PGOPT Manual

Huanchen Zhai

October 18, 2018

0.1 Installation

1. Download `anaconda2`. See `https://www.continuum.io/downloads`.

```
curl https://repo.continuum.io/archive/Anaconda2-4.2.0-Linux-x86_64.sh \
-o Anaconda2-4.2.0-Linux-x86_64.sh
bash Anaconda2-4.2.0-Linux-x86_64.sh
source ~/.bashrc
```

2. Upgrade `git`.

```
conda install git
source ~/.bashrc
```

3. Install `theano`, `reportlab`, `dill`.

```
pip install --upgrade theano
pip install --upgrade reportlab
pip install --upgrade dill
echo 'import theano;print "ok"' | python
```

4. Install `VASP` (optional). Add the following to `~/bashrc`.

```
export VASPHOME=???/vasp.5.4.1/bin
export VASP_PP_PATH=???/potentials
```

5. Install `TURBOMOLE` (optional). Add the following to `~/bashrc`.

```
export TURBOMOLE_HOME=???/TURBOMOLE
```

6. Add the following to `~/bashrc`.

```
module load intel/16.0.2
BASE=~/program
export STMOLE_HOME=$BASE/STMOLE
export ACNNHOME=$BASE/ACNN
export PGOPTHOME=$BASE/PGOPT
export PATH=$STMOLE_HOME:$PATH
export PATH=$PGOPTHOME:$ACNNHOME:$PATH
```

7. Add the following to `~/bashrc` (optional, for zs/zr and job submission in PBS systems).

```
export TMP_HOST=...
export PROJECT_NAME=...
```

8. Compile Fortran modules for `ACNN`.

```
module load intel/16.0.2
cd $ACNNHOME/formod
make
```

0.2 Basic Usage

1. Make a directory, called "local directory".

```
mkdir Au3Pd3-MGO
cd Au3Pd3-MGO
```

2. Initiation of a calculation.

- Gas phase cluster Pt13, generating 100 structures.

```
pgopt init Pt13 100
```

Recommended additional settings for gas phase accuracy (increasing squares and planar structures):

```
pgopt set creation order 3
pgopt set creation 2d 0.3
```

Note that `order 1` means the added atom will bind to at least one (usually one) existing atom. `order 2` binds to at least two. `order 3` means the first half of atoms are `order 1` and the remaining `order 2`. The default is `order 2`. `2d` gives the ratio of 2D structures.

- Gas phase cluster Pt13B2, B atoms can be anywhere.

```
pgopt init Pt13B2 100
```

or equivalently

```
pgopt init B2Pt13 100
```

- Gas phase cluster Pt13B2, B atoms must be added finally.

```
pgopt init "Pt13|B2" 100
```

- Gas phase cluster Pt9, with two CO molecule attached (added finally).

```
pgopt init "Pt9|(CO)2" 100
```

- Pt5 deposited at MgO surface.

```
pgopt init Pt5 100 --surface=mgo
```

For custom surface, for example, `--surface=mgox`, files named `mgox.xyz` and `mgox.json` are required at local directory. The surface files can be generated using `pgopt surfgen` from VASP CONTCAR or XYZ files (see the example section). Sample `mgox.xyz`:

```
72
CELL = 12.878     12.878     14.595 (4.2927)
  Mg       2.045958       8.705236      10.669160
  Mg       4.416078       6.334879      10.667900
   O       2.046976       6.337625      10.618636
   O       4.413058       8.705486      10.616889
... ...
```

Sample `mgox.json`:

```
{
    "space_group": "-F 4 2 3",
    "space_group_ref": [0.0, 0.0],
    "unit_cell": [4.2927, 4.2927, 4.2927],
    "citation": "",
    "fix": "0-4"
}
```

where the first five atoms are fixed. `fix` is optional. `fix` value can also be written as `0,1,2,5-7`. No `fix` (here) means fixing all surface element types. `"fix": "-1"` means not fixing any atoms. If `fix` is removed or changed later by `set opts`, the surface atoms will not be fixed or the fixing will be changed accordingly.

- Pt5 deposited at MgO surface, but Pt5 will be read from given structures.

```
pgopt init "(Pt5)" 100 --surface=mgo
pgopt set molecules+ Pt5
```

For custom clusters or molecules, the name `Pt5` will be arbitrary, like `ptcl` or `cytosine`. A file named `Pt5.xyz` is required at local directory. Multiple structures series in a single file is acceptable. Note the comment line format (for surface case, the number of surface atoms must be indicated). Common organic and small molecules, like C2H4, do not require user provided `xyz` file. Sample `Pt5.xyz`:

```
221
MAG:0:0.0:1 = 0.0 SURF = 216
   Mg       0.000000       0.000000       0.000000
   Mg       2.149873       0.000000       2.149873
   Mg       0.000000       2.149873       2.149873
... ...
```

- Pt5 deposited at MgO surface, read from given structures, with C2H4 attached.

```
pgopt init "(Pt5)|(C2H4)" 100 --surface=mgo
pgopt set molecules+ Pt5
```

Common organic and small molecules, like C2H4, do not require user provided `xyz` file. These molecule files are in `$ACNNHOME/library/molecules.zip` file. The provided molecules are

```
2-butyne,AlCl3,AlF3,BCl3,BF3,BeH,C2Cl4,C2F4,C2H2,C2H3,
C2H4,C2H5,C2H6,C2H6CHOH,C2H6NH,C2H6SO,C3H4_C2v,C3H4_C3v,
C3H4_D2d,C3H6_Cs,C3H6_D3h,C3H7,C3H7Cl,C3H8,C3H9C,C3H9N,
C4H4NH,C4H4O,C4H4S,C5H5N,C5H8,C6H6,CCH,CCl4,CF3CN,CF4,
CH,CH2NHCH2,CH2OCH2,CH2SCH2,CH2_s1A1d,CH2_s3B1d,CH3,
CH3CH2Cl,CH3CH2NH2,CH3CH2O,CH3CH2OCH3,CH3CH2OH,CH3CH2SH,
CH3CHO,CH3CN,CH3CO,CH3COCH3,CH3COCl,CH3COF,CH3CONH2,
CH3COOH,CH3Cl,CH3NO2,CH3O,CH3OCH3,CH3OH,CH3ONO,CH3S,
CH3SCH3,CH3SH,CH3SiH3,CH4,CN,CO,CO2,COF2,CS,CS2,Cl2,
ClF,ClF3,ClNO,ClO,F2,F2O,H2,H2CCHCN,H2CCHCl,H2CCHF,
H2CCO,H2CCl2,H2CF2,H2CO,H2COH,H2O,H2O2,H3CNH2,HCCl3,
HCF3,HCN,HCO,HCOOCH3,HCOOH,HCl,HF,HOCl,Li2,LiF,LiH,N2,
N2H4,N2O,NCCN,NF3,NH,NH2,NH3,NO,NO2,Na2,NaCl,O2,O3,
OCHCHO,OCS,OH,P2,PF3,PH2,PH3,S2,SH,SH2,SO,SO2,Si2,
Si2H6,SiCl4,SiF4,SiH2_s1A1d,SiH2_s3B1d,SiH3,SiH4,
SiO,bicyclobutane,butadiene,cyclobutane,cyclobutene,
isobutane,isobutene,methylenecyclopropane,trans-butane
```

- Surface/Bulk-only calculation.

  ```
  pgopt init H0 1 --surface=bulk --cores=32
  ```

  Where `bulk.xyz` and `bulk.json` are required in local directory. In `hoffman2`, use `--cores` to specify how many cores should be used for each configuration. For other hosts, `--nodes` might be used.

- Other creation parameters include `~sampling`, `~hydro`, etc.

- `~sampling` can be `random/all/grid`. `all`: for adsorbates. generate <`given number`> of initial structures for each isomers in the mole file. In total, <`given number`> * <`number of isomers`> structures will be generated. `random` (default): for adsorbates. In total <`given number`> will be generated. Each time it will select a random isomer from the given bare cluster isomers. `grid.`<`m`>`.`<`n`>: for PES sampling. Must use xyfix in DFT parameter. No molecules.

- `~hydro` (boolean, default `false`): if it is true, then hydrogen atoms will be added to the surface. The number of H atoms will be determined based on the number of top layer surface atoms (within 0.5 Angstrom).

- (deleted) `~mnohydro` (boolean, default `false`): if it is true, then the molecule will not bind to the cluster via H atom. This parameter is now deleted. `~mnohydro` = false is `~nomoleelems` = [] and `~mnohydro` = true is `~nomoleelems` = ['H'].

- `~rmhydro` (integer, default 3): only valid when `~hydro` is true. This number determines how many H atoms should be removed because cluster occupies some space.

- `~lowpos` (double, default -999.0): Usually this is negative. It indicates cluster atoms below `max_z + lowpos` z position will be rejected.

- `morder` (integer, default 1). Can be 1 or 2. When the added element is not a atom, but a molecule, then `morder` will be used. It means that the 1 or 2 atoms in the added molecule will bind to one existing atoms at once. Note that this has the different meaning from `order`. The number indicates the number of atoms in the molecule, not the cluster.

- `~nocluselems` (string list, default empty). Only works with `morder = 1` for adding molecules or `order = 1` for surface cases (no 2D). The molecule will not bind to the cluter/surface elements that are in the list. If an element name starts with `-`, for example, `-X`, then the element names after this notation will only be applied when the current adding atom is of element `X`.

- `~nomoleelems` (string list, default empty). Only works with `morder = 1` for adding molecules. The molecule will not bind to the cluster via the atoms in molecule that are listed. Can use the same notation `-X` as `~nocluselems`.

- The combination of `~mnohydro` and `morder` can enforce binding patterns. For example, `~mnohydro = false, morder = 2` will gives H-pi binding.

- `~loworderelems` (string list, default empty). If the `order` is two, then only the order of elements in this list will be change to 1. This will be useful when considering adding H atoms to the shell of cluster. H atoms require first order. For example,

  ```
  pgopt set creation ~loworderelems H,O
  ```

3. Parameter settings for a stage. The calculation is divided into several stages. Later stages may depend on data analysis of previous stages. Each stage may consist of several stage-steps. Different steps have different computational parameters or accuracy. During each stage, the work will be assigned for all structures for the first stage-step, then for all structures for the second stage-step, and so forth. At some time, there may be some workers work for the first step, while some others work for the second.

- Single step relaxation using `VASP`.

```
pgopt set relax
pgopt set opts+ "encut=280;lwave=T"
```

The default max number of relaxation steps is 500. The default number of relaxation steps for checking is 100. These can be changed using

```
pgopt set args max_step 100
pgopt set args step 20
```

The default number of SCF steps is 100. This can be changed using

```
pgopt set opts^ "scf(iter=400)"
```

- Multiple steps relaxation using `VASP`, for example, using smaller `encut` and `prec` at the beginning.

```
pgopt set 0 relax
pgopt set 0 opts+ "encut=280;lwave=T;cell=15"
pgopt set 1 relax
pgopt set 1 opts+ "encut=400;lwave=T;cell=15;PREC=Accurate"
pgopt set 1 sources- create
pgopt set 1 sources+ inherit,local,max
```

Note that for gas phase, when `VASP` is used the cell size must be set manually here. For surface case, this is not necessary and it will be automatically read from surface file comment line.

- Setting k-points or DFT+U for `VASP`. Note that the element order will be alphabetic.

```
pgopt set 0 opts+ "kmethod=MP;kgrid=1:1:1"
pgopt set 1 opts+ "kmethod=Gamma;kgrid=8:8:2"
pgopt set 0 opts+ "LDAU=T;LDAUTYPE=2;LDAUL(-1,2);LDAUU(0.0,3.5);LDAUJ(0.0,0.0)"
```

The number after `set` is the step number. If omitted, it will be `0`. No space is allowed for options. For bulk relaxation, make sure there is no implicit `fix`.

```
pgopt set 0 opts- "fix(O,Ti)"
pgopt set 0 opts+ "isym=0;isif=3"
```

- Remove previous defined steps.

```
pgopt set none
pgopt set none
```

- Energy and frequency calculation using `VASP`. For single point energy,

```
pgopt set 0 energy
pgopt set 0 opts+ "encut=400;lwave=T;cell=15;PREC=Accurate"
pgopt set 0 sources^ restart,1.0-filter.1
```

For Vertical Ionization Potential (VIP) calculation,

```
pgopt set 1 energy
pgopt set 1 args charge 1
pgopt set 1 opts+ "encut=400;lwave=T;cell=15;PREC=Accurate"
pgopt set 1 sources^ restart,1.0-filter.1
```

For frequency calculation, default `ibrion` is `5`.

```
pgopt set 0 freq
pgopt set 0 args read_restart true
pgopt set 0 opts+ "encut=400;lwave=T;cell=15;PREC=Accurate;IBRION=8"
pgopt set 0 sources^ restart,1.0-filter.1
```

- Calculation using `TURBOMOLE`.

```
pgopt set relax-tm
```

- Add additional structures to previous calculation stage.

```
pgopt set relax
pgopt set 0 opts+ "encut=400;lwave=.TRUE.;cell=15;PREC=Accurate"
pgopt set 0 sources- create
pgopt set 0 sources+ read,../../square.xyz
pgopt set parallel secondary 1
```

where `square.xyz` is a series of structures provided at local directory. It will be copied to scratch directory automatically when issuing the master script command. The name after `secondary` is the stage name.

- Change surfaces during the steps.

```
pgopt set 0 relax
pgopt set 0 surf mgox
pgopt set 1 relax
pgopt set 1 surf mgoy
```

Note that the surface atoms at former steps must be all fixed, otherwise when changing the surface the cluster position will be shifted (the $x_{min}, y_{min}, z_{max}$ will be aligned). The top layers of several surfaces must be aligned before being used. In this case, `mgox.xyz`, `mgox.json`, `mgoy.xyz`, and `mgoy.json` must be provided at local directory or already in the program library. The `json` file provides the fixed atoms and cell size information. These information will be automatically updated in the `opts` arguments. The updated `opts` will be printed on the screen. To remove the surface for one step,

```
pgopt set 0 surf none
```

If no surface is set for the step, the surface will be that specified in `pgopt init`. If no surface is set at `pgopt init`, then the program will create gas phase structures instead of surface structures. Therefore, if you want to create initial structures (instead of reading existing files), there must be a surface set at `pgopt init`.

- Generating structures without doing optimization.

```
pgopt set tasks create
pgopt set relax
pgopt relax 1
pgopt para 1
```

The first command is to switch the task from `global` to `create`. The last command is directly running the master script instead of submitting the script. The directory of generated structures will be printed on the screen when finished.

4. Job script generation and submission.

- Master script generation and submission.

```
pgopt relax 1 --time=24:00:00
pgopt submit relax 1
```

`relax`, `energy` and `freq` are stage task names. They can be used directly as commands. For custom stage task names, `master` should be the command name, and task name can be specified using `--runt=`

```
pgopt master 2 --time=24:00:00 --runt=mixed
pgopt submit mixed 2
```

The number after `master` is stage name. It can also be non-numeric. The default time of master is 24 hours.

- Worker (torun) script generation and submission (with no dependence, like in `hoffman2`).

```
pgopt torun para 0 24 --time=24:00:00 --x
pgopt submit torun para
```

`--x` indicates job array, which is recommended for `hoffman2` and `cascade`. It should not be used for other hosts.

- Job dependence (type I): worker jobs should be executed after the master job starts (this is not a requirement). This is not useful and is not implemented for `hoffman2`. Under this case, the depend command should be inserted to modify the worker scripts.

```
pgopt torun para 0 24 --time=24:00:00
pgopt depend torun para --jobid=10001
pgopt submit torun para
```

The `jobid` number should be looked up from the result of previous issued `pgopt submit` command.

- Job dependence (type II): master script for later stages (or extending current stage time) should executed after the current master job finishes (this is a requirement, if one wants to submit two master scripts at the same time).

```
pgopt relax 1 --time=24:00:00
pgopt relax 2 --time=24:00:00
pgopt submit relax 1
pgopt depend relax 2 --jobid=10002
pgopt submit relax 2
```

The `jobid` number is looked up from the result of `pgopt submit relax 1` command. The following will automatically determine the depended jobid.

```
pgopt depend relax 1 --autodep
pgopt submit relax 1
pgopt depend torun para 0 10 --autodep
pgopt submit torun para 0 10
```

- Notes: If several jobs for the same master stage or the same individual worker are running at the same time, it will be problematic. However, after master or worker jobs finish their time and are killed, the same `submit` command can be used again to continue the calculation. Ideally, to continue a calculation, both `master` and `torun` are needed to re-submitted.

- When master job is running but no workers are running, the master job is consuming time but does nothing.

- When some workers are running but master is not, the workers are either doing their current task (for a specific assigned atomic structure) or, if they already finished their current task, consuming time but do nothing.

- Master/workers belonging to different local directories will not influence each other's job.
- For `TURBOMOLE`, different multiplicities are specified.

  ```
  pgopt relax 1 1-9 --time=24:00:00
  pgopt submit relax 1 1-9
  ```

  The number list `1-9` indicates the multiplicity `1,3,5,7,9` will be calculated separately. If no number list is given, it will be `0`, which means the multiplicity will be automatically relaxed, which is only available in periodic softwares like `VASP`.
- The automatically generated file `CMD-HISTORY` at local directory includes all commands typed. If another similar calculation for a different system should be done in the future, one can `source` the copy of this file in a new directory.

  ```
  mkdir Au5Pd1-MGO
  cd Au5Pd1-MGO
  cp ../Au3Pd3-MGO/CMD-HISTORY ./cmd
  vim cmd
  source cmd
  ```

  But be careful when using `jobid` feature because the `jobid` is dynamic. And all `pgopt log/report` command should be removed from the `cmd` file.
- Directly run master without submitting the job (not recommended).

  ```
  pgopt para 1 0 --runt=relax
  ```

  where the first number is stage name, the second number is the multiplicity (can be omitted). `--runt` specify the stage name and can be omitted. This does not provide several features of the `pgopt submit relax`:

  (1) This will not auto-sync files from scratch to local. But the sync can be done manually by

  ```
  pgopt sync
  ```

  (2) This will not auto-check workers that are not responding for more than one hour and kill them. But the checking can be done manually by

  ```
  pgopt check --runt=para --maxt=3600 --delete
  ```

  where 3600 is the minimal seconds for no responding, and with `--delete` the jobs of detected workers will be deleted, otherwise it will only print the workers' id on the screen but taking no further action.

  (3) This will not stop all worker jobs when all structures are finished. You may need to kill the worker jobs manually.

5. Result checking and filtering.

- Show summary information when the jobs are running (or at any time, if you just want to check the status).

  ```
  pgopt log
  ```

  The columns are: stage name, multiplicity (0 means multiplicities are not distinguished), total/remaining/converged/failed/max-number-of-steps-reached/duplicated number of structures, current step index (count from zero), total number of step indices, current number of running processes (workers), total number of processes, state (can be init/paused/running/waiting/finished, time passed since the start time of current state, time since receiving last report from any workers.

- Check detailed information for master and individual workers when the jobs are running (or at any time). Assume the current stage name is `1` and multiplicity is `0` and the current directory is somewhere under local directory.

  ```
  cd `pgopt show`
  tail -f -n 100 master/1.0/par_log.txt.0
  ```

  The command `cd pgopt show` switches between local/scratch directories. The `par_log.txt` file is updated every 8 seconds at scratch directory and every 2 minutes at local directory. So it is advantageous to look at it at scratch directory.

- Filter results when the jobs are finished. Assume the current stage name is `1`.

  ```
  pgopt filter 1
  ```

  The full form with omitted options is `pgopt filter 1 0 --runt=relax`. See `pgopt para` for the meaning of options. During the runtime a smaller (stricter) similarity threshold is used. When filtering a bigger value is often used. By this way, the lowest energy isomer for a series similar structures will be retained. Only the result from the last step of that stage will be filtered. The alignment may also be done after the filtering.

- Final report generation.

  ```
  pgopt report
  ```

  Final PDF report and filter structures are in `report` (local) directory. Intermediate files generated by `VASP` or `TURBOMOLE` are in `restarts` (scratch) directory (compressed).

6. Notes.

- For bader charge/single point energy calculation, remember to set `rescue_bad` parameter to false (the default is false now). When running parallel frequency (FD), this parameter will be automatically set to false.

  ```
  pgopt set parallel rescue_bad false
  ```

- If `rescue_bad` is true, when a calculation has a scf-non-convergence, the program will automatically change the geometry slightly and recalculate. This is good only for global optimization. But dangerous for single point energy calculations.

- make super cells for a xyz file (structure series):

  ```
  pgopt enlarge abc.xyz --size=3:3:2 --cell=3.0342942:0:-1.5171471:2.6277765:3.2235271698
  ```

- generate a pdf for a xyz file (structure series):

  ```
  pgopt draw xx.xyz
  pgopt draw xx.xyz --plot-force --number=6 --force-factor=0.3 --rotmat=0:1:0:0 --ortho
  ```

  `plot-force` will take the forth to sixth column as vibration data and plot the vibration modes.

- select some xyz from a xyz series:

  ```
  pgopt select abc.xyz 4,9,12,14,17,44 "--output=selected.xyz"
  ```

0.3 Examples

1. Pt7 gas phase with VASP.

```
pgopt init Pt7 500
pgopt set relax
pgopt set opts+ "encut=300;lwave=T;cell=18"
pgopt set creation order 3
pgopt set creation 2d 0.2
pgopt relax 1
pgopt torun para 0 24 --x
pgopt submit relax 1
pgopt submit torun para
```

2. Pt7 gas phase with VASP, using Basin-Hopping (Monte Carlo and Relaxation) Algorithm.

```
pgopt init Pt7 10 "--cores=64"
pgopt set mc
pgopt set mc temperature 1500
pgopt set mc short-distance-factor 0.9
pgopt set relax
pgopt set do-monte-carlo T
pgopt set opts+ "encut=300;lwave=T;cell=18"
pgopt set opts^ "scf(iter=100)"
pgopt set args step 50
pgopt set creation order 3
pgopt set creation 2d 0.2
pgopt relax 1
pgopt torun para 0 10 --x
pgopt submit relax 1
pgopt submit torun para
pgopt mclog
```

3. Pt5 deposited on MgO Surface, switching between small/big surfaces.

```
pgopt surfgen prep.xyz "--sname=mgo" "--fixz=5" "--gapz=14"
pgopt surfgen prep.xyz "--sname=mgor" "--fixz=11" "--gapz=14" "--removez=7"
pgopt init Pt5 10 "--surface=mgor" "--cores=32"
pgopt set mc
pgopt set mc temperature 1500
pgopt set mc short-distance-factor 0.85
pgopt set monte-carlo max-iter 20
pgopt set 0 relax
pgopt set 0 do-monte-carlo T
pgopt set 0 opts+ "encut=300;lwave=T;EDIFF=1E-4;EDIFFG=-0.05"
pgopt set 0 args max_step 350
pgopt set 0 args step 50
pgopt set 1 relax
pgopt set 1 surf mgo
pgopt set 1 opts+ "encut=400;lwave=T;EDIFF=1E-5;EDIFFG=-0.01"
pgopt set 1 args step 50
pgopt set 1 sources- create
pgopt set 1 sources+ inherit,local,max
pgopt set 1 args max_config 50
pgopt relax 1
pgopt torun para 0 10 --x
```

```
pgopt submit relax 1
pgopt submit torun para
```

4. Bader analysis of benzene.

```
pgopt init "(C6H6)" 1 "--cores=32"
pgopt set 0 relax
pgopt set 0 opts+ "encut=280;lwave=T;cell=15"
pgopt set 1 energy
pgopt set 1 opts+ "encut=280;lwave=T;cell=15;bader"
pgopt set 1 sources- restart,?
pgopt set 1 sources+ inherit,local,max
pgopt relax 1
pgopt torun para 0 1 --x
pgopt submit relax 1
pgopt submit torun para
```

5. Finite difference calculation (frequency calculation that is parallelized, restartable).

```
pgopt init Pt7 0 "--surface=alphaz" "--nodes=32"
pgopt set fd
pgopt set 0 energy
pgopt set 0 optsˆ "sigma=0.1;encut=400;ediff=1E-6;ediffg=1E-5;lreal=A;"
pgopt set 0 optsˆ "scf(iter=300);lwave=T;algo=Fast;npar=32"
pgopt set 0 sources- restart,?
pgopt set 0 sources+ read,init.xyz
pgopt set 1 energy
pgopt set 1 do-finite-diff T
pgopt set 1 args read_restart T
pgopt set 1 optsˆ "sigma=0.1;amix=0.2;bmix=2.0;encut=400;ediff=1E-7;"
pgopt set 1 optsˆ "prec=A;lreal=A;scf(iter=300);lmaxmix=4;lwave=T;algo=Fast;potim=0.01;"
pgopt set 1 optsˆ "npar=32;randmix;icharg=1;lcharg=T"
pgopt set 1 sources- restart,?
pgopt set 1 sources+ inherit,structs
pgopt relax 1 "--time=24:00:00"
pgopt torun para 0 24 "--time=1:00:00"
pgopt submit relax 1
pgopt submit torun para
pgopt fdlog
```

6. Pt7H14CH3 generation on alpha-Al2O3. Need to manually prepare a CH3 from CH4, because the default CH3 is the structure of CH3-.

```
pgopt init "Pt7|H14(CH3)" 100 "--surface=alphaz" "--nodes=12"
pgopt set molecules+ CH3
pgopt set creation ~lowpos 0.1
pgopt set creation ~loworderelems H
# pgopt set creation ~mmohydro T
pgopt set creation ~nomoleelems -CH3,H
pgopt set creation ~nocluselems -H,Al,H,O,C,-CH3,Al,H,O
pgopt set 0 relax
pgopt set 0 optsˆ "sigma=0.1;encut=400;ediff=1E-6;ediffg=1E-5;lreal=A"
pgopt set 0 optsˆ "scf(iter=300);lwave=T;algo=Fast;npar=12"
pgopt set 0 args step 50
pgopt set 0 args max_step 500
```

```
pgopt relax 1
pgopt torun para 0 10
pgopt submit relax 1
pgopt submit torun para 0 10
```

7. Pt7(CO)5 generation on alpha-Al2O3. Here in order to reduce number of configurations, we can let only C in CO bind to the cluster.

```
pgopt init "Pt7|(CO)5" 100 "--surface=alphaz" "--nodes=12"
pgopt set creation ~lowpos 0.1
pgopt set creation ~nomoleelems -CO,O
pgopt set creation ~nocluselems -CO,Al,C,O
pgopt set 0 relax
pgopt relax 1
pgopt torun para 0 10
pgopt submit relax 1
pgopt submit torun para 0 10
```

8. Pt7 MD calculation (temperature/tebeg = 1000 K, time step/potim = 1.0 fs).

```
pgopt init Pt7 0 "--surface=alphaz" "--nodes=32"
pgopt set 0 relax
pgopt set 0 opts^ "sigma=0.1;encut=400;ediff=5E-6;lreal=A;scf(iter=300)"
pgopt set 0 opts^ "lwave=F;algo=Fast;ibrion=0;potim=1.0;tebeg=1000;smass=0;iwavpr=11"
pgopt set 0 sources- create
pgopt set 0 sources+ read,sec-min.xyz
pgopt set 0 args max_step 11000
pgopt set 0 args step 500
pgopt set 0 args filter_type none
```

9. Pt7(CO)5 MC calculation.

```
pgopt init "Pt7|(CO)5" 10 "--surface=alphar" "--nodes=4"
pgopt set creation ~lowpos 0.1
pgopt set creation ~nomoleelems -CO,O
pgopt set creation ~nocluselems -CO,Al,C,O
pgopt set mc
pgopt set mc temperature 1500
pgopt set mc short-distance-factor 0.85
pgopt set mc max-iter 300
pgopt set mc detailed-balance F
pgopt set mc swap-site T
pgopt set mc keep-co T
pgopt set mc swap-site-make-space CO
pgopt set mc solid-move CO
pgopt set mc light-shell T
pgopt set 0 relax
pgopt set 0 do-monte-carlo T
pgopt set 0 opts^ "sigma=0.1;encut=300;ediff=1E-4;ediffg=1E-3;lreal=A;"
pgopt set 0 opts^ "scf(iter=300);lwave=F;algo=Fast"
pgopt set 0 args step 50
pgopt set 0 args max_step 500
pgopt relax 1 "--time=24:00:00"
pgopt transfer relax 1
pgopt torun para 0 10 "--time=24:00:00"
```

```
pgopt submit relax 1
pgopt submit torun para 0 10
```

10. Pt7H10(CH3) MC calculation.

```
pgopt init "Pt7|H10(CH3)" 1 "--surface=alpha4rz" "--nodes=8"
pgopt set molecules+ CH3
pgopt set creation ~lowpos 0.1
pgopt set creation ~loworderelems H
pgopt set creation ~nomoleelems -CH3,H
pgopt set creation ~nocluselems -H,Al,H,O,C,-CH3,Al,H,O
pgopt set creation number 10
pgopt set mc
pgopt set mc temperature 1500
pgopt set mc short-distance-factor 0.85
pgopt set mc max-iter 300
pgopt set mc detailed-balance F
pgopt set mc swap-site T
pgopt set mc swap-site-make-space CH3
pgopt set mc swap-site-rate 0.25
pgopt set mc keep-ch3 T
pgopt set mc solid-move CH3
pgopt set mc light-shell T
pgopt set 0 relax
pgopt set 0 do-monte-carlo T
pgopt set 0 opts^ "sigma=0.1;encut=300;ediff=1E-4;ediffg=1E-3;"
pgopt set 0 opts^ "lreal=A;scf(iter=300);lwave=F;algo=Fast;npar=16"
pgopt set 0 args step 50
pgopt set 0 args max_step 750
pgopt relax 1
pgopt transfer relax 1
pgopt torun para 0 10 "--time=12:00:00"
pgopt submit relax 1
pgopt submit torun para 0 10
```

11. Type `pgopt connect 1` in a converged global optimization calculation will generate the images for neb calculation. Then one can do neb calculation using the following.

```
pgopt init Pt7 0 "--surface=alphaz" "--nodes=20"
pgopt set 0 neb
pgopt set 0 sources^ read,images.xyz
pgopt set 0 opts^ "sigma=0.1;encut=400;ediff=5E-6;ediffg=-0.05;lclimb=F;"
pgopt set 0 opts^ "lreal=A;scf(iter=300);lwave=T;algo=Fast"
pgopt torun para 0 10 "--time=24:00:00"
pgopt neb 1 "--time=24:00:00"
pgopt submit neb 1
pgopt submit torun para 0 10
```