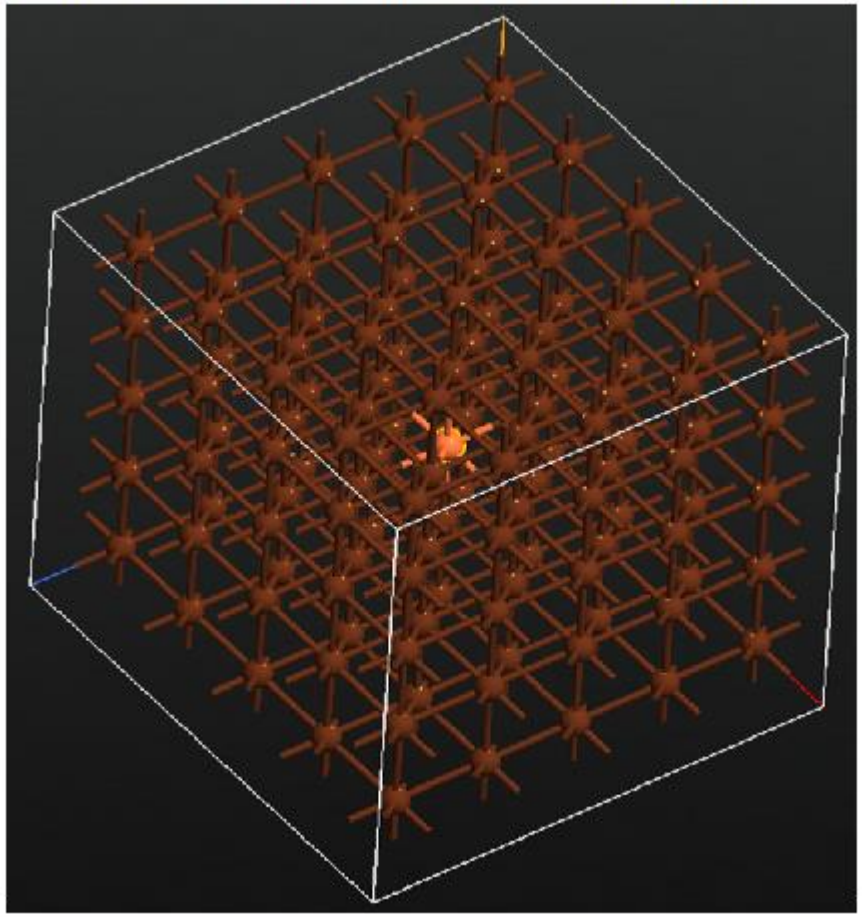
6. When the calculation is finished, run the jx using the following command,

```
nohup mpirun -np 5 jx Fe.scfout Fe.config > Jx.log -f openmp &
```

Fe: Adiabatic Magnon Dispersion

Fe.config

```
Flag.PeriodicSum        Off         # default - off  (NOTE: No role in cluster calculations.)
Num.Poles               60          # number of poles NP for the nite pole approximation of Fermi function
Num.Kgrid               27 27 27    # These values should be same or a bit larger than those in OpenMX calculation.
Num.ij.pairs            124         # NOTE: Number of ij pairs.
Bunch.ij.pairs          124         # default - 1

<ijpairs.cellid
  1   1 -2 -2 -2
  1   1 -2 -2 -1
  1   1 -2 -2  0
  1   1 -2 -2  1
  1   1 -2 -2  2
  1   1 -2 -1 -2
  1   1 -2 -1 -1
  1   1 -2 -1  0
  1   1 -2 -1  1
  1   1 -2 -1  2
  1   1 -2  0 -2
  1   1 -2  0 -1
  1   1 -2  0  0
  1   1 -2  0  1
  1   1 -2  0  2
  1   1 -2  1 -2
  1   1 -2  1 -1
  1   1 -2  1  0
  1   1 -2  1  1
  1   1 -2  1  2
```



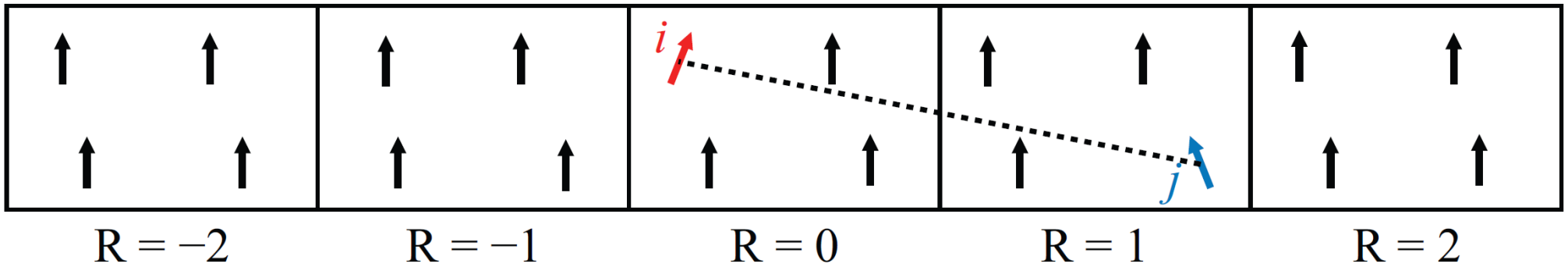
Fe: Adiabatic Magnon Dispersion

The code calculates the Heisenberg constants as,

$$J_{i\mathbf{0},j\mathbf{R}} = \frac{1}{2}\sum_{p=1}^{N_\mathrm{P}} \tilde{R}_p \sum_{\mu,\nu\in i}\sum_{\mu',\nu'\in j} \mathrm{Re}\left\{[\hat{P}_i]_{\nu\mu} G^+_{i\mu,j\nu'}(\downarrow,\tilde{z}_p,\mathbf{R})[\hat{P}_j]_{\nu'\mu'} G^+_{j\mu',i\nu}(\uparrow,\tilde{z}_p,-\mathbf{R})\right\}$$

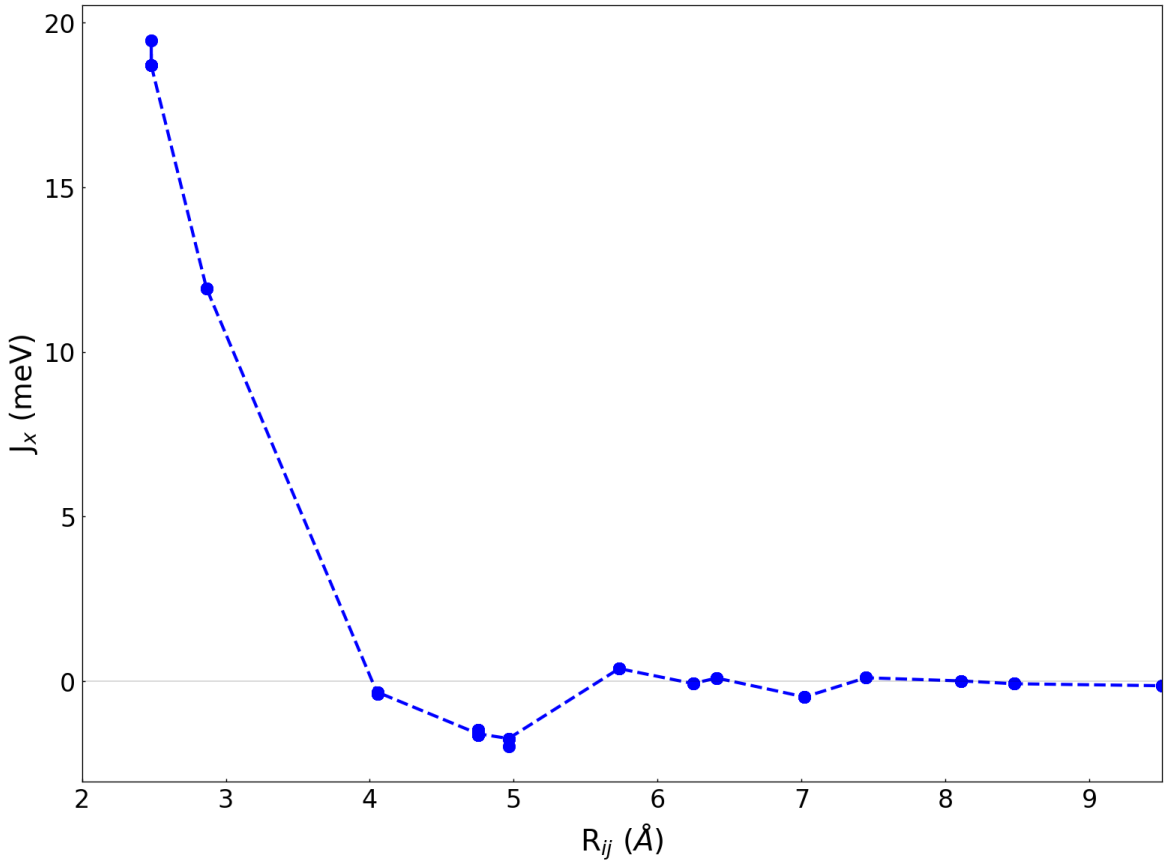
$$G^+_{j\mu',i\nu}(\uparrow,\varepsilon,-\mathbf{R}) \equiv \int d^3\left(\frac{ka}{2\pi}\right) e^{i\mathbf{k}\cdot\mathbf{R}} \sum_n \frac{C_{j\mu',n\uparrow}(\mathbf{k}) C_{i\mu,n\uparrow}(\mathbf{k})}{\varepsilon + i\eta - \varepsilon_{n\uparrow}(\mathbf{k})}$$

$$G^+_{i\mu,j\nu'}(\downarrow,\varepsilon,\mathbf{R}) \equiv \int d^3\left(\frac{ka}{2\pi}\right) e^{-i\mathbf{k}\cdot\mathbf{R}} \sum_{n'} \frac{C_{i\mu,n'\downarrow}(\mathbf{k}) C_{j\nu',n'\downarrow}(\mathbf{k})}{\varepsilon + i\eta - \varepsilon_{n'\downarrow}(\mathbf{k})}.$$



Fe: Adiabatic Magnon Dispersion

Here, we calculated the Heisenberg exchange constants for all neighbors in the demonstrated supercell of the Fe atom in the middle of the supercell (The supercell contains 124 atoms with one centered and 123 neighboring Fe atoms). When the calculation is done run: python3 Plot-HeisP.py

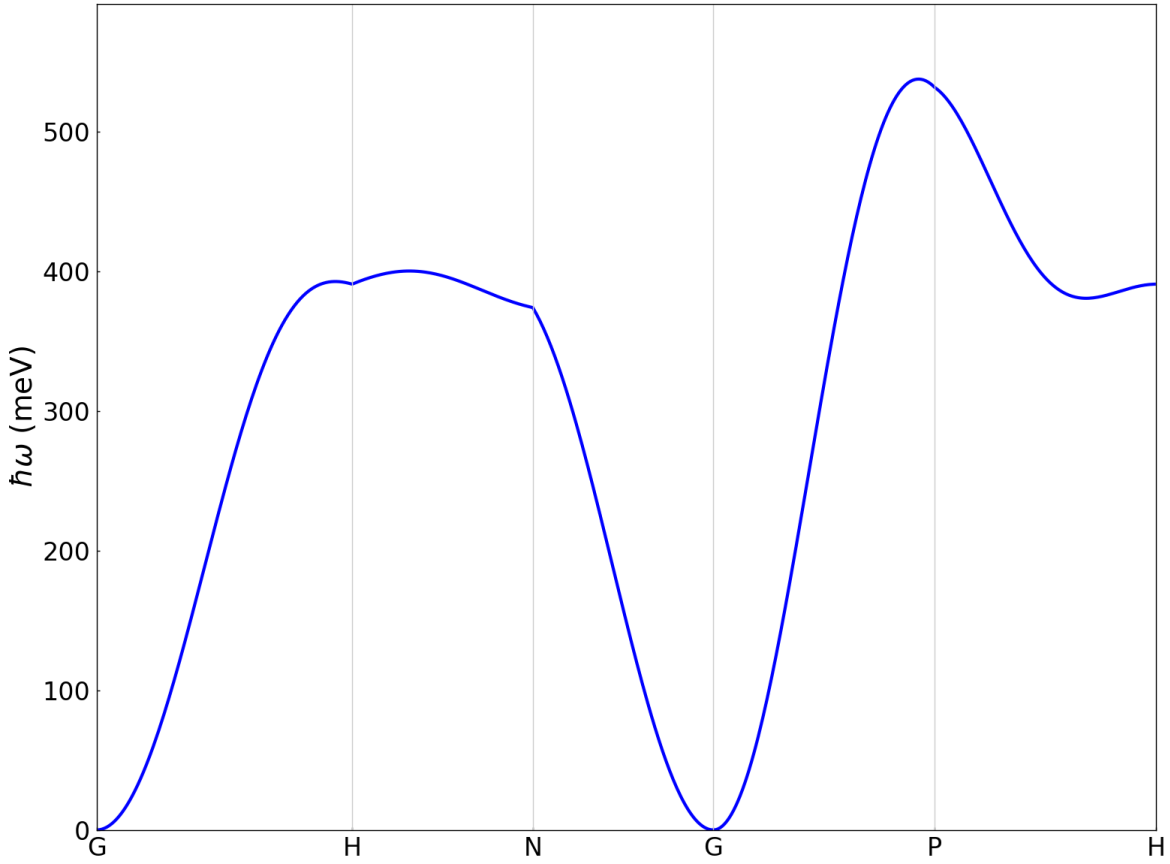


Fe: Adiabatic Magnon Dispersion

In the next step, run the following script: python3 Plot-Magnon.py

It plots the calculated exchange constants as well as the adiabatic magnon dispersion along high symmetry lines in the Brillouin zone using the spin wave theory:

$$\hbar\omega_{\boldsymbol{k}} = 2S \sum_{j} J_{ij} \left[1 - \mathrm{e}^{\mathrm{i}\boldsymbol{k}\cdot(\boldsymbol{R}_i - \boldsymbol{R}_j)}\right]$$



Magnon Dispersion of Fe:
OpenMX + TB2J + UppASD

OpenMX + TB2J + UppASD

In this exercise, you will learn how to calculate the Heisenberg exchange parameters of ferromagnetic Fe using

OpenMX + TB2J + UppASD

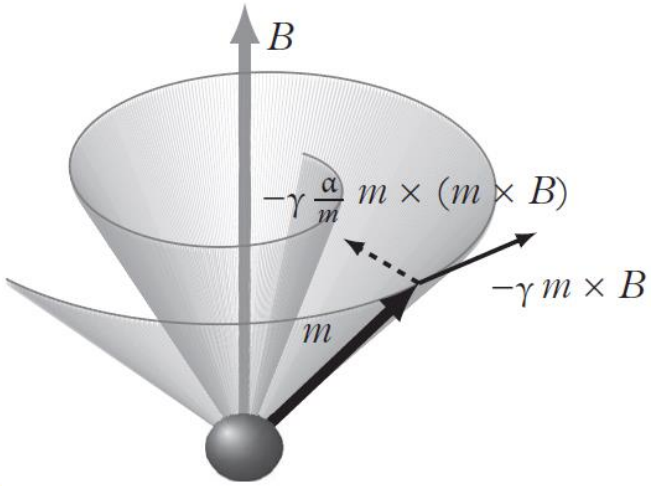
TB2J: A python package that calculates the Heisenberg exchange parameters using tight-binding Hamiltonians. It also works with Hamiltonians with Wanier functions or Atomic Orbitals as the basis set (tutorials here).

UppASD: A code Atomistic Spin Dynamics based of the Landau–Lifshitz–Gilbert equation (LLG). The user provides the Heisenberg exchange parameters, the Atomic Structure, and the local magnetic moments in the structure and the code calculates adiabatic magnon dispersion, static and dynamical structural functions, thermodynamics of the spin system, …

LLG equation:

$$\frac{d\mathbf{m}_i}{dt} = -\gamma_{\mathrm{L}}\mathbf{m}_i \times \left(\mathbf{B}_i + \mathbf{B}_i^{\mathrm{fl}}\right) - \gamma_{\mathrm{L}}\frac{\alpha}{m_i}\mathbf{m}_i \times \left[\mathbf{m}_i \times \left(\mathbf{B}_i + \mathbf{B}_i^{\mathrm{fl}}\right)\right]$$

$$\gamma_{\mathrm{L}} = \frac{\gamma}{(1+\alpha^2)},$$



OpenMX + TB2J + UppASD

1. Create a subdirectory in your directory and call it ‘OMX-Fe-OTU’

```
mkdir    ./OMX-Fe-OTU
```

2. First copy the following directory to you own directory using cp command and then ‘cd’ to the subdirectory:

```
cp    /home/cmpb/OpenMX/Fe-OTU/*    ./OMX-Fe-OTU
cd    ./OMX-Fe-OTU
```

3. Open Fe.in file and try to understand all the keywords. As you can see, we are using many k-points for this calculation to get good convergence and accuracy for the Heisenberg exchange parameters that will be calculated using TB2J package. In addition, the spin polarization of the calculation is set to noncollinear (NC).
4. Activate the module using:

```
module    unload    openmpi/4.1.4-intel-2023.0.0    intel/2023.0.0
module    load    intel/2020.3
module    load    openmx
```

Fe: Adiabatic Magnon Dispersion

5. Now you can run the job. This time, we want to save the output files in a directory called ‘Out’,

```
mkdir Out
nohup mpirun -np 5 openmx Fe.in > Fe.out -f openmp &
```

6. When OpenMX run is finished, let’s before going further take care of the modules. Use the following commands,

```
module unload openmx/3.9.9-intel-2020.3 intel/2020.3
module load intel/2023.3
module load tb2j
module load uppasd
```

7. When the calculation is done, create a subdirectory and call it ‘TB2J’, copy the Hamiltonian of the OpenMX into it and ‘cd’ to the subdirectory,

```
mkdir ./TB2Jrun
cp ./Out/Fe.scfout ./TB2Jrun
cp ./Out/Fe.xyz ./TB2Jrun
cd ./TB2Jrun
```

OpenMX + TB2J + UppASD

8. We need to create a python virtual environment for TB2J calculation. To do it, follow the instructions,

```
python3  -m  venv  ./TB2Jenv
source  ./TB2Jenv/bin/activate
```

9. Now, we are ready to run the TB2J code. Run the following script,

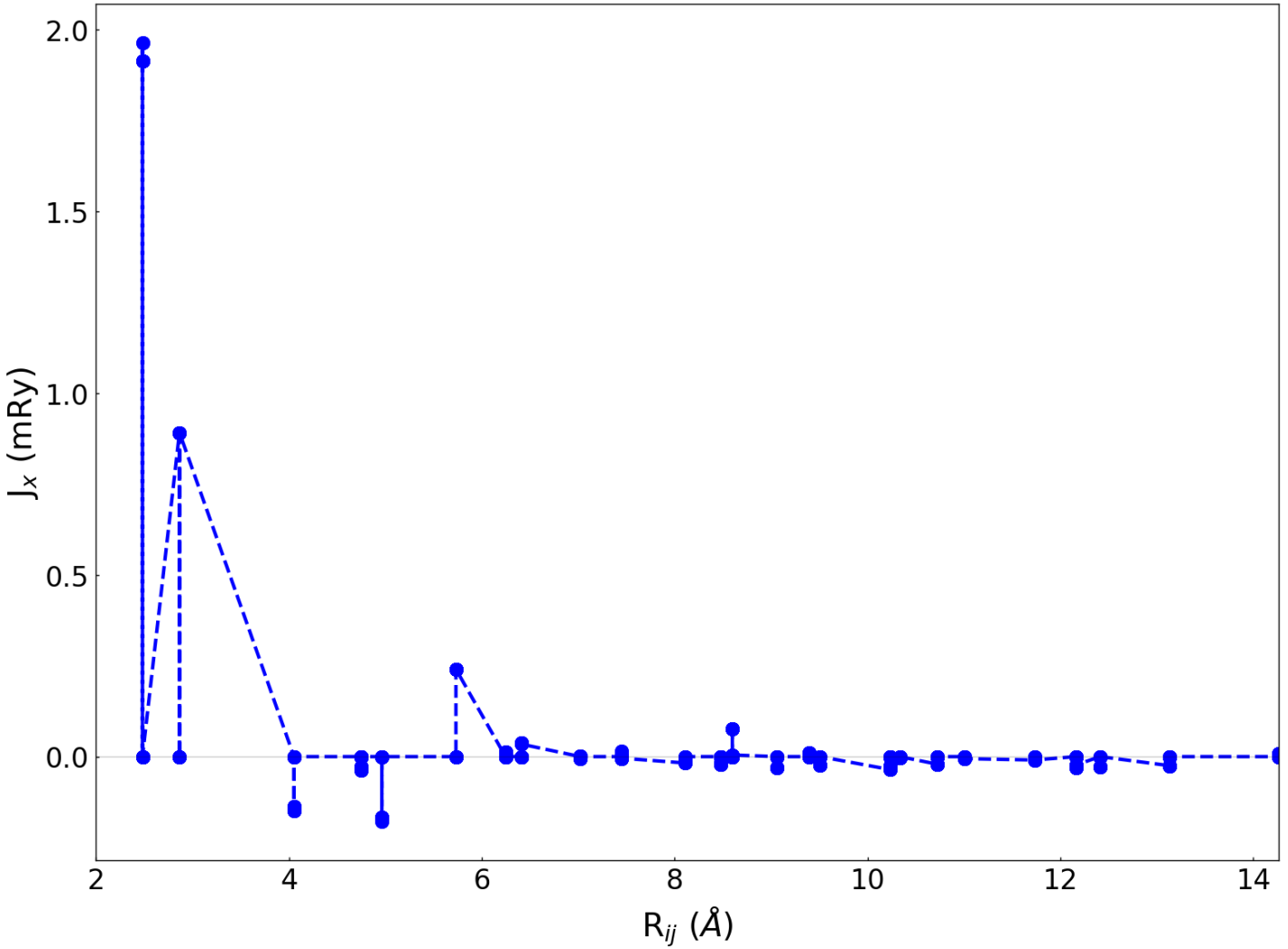
```
openmx2J.py  --prefix  Fe  --elements  Fe  --kmesh  7  7  7
```

10. When the calculation is done, deactivate the python environment, 'cd' back to the 'OMX-Fe-OUT' directory and use the following script to extract the Heisenberg parameters for the UppASD calculation.

```
deactivate
cd  ..
python3  Exchange.py
```

✓ The Exchange code extract the Heisenberg parameters from the output file of TB2J and saves it as 'jfile' and plots it ('Exchange.png'). The unit of the exchange parameters for the ASD calculation with UppASD must be mRy.

OpenMX + TB2J + UppASD



OpenMX + TB2J + UppASD

11. Next, create a subdirectory called ‘UppASDrun’,

```
mkdir     UppASDrun
```

12. Next, copy ‘jfile’ that is created by running ‘Exchange.py’ to UppASDrun,

```
cp  ./jfile       ./UppASDrun
```

13. Then, copy file from the home/cmpb/OpenMX/UppASD to the subdirectory. These are input files of UppASD.

```
cp  /home/cmpb/OpenMX/UppASD/*    ./UppASDrun
```

14. Now, we are ready to run UppASD. ‘cd’ to UppASDrun and open all input files and try to understand each file and contemplate the keywords in the input file,

```
cd       ./UppASDrun
```

OpenMX + TB2J + UppASD

15. There are four input files for UppASD (python files are for pre or post processings),

- ✓ Inspad.dat: is the input file containing all the settings for solving the stochastic LLG equations as well as the data related to the structure.
- ✓ Momfile: is the file containing the magnetic moment of each atom and its direction.
- ✓ Posfile: is a file containing the atomic position of the magnetic structure.
- ✓ Qfile: is a file containing k-pints of the high symmetry lines of the BZ for plotting the magnon dispersion.

16. Run the code using this command,

```
sd
```

17. Finally, you can plot the magnon dispersion using the postQ.py script. The dispersion is saved as ams.png in you directory and is shown below,

```
python3   postQ.py
```

OpenMX + TB2J + UppASD