Hands-on introduction of OpenMX
- Practice -

Ozaki group
Mitsuaki Kawamura

Schedule:
14:30～15:30 Lecture
15:50～17:20 Practice

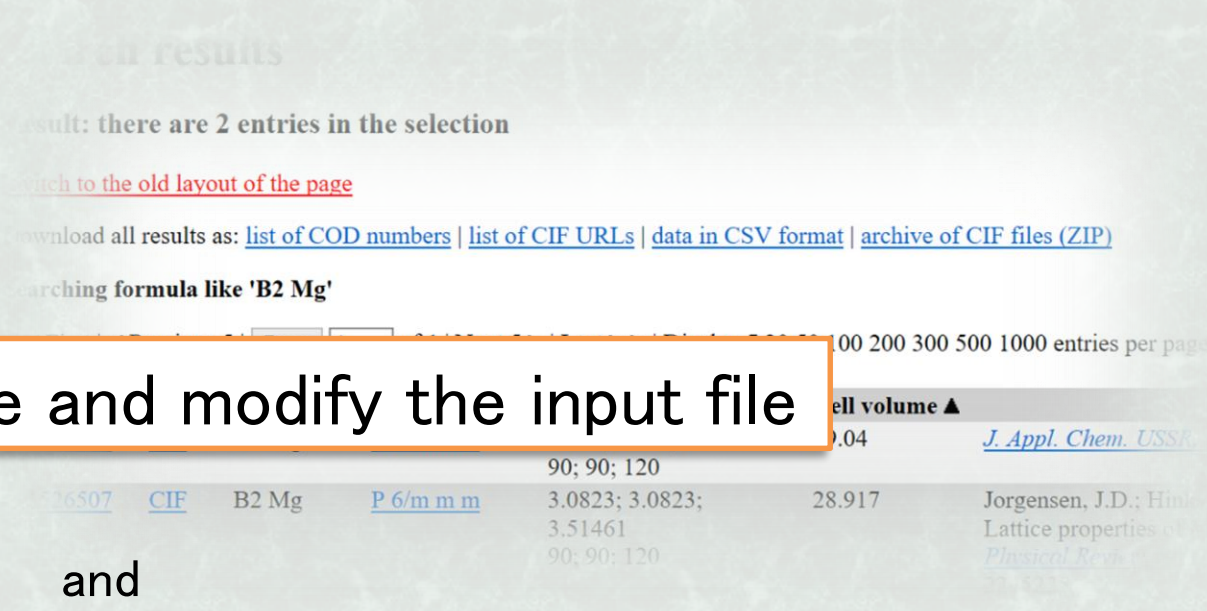
2018/7/2
ISS2018 1st part

Outline

- Introduction
- Computer in this tutorial
  - Configuration
  - Login
- Input file generation
- Job submit
- Check results
- Post processing and visualization
  - Band structure
  - Kohn–Sham orbitals
  - DOS, PDOS, Fermi surfaces
- Free excise

Introduction : Purpose

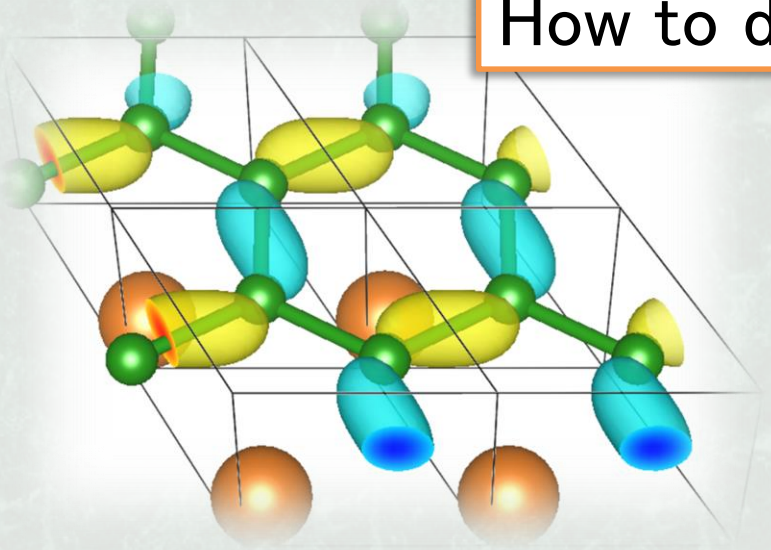
We will overview

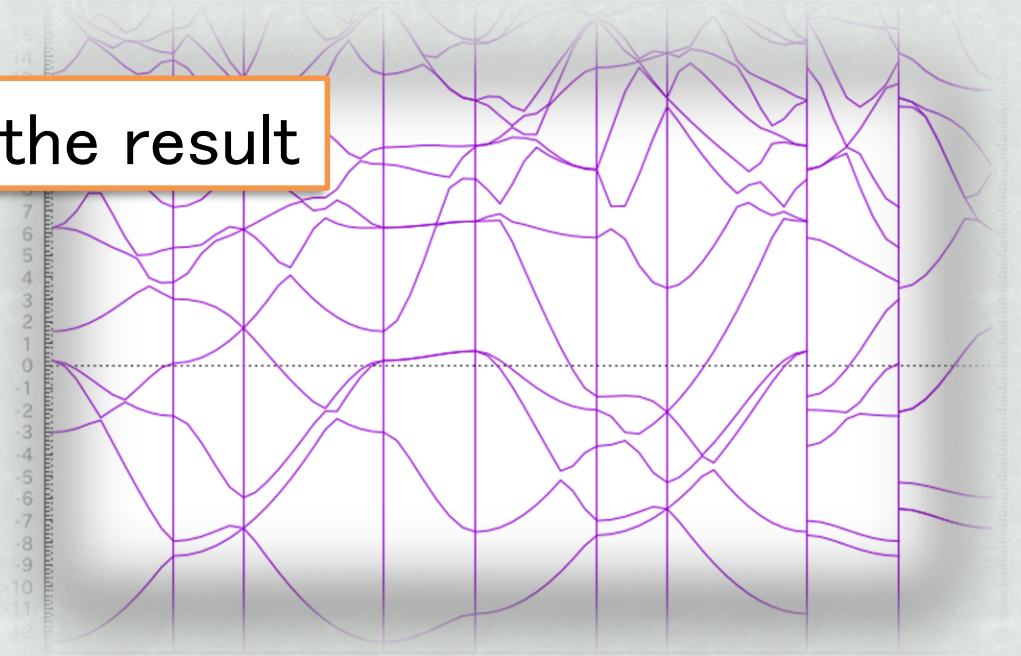


How to generate and modify the input file

and

How to display the result





Flow of this practice

Intro

Log-in workstation

Input file

Get CIF file from database → Convert cif to input file → View structure

SCF

Submit job for SCF → Check the result

Band structure

Modify input for band → Submit job → Plot band structure

Kohn-Sham orbital

Modify input for displaying orbital → Submit job → Plot Kohn-Sham orbital

DOS and Fermi surface

Modify input for DOS → Submit job → Interactive queue for DOS

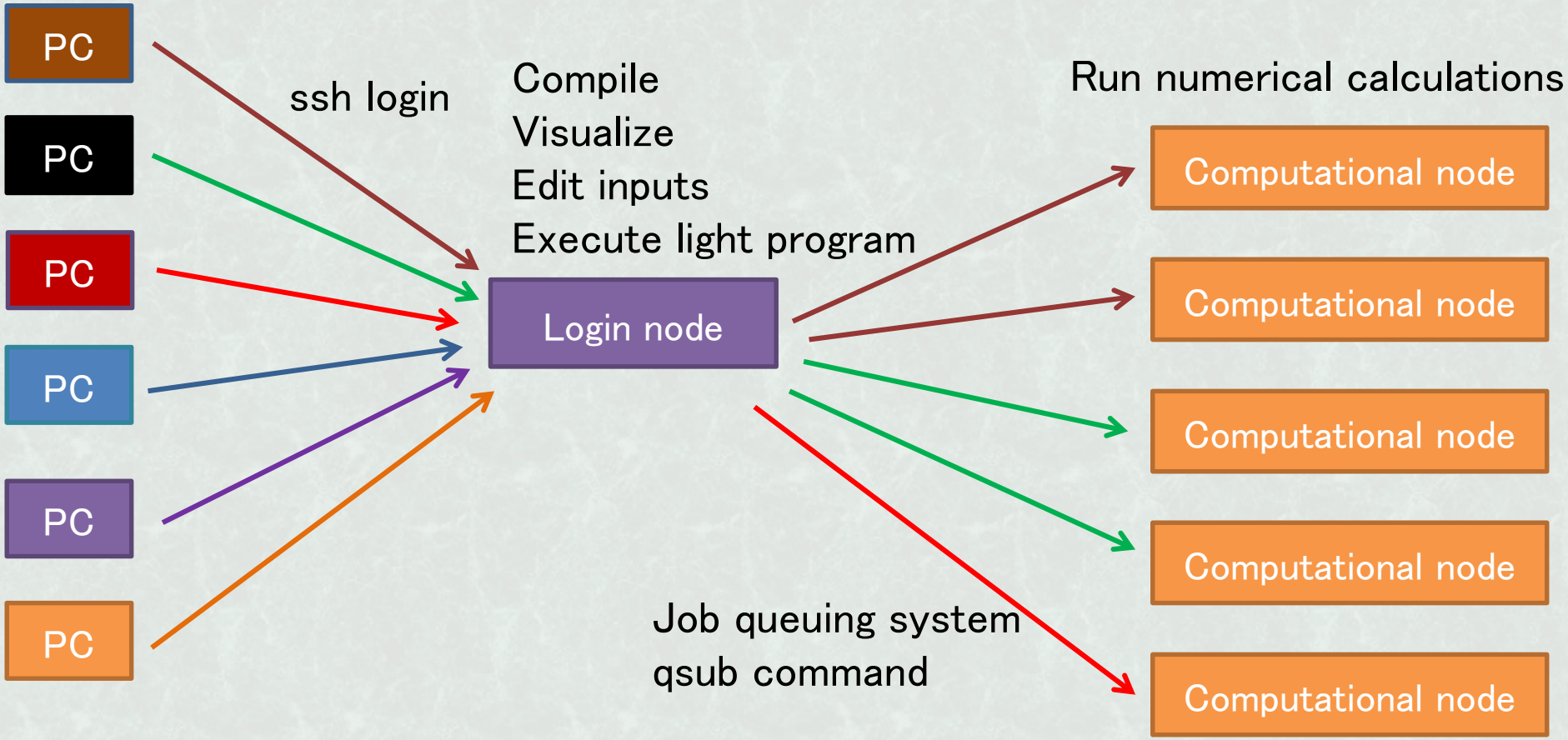
→ Plot DOS and Fermi surface

Free exercise time

Login

Computer for this summer school

CPU : Intel Xeon Broadwell 14cores × 2 (28 cores/node)
RAM : 64 GB / node
Compiler : Intel compiler & MKL library
Utilities : /home/public/bin/ (already added to PATH)



Login

Login to workstation

Case 1: Use ssh command (Linux, UNIX, macOS, WSL)

```
$ ssh -Y user-name@server-name
```

Your PC may ask you to add this workstation to “known-hosts”.

Enter your password.

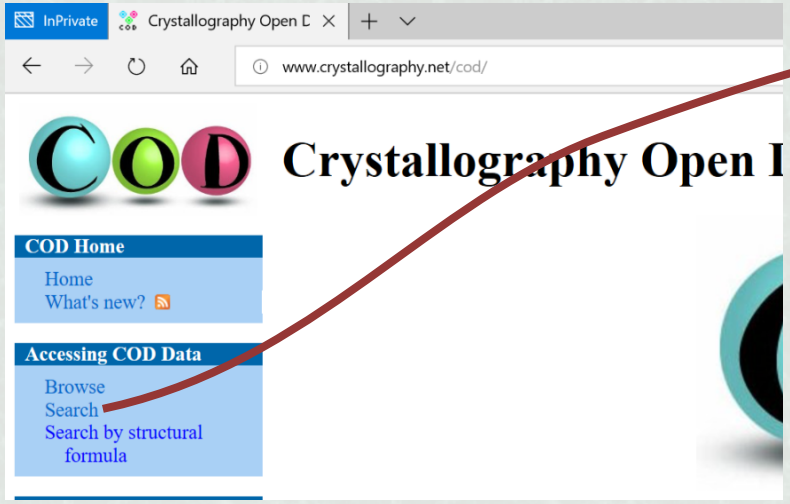
Case 2: GUI ssh-client in Windows

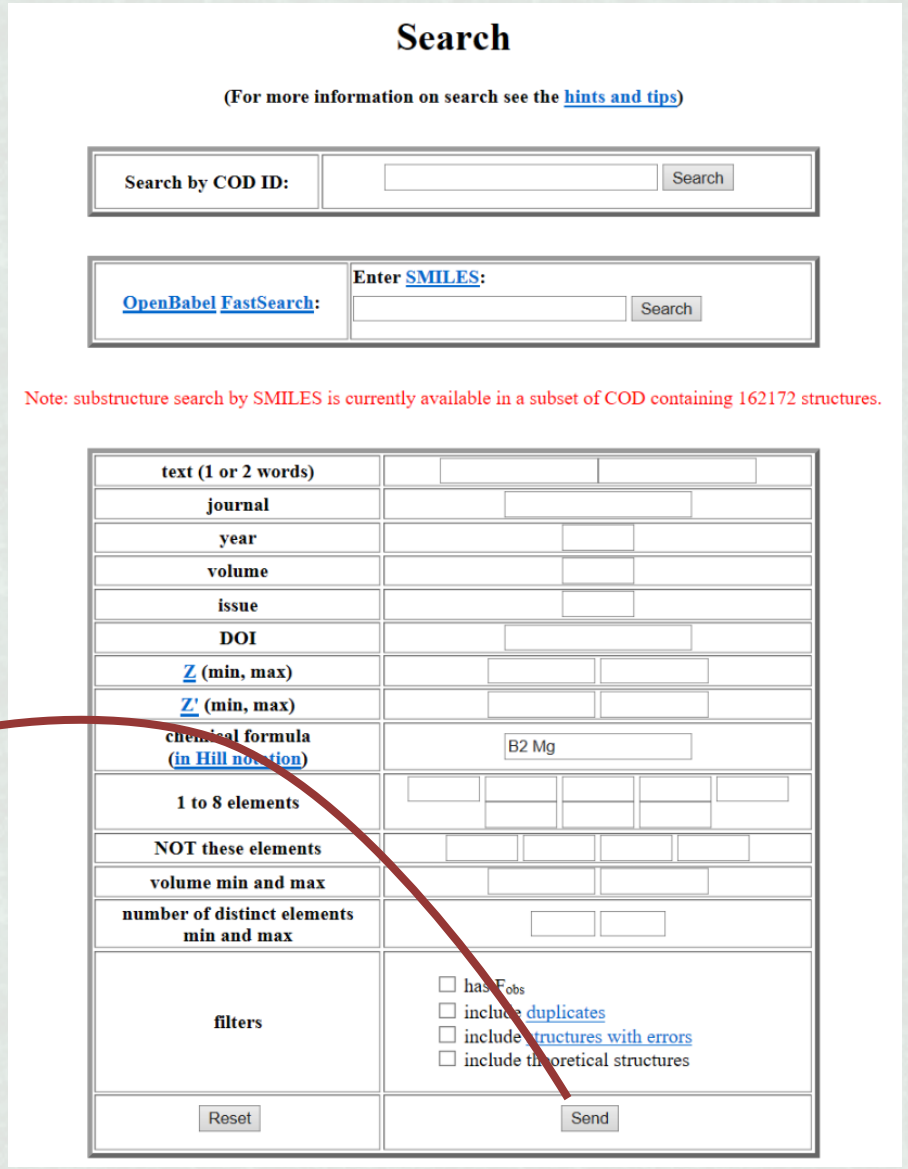
Google “xming putty”

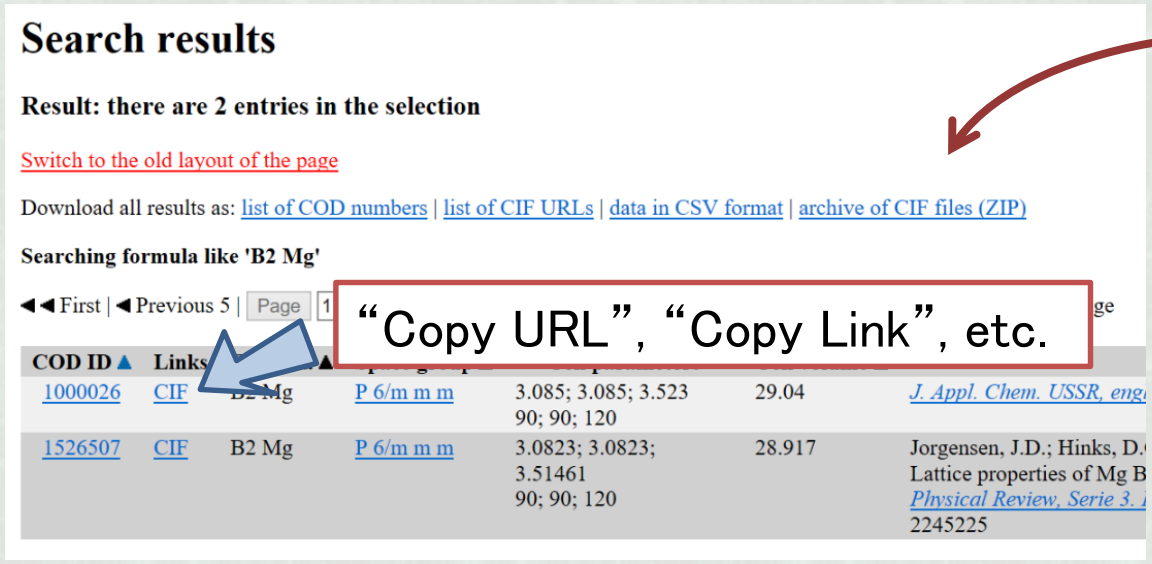
WinSCP is also required in this case.

Get CIF file from COD

Google “Crystallography Open Database”







“Copy URL”, “Copy Link”, etc.

“B2 Mg” into “chemical formula”
Alphabetic order

In the workstation:

```
$ wget paste-url-here
$ fixcod.sh 1000026.cif
$ cif2input.py 1000026.cif mgb2
```

→ Input file “openmx.in” is generated.

```
$ VESTA 1000026.cif
```

Queuing system

```
$ cp /home/public/iss2018/openmx.sh .
$ qsub openmx.sh
```

MPI processes per node

```
#!/bin/sh
#PBS -l nodes=1:ppn=7
#
source ~/.bashrc
#
export OMP_NUM_THREADS=4
#
cd $PBS_O_WORKDIR
#
mpiexec -hostfile $PBS_NODEFILE /home/public/openmx openmx.in -nt $OMP_NUM_THREADS ¥
   > openmx.out
```

Number of OpenMP threads

ppn × OMP_NUM_THREADS ≦ number of cores par node (28)

Job status

```
$ qstat -a

gauss:

                                                                  Req'd    Req'd       Elap
Job ID                  Username    Queue    Jobname          SessID  NDS   TSK    Memory   Time    S   Time
----------------------- ----------- -------- ---------------- ------ ----- ------ ------ --------- - ---------
168.gauss               kawamura    default  openmx.sh         23466     1      7    --         --  R  00:00:00
```

Nodes

R: Running

Q: Waiting to start

```
$ qdel job-ID
```

Stop job

Check results

```
$ less mgb2.out
$ less openmx.out
```

```
*******************************************************
*******************************************************
                  SCF history at MD= 1
*******************************************************
*******************************************************

   SCF=   1  NormRD=  1.000000000000  Uele= -14.025399913227
   SCF=   2  NormRD=  0.589538957582  Uele= -14.093889873889
   SCF=   3  NormRD=  0.249943571587  Uele= -14.122239227622
                          :
   SCF=  16  NormRD=  0.000350260351  Uele= -14.150623623150
   SCF=  17  NormRD=  0.000216964658  Uele= -14.150651456947
   SCF=  18  NormRD=  0.000134457940  Uele= -14.150668805936
   SCF=  19  NormRD=  0.000083360095  Uele= -14.150679594319
   SCF=  20  NormRD=  0.000051699263  Uele= -14.150686290499
   SCF=  21  NormRD=  0.000032833716  Uele= -14.150689700063
   SCF=  22  NormRD=  0.000020911680  Uele= -14.150692116404
   SCF=  23  NormRD=  0.000000083022  Uele= -14.150697139160
   SCF=  24  NormRD=  0.000000081048  Uele= -14.150697139607
```

Band structure

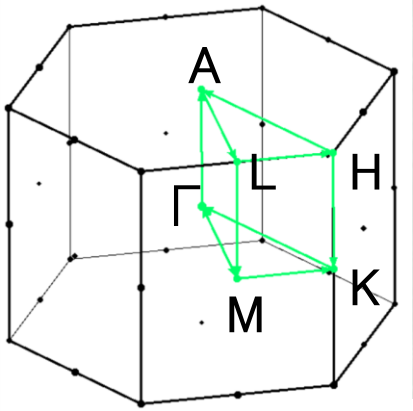
Edit “openmx.in” with emacs, vim, etc.

```
scf.restart      off
orbitalOpt.Force.Skip    off
Band.dispersion          off
```

Line 10
Line 100
Line 137

```
scf.restart      on
orbitalOpt.Force.Skip    on
Band.dispersion          on
```

```
Band.Nkpath  9
<Band.kpath
11  0.000000 0.000000 0.000000 0.500000 0.000000 0.000000 GAMMA M
6  0.500000 0.000000 0.000000 0.333333 0.333333 0.000000 M K
13  0.333333 0.333333 0.000000 0.000000 0.000000 0.000000 K GAMMA
8  0.000000 0.000000 0.000000 0.000000 0.000000 0.500000 GAMMA A
11  0.000000 0.000000 0.500000 0.500000 0.000000 0.500000 A L
6  0.500000 0.000000 0.500000 0.333333 0.333333 0.500000 L H
13  0.333333 0.333333 0.500000 0.000000 0.000000
8  0.500000 0.000000 0.500000 0.500000 0.000000
8  0.333333 0.333333 0.500000 0.333333 0.333333
Band.kpath>
```



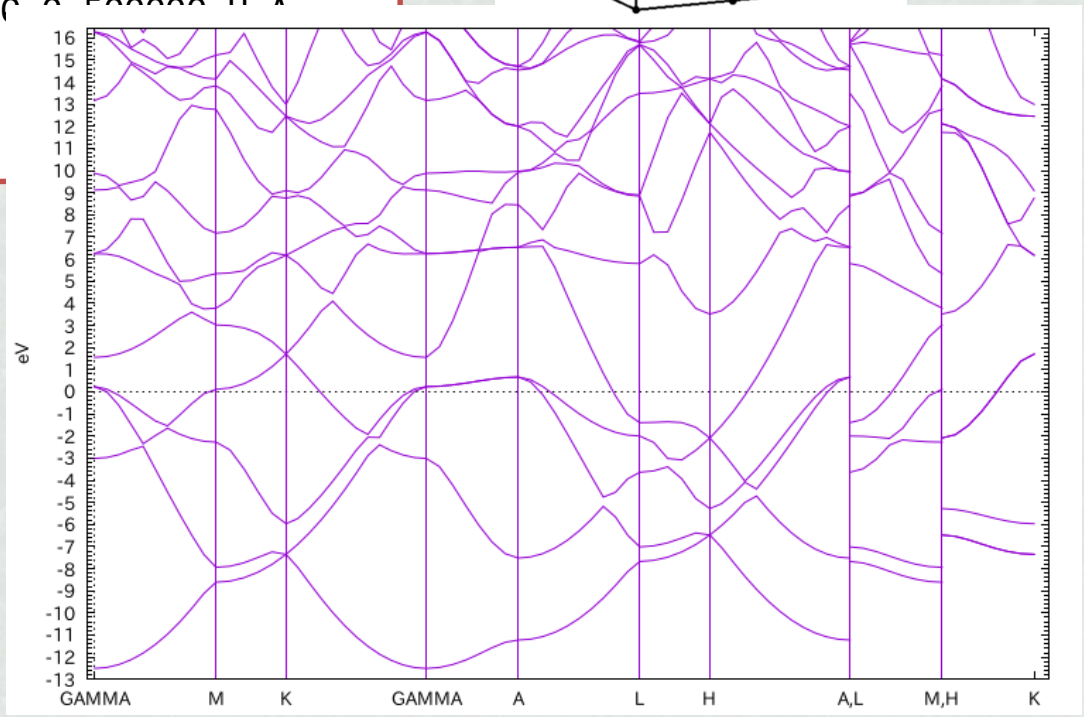

```
$ qsub openmx.sh
```

mgb2.Band

```
$ bandgnu13 mgb2.Band
```

mgb2.GNUBAND

```
$ gnuplot mgb2.GNUBAND
```

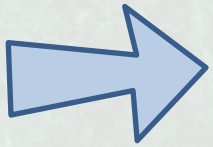


Kohn-Sham orbitals

Modify openmx.in

```
Band.dispersion     on
MO.fileout          off
num.LUMOs   1
```

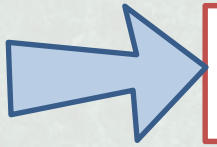
Line 137
Line 311
Line 313



```
Band.dispersion     off
MO.fileout          on
num.LUMOs   2
```

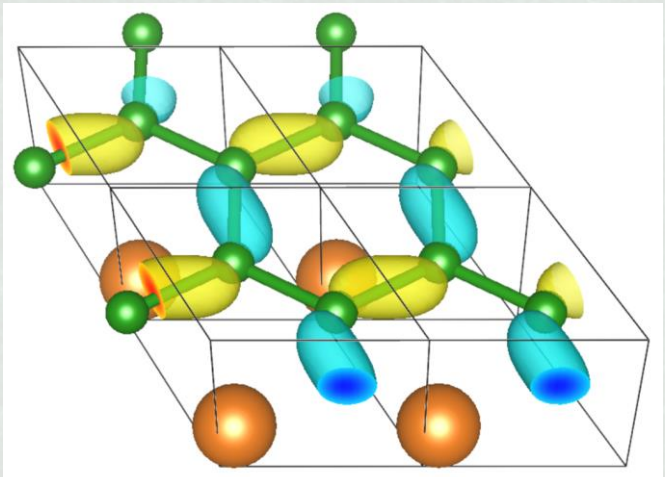
```
num.HOMOs   1
num.LUMOs   2
MO.Nkpoint   1
<MO.kpoint
0.0   0.0   0.0
MO.kpoint>
```

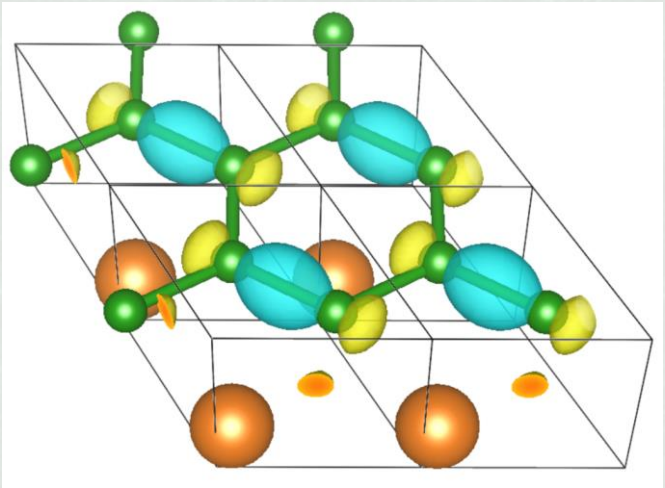
```
$ qsub openmx.sh
```

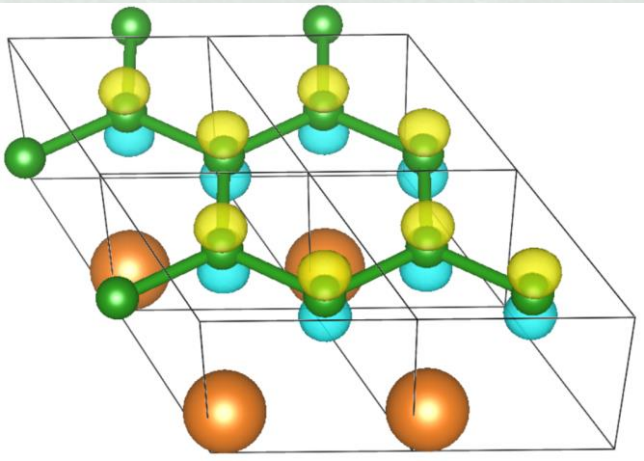


```
mgb2.homo0_0_0_i.cube   mgb2.lumo0_0_0_i.cube   mgb2.lumo0_0_1_i.cube
mgb2.homo0_0_0_r.cube   mgb2.lumo0_0_0_r.cube   mgb2.lumo0_0_1_r.cube
```

```
$ cube2xsf *_r.cube
$ VESTA *_r.xsf
```







Fermi surface, DOS, PDOS (1)

```
MO.fileout      on
DOS.fileout     off
```

Line 311
Line 207

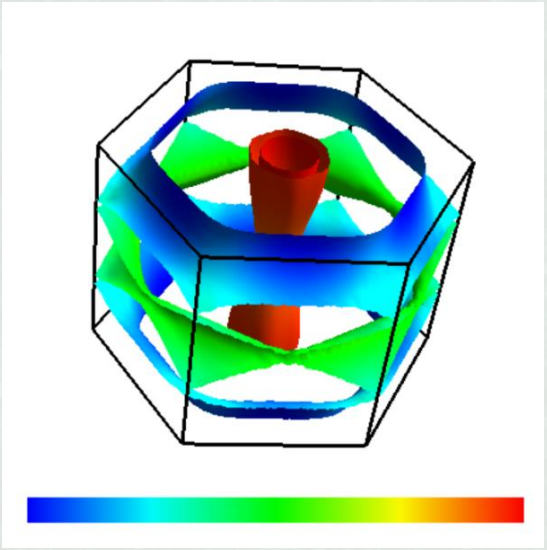
```
MO.fileout      off
DOS.fileout     on
```

```
Dos.Erange      -20.0  20.0
Dos.Kgrid       28 28 20
```

```
$ qsub openmx.sh
```

```
mgb2.Dos.val, mgb2.Dos.vec,
mgb2.FermiSurf_s0_a1.frmsf,
mgb2.FermiSurf_s0_a2.frmsf,
mgb2.FermiSurf_s0_a3.frmsf
```

```
$ fermisurfer mgb2.FermiSurf_s0_a2.frmsf
```

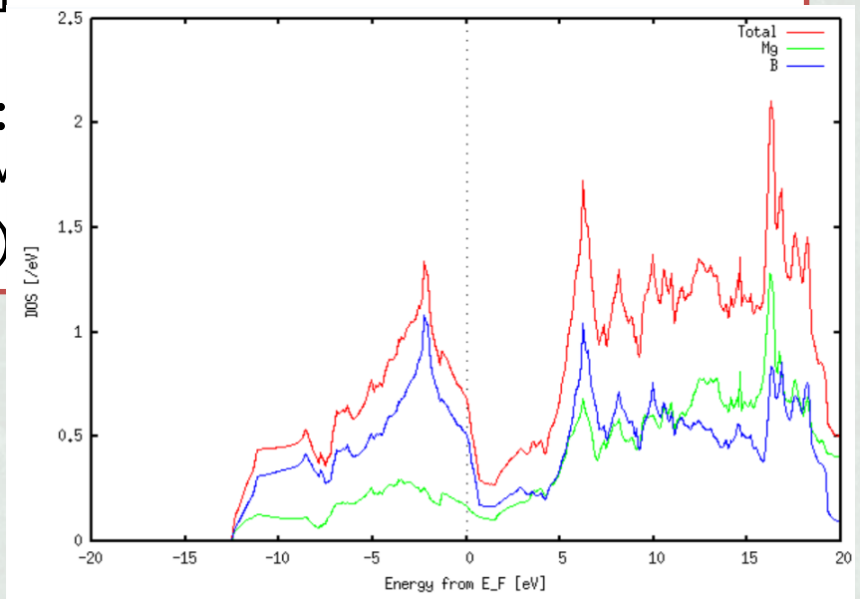


Fermi surface, DOS, PDOS (2)

Interactive queue

```
[XXX@gauss] $ qsub -I
[XXX@gaussXX] $ DosMain mgb2.Dos.val mgb2.Dos.vec
:
Which method do you use?, Tetrahedron(1), Gaussian Broadening(2)
1 ⏎
Do you want Dos(1) or PDos(2)?
2 ⏎
Which atoms for PDOS : (1,...,3), ex 1 2
1 2 3 ⏎
[XXX@gaussXX] $ exit
$ gnuplot
gnuplot> set yzeroaxis
gnuplot> set xlabel "Energy from E_F [eV]"
gnuplot> set ylabel "DOS [/eV]"
gnuplot> plot "mgb2.DOS.Tetrahedron" u 1:
 "mgb2.PDOS.Tetrahedron.atom1" w l tit "M
 "mgb2.PDOS.Tetrahedron.atom2" u 1:($2*2)
```

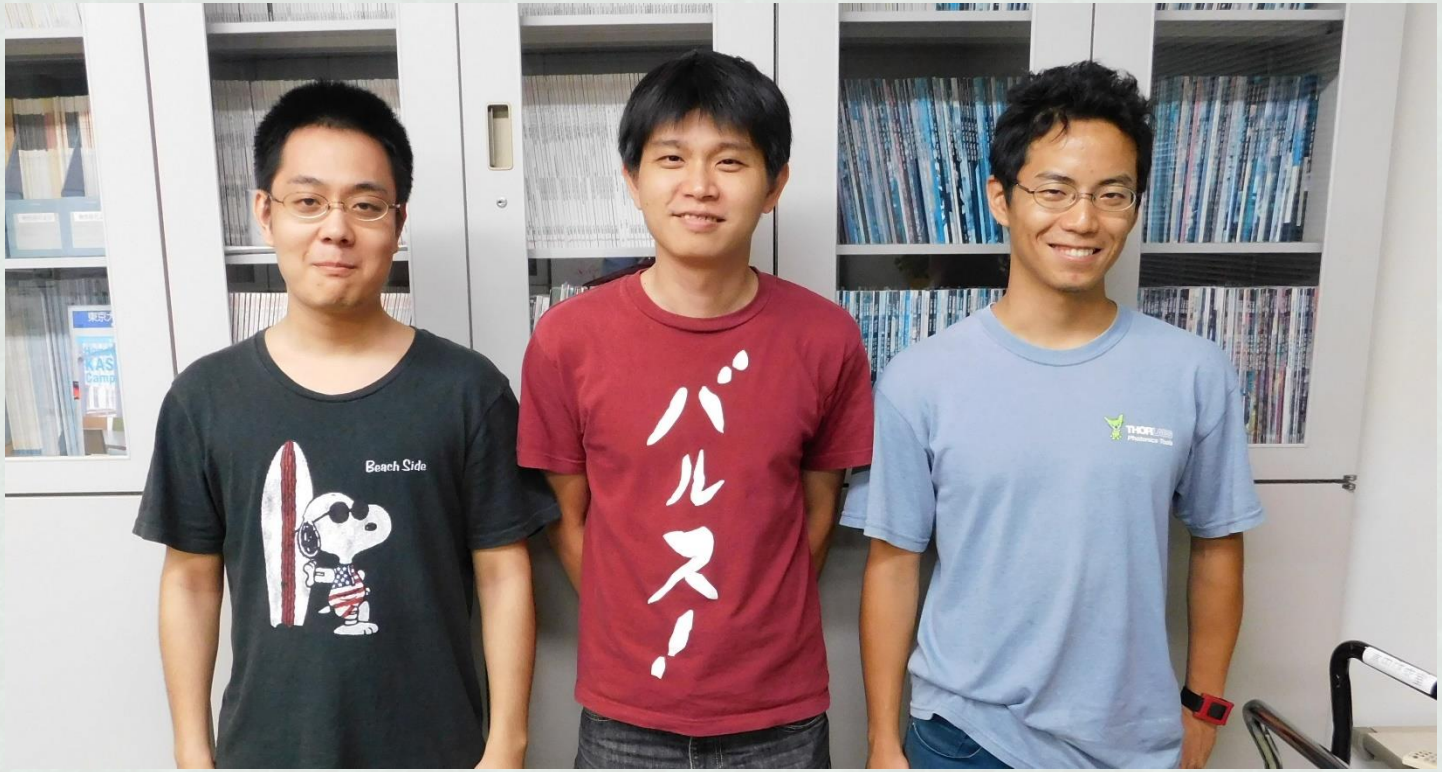
As well as DOS



Free exercise time

- Compute other materials from COD database. Explain the result of band structure, PDOS, Fermisurface.
- Install the following visualization tools into your own PC.
  - VESTA
  - XCrysDen
  - FermiSurfer
- Try other functions of OpenMX (Wannier function, etc.).
- Etc.

Tutors



Masahiro Fukuda　　Yung-Ting Lee　　Mitsuaki Kawamura