cDFT input check-list

[gibo, g.teobaldi@liv.ac.uk, November 2013]

This is a short check-list meant to help the setting up of a constrained-DFT (cDFT) simulation. ONETEP implements some rather extensive check of the input (as specified in .dat file). In spite of this, users may be still capable of creating erroneous inputs which I could not think of. If you experience disaster (commutators larger than 1.E-3, NWGFs- and cDFT-optimisation stuck at the same value of the gradient for many iterations, or NaN (Not A Number) and '***' outputs from ONETEP), please do get in touch.

● Can I start by running a cDFT geometry-optimisation and/or Molecular Dynamics straightaway?

Currently, this is **not** possible. ONETEP will not allow you to run a geometry-optimisation/molecular dynamics simulation unless a `.cdft` file (containing the cDFT-potentials, `Uq/s`) is present. This is meant to force you to first make sure that your cDFT-settings let you obtain a good convergence in a single-point calculation for the system and constrained solution you are interested in.

● What projectors am I telling ONETEP to use?

Is `cdft_read_projector=F` (this is the default) and `cdft_multi_proj=F` (this is the default) in your input?

If so, you will be using as cDFT-projectors the orbitals, of angular momentum *L-projectors*, of a hydrogenic atom of atomic number *Z-projector*, provided *Z-projector* is positive in the `%block constrained_dft` (Z>0). Conversely, if you have entered a negative *Z-projector* value in the `%block constrained_dft` (Z<0), you will be using the numerical orbital (of angular momentum *L-projectors*) for the given cDFT-atom as cDFT-projectors.

Is `cdft_multi_proj=T` (this is NOT the default) in your input?

If so, you will be using as cDFT-projectors the numerical orbitals, obtained from the pseudo-atom solver. You will be using as many cDFT-projectors as NGWFs for the given cDFT-site. Thus, to know the number and angular momentum of your cDFT-projectors, you need to refer to the specific settings of the cDFT-site in the `%block species_atom_set` and the relevant output in the standard .out file.

Is `cdft_read_projector=T` in your input?

If so, you will use the projectors stored in the `.tightbox_hub_projs` file (or experience a crash if this file is not present). Mind that the file will contain different classes of projectors (hydrogenic orbitals, numerical orbitals, self-consistent projectors) depending on the settings you used for the cDFT-run which generated the `.tightbox_hub_projs` file. This means that, unless one wants to experience a crash, the projectors in the `.tightbox_hub_projs` file should have been generated with the same entries for *Z-projector and L-projector* in `%block constrained_dft` **OR** `cdft_multi_proj` and `%block species_atom_set` as in the current calculation.

● Is this the first single-point cDFT submission or do I want to restart a single-point calculation?

For submission from scratch the user can chose between starting the cDFT-run from previously obtained DFT-NGWFs and Density Kernel (DKN), by setting (`read_denskern=T` and `read_tightbox_ngwfs=T`), **or not** (`read_denskern=F` and `read_tightbox_ngwfs=F`).

To restart a cDFT-single point run (from the latest Uq/s-potentials) or to run cDFT-geometry optimisations/Molecular Dynamics it is necessary to activate `cdft_continuation=T`. **Mind** that if, instead of hydrogenic or numerical orbitals as cDFT-projectors (see above) one wants to use/keep using pre-optimised projectors (stored in `.tightbox_hub_projs` file) it is necessary to activate also `cdft_read_projector=T` in a restart.

● Are the columns of the %block constrained_dft requesting the right targets for the given cDFT-flavour?

MIND that, as explained in the description of the keywords, different columns of the %block constrained_dft are used depending on the selected cDFT-modes (cdft_atom_charge, cdft_atom_spin, cdft_group_charge/spin_acceptor/donor, cdft_group_charge/spin_diff). Are you requesting the right targets in terms of atomic-population, group-populations, atomic magnetic-moments and group magnetic-moments?

Mind also that the cdft_group_charge/spin_acceptor/donor, cdft_group_charge/spin_diff cDFT-modes require that targeted population, magnetic-moment and differences are entered explicitly with the corresponding _TARGET keyword. If you have forgotten entering the relevant _TARGET keyword in your input, the simulation will stop and ONETEP will tell you about it.

● Have I set CDFT_GROUP_CHARGE_UP/DOWN_ONLY=T?

Remember that, for CDFT_GROUP_CHARGE_UP/DOWN_ONLY=T, only the corresponding spin-channel will be constrained in cdft_group_charge_acceptor/donor and cdft_group_charge_diff runs. Accordingly, you should target populations for one-spin channel only (**not** for the UP+DOWN channels).

● Is the sign of $U_{q/s}$ correct for the group_charge/spin_acceptor/donor/diff run?

Remember that atoms in acceptor- and donor-group are identified by mean of the sign of the Uq/s in the %block constrained_dft. Thus, **acceptor-group atoms** need to be assigned negative (e-attractive) $U_{q/s}$ (**$U_{q/s}$<0**), whereas **donor-group atoms** need to be assigned positive (e-repulsive) $U_{q/s}$ (**$U_{q/s}$>0**).

● Are my constraints compatible with the spin (=N_{UP}-N_{DOWN}) keyword in the input (.dat) file?

Remember that ONETEP optimises the Density Kernel keeping the number of UP (N_{UP}) and DOWN (N_{DOWN}) electrons fixed. As a result, the net magnetization of the system (N_{UP}-N_{DOWN}) is also fixed. Have you introduced any incongruence between the spin=0,1,2,etc keyword and the cDFT-ones in your input? If your targeted cDFT-solution results in a high-value (i.e. > 0 μ_B) magnetization of the **total** system, the spin keyword should reflect it.

● What values of min/maxit_lnv, maxit_ngwf_cg and maxit_cdft_u_cg should I use?

Based on the tests run so far, and considering that the cDFT-potentials (Uq/s) are iteratively optimised at each (1st) step of the NGWFs-optimisation line-search, the following choices seem to be reasonable:

```
maxit_palser_mano      : -1
kerfix                 : 2
maxit_pen              : 0
minit_lnv              : 5
maxit_lnv              : 10
maxit_ngwf_cg          : 60
lnv_check_trial_steps  : T
lnv_threshold_orig     : 1.0e-10

maxit_cdft_u_cg        : 5
```

Increasing maxit_cdft_u_cg above 5 is hardly going to accelerate the convergence of the cDFT-run (the NGWFs will change at the following step of NGWFs-optimisation), decreasing minit_lnv below 5 might be risky.

• To what value should I initialise the cDFT-potentials (Uq/s)?

In the current implementation, unless cdft_guru=T, in which case the Uq/s entered in the %block constrained_dft will be used, the absolute value of the Uq/s cDFT-potentials are internally initialised to 1 eV (their original sign is, of course, maintained). For spin-excitation |Us|=1eV may be too large, and initialising the |Us| with 0.1-0.3 eV may accelerate convergence (this has been tested only on triplet excitations in benzene dimers).

• What is the difference between a cdft_atom_charge run and a cdft_group_charge_acceptor/donor one with one-atom group?

Whereas the cdft_atom_charge=T mode allows independent (i.e. potentially different) constraining potentials to be applied to the UP and DOWN spin-channels, for **one-atom** cdft_group_charge_acceptor/donor=T the same Uq will be applied to both the UP and DOWN spin-channel. Activation of cdft_group_charge_up/down_only=T in cdft_group_acceptor/donor modes allows to optimise Uq for only a spin-channel, leaving the other spin-channel unconstrained.

• Have I chosen a meaningful cdft_cg_max for the cDFT-mode I wish to use?

For cDFT-runs with only one cDFT-group in the system (cdft_group_charge/spin_acceptor/donor modes) and group_charge/spin_diff runs, **only one** cDFT-potential (Uq/s) will be optimised in the cDFT-loop. Accordingly, for these cases it is recommended to perform the cDFT-optimisation via a steepest descendent algorithm (cdft_cg_max=1).

• How do I obtain the population of the cDFT-sites for a standard DFT-run?

Performing a single-point (task=singlepoint) fixed-Uq/s (maxit_cdft_u_cg=0) cDFT-run using very small (e.g. 1.E-60) Uq/s-potentials in the %block constrained_dft and setting output_detail=VERBOSE will result in the cDFT-population of all the cDFT-sites being printed in the standard output (.out) file. To obtain atom-specific (instead of atomic_species-specific) information on the population of the cDFT-sites, it is necessary to set CDFT_PRINT_ALL_OCC=T.

Mind. The population of a given cDFT-site depends critically on the projector used. Make sure you decide your cDFT-targets from the DFT-populations obtained with the same set of projectors!

• How do I optimise self-consistently the projectors for a given geometry?

In analogy with DFT+U simulations, self-consistent optimisation of the cDFT-projectors (for a fixed geometry) is activated by setting task=HUBBARDSCF in the .dat file. It is recommended to start the task=HUBBARDSCF run from pre-optimised PAO-cDFT projectors (Z<0 in %block constrained_dft), cDFT-potentials (Uq/s), NGWFs and DKN. This is accomplished, regardless of the keywords specific to the chosen cDFT-flavour, by making sure the input file (.dat) contains:

```
task                    : HUBBARDSCF
hubbard_max_iter        : 40  # perform 40 Hubbard-SCF iterations
read_denskern           : T
```

```
read_tightbox_ngwfs    : T
cdft_read_projectors   : T
cdft_continuation      : T
```

The percentage of the latest-optimised cDFT-NGWFs to be used as new cDFT-projectors is controlled by the `hubbard_proj_mixing` keyword [0 < |`hubbard_proj_mixing`| < 1], with `hubbard_proj_mixing=1` meaning that the latest optimised-NGWFs are used entirely as new cDFT-projectors.

Mind. Provided `cdft_read_projector=T`, the cDFT-projectors in the `.tightbox_hub_projs` file are used (entirely) as new projector during the 1st HUBBARDSCF iteration.

● Is self-consistent optimisation of the cDFT-projectors at the DFT-geometry a good idea?

From the tests so far, this is **not a good idea**. For tightly constrained systems (for instance an hypothetical $N^{(+)}=N^{(-)}$ excitation) a cDFT `task=HUBBARDSCF` run at the DFT-optimised geometry may result in very slow convergence. The recommended procedure is to **first** optimise the cDFT-geometry using numerical orbitals (PAO) as projectors (Z-projector <0 in `%block constrained_dft`) and **then** optimise the cDFT-projectors at the PAO-cDFT optimised geometry.