

# Embedded mean-field theory in ONETEP

Robert J. Charlton and Joseph C.A. Prentice  
Imperial College London

January 2019

## 1 Embedded Mean-Field Theory (EMFT)

Often in simulations of materials we wish to consider the impact of a host environment on a system of interest, such as chromophores in solvent or doped molecular crystals. While the interesting physics or chemistry may be associated with the subsystem, the effects of the environment can be significant and warrant description at the quantum level of theory. However, the cost of applying accurate quantum methods such as hybrid functionals to potentially large environments can be restricted by the cost of such methods. Quantum embedding [1, 2] methods are intended to combine an accurate, high-level description of the subsystem of interest (active) with a cheaper, low-level method for the host environment.

Embedded mean field theory (EMFT) [3] is an approach to quantum embedding based on the one-electron density matrix. To begin, we partition the density matrix into subsystem components,

$$\rho = \begin{pmatrix} \rho_{AA} & \rho_{AB} \\ \rho_{BA} & \rho_{BB} \end{pmatrix}, \quad (1)$$

where A is the active region and B is the inactive environment. The total energy can be written as

$$E[\rho] = \text{tr}[\rho H_0] + G[\rho], \quad (2)$$

where  $H_0$  contains the one-electron terms of the Hamiltonian and  $G[\rho]$  contains all two-electron terms (local, Hartree and exchange-correlation effects). In embedded mean-field theory (EMFT), the two-electron interaction for the active subsystem A is constructed at a higher level of theory to the rest of the system,

$$E^{\text{EMFT}}[\rho] = \text{tr}[\rho H_0] + G^{\text{low}}[\rho] + \left( G^{\text{high}}[\rho_{AA}] - G^{\text{low}}[\rho_{AA}] \right), \quad (3)$$

where  $G^{\text{low}}$  and  $G^{\text{high}}$  are the two-electron interaction energies at the lower and higher levels of theory, respectively. For example, the low level theory

could be LDA while the higher level uses a hybrid functional such as B3LYP. We assume here that the core Hamiltonian  $H_0$  is the same at both levels of theory, though this need not necessarily be the case. The ground state of the embedded system can thus be obtained by minimising (3) with respect to the elements of the density matrix.

### 1.1 Block orthogonalisation

Normalisation is maintained provided the trace of the density matrix with the overlap matrix satisfies

$$\text{Tr}[\rho\mathbf{S}] = N. \quad (4)$$

EMFT partitioning can result in unrealistic charge spillover from the low-level to the high-level region, producing large negative results for the off-diagonal terms  $\text{Tr}[\rho_{AB}\mathbf{S}_{BA}]$  and  $\text{Tr}[\rho_{BA}\mathbf{S}_{AB}]$ . One possible remedy is to impose a block-orthogonalisation (BO) between the subsystem orbitals [4],

$$|\tilde{\phi}_i^B\rangle = (1 - \hat{P}^A)|\phi_i^B\rangle, \quad (5)$$

$$\hat{P}^A = \sum_{j,k \in A} |\phi_j^A\rangle (\mathbf{S}^{AA})_{jk}^{-1} \langle \phi_k^A|. \quad (6)$$

By construction  $\text{Tr}[\rho_{AB}\mathbf{S}_{BA}]$  and  $\text{Tr}[\rho_{BA}\mathbf{S}_{AB}]$  are strictly zero and all electrons are associated with the diagonal blocks.

## 2 Implementation in ONETEP

Quantum embedding as implemented in ONETEP is based around EMFT. Here we denote the active system NGWFs as  $|\chi_i^A\rangle$  and the environment NGWFs as  $|\phi_j^B\rangle$ . The fundamental quantity of interest is the Hamiltonian,

$$\mathbf{H}^{\text{EMFT}} = \begin{pmatrix} \mathbf{H}_{AA}^{\text{high}} & \mathbf{H}_{AB}^{\text{low}} \\ \mathbf{H}_{BA}^{\text{low}} & \mathbf{H}_{BB}^{\text{low}} \end{pmatrix}, \quad (7)$$

where the high- and low-level Hamiltonian operators are given as

$$\hat{H}^{\text{high}} = \hat{T} + \hat{V}_{\text{local}} + \hat{V}_{\text{Hartree}} + \hat{V}_{\text{XC}}^{\text{high}}, \quad (8)$$

$$\hat{H}^{\text{low}} = \hat{T} + \hat{V}_{\text{local}} + \hat{V}_{\text{Hartree}} + \hat{V}_{\text{XC}}^{\text{low}}. \quad (9)$$

The total energy can thus be found by minimising the quantity

$$E^{\text{EMFT}} = \min_{\{K^{\alpha\beta}\}, \{\chi_\alpha\}} \text{Tr}[\mathbf{K}\mathbf{H}^{\text{EMFT}}], \quad (10)$$

with respect to the NGWFs and elements of the density kernel  $\mathbf{K}$ , using the conventional methods available in ONETEP. The Hamiltonian is constructed as follows,

1. The total electron density  $n(\mathbf{r})$  is constructed from the full system NGWFs and kernel, from which  $V_{XC}^{low}(\mathbf{r})$  is calculated.
2. The active subsystem density  $n^{AA}(\mathbf{r})$  is constructed using the subsystem terms and the subsystem XC potentials  $V_{XC}^{low,A}(\mathbf{r})$  and  $V_{XC}^{high,A}(\mathbf{r})$  calculated.
3. Final EMFT potential can be written as

$$V_{XC}^{high}(\mathbf{r}) = V_{XC}^{low}(\mathbf{r}) + \left( V_{XC}^{high,A}(\mathbf{r}) - V_{XC}^{low,A}(\mathbf{r}) \right). \quad (11)$$

with which we can construct the high-level Hamiltonian.

Although block orthogonalisation is found to work when just the density kernel is being optimised, it does not do so generally for the optimisation of the NGWFs. Because of this, there is an option to optimise the NGWFs at the lower level of theory first (in all regions), and then fix them for an optimisation of the kernel under EMFT.

If you would like to use hybrid functionals with embedding, there are two things to bear in mind. Firstly, only hybrid-in-semi local DFT calculations are currently supported – hybrid-in-hybrid calculations are not possible. Secondly, the species in the `species_swri-[swri name]` block must match the species in the active region exactly. Anything else will give incorrect results. Otherwise, the set-up of the hybrid functional calculation is identical to a normal ONETEP calculation.

### 3 Keywords

- `species_ngwf_regions` (block): This block defines which species are in which region. Each line of the block corresponds to a distinct region. The species within each region do not necessarily have to be physically next to one another. If this block is not defined, it is assumed that there is only one region, containing all the species in the system.
- `do_fandt` (logical): Controls whether a freeze-and-thaw (F+T) optimisation of the NGWFs is performed or not. This is a cruder form of embedding, where all regions are treated at the same level of theory, but each region’s NGWFs are optimised in turn, with the others frozen. Default F.
- `freeze_switch_steps` (integer): How many NGWF CG optimisation steps should be spent on each region before moving onto the next in a F+T calculation. `maxit_ngwf_cg` represents the total number of NGWF optimisation steps across all regions. A value less than 0 means that all NGWFs are optimised together i.e. no F+T takes place. Default -1.

- `use_emft` (logical): Controls whether an EMFT calculation is performed, as described above. Default `F`.
- `active_region` (integer): Defines which region is the active region – 1 means the species on the first line in the `species_ngwf_regions` block constitute the active region, 2 means the second line, and so on. Default 1.
- `active_xc_functional` (string): Defines what functional is used as the higher level of theory within EMFT. Default is the value of `xc_functional` i.e. no difference between the regions.
- `freeze_envir_ngwfs` (logical): Controls whether the environment NGWFs should ever be optimised or not. Default `F`.
- `use_emft_follow` (logical): Controls whether the EMFT calculation is only performed after a regular calculation, so the NGWFs are optimised at the lower level of theory first, before applying EMFT. Default `F`.
- `use_emft_lnv_only` (logical): Controls whether only the kernel is optimised within EMFT, with the NGWFs optimised at the lower level of theory and then fixed. Usually used in conjunction with `use_emft_follow`. Default `F`.
- `emft_lnv_steps` (integer): Controls the number of LNV kernel optimisation steps to be used in conjunction with `use_emft_lnv_only`. Default 10.
- `block_orthogonalise` (logical): Controls whether the environment NGWFs are orthogonalised with respect to the active region NGWFs, as described above. Default `F`.
- `parallel_scheme` (string): Defines the parallel scheme used for the calculation. See Appendix for more information. Default `NONE`.
- `read_sub_denskern` (logical): Controls whether only diagonal blocks of the density kernel are read in when restarting. This is useful for starting an embedding calculation from two separate calculations on the individual regions, so you only have the diagonal blocks of the density kernel. Default `F`.
- `embed_debug` (logical): Turns on verbose printing for debugging of embedding functionalities. Default `F`.

The most reliable way to run EMFT calculations is to have `use_emft`, `use_emft_follow`, `use_emft_lnv_only` and `block_orthogonalise` all set to

T. These can be set to F (most sensibly in reverse order i.e. `block_orthogonalise` first), but the calculation may become more unstable, depending on the system, the regions chosen and the functionals chosen.

## 4 Example input file

```
!=====  
! Input for calculation with the ONETEP program      !  
!                                                    !  
! O2 and H2 form the embedded system to be treated  !  
! at the higher level of theory, O1 and H1 are the  !  
! environment treated at the low-level.            !  
!=====  
  
%block species_ngwf_regions  
O2 H2  
O1 H1  
%endblock species_ngwf_regions  
  
task: SINGLEPOINT  
cutoff_energy 1000 eV  
write_forces: T  
xc_functional: LDA  
active_xc_functional: PBE  
  
use_emft: T  
use_emft_follow: T  
use_emft_lnv_only: T  
block_orthogonalise : T  
parallel_scheme: HOUSE  
  
%block species_atomic_set  
H1 "SOLVE"  
O1 "SOLVE"  
H2 "SOLVE"  
O2 "SOLVE"  
%endblock species_atomic_set  
  
%block species  
H1 H 1 1 7.0  
O1 O 8 4 7.0  
H2 H 1 1 7.0  
O2 O 8 4 7.0
```

```

%endblock species

%block species_pot
H1 "pseudo/hydrogen.recpot"
O1 "pseudo/oxygen.recpot"
H2 "pseudo/hydrogen.recpot"
O2 "pseudo/oxygen.recpot"
%endblock species_pot

%block lattice_cart
      30.000000000      0.000000000      0.000000000
      0.000000000     30.000000000      0.000000000
      0.000000000      0.000000000     30.000000000
%endblock lattice_cart

%block positions_abs
O1      16.203224001      15.100000000      11.536063353
H1      15.100000000      15.100000000      10.100000000
H1      15.100000000      15.100000000      12.991451046
O2      12.600158789      15.100000000      17.306583960
H2      13.051873252      13.656398529      18.308114239
H2      13.051873252      16.543601471      18.308114239
%endblock positions_abs

```

## 5 Interaction with other functionalities

### 5.1 Fully tested

- Energy and forces calculations
- Hybrid-in-semi local DFT
- Restarting calculations

### 5.2 Should work, not thoroughly tested

- Geometry optimisation
- Finite displacement phonons
- Molecular dynamics
- Conduction NGWF optimisation
- LR-TDDFT
- Ensemble DFT

- Kernel DIIS
- Implicit solvent
- QNTO
- NAO
- Cutoff Coulomb
- Spin polarised calculations
- Some properties calculations (eigenstates, Mulliken charges, plotting, DoS)

### **5.3 Not compatible with embedding**

- Hubbard calculations
- DMFT
- PAW
- cDFT
- Bandstructure calculations
- DMA
- EDA
- Electronic transport
- Hybrid-in-hybrid DFT
- NEB
- EELS
- Polarisable embedding
- Transition state searching
- DDEC

Any functionalities missed above are likely to not work with embedding.

## 6 Appendix: Parallel strategies with embedding

In a normal ONETEP calculation, atoms are distributed across the available MPI processes according to a ‘parallel strategy’. This determines how resources such as matrix elements will be spread across the MPI environment in order to reduce the communication between nodes and maximise the efficiency of the calculation. Details on maximising parallel efficiency are available via the ONETEP documentation and website.

As part of the embedding infrastructure, each subsystem is given its own parallel strategy. This contains all information relating to the distribution of resources across the MPI nodes available to the calculation, which are determined by the parameter `PARALLEL_SCHEME`. There are three settings for the distribution of resources during an embedding calculation:

- **NONE**: All subsystems are treated completely independently, with atoms distributed across all available processors as though the other subsystems do not exist. The number of MPI processes cannot be greater than the number of atoms in the smallest subsystem. For example, if there are 8 processors available then each will hold atoms and data from all subsystems, though the calculation will fail if any subsystem has less than 8 atoms (or possibly slightly more if the space-filling curve is in use). This is the default setting for testing but is not recommended for practical calculations due to the constraint on the number of processors.
- **SENATE**: Nodes are partitioned evenly between all subsystems. For example, if there are 8 processors and 2 subsystems, then each will be allocated 4 processors, regardless of the number of atoms in each subsystem. Unlike the **NONE** setting, there is no upper bound on the number of processors which may be used, so user discretion is advised.
- **HOUSE**: Divides the processors proportionally between all subsystems, with a minimum of 1 processor per subsystem. For example, if we have two subsystems consisting of 15 and 5 atoms each, then with 8 processors each subsystem will be allocated 6 and 2 nodes respectively. At a minimum all subsystems are granted 1 processor — if we had two subsystems with 1 and 100 atoms in our 8 processor example, then they will receive 1 and 7 processor respectively. Like **SENATE**, there is no upper bound on the number of processors that can be allocated and finding a sensible setting is left to the user.

**HOUSE** is the recommended setting for running calculations, the others are mainly of use for testing. Since they should all produce the same results, any significant differences may be a sign of an underlying problem, so comparing them is a useful consistency check.



## References

- [1] P. Huang and E. M. Carter, *Annu. Rev. Phys. Chem.*, **59**, 261–290, (2008).
- [2] A. S. P. Gomes and C. R. Jacob, *Annu. Rep. Prog. Chem., Sect. C: Phys. Chem.*, **108**, 222–277, (2012).
- [3] M. E. Fornace, J. Lee, M. Kaito, F. R. Manby, T. F. Miller, *J. Chem. Theory Comput.*, **11**, 568–580, (2015).
- [4] F. Ding, F. R. Manby and T. F. Miller, *J. Chem. Theory Comput.*, **13**, 1605–1615, (2017).