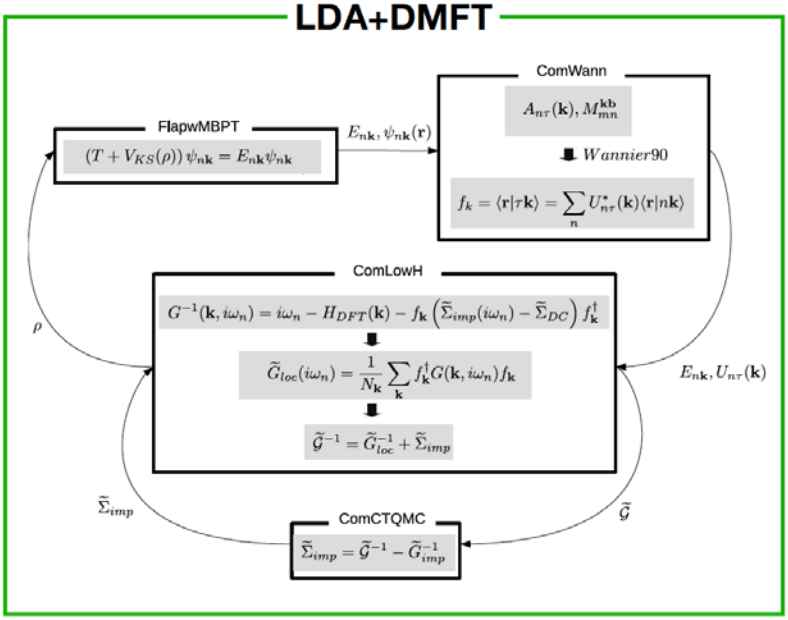
Example: MnO

LDA+DMFT in COMSUITE

We will calculate the electronic structure of a charge transfer insulator MnO within LDA+DMFT. COMSUITE package for LDA+DMFT is composed of four components (software packages). Its work flow is described in the figure below:



1. Construction of a Kohn-Sham Hamiltonian within LDA by **FlapwMBPT** (see https://doi.org/10.1016/j.cpc.2017.06.012, https://www.bnl.gov/cmpmsd/flapwmbpt)
2. Construction of the atom-centered local basis set spanning the low energy Hilbert space by **ComWann utilizing the Wannier90 package.**
3. Wannier interpolation of the mean-field Hamiltonian and solving the DMFT self-consistent equation by **ComLowH** and **ComCTQMC.**
4. Updating the electron density.

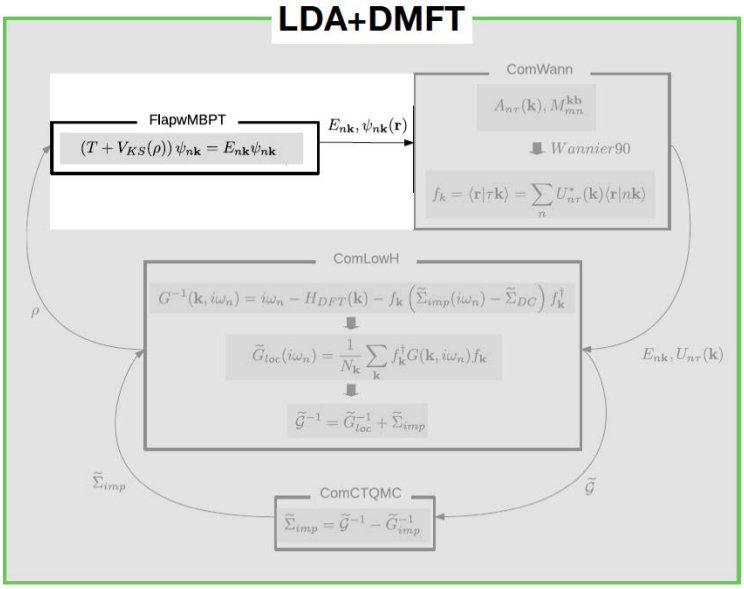
These components along with Wannier90 library are located at install_directory ('install_directory' is described in 'Build and Install' section in Installation of COMSUITE page). To access COMSUITE executables, you should export bin path in your startup shell script.

```
export COMSUITE_BIN=install_directory/bin
```

This tutorial consists of three parts: LDA prerun, LDA+DMFT run, and analysis.

MnO LDA prerun

To run LDA+DMFT, we start with the LDA prerun (see unshaded part of following figure).



To start the LDA calculation, you need to create a directory named “lda” (or a name of your own choice). Note that this name should be specified in ‘comdmft.ini’ as will be explained in the next section. Having done this, move to the created directory:

```
$ mkdir lda
$ cd lda
```

Create an input file “ini” using MnO cif file by executing cif2matdelab.py:

```
$ $COMSUITE_BIN/cif2matdelab.py –m dft –k 2.5 MnO.cif
```

The generated ini file of MnO looks as follows:

```
TEXT band structure calculation
CONTROL   iter_dft= 150  iter_hf=  0  iter_gw=  0 iter_qp=  0
          admix=0.100  adspin=0.600 adm_gw=0.100 acc_it_gw=0.15
          iexch=005 scal_spin= 1.0000
          nproc_tau=   1 nproc_k=   16
          irel=1 clight=274.074e+00 rel_interst=F irel_core=1
          temperature=   1000.00  restart=F
FILES
  allfile=mdl.
SYM symgen=I_R4[0.0,1.0,0.0]_R2[1.0,0.0,0.0]_R3[1.0,-1.0,1.0]_
STRUCTURE  par=  1.0000000  natom=  2 nsort=  2 istruct= -3
      is=  1  2
      b/a= 1.000000  c/a= 1.000000
      a=   0.0000000000000000  -4.2005777179643129  -4.2005777179643129
      b=   4.2005777179643129   0.0000000000000000  -4.2005777179643129
      c=   4.2005777179643129  -4.2005777179643129   0.0000000000000000
    tau=   0.0000000000000000   0.0000000000000000   0.0000000000000000
    tau=   0.5000000000000000   0.5000000000000000   0.5000000000000000
REAL SPACE MESHES mdiv=  14   14   14
                  nrdiv=  10   10   10
BASIS  cut_lapw_ratio=0.610 cut_pb_ratio=0.980
       eps_pb=1.e-03
ZONES nbndf=   0
DOS   emindos=-15.000  emaxdos= 15.000   ndos=  800
      n_cont_frac=  30 e_small=2.e-02
      dos= T           bandstructure= T
K_POINT  ndiv=   6   6   6  metal=T n_k_div=  9 k_line=010
MULTI_SCF vv0=  1.00
MAGNET  b_extval=   0.000000 iter_h_ext=0000100
        b_ext=  0.000  0.000  1.000
TAU MESH n_tau=   46 n_tau_int= 1200
OMEGA MESH n_omega_exa=   29 n_omega_asy=   18 omega_max=  200.00
          interp_omega_d= 2
NU MESH n_nu_exa=   29 n_nu_asy=   18 nu_max=  200.00
        interp_nu_d= 2
ATOMIC DATA ------------------------------------------------------
  txtel=mn    z= 25.0 magn_shift= 0.000
  smt= 2.32800 h= 0.0120 nrad= 1216 z_dop=0.000
  lmb= 6  lmpb= 6
  lim_pb_mt= 30 30 30 30 30 30 30
  ntle=  4  3  3  1  1  1  1
  l  augm  atocc   ptnl  corr idmd
  0   LOC  2.000  3.800     N    0
  0   APW  2.000  4.800     N    0
  0   LOC  0.000  5.800     N    1
  0   LOC  0.000  6.800     N    1
  1   LOC  6.000  3.800     N    0
  1   APW  0.000  4.800     N    0
  1   LOC  0.000  5.800     N    1
  2   APW  5.000  3.800     N    0
  2   LOC  0.000  4.800     N    1
  2   LOC  0.000  5.800     N    1
  3   APW  0.000  4.800     N    0
  4   APW  0.000  5.800     N    0
  5   APW  0.000  6.800     N    0
  6   APW  0.000  7.800     N    0
  txtel=o     z=  8.0 magn_shift= 0.000
  smt= 1.74600 h= 0.0120 nrad= 1216 z_dop=0.000
  lmb= 4  lmpb= 4
  lim_pb_mt= 30 30 30 30 30
  ntle=  3  3  1  1  1
  l  augm  atocc   ptnl  corr idmd
  0   APW  2.000  2.800     N    0
  0   LOC  0.000  3.800     N    1
  0   LOC  0.000  4.800     N    1
  1   APW  4.000  2.800     N    0
  1   LOC  0.000  3.800     N    1
  1   LOC  0.000  4.800     N    1
  2   APW  0.000  3.800     N    0
  3   APW  0.000  4.800     N    0
  4   APW  0.000  5.800     N    0
```

One should modify several input keywords as follows:

- Iter_dft: The number of DFT iterations. Set to 150.
- nproc_tau: The number of MPI processes for imaginary time and frequency parallelization. It is set to 1 as we do not need this parallelization within DFT.
- nproc_k: The number of MPI processes associated with parallelization over Brillouin zone k-points. For the present case, set to 16.
- ndiv: The number of k-mesh in Brillouin zone. $6 \times 6 \times 6$ k-mesh for the present case.

Note that the total number of MPI processes is nproc_tau $\times$ nproc_k = 16 for the present case.

The next step is to run LDA by executing rspflapw.exe. An example of the job script to run rspflapw.exe using SLURM is

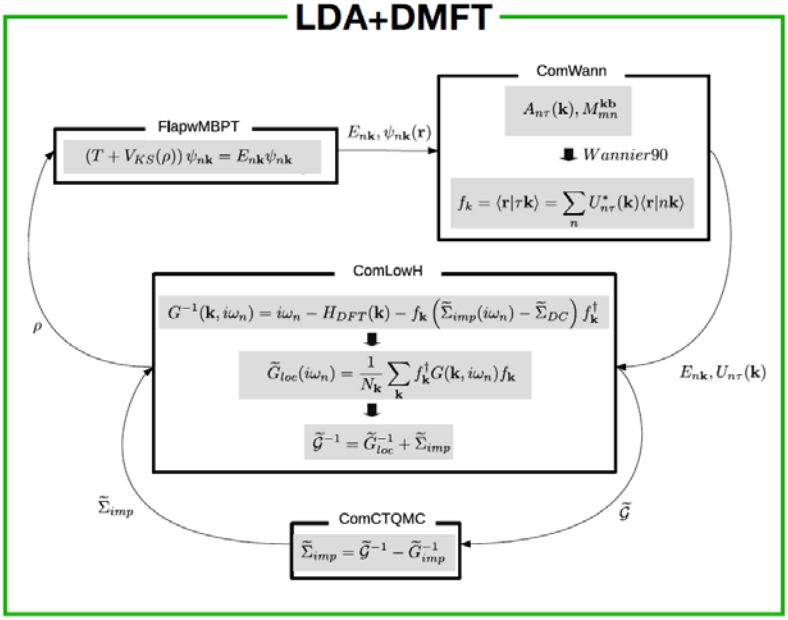
```
#!/bin/bash -l
#SBATCH -J temp
#SBATCH -p regular
#SBATCH -N 1
#SBATCH -e temp.%j.err
#SBATCH -o temp.%j.out
#SBATCH -L SCRATCH
```

```
#SBATCH -C haswell
#SBATCH -t 01:00:00
srun -n 16 $COMSUITE_BIN/rspflapw.exe
```

The total number of processes should be 16 as specified in ini file.

MnO LDA+DMFT run

Once the prerun is finished successfully, the next step is to run the LDA+DMFT calculation (the entire part of the figure below). To run LDA+DMFT, the input file named “comdmft.ini” is needed (see input file section). The calculation of MnO within LDA+DMFT reads output data from the LDA prerun. If you specify the prerun path in comdmft.ini (e.g., ‘../lda’ in this case) correctly, it will read necessary data automatically.



To run LDA+DMFT, move to your work directory, then create lda_dmft directory (you can name this directory what you want) and move to the folder as follows:

```
$ cd ..
$ mkdir lda_dmft
$ cd lda_dmft
```

Then create comdmft.ini (see input file section) file for the LDA+DMFT calculation and execute ‘comdmft.py’ python file in $COMSUITE_BIN. An example of job script doing so using SLURM is

```
#!/bin/bash -l
#SBATCH -J temp
#SBATCH -q regular
#SBATCH -N 12
#SBATCH -e temp.%j.err
#SBATCH -o temp.%j.out
#SBATCH -L SCRATCH
#SBATCH -C haswell
#SBATCH -t 05:00:00
$COMSUITE_BIN/comdmft.py
```

Here ‘comdmft.py’ is a python script which controls the LDA+DMFT calculation. Based on the ‘comdmft.ini’ input file, ‘comdmft.py’ generates all necessary input files to run the individual programs and execute jobs.

Input file (comdmft.ini)

In order to perform the LDA+DMFT calculation, we need only a single input file ‘comdmft.ini’. This input file should be written in python dictionary format. All dictionary keys are in small letters. ‘comdfmt.ini’ is composed of three python dictionaries: ‘control’, ‘wan_hmat’, and ‘imp’:

```
control={'initial_lattice_dir'          : '../lda',
```

```
        'method'                : 'lda+dmft',
        'spin_orbit'            : False,
        'mpi_prefix': "srun -n 384",
        'impurity_problem':[[1, 'd']],
        'impurity_problem_equivalence':[1],
        }

wan_hmat={
        'kgrid': [15, 15, 15],
        'froz_win_min': -10.0,
        'froz_win_max': 10.0,
                    }

imp={'temperature'          : 300, # temperature (in K)
     '1':
     {
    'F0': 9.0,
    'F2': 9.821538461594,
    'F4': 6.13846153846,
    'nominal_n': 5.0,
    'impurity_matrix': [
        [1,0,0,0,0],
        [0,1,0,0,0],
        [0,0,2,0,0],
        [0,0,0,1,0],
        [0,0,0,0,2]
        ],
     'thermalization_time': 1,
     'measurement_time': 20,
     'green_cutoff':  10
     'susceptibility_cutoff': 300,
     'coulomb': 'full',
     }
     }
```

■ In Control

These fields contain basic parameters to control the LDA+DMFT run.

- 'initial_lattice_dir' : '../lda'
  Enter the path which contains the LDA output including the Kohn-Sham eigenvalues and eigenfunctions. This is the LDA prerun folder.

- 'method' : 'lda+dmft'
  Either lda+dmft or lqsgw+dmft. Currently COMSUITE has these two options. Choose 'lda+dmft' for the present case (LDA+DMFT approximation).

- 'spin_orbit': False
  Enter True of False. If False, correlated orbitals correspond to cubic spherical harmonics

$$Y_{lm} = \begin{cases} \frac{i}{\sqrt{2}}\left(Y_l^{-|m|} - (-1)^m Y_l^{|m|}\right), & m < 0 \\ Y_l^0, & m = 0 \\ \frac{1}{\sqrt{2}}\left(Y_l^{-|m|} + (-1)^m Y_l^{|m|}\right), & m > 0 \end{cases}$$

where Y_l^m is a spherical harmonics.

  If True, correlated orbitals chosen at each correlated atom correspond to spin-angular functions |l,i,m>

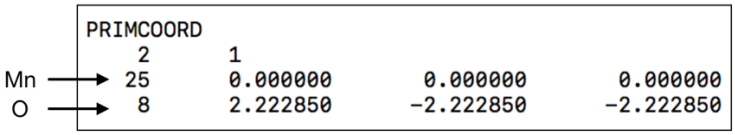
$$\Omega_{l,i=\pm\frac{1}{2},m} = \sum_{s\pm 1/2} C_{i,s}^{l,m} Y_l^{m-s}(\hat{r}) u_s$$

where u_s is a spinor, and $C_{i,s}^{l,m} = \langle l, m-s, \frac{1}{2}, s | l+i, m\rangle$.

- 'mpi_prefix': 'srun -n 384'
  MPI prefix used for FlapwMBPT, ComLowH, ComWann, and ComCTQMC. If a different prefix is required for individual program, specifiy the number using

'mpi_prefix_lattice','mpi_prefix_lowh', 'mpi_prefix_wannier', and 'mpi_prefix_impurity'. Note that 384 is the number of total processes we want to use for the present case.

- 'impurity_problem': [[1,'d']]
  A python list to specify correlated orbitals. The first and second indices indicates the atom index and shell type, respectively. Atom index: in the order listed in the "../lda/coord.xsf"



  Shell index is either "d" or "f".

- 'impurity_problem_equivalence': [1]
  Equivalence of each impurity problem. The value is identified by a positive integer starting from 1. If an impurity has the same value the one for another impurity, the problems are equivalent.

- 'restart': False
  Enter True or False. If True, it will resume the calculation from the previous LDA+DMFT run. The default value is False.

- 'mpi_prefix_lowh':
  MPI prefix for ComLowH. The default value is the one specified in control['mpi_prefix']

- 'mpi_prefix_impurity':
  MPI prefix for the impurity solver. The default value is the one specified in control['mpi_prefix']

- 'mpi_prefix_wannier':
  MPI prefix for ComWann. The default value is the one specified in control['mpi_prefix'].

- 'sigma_mix_ratio':
  Self-energy linear mixing ratio. You can specify any number within 0.0 – 1.0. The default value is 0.5.

- 'max_iter_num_impurity':
  Maximum iteration for the DMFT self-consistent loop. The default value is 50.

- 'proj_win_min':
  Low-energy cutoff to renormalize the projectors. The default value is the one specified in wan_hmat['dis_win_min']

- 'proj_win_max':
  High-energy cutoff to renormalize the projectors. The default value is the one specified in wan_hmat['dis_win_max']

■ **In wan_hmat**:
These fields define the frozen window, the disentanglement window, and the ab initio calculation from which the maximally localized Wannier functions (MLFWs) are constructed.

- 'kgrid': [15,15,15]
  Crystal momentum grid for the Wannier interpolation of the LDA band structure.

- 'froz_win_min': -10.0 eV
  Lower boundary of the inner (frozen) window in eV.

- 'froz_win_max': 10.0 eV
  Upper boundary of the inner (frozen) window in eV.

- 'dis_win_min':
  Lower boundary of the outer (disentanglement) window in eV. The default value is same with wan_hmat['froz_win_min'].

- 'dis_win_max':
  Upper boundary of the outer (disentanglement) window in eV. The default value is wan_hmat['froz_win_max'] +40.0

- 'num_iter':
  The number of minimization steps for the wannierization process (gauge dependent part of total spreading). The default value is 0.

- 'dis_num_iter':
  The number of minimization steps for the disentanglement process (gauge independent part of total spreading). The default value is 100.

■ **In imp**:
These fields are related with the Monte-Carlo algorithm and sampling of observables.

- 'temperature': 300
  Simulation temperature in K

- 'F0', 'F2', 'F4': The values of Slater integrals in eV. Note that for "f" shells, 'F0', 'F2', 'F4', and 'F6' should be specified.

- 'nominal_n': Nominal occupancy associated with the impurity shell. This value is required since COMSUITE adopts the so-called "nominal double counting" for LDA+DMFT which reads:

$$\tilde{\Sigma}^{DC} = U(N_0 - \frac{1}{2}) - J(\frac{N_0}{2} - \frac{1}{2})$$

  where N_0 is the nominal occupancy specified by 'nominal_n'.

- For each distinct impurity problem indexed by the value in control ["impurity_problem_equivalence"],
  - 'impurity_matrix': [
    [1,0,0,0,0],
    [0,1,0,0,0], 1:t_{2g}
    [0,0,2,0,0], 2:e_g
    [0,0,0,1,0],
    [0,0,0,0,2]
    ],
    Equivalence of the matrix element of the hybridization function and impurity self-energy. Starting from "1", you can set any positive number. For those values that are the same, the corresponding hybridization functions and impurity self-energies will be identical. If an element in the matrix is zero, then it will not be sampled by the impurity solver. Each column and row correspond to the Wannier orbitals in the following order: |xy>,|yz>,|z^2>,|xz>, |x^2-y^2> if control['spin_orbit']==False. If control['spin_orbit']==True, the most rapidly changing index is "m" and the next one is "i". They are sorted in ascending order. For the case of the "f" shell, for example, they are ordered as:|3,-0.5, -2.5>,|3,-0.5, -1.5>,|3,-0.5, -0.5>, |3,-0.5, 0.5>,|3,-0.5, 1.5>,|3,-0.5, 2.5>, |3,0.5, -3.5>,|3,0.5, -2.5>,|3,0.5,-1.5>,|3,0.5, -0.5>,|3,0.5, 0.5>,|3,0.5, 1.5>,|3,0.5, 2.5>, |3,0.5, 3.5>,

  - 'thermalization_time': 1,
    Wall time for the thermalization in minutes.

  - 'measurement_time': 20,
    Wall time for the measurement in minutes.

  - 'green_cutoff': 10,
    Cutoff-energy in eV to sample the Green's function and self-energy. The values beyond this energy are provided by an analytical form.

  - 'susceptibility_cutoff': 300,
    Cutoff-energy to sample susceptibility. The default value is 300 eV.

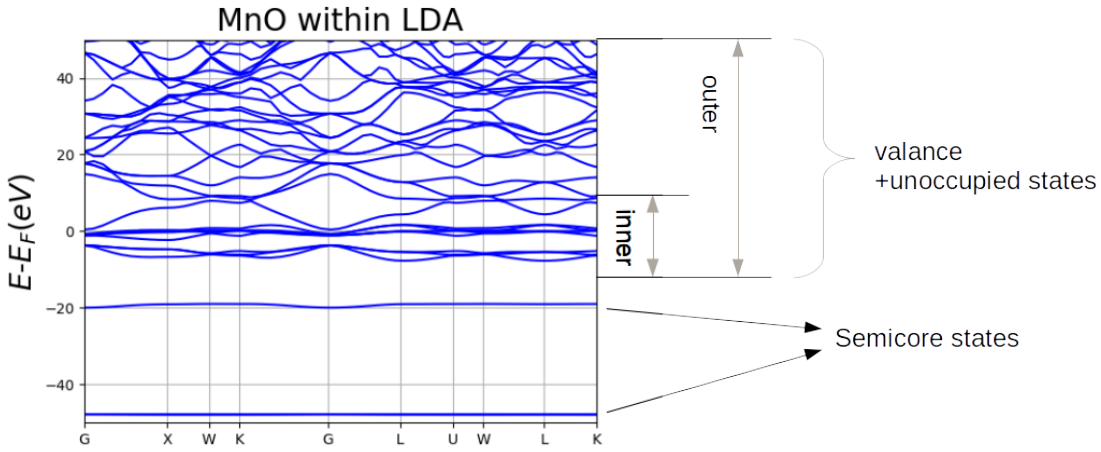
  - 'Coulomb': 'full',

'full' or 'ising' are available. We construct the Coulomb matrix in the following way:

If 'full', no additional approximation is considered. If 'ising', only U_{abba} or U_{abab} are non-zero.

Input file (comdmft.ini)-Important concepts for wan_hmat

COMSUITE uses localized orbitals such as Wannier functions to represent the low-energy Hilbert space. To construct the Wannier functions, the inner (frozen) energy window can be set to range from E_F-10 eV to E_F+10 eV, and the outer (disentanglement) energy window can range from E_F-10eV to E_F+50eV; see the figure below.

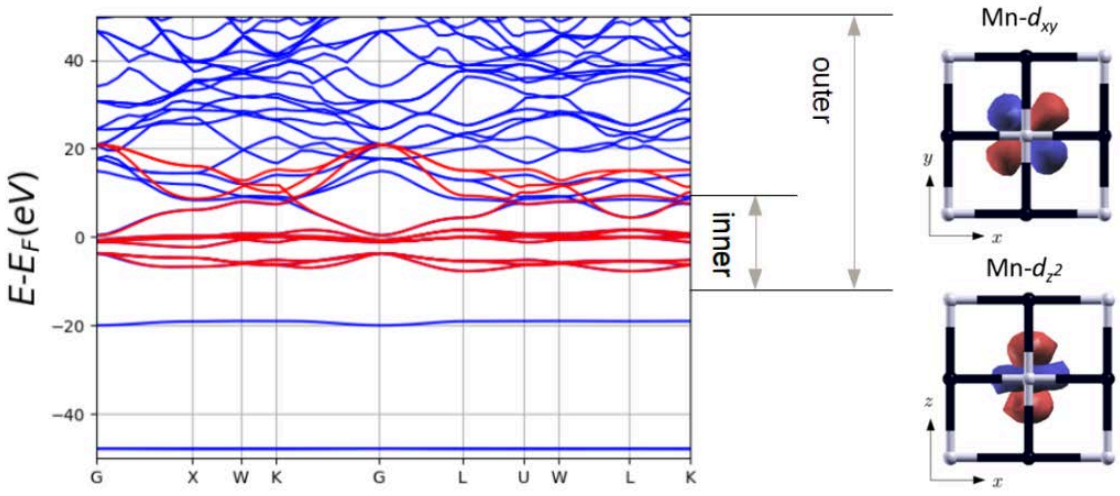


We choose initial trial orbitals $|\tau\mathbf{R}=0\rangle_t$ using muffintin orbitals with the desired angular momentum character. The radial functions of $|\tau\mathbf{R}=0\rangle_t$ are chosen in such a way to maximize

$$\frac{1}{N_{\mathbf{k}}} \sum_{n\mathbf{k}}^{E_{min}^{inner}<E_{n\mathbf{k}}<E_{min}^{inner}} |\langle n\mathbf{k}|\tau\mathbf{k}\rangle_t|^2$$

, where $|\tau\mathbf{k}\rangle_t = \frac{1}{N_{\mathbf{k}}} \sum_{\mathbf{R}} |\tau\mathbf{R}\rangle_t e^{i\mathbf{k}\cdot\mathbf{R}}$. Among the MT orbitals above, we chosse ones which are larger than 0.15. For correlated orbitals, the final Wannier functions $|\tau\mathbf{R}=0\rangle_f$ usually satisfy a condition of ${}_f\langle\tau\mathbf{R}=0|\tau\mathbf{R}=0\rangle_t > 0.95$. This means that $|\tau\mathbf{R}=0\rangle_f$ are strongly localized and are regarded as atom-like wavefunctions.

The figure below shows the Wannier functions and the interpolated band structure of MnO in comparison with the LDA band structure. The number of bands in the inner window is 10, while the number of bands in the outer window is 29. The number of trial orbitals is 12 (Mn-*s*, Mn-*p*, Mn-*d*, O-*p*).



Output files

COMSUITE places important output files generated from individual programs in the work directory (lda_dmft in this example). The list of files is

cmd.log
convergence.log : convergence log filesig.dat : impurity self-energy
delta.dat : hybridization function

The format of each file and meaning of fields are discussed below. The corresponding results of the MnO LDA+DMFT calculation are presented with illustrative plots.

■ convergence.log

step	i_outer	i_latt	i_imp	causality	delta_rho	w_sp_min	w_sp_max	mu	std_sig	n_imp	histo_1	histo_2
wannier	1					0.49222527	2.20331132					
delta	1		1	good				-0.003306711437				
impurity_1	1		1	good					1.18650371541	5.05871	70.3561947683	75.6208508652
dft	2	1			1.986442e-05							
wannier	2					0.49222527	2.20331136					
delta	2		1	good				-0.186608653325				
impurity_1	2		1	broken					0.340370049484	5.06588	65.5153172627	65.8529794209
dft	3	1			0.02857597							
wannier	3					0.49209252	2.20332784					
delta	3		1	good				-0.286551210778				
impurity_1	3		1	good					0.651736929279	5.067	65.2825222364	65.4645298718

- i_outer: The scf step number for a given charge density and impurity self-energy.
- i_latt: The iteration number for solving the Kohn-Sham equation with a given charge density obtained from ComLowH.
- i_imp: The iteration number for solving the impurity problem through ComLowH + ComCTQMC
- causality: causality of hybridization function/self-energy
- delta_rho: The norm difference between the current charge density and the one from the previous scf step.
- w_sp_min: minimum spread of the Wannier functions
- w_sp_max: maximum spread of the Wannier functions
- mu: LDA+DMFT chemical potential w.r.t. LDA chemical potential
- std_sig:

$$\sqrt{\frac{\sum_i (\Sigma_i^j(i\omega_n) - \Sigma_i^{j-1}(i\omega_n))^2}{n_\omega n_{orb}}}$$

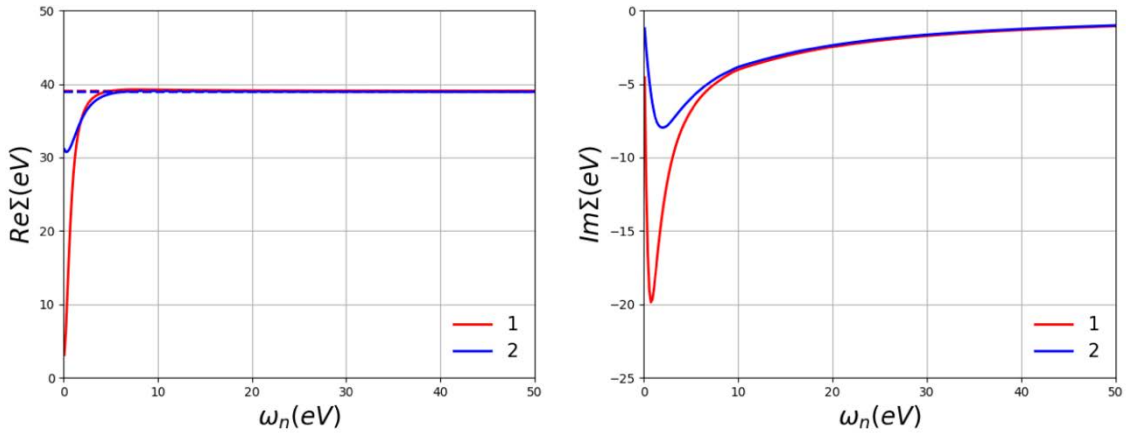
- n_imp: occupation of the impurity orbitals
- histo_1: the first moment of the perturbation order histogram in the CTQMC solver.
- histo_2: the second moment of the perturbation order histogram in the CTQMC solver.

■ sig.dat

'sig.dat' contains the impurity self-energies. The first column lists Matsubara frequencies and the next columns are the real and imaginary parts of self-energies for "t_{2g}" (Sig_{1,1}) and "e_g" (Sig_{1,2}) of the Mn d-orbitals.

# omega(eV)	Re Sig_{1,1}(eV)	Im Sig_{1,1}(eV)	Re Sig_{1,2}(eV)	Im Sig_{1,2}(eV)
0.081216424692	3.114863877833	-4.554297039654	31.108323219061	-1.217609297270
0.243649274076	6.228220525507	-12.006424175023	30.747977833336	-2.632684734601
0.406082123461	11.353047666223	-16.569603244928	30.779320340335	-3.791321447075
0.568514972845	16.544277720346	-19.055213906760	31.001772520541	-4.783451177032
0.730947822229	21.151079976456	-19.858731880712	31.393889655396	-5.625962702023
0.893380671613	24.904718921905	-19.663388647230	31.883643841680	-6.318895663965
1.055813520997	27.843554786113	-18.954941724430	32.430290141185	-6.876858394169
1.218246370382	30.110743137325	-18.024346609681	32.979690026641	-7.303327849460
1.380679219766	31.858022374455	-17.029724664888	33.523386380996	-7.601462820372
1.543112069150	33.213053035138	-16.051058133558	34.058869437732	-7.793160652151
1.705544918534	34.274620346168	-15.125733546836	34.570096307137	-7.910013093578
1.867977767919	35.116209452671	-14.269012317915	35.043520851238	-7.965418174840
2.030410617303	35.791542677236	-13.484729369630	35.479279740080	-7.970893126714
2.192843466687	36.339895692266	-12.771048686743	35.876163900288	-7.947067853456
2.355276316071	36.790056215904	-12.123654362640	36.222036743753	-7.899450091154
2.517709165455	37.163107586834	-11.537239692632	36.520261464254	-7.819047100076

Plots of impurity self-energy:

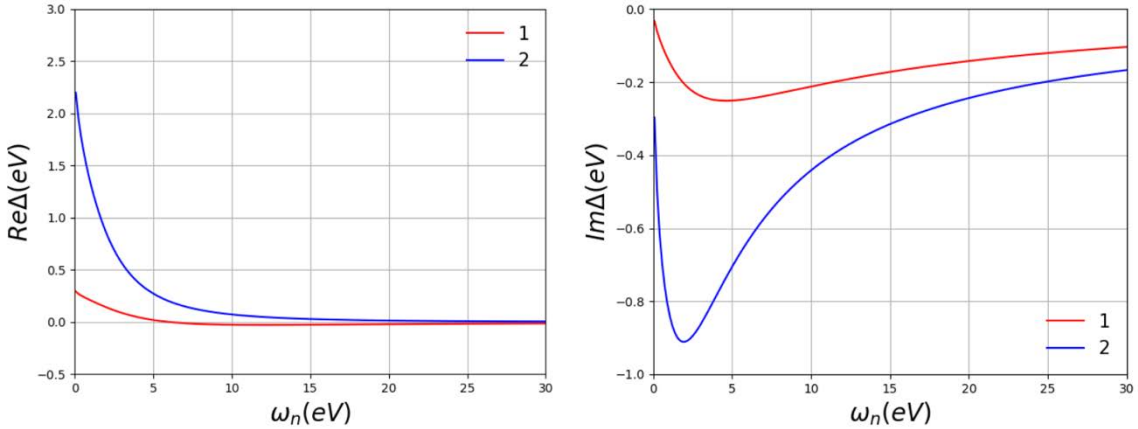


Impurity self-energy associated with the Mn-d orbitals on the imaginary frequency axis. The red and blue lines represent Mn-t_{2g} and Mn-e_g characters, respectively.

■ **delta.dat**

'delta.dat' is in the same format of 'sig.dat'

Plots of the hybridization function:



Hybridization function associated with the Mn-d orbitals on imaginary frequency axis. The red and blue lines represent the Mn-t_{2g} and Mn-e_g characters, respectively. Mn-t_{2g} orbitals experience less hybridization with the rest of the electrons than the Mn-e_g orbitals.

Analytical Continuation of Self-energy

To obtain the density of states (DOS), we need to perform an analytic continuation of 'sig.dat' to produce the impurity self-energy on the real frequency axis. To do this, we will adopt the maximum entropy (maxent) method. Any publicly available maxent code can be employed. For the purposes of this tutorial, however, we will use K. Haule's maxent code (freely available at http://www.physics.rutgers.edu/~haule.)

To access the maxent code, you should export the path to the executable in your startup shell script.

```
export WIEN_DMFT_ROOT=[path to Haule's code bin directory where maxent_run.py is located]
```

To run the maxent code, move to your working directory, create the maxent directory in the "lda_dmft" directory and then cd to it:

```
$ mkdir maxent
$ cd maxent
```

By executing ‘maxent_wrapper.py’, we can obtain the self-energy on the real axis by automatically calling maxent_run.py:

```
$ $COMSUITE_BIN/maxent_wrapper.py ../sig.dat
```

You can see its options with the -h option:

```
$COMSUITE_BIN/maxent_wrapper.py –h
```

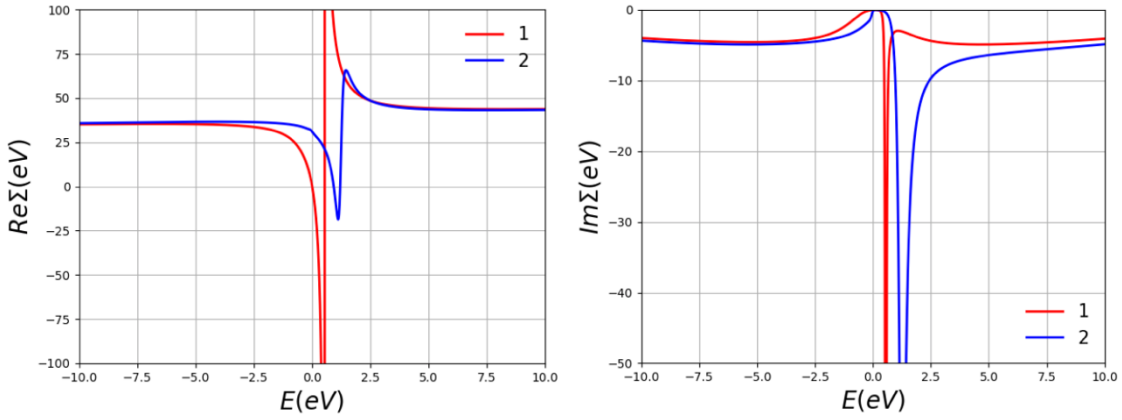
```
usage: maxent_wrapper.py [-h] sig [error]

call maxent_run.py from EDMFTF package and return self-energy on real axis. If
maxent_params.dat is not present in the directory, it generates one. The name
of the output file will be sig_realaxis.dat

positional arguments:
  sig         self-energy file on imaginary axis
  error       Errors for the maxent.Optional. Default value=0.05

optional arguments:
  -h, --help  show this help message and exit
```

Here note that in order to perform maxent, the `maxent_params.dat’ file is needed. This file should be written in the python dictionary format as follows:

```
params={'statistics': 'fermi', # fermi/bose
        'Ntau'      : 3000,     # Number of time points
        'L'         : 70.0,     # cutoff frequency on real axis
        'Nw'        : 501,     # number of frequency points on real axis
        'gwidth'    : 140.0,  # width of gaussian
        'idg'       : 1,       # error scheme: idg=1 -> sigma=deltag ; idg=0 -> sigma=deltag*G(tau)
        'deltag'    : 0.05,   # error
        'Asteps'    : 4000,    # anealing steps
        'alpha0'    : 1000,    # starting alpha
        'x0'        : 0.01,    # low energy cutoff
        'min_ratio' : 0.001,    # condition to finish, what should be the ratio
        'iflat'     : 1,       # iflat=0 : constant model, iflat=1 : gaussian of width gwidth, iflat=2 : input using file model.dat
        'Nitt'      : 1000,    # maximum number of outside iterations
        'Nr'        : 0,       # number of smoothing runs
        'bwdth'     : 0.03,    # smoothing width
        'Nf'        : 5,      # to perform inverse Fourier, high frequency limit is computed from the last Nf points
         }
```

If the ‘maxent_params.dat’ file is not present in the directory, maxent_wrapper.py will automatically generate the file with default options and execute subsequent jobs. If maxent is finished successfully, you will obtain an analytically continued self-energy file, ‘sig_realaxis.dat’. ‘sig_realaxis.dat’ is in the same format as ‘sig.dat’ except that the first column corresponds to real frequency values.

Plots of the impurity self-energy on the real frequency axis:



The impurity self-energy on the real frequency axis as obtained by using the maximum entropy method. Electronic self-energy for both Mn-t_{2g} and Mn-e_g has a pole near Fermi level indicative of the paramagnetic Mott gap in MnO.

LDA+DMFT density of states

To obtain the DOS, we must post-process the data by executing ComLowH again. First, create a directory for the DOS calculation in the “lda_dmft” directory and move to it:

```
$ mkdir realgrid
$ cd realgrid
```

Copy the necessary files to calculate DOS and partial DOS by using prepare_realaxis.py which is in the $COMSUITE_BIN folder.

```
$ $COMSUITE_BIN/prepare_realaxis.py 0.1 ../lowh/ ../wannier/ ../maxent/sig_realaxis.dat 30 30 30 -m 2
```

You can see its options with the -h option:

```
$COMSUITE_BIN/prepare_realaxis.py –h
```

```
usage: prepare_realaxis.py [-h] [-m MODE]
                           broadening lowh_directory wan_directory self_energy
                           [kmesh_b1_for_dos] [kmesh_b2_for_dos]
                           [kmesh_b3_for_dos]

prepare inputs of comlowh calculation on real axis

positional arguments:
  broadening            broadening
  lowh_directory        lowh directory
  wan_directory         wannier directory
  self_energy           real-axis self-energy
  kmesh_b1_for_dos      finer kmesh along b1 axis for the DOS. Optional
  kmesh_b2_for_dos      finer kmesh along b2 axis for the DOS. Optional
  kmesh_b3_for_dos      finer kmesh along b3 axis for the DOS. Optional

optional arguments:
  -h, --help            show this help message and exit
  -m MODE, --mode MODE  If 3, code calculates spectral function along the high
                        symmetry line defined in 'kpath.dat'. If it is 2, it
                        calculates projected density of states. Default: 3
```

Run ComLowH with the job submission script. An example of such a script using SLURM is

```
#!/bin/bash -l
#SBATCH -J temp
#SBATCH -q debug
#SBATCH -N 1
#SBATCH -e temp.%j.err
#SBATCH -o temp.%j.out
#SBATCH -L SCRATCH
#SBATCH -C haswell
#SBATCH -t 00:30:00
srun -n 30 $COMSUITE_BIN/ComLowH
```

Having done so, you will obtain files tdos.dat and pdos.dat.

The format of tdos.dat file is:

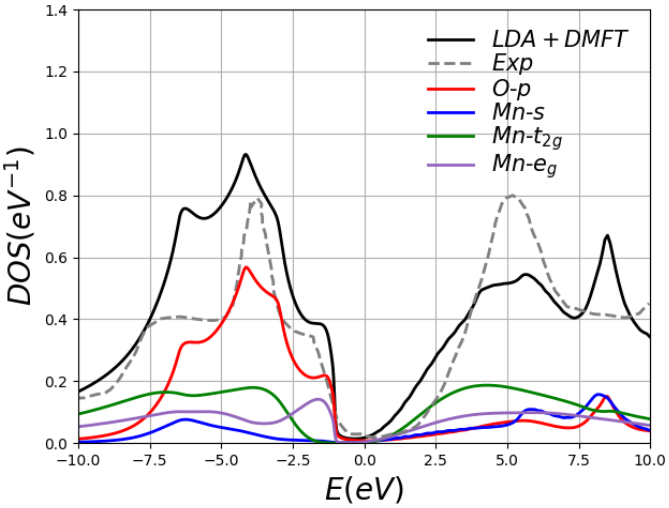
```
#      omega (eV)          TDOS (1/eV)
   -70.000000000000     0.000073395648
   -65.733023713200     0.000083146001
   -61.955278682000     0.000093507756
   -58.587128546700     0.000104525035
   -55.565331497900     0.000116170098
   -52.839027896300     0.000128466779
   -50.366851286200     0.000141438351
   -48.114812211200     0.000155121390
   -46.054723948000     0.000169469789
   -44.163014879300     0.000184613333
   -42.419821079200     0.000200601195
```

The format of pdos.dat file is:

```
#     omega (eV)            (1,0,0)          (1,1,-1)          (1,1,0)           (1,1,1)          (1,2,-2)
   -70.000000000000     0.000005845078    0.000004956871    0.000004956870    0.000004956871    0.000006630582
   -65.733023713200     0.000006595095    0.000005539974    0.000005539973    0.000005539973    0.000007542111
   -61.955278682000     0.000007387974    0.000006147944    0.000006147943    0.000006147944    0.000008520099
   -58.587128546700     0.000008223617    0.000006780038    0.000006780037    0.000006780038    0.000009565748
   -55.565331497900     0.000009101985    0.000007435583    0.000007435582    0.000007435583    0.000010675665
   -52.839027896300     0.000010023105    0.000008113983    0.000008113982    0.000008113982    0.000011856762
   -50.366851286200     0.000010987062    0.000008814708    0.000008814706    0.000008814707    0.000013109745
   -48.114812211200     0.000011994007    0.000009537296    0.000009537294    0.000009537295    0.000014444438
   -46.054723948000     0.000013044138    0.000010281331    0.000010281328    0.000010281330    0.000015848555
   -44.163014879300     0.000014137740    0.000011046486    0.000011046483    0.000011046485    0.000017346916
```

- (atom index, l, m) if spin_orbit==False and (atom index, l, i,m) if spin_orbit==True.

A plot of pdos.dat along with experimental data is given below:



MnO DOS within LDA+DMFT. For comparison, experimental data is plotted with a gray dashed line. LDA+DMFT reasonably captures the experimentally observed peaks at 5eV, -2eV, -4eV and -7eV. PDOS data attributes each peak to Mn-t_{2g}, Mn-e_g, Mn-t_{2g}, and O-p orbitals, respectively. Substantial contribution of the O-p to the top of the valence band is a manifestation of the strong hybridization between the O-p and Mn-e_g, which is consistent with the Zhang-Rice picture.

LDA+DMFT spectral function

To obtain spectral function, we need to post-process with ComLowH again. First create a directory for the spectral function in “lda_dmft” directory and move to it:

```
$ mkdir realaxis
$ cd realaxis
```

Having done so, copy the necessary files to calculate the spectral function by using prepare_realaxis.py which is in $COMSUITE_BIT directory. Note that the option should be ‘-m 3’.

```
$ $COMSUITE_BIN/prepare_realaxis.py 0.1 ../lowh/ ../wannier/ ../maxent/sig_realaxis.dat -m 3
```

You then need to create a k-path file (kpath.dat). In the first line, put the number of k points. And in the following lines, k points are specified in terms of the reciprocal lattice vector.

```
186       # (kx,  ky,  kz)
  0.00000000   0.00000000   0.00000000
  0.01666667   0.00000000   0.01666667
  0.03333333   0.00000000   0.03333333
  0.05000000   0.00000000   0.05000000
  0.06666667   0.00000000   0.06666667
  0.08333333   0.00000000   0.08333333
  0.10000000   0.00000000   0.10000000
  0.11666667   0.00000000   0.11666667
  0.13333333   0.00000000   0.13333333
  0.15000000   0.00000000   0.15000000
  0.16666667   0.00000000   0.16666667
  0.18333333   0.00000000   0.18333333
  0.20000000   0.00000000   0.20000000
  0.21666667   0.00000000   0.21666667
  0.23333333   0.00000000   0.23333333
  0.25000000   0.00000000   0.25000000
```

Run ComLowH with a job submission script. An example of such a script using SLURM is

```
#!/bin/bash -l
#SBATCH -J temp
#SBATCH -q debug
#SBATCH -N 1
#SBATCH -e temp.%j.err
#SBATCH -o temp.%j.out
#SBATCH -L SCRATCH
#SBATCH -C haswell
#SBATCH -t 00:30:00
srun -n 30 $COMSUITE_BIN/ComLowH
```

Having done so, you will have obtained a file named spectral.dat. The format of this file is

# kpoint	E (eV)	A (1/eV)
1	-70.000000000000	0.000071405488
2	-70.000000000000	0.000071413398
3	-70.000000000000	0.000071436882
4	-70.000000000000	0.000071475244
5	-70.000000000000	0.000071527437
6	-70.000000000000	0.000071592209
7	-70.000000000000	0.000071668238
8	-70.000000000000	0.000071754230
9	-70.000000000000	0.000071848944
10	-70.000000000000	0.000071951141
11	-70.000000000000	0.000072059484

Plot of spectral.dat file along a high symmetry line in the first Brillouin zone.: