# DFT Embedded Potential Construction V1.0

By Chen Yilin

## Replace the corresponding files in PySCF (Be sure to backup the original files before replacing)

- Replace hf.py, uhf.py, rohf.py in the pyscf/scf/ path with the files from step 0 folder.

- Replace rks.py, uks.py, roks.py in the pyscf/dft/ path with the files from step 0 folder.

## Note

After replacement, the "symmetry" setting in PySCF will have a bug that is not yet fixed.

The mol.symmetry = True setting cannot be used.

Symmetry must always be turned off.

All HF and DFT calculations in PySCF require three empty files named hf\_mo\_occ.txt, hf\_mo\_occ\_a.txt, and hf\_mo\_occ\_b.txt in the input files to avoid errors. If errors occur, add the corresponding empty files (see section 3-2).

## Current Solutions:

1. Backup the original files before replacement, and replace them back when symmetry calculations are needed. (Constructing the embedding potential does not require symmetry, but calculations for individual systems might).

2. Fix the bug, possibly in pyscf/scf/hf\_symmetry.py and pyscf/dft/rks\_symmetry.py.

## Overall Steps:

1. Specify the "embedding region" atoms in the system, and the rest as "environment region".

2. Perform DMFET with a larger smearing width (0.02 or other values, testing needed).

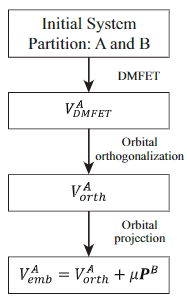
3. Perform DMFET with a smaller smearing width (0.005), converge, and check the final molecular occupancy numbers in the molden file, which should be integers.

4. (Optional) For systems with degenerate orbitals or persistent overlap between "embedding region" and "environment region" orbitals, perform orthogonalization in the overlapping orbital space.

5. Construct the projection operator P based on the density matrices dma and dmb of the "embedding region" obtained above, choose the projection term μ, and complete the projection and final embedding potential construction.

Note: Detailed comments are available in the example code, please read them carefully. This section only explains the framework and basic process. Read with the code and comments!

## Simple Flowchart:



## First Calculation Steps (Using Ethanol C2H6O as an Example):

3-1. Open the XYZ coordinate file and specify the atoms to be the "embedding region" based on their line numbers (starting from 0).

Example: I specified atom 0 (Oxygen) and atom 8 (Hydrogen) as the "embedding region", and the rest are automatically "environment region".

Enter [0 8] in the example.



3-2. In the input file of section 1-1:

charge\_A.txt and charge\_B.txt contain the coordinates (XYZ) of the "Capping Charges" added to the embedding region and environment region.

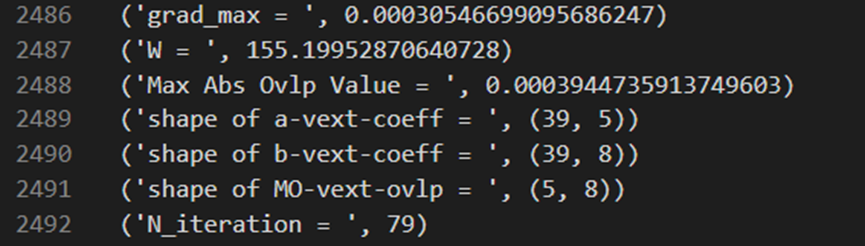
hf\_mo\_occ.txt, hf\_mo\_occ\_a.txt, hf\_mo\_occ\_b.txt are empty files required for all PySCF calculations. Ensure these files exist to avoid errors.

3-3. In the settings of "step 1-DMFET.py" in section 1-1:

For the first calculation, set isFirstOEP to True.

Adjust other parameters as per the detailed comments in the code.

3-4. Start the first DMFET calculation and check the output file of section 1-1.

Check the OUT file ‘Step 1 – DMFET.out’ which records the OEP iteration process:

Grad\_max is the maximum absolute value in the current Gradient matrix.

W is the current functional value.

Max Abs Ovlp Value is the maximum overlap value between "embedding region" and "environment region" orbitals.

N\_iteration is the current OEP iteration count.

After this calculation, V\_DMFET.txt stores the current embedding potential Vemb.

## Continuation Calculation:

See section 1-2.

Copy the output files of section 1-1 to a new folder as the new input files for continuation.

Change only one thing to continue the calculation: Set isFirstOEP to False in "step 1-DMFET.py".

The smearing width value can be changed as needed, either keeping the original value or reducing it.

Example 1-2 reduces the smearing width to 0.005, finalizes the potential construction, and converges.

After this calculation, V\_DMFET.txt stores the current embedding potential Vemb.

## Orbital Orthogonalization:

This step is for degenerate orbitals or special cases requiring a large smearing width.

Since actual examples are rare and the code is highly specific, it's recommended to understand the principle and design and handle each system individually.

Attached is part of the original code and data for NV centers:

The basic idea is to reconstruct the orthogonalized dma and dmb using Reformorbitals.py and then perform OEP for the embedding region a with the new dma.

## Orbital Projection:

Orbital projection is universal.

Use the output files of section 1-2 as the new input files, and run "step 3-Projection.py" to complete the projection.

After this calculation, Vemb\_with\_P.txt stores the final embedding potential with the projection term.

Thus, the DFT embedded potential construction is complete.