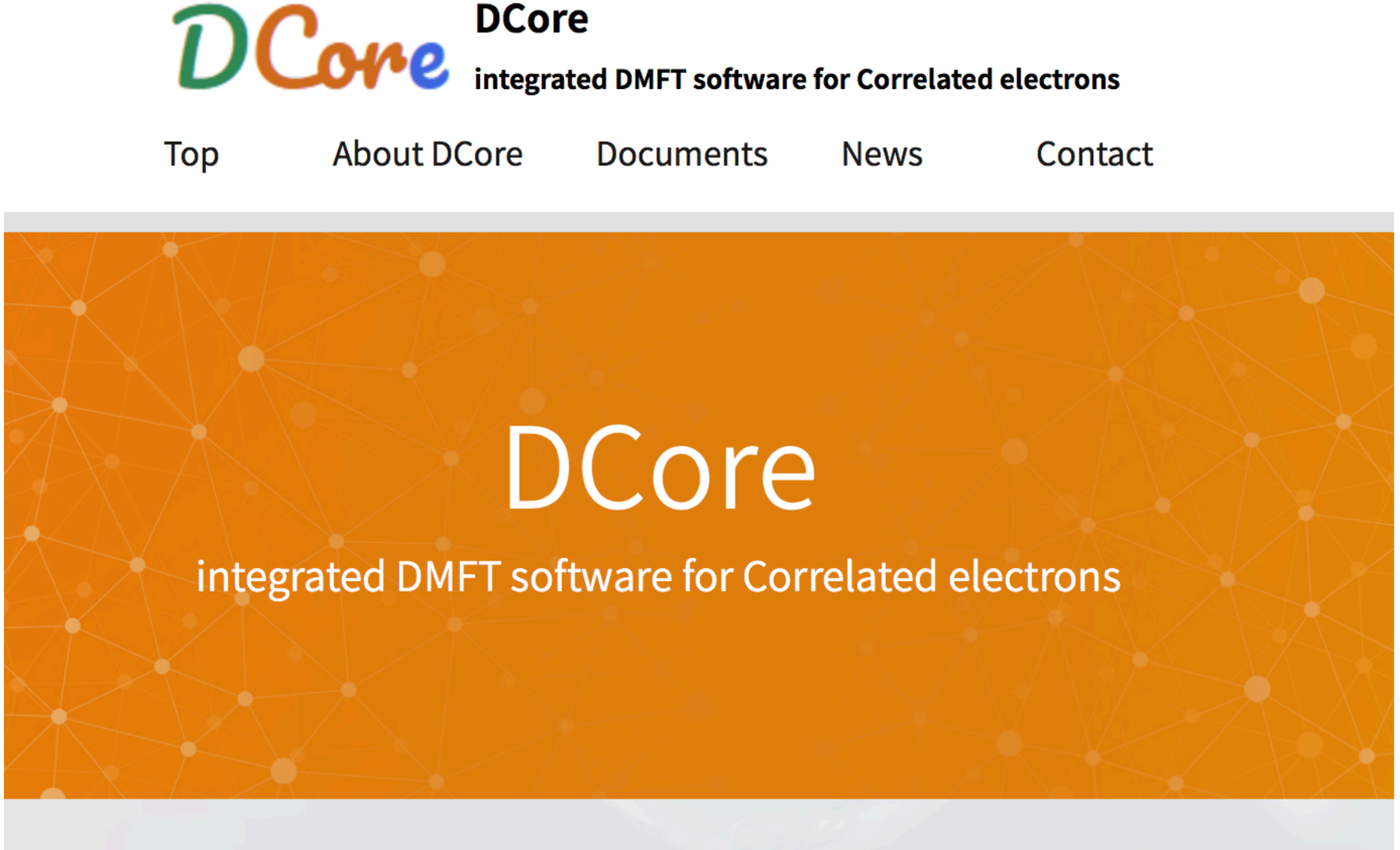
Tutorial for DCore

2019/12/26 Kazuyoshi Yoshimi
ISSP, Univ. of Tokyo



ref.) http://www.pasums.issp.u-tokyo.ac.jp/dcore/en/

Contents

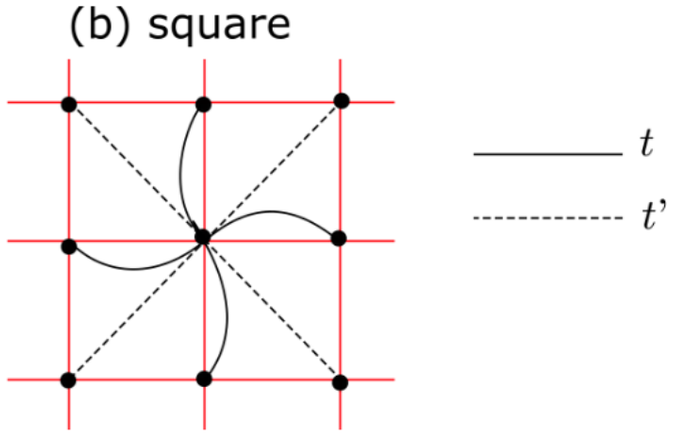
- 2D Hubbard Model
- Antiferromagnetic state in 3D Hubbard model
- DFT+DMFT

2D Hubbard model (1)

- Sample file : PathToDCore/doc/tutorial/square

[model] section

- About lattice



ref.) dmft_square.ini

```
[model]
seedname = square
lattice = square
norb = 1
nelec = 1.0
t = -1.0
kanamori = [(2.0, 0.0, 0.0)]
nk = 8
```

- About Interaction term

$$\mathcal{H}_{\mathrm{int}} = \frac{1}{2} \sum_{i,\alpha\beta\gamma\delta\sigma\sigma'} U^{i}_{\alpha\beta\gamma\delta} c^{\dagger}_{i\alpha\sigma} c^{\dagger}_{i\beta\sigma'} c_{i\delta\sigma'} c_{i\gamma\sigma}$$

Interaction = kanamori

$$U^{i}_{\alpha\alpha\alpha\alpha} = U_i, U^{i}_{\alpha\beta\alpha\beta} = U'_i (\alpha \neq \beta), U^{i}_{\alpha\beta\beta\alpha} = J_i (\alpha \neq \beta), U^{i}_{\alpha\alpha\beta\beta} = J_i (\alpha \neq \beta)$$

```
kanamori = [(U_1, U'_1, J_1), (U_2, U'_2, J_2), ... ]
```

define U, U', J at each correlated shell (default: ncorr = 1)

2D Hubbard model (2)

```
[system]
beta = 40.0                 Inverse temperature
n_iw = 1000                 Number of Matsubara frequencies
prec_mu = 0.001             Threshold for calculating chemical potential
[impurity_solver]           with the bisection method.
name = TRIQS/hubbard-I
```

```
[control]
max_step = 5                Maximum steps of DMFT loops
sigma_mix = 1.0             Mixing parameter for self-energy
[tool]                      → for dcore_post
broadening = 0.4            An additional Lorentzian broadening
knode = [(G,0.0,0.0,0.0),(X,0.5,0.0,0.0),(M,0.5,0.5,0.0),(G,0.0,0.0,0.0)]
nk_line = 100               Number of k along each line
omega_max =6.0              Max value of real frequency
omega_min =-5.0             Minimum value of real frequency
Nomega = 400                Number of real frequencies
[mpi]
command = “mpijob”          Command for executing a MPI job.
                            # will be relaced by the number of processes.
```

2D Hubbard model (3)

- Execute on sekirei (finish in a few minutes)

```
$ mkdir square
$ cp -rf /home/issp/materiapps/dcore/dcore-2.0.2-1/examples/square .
$ cp /home/issp/materiapps/dcore/sample_jobscript/dcore.sh .
$ qsub dcore.sh
```

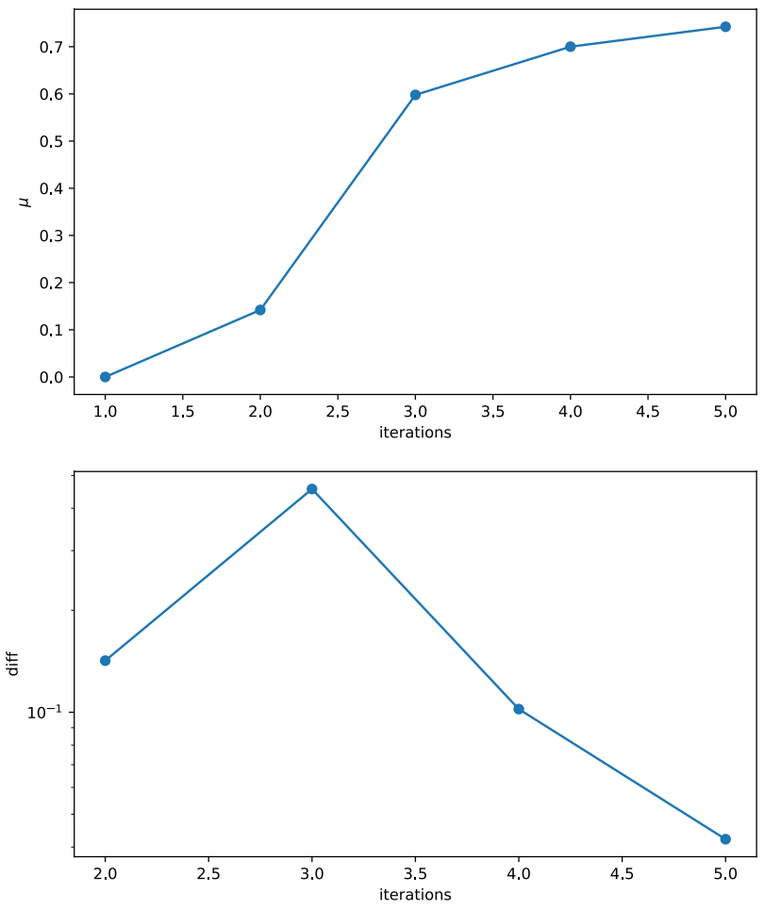
- Execution of DCore

	Output files
`$ dcore_pre dmft_square.ini`	→ square.h5
`$ dcore  dmft_square.ini --np 4`	→ square.out.h5
`$ dcore_check dmft_square.ini --ext=eps`	→ check/*.dat, *.eps
`$ dcore_post dmft_square.ini --np 4`	→ post/*.dat

2D Hubbard model (4)

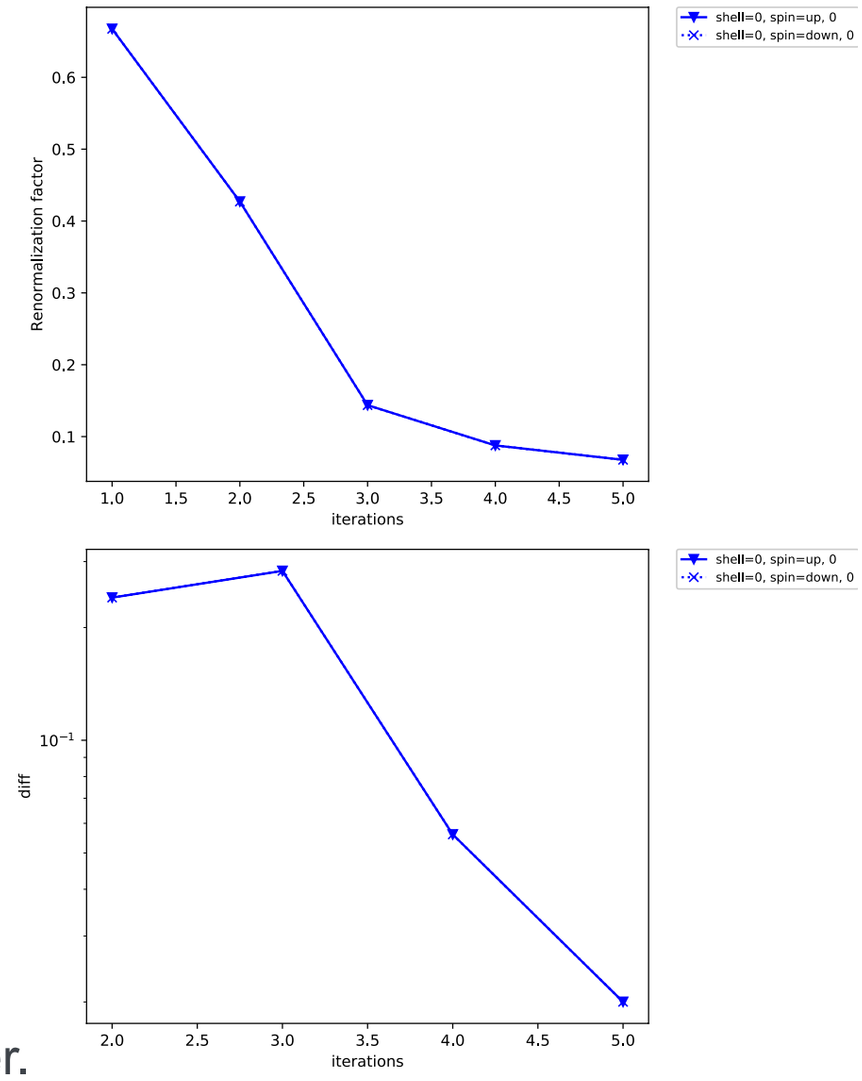
check/

iter_mu.eps



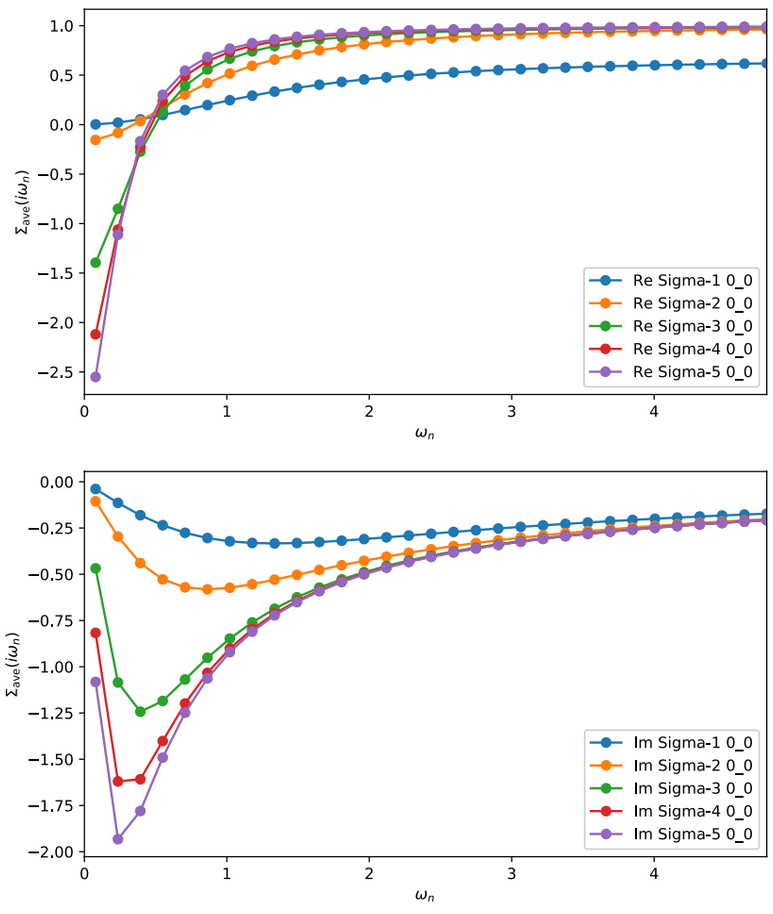
The chemical potential
as a function of iteration number.

iter_sigma-ish0.eps



The renormalization factor
as a function of iteration number.

sigma_ave.eps



The averaged self-energy
at the last seven iterations.

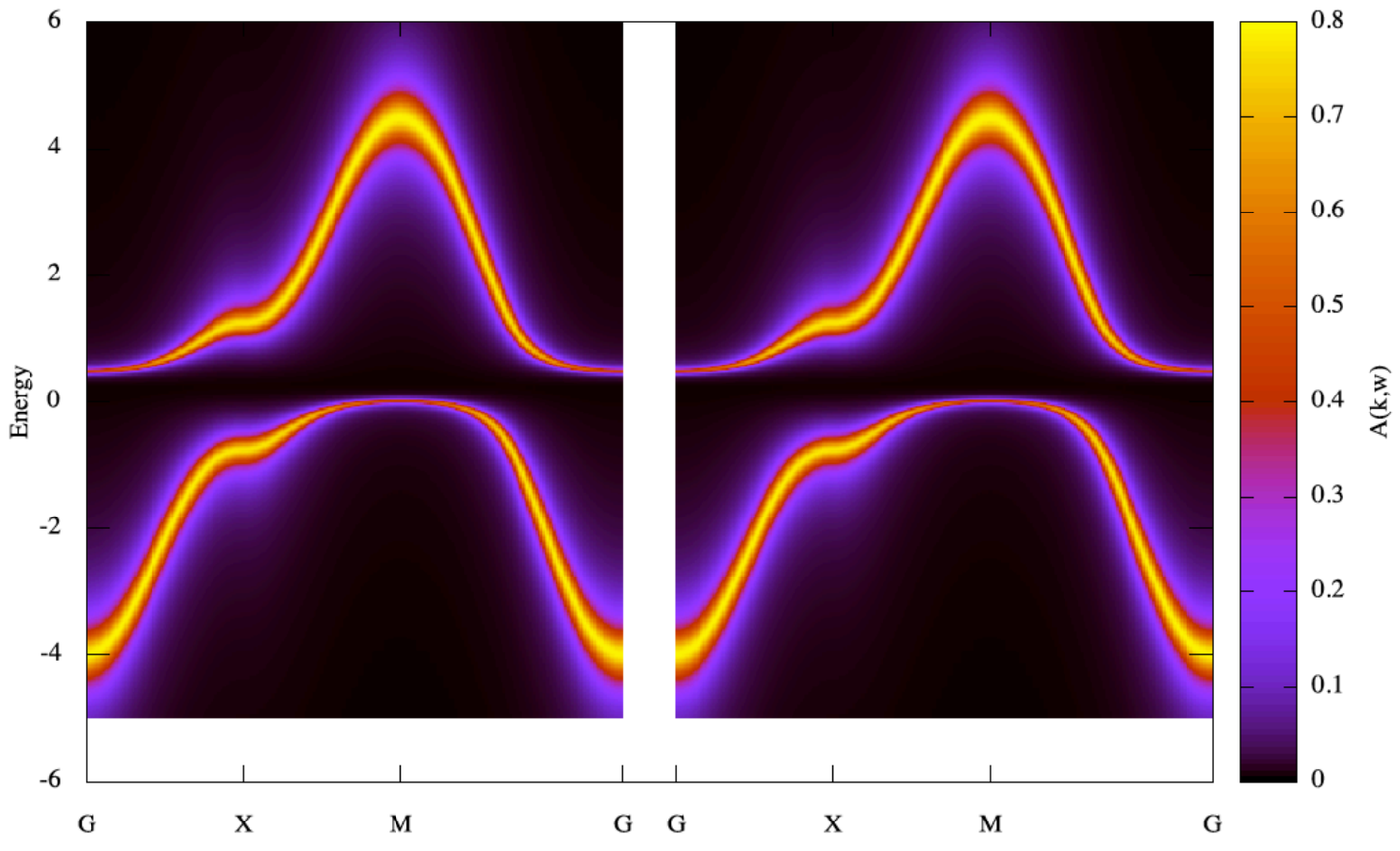
$$\Sigma_{\mathrm{Ave}}(i\omega_n) = \left[\sum_{i}^{\mathrm{shell}} \sum_{\alpha\beta}^{N^i_{\mathrm{orb}}} \Sigma_{\alpha\beta}(i\omega)\right] / \left[\sum_{i}^{\mathrm{shell}} N^i_{\mathrm{orb}}\right],$$

2D Hubbard model (5)

post/

square_akw.dat

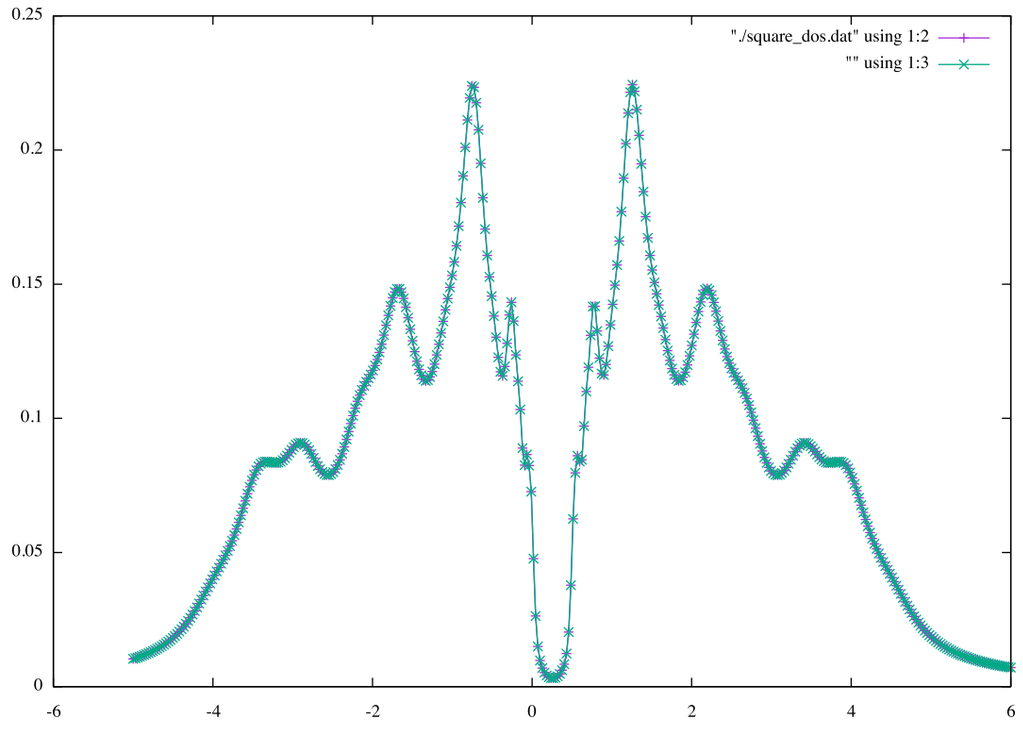
Momenum-resolved spectral function




```
$ gnuplot square_akw.gp
```

square_dos.dat

Density of States



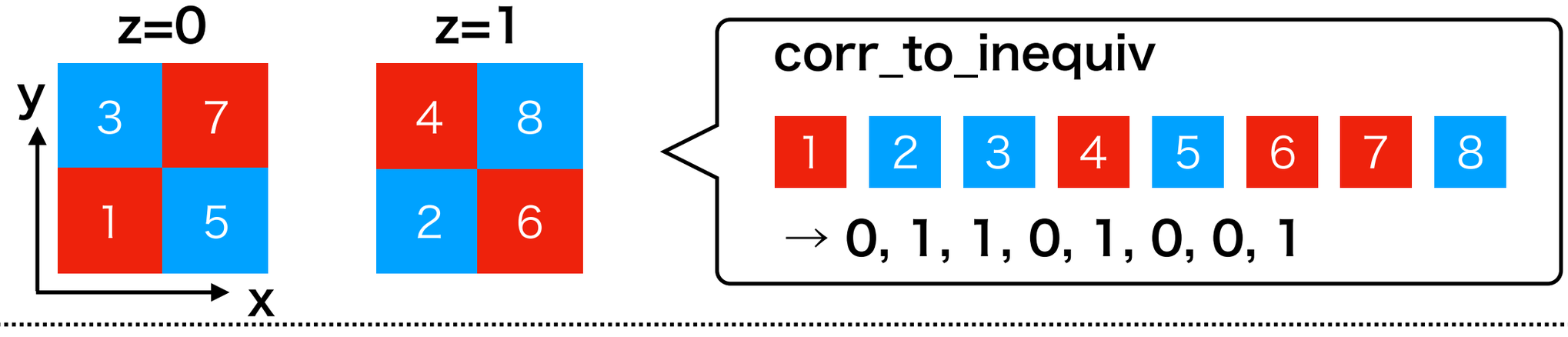
Antiferromagnetic state in 3D Hubbard model (1)

- Sample file : PathToDCore/doc/tutorial/afm

ref.) cubic.ini

```
[model]
seedname = cubic
lattice = wannier90
ncor = 8
#Numbers of orbitals on two inequivalent shells
norb = 1, 1
nelec = 8.0
kanamori = [(10.0, 0.0, 0.0), (10.0, 0.0, 0.0)]
#Mapping from correlated shells to two inequivalent shells
corr_to_inequiv =  0, 1, 1, 0, 1, 0, 0, 1
nk = 16
```

ncor = 8 (2 inequivalent shells)



Antiferromagnetic state in 3D Hubbard model (2)

Wannier90 format

- **Line 1**
  File Header
- **Line 2**
  Total number of wannier functions.
- **Line 3**
  Total number of super cells `nrpts`
- **Line 4- Line 5 + int(nrpts/15)**
  The degeneracy at each super cell (basically, the number is set as 1).
- **Line 6 + int(nrpts/15) -**
  1-3rd columns: The supercell lattice vectors.
  4-th column: The index of wannier functions at the original cell.
  5-th column: The index of wannier functions at the supercell.
  6(7)-th column: The real (imaginary) value.

```
wannier90 format for mvmcdry
8
343
1    1    1    1    1    1    1    1    1    1    1    1    1
...
 -3   -3   -3    1    1   0.0004104251  -0.0000000000
 -3   -3   -3    1    2   0.0001515941  -0.0000000006
 -3   -3   -3    1    3  -0.0001515941   0.0000000002
```

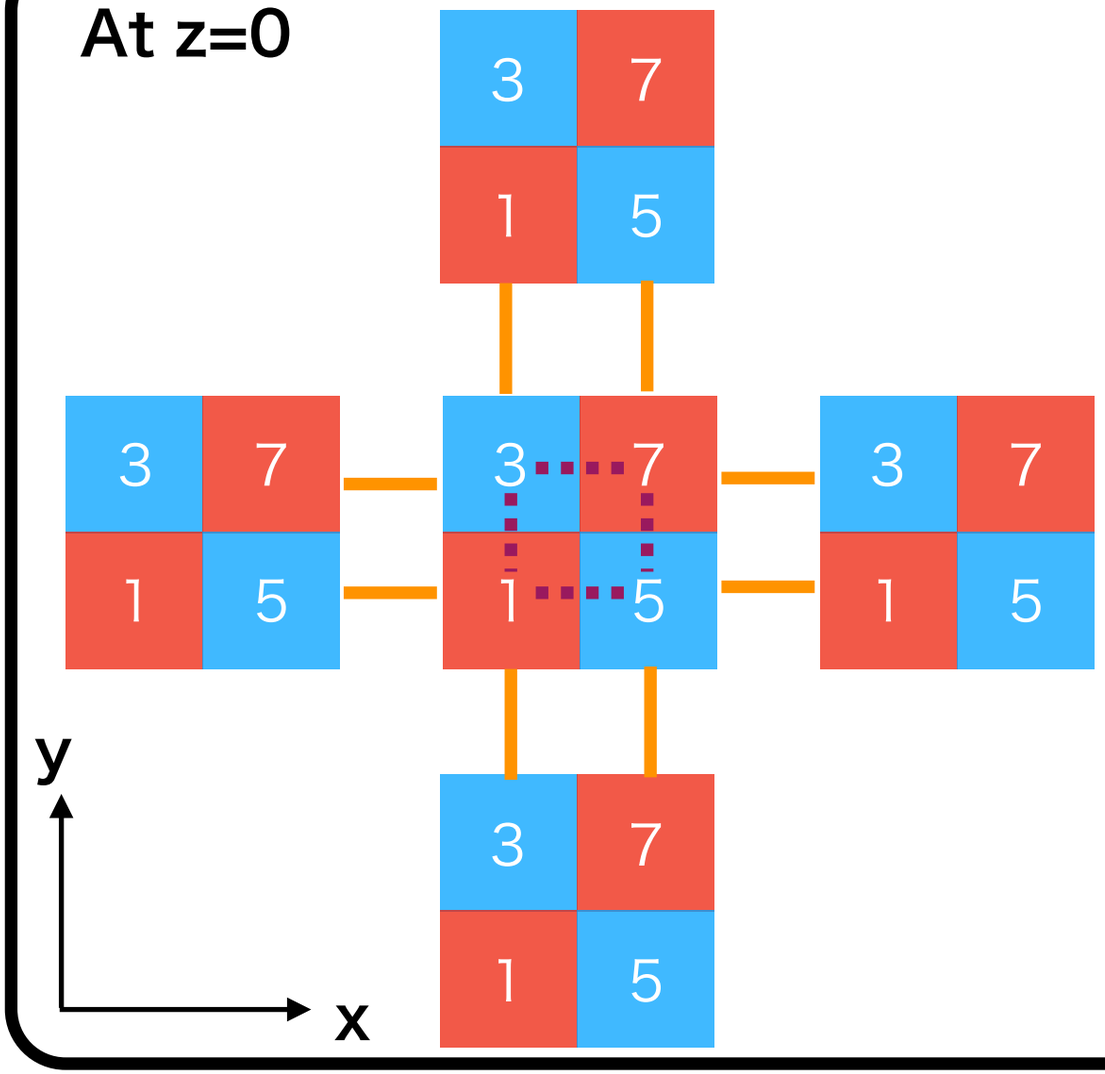
(See details in the `8.19 seedname_hr.dat` in the user_guide for wannier90.)
ref.) http://issp-center-dev.github.io/HPhi/manual/master/en/html/wannier/format.html

Antiferromagnetic state in 3D Hubbard model (3)

Make cubic_hr.dat before running DCore.

```
$ python mk_hr.py → cubic_hr.dat
```

At z=0



In cubic_hr.dat

In unit cell

```
0 0 0 1 3 -1.0 0.0
0 0 0 3 1 -1.0 0.0
0 0 0 3 7 -1.0 0.0
0 0 0 7 3 -1.0 0.0
0 0 0 7 5 -1.0 0.0
0 0 0 5 7 -1.0 0.0
0 0 0 1 5 -1.0 0.0
0 0 0 5 1 -1.0 0.0
```

Super cell

```
-1  0 0  1 5  -1.0 0.0
 1  0 0  5 1  -1.0 0.0
-1  0 0  3 7  -1.0 0.0
 1  0 0  7 3  -1.0 0.0
 0  1 0  3 1  -1.0 0.0
 0 -1 0  1 3  -1.0 0.0
 0  1 0  7 5  -1.0 0.0
 0 -1 0  5 7  -1.0 0.0
```

Antiferromagnetic state in 3D Hubbard model (4)

```
[system]
beta = 5.0                      Inverse temperature
#half filling (mu=U/2)
mu = 5.0                        Chemical potential
fix_mu = True                   Fix chemical potential

[impurity_solver]
name = ALPS/cthyb
timelimit{int} = 60             The time (in units of seconds)
exec_path{str} = hybmat         which the program should run before exiting.
```

[impurty_solver] specifies an impurity solver to be used and necessary parameters for running the solver program.
For ALPS/cthyb, see https://github.com/ALPSCore/CT-HYB/wiki/Input-parameters

Antiferromagnetic state in 3D Hubbard model (5)

Generate initial guess for self-energy

```
[control]
max_step = 5
initial_static_self_energy = {0: 'init_se_up.txt', 1: ‘init_se_down.txt'}
```

Make init_se_up.txt and init_se_down.txt before running DCore.

```
$ python mk_init_se.py → init_se_up.txt, init_se_down.txt
```

init_se_up.txt

```
# spin orb1 orb2   Re Im
0 0 0 1.0 0.0
1 0 0 -1.0 0.0
```

init_se_down.txt

```
# spin orb1 orb2   Re Im
0 0 0 -1.0 0.0
1 0 0 1.0 0.0
```

Antiferromagnetic state in 3D Hubbard model (6)

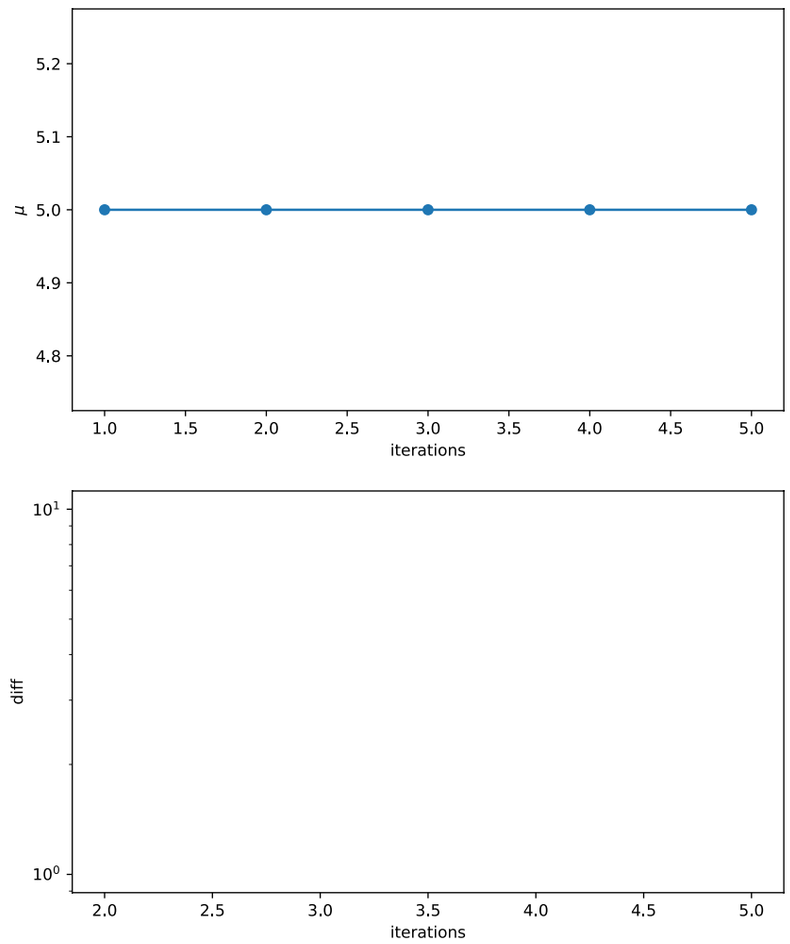
- Execute on sekirei (about 15 minutes)

  example of job script

```
#!/bin/sh
#QSUB -queue i18cpu
#QSUB -node  4
#QSUB -mpi   96
#QSUB -omp   1
#QSUB -place pack
#QSUB -over false
#PBS -l walltime=00:30:00
#PBS -N dcore

## https://issp-center-dev.github.io/DCore/tutorial/square/square.html
source /home/issp/materiapps/dcore/dcorevars.sh
source /home/issp/materiapps/alpscore-cthyb/alpscore-cthybvars.sh
cd ${PBS_O_WORKDIR}
dcore_pre cubic.ini
dcore  cubic.ini --np 96
dcore_check cubic.ini --ext=eps
dcore_post cubic.ini --np 96
```

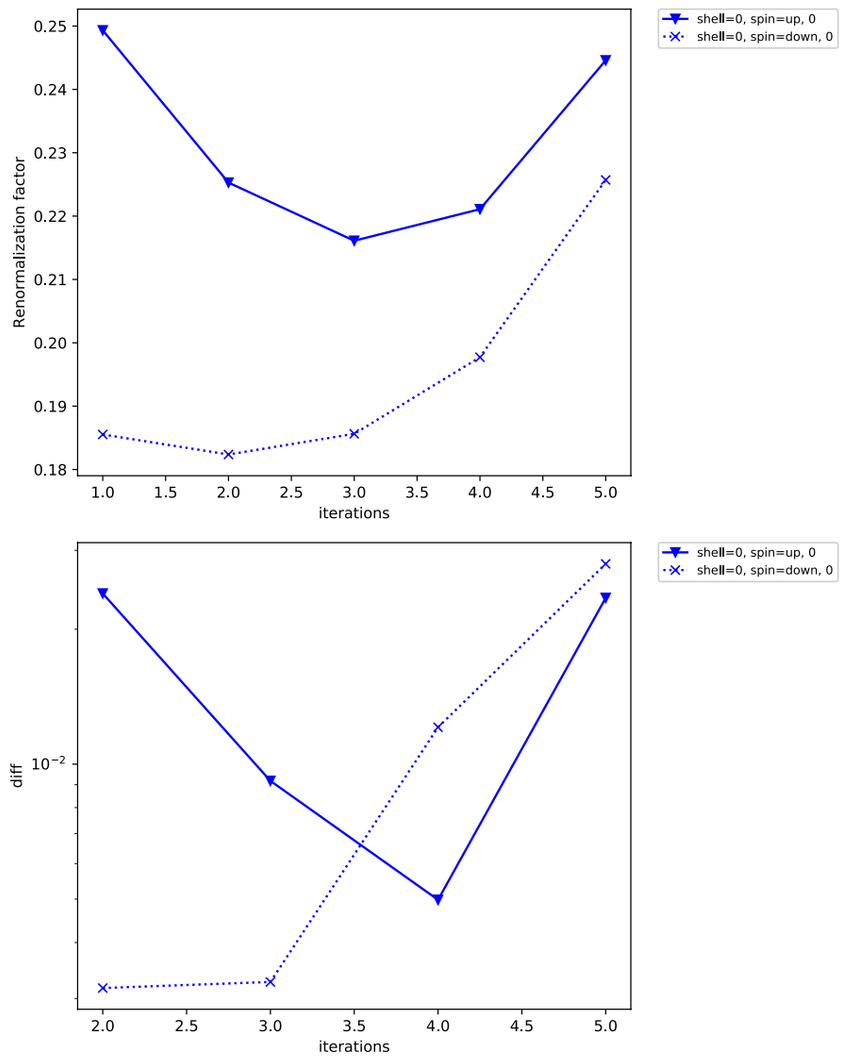
Antiferromagnetic state in 3D Hubbard model (7)

check/iter_mu.eps



Chemical potential is fixed.

check/iter_sigma-ish0.eps



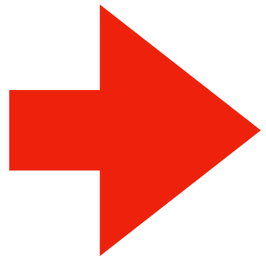
not sufficiently converged

DFT+DMFT

Using wannier90 format, we can treat realistic materials.

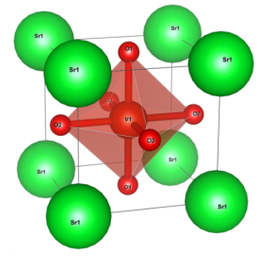
First principles calculation

Wannier90



DMFT calculation

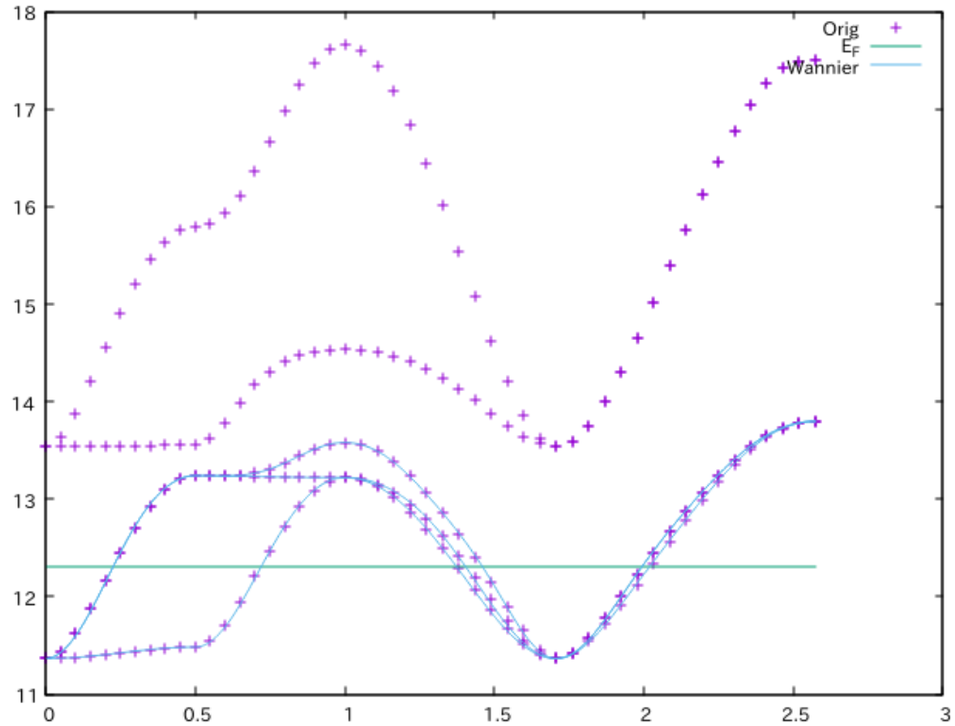
- ex.) $SrVO_3$
  Sample file : PathToDCore/doc/tutorial/srvo3



$SrVO_3$(1)

0. Construction of Wannier functions

Maximally localized Wannier functions
for the t_{2g} manifold can be constructed by using DFT code.



The procedure of Wanniernization is detailed in
Wanniernization using Quantum ESPRESSO.

$SrVO_3$(2)

DMFT calculation

ref.) srvo3.ini

```
[model]
lattice = wannier90
seedname = srvo3
nelec = 1.0
ncor = 1
norb = 3
kanamori = [(3.419, 2.315, 0.530)]
bvec=[(1.627091,0.0,0.0),(0.0,1.627091,0.0),(0.0,0.0,1.627091)]
nk0 = 10
nk1 = 10
nk2 = 10
```

Interaction = kanamori

$$\mathcal{H}_{\rm int} = \frac{1}{2} \sum_{i,\alpha\beta\gamma\delta\sigma\sigma'} U^i_{\alpha\beta\gamma\delta} c^\dagger_{i\alpha\sigma} c^\dagger_{i\beta\sigma'} c_{i\delta\sigma'} c_{i\gamma\sigma}$$

$$U^i_{\alpha\alpha\alpha\alpha} = U_i, U^i_{\alpha\beta\alpha\beta} = U'_i(\alpha \neq \beta), U^i_{\alpha\beta\beta\alpha} = J_i(\alpha \neq \beta), U^i_{\alpha\alpha\beta\beta} = J_i(\alpha \neq \beta)$$

```
kanamori = [(U_1, U'_1, J_1), (U_2, U'_2, J_2), ... ]
```

define U, U’, J at each correlated shell (default: ncorr = 1)

$SrVO_3$(3)

```
[system]
beta = 40.0
mu = 12.290722
with_dc = True

[impurity_solver]
name = ALPS/cthyb
timelimit{int} = 300
exec_path{str} = hybmat

[control]
max_step = 15
time_reversal = True
sigma_mix = 0.8

[tool]
broadening = 0.1
nk_line = 50
nnode = 5
knode=[(G,0.0,0.0,0.0),(X,0.5,0.0,0.0),(M,0.5,0.5,0.0),(G,
0.0,0.0,0.0),(R,0.5,0.5,0.5)]
omega_max =2.0
omega_min =-2.0
Nomega = 400
omega_check = 30.0
omega_pade = 2.0
```

$SrVO_3$(4)

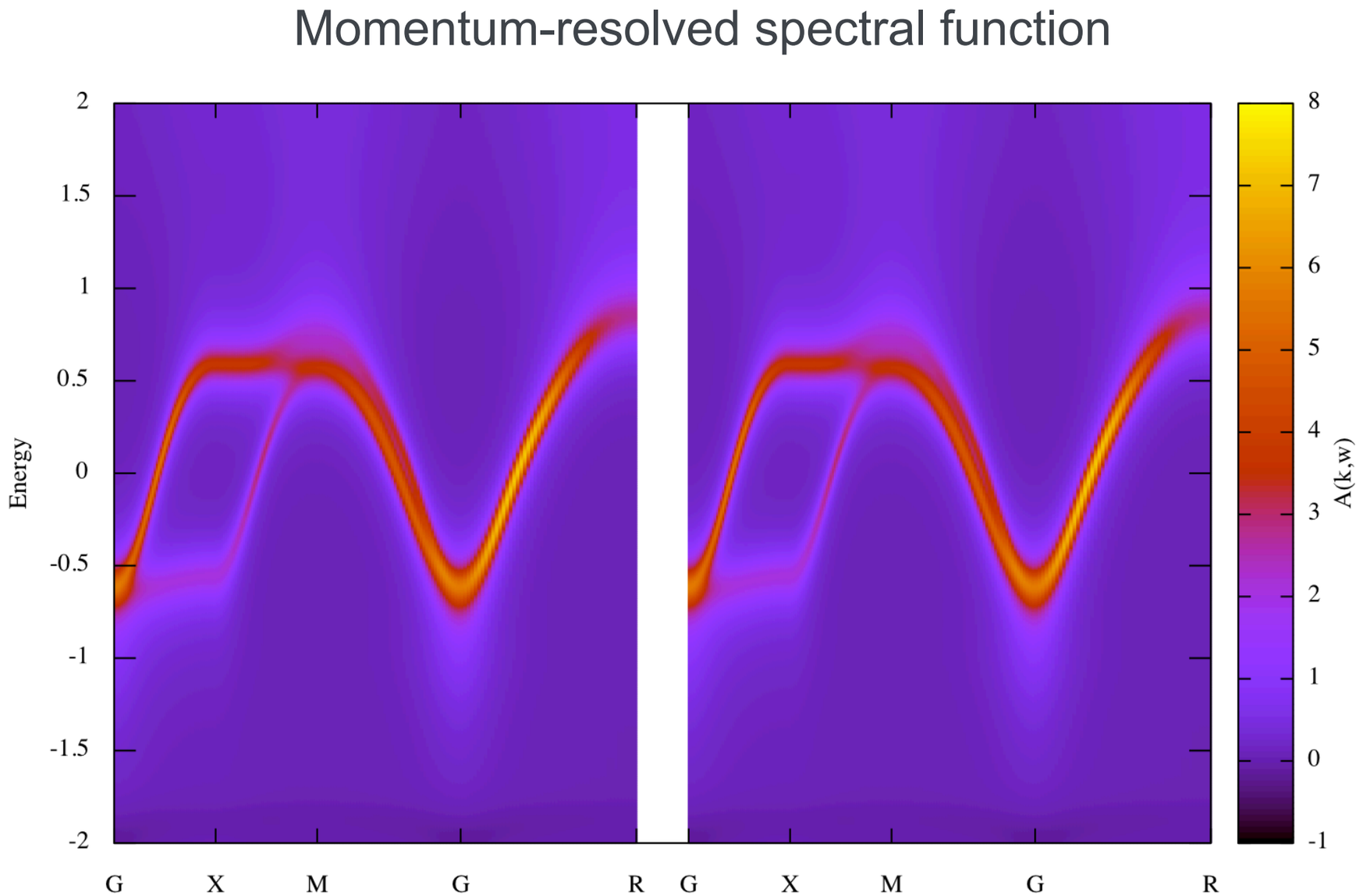
- Execute on enaga (finish about an hour)
  example of job script

```
#!/bin/sh
#QSUB -queue F4cpu
#QSUB -node 2
#QSUB -mpi 80
#QSUB -omp 1
#QSUB -place pack
#QSUB -over false
#PBS -l walltime=2:00:00
#PBS -N srvo3
cd $PBS_O_WORKDIR

source /home/issp/materiapps/dcore/dcorevars.sh
source /home/issp/materiapps/alpscore-cthyb/alpscore-cthybvars.sh
export MPIRUN="mpijob"

export MPI_LAUNCH_TIMEOUT=40
date
dcore_pre srvo3.ini > output-pre
dcore srvo3.ini --np 80 >> output
dcore_post srvo3.ini --np 80 > output-post
dcore_check srvo3.ini --ext=eps
date
```

$SrVO_3$(5)

post/

square_akw.dat



Not converged at 15-th step → use restart function (practice)