MOMAP

Tutorial 12

Numfreq Calculation with MOMAP

Version 2021B

December, 2021

MOMAP Tutorial 12

Version 2021B edited by:

Dr. Qikai Li

Last modified on Mar 8, 2025 by:

Dr. Qikai Li

Released by Hongzhiwei Technology (Shanghai) Co., Ltd

and Z.G. Shuai Group

The information in this document applies to version 2021B of MOMAP

Table of Contents

Numfreq Calculation with MOMAP

Note:

With g16, by default, the force constants are determined analytically if possible, by single numerical differentiation for methods for which only first derivatives are available, and by double numerical differentiation for those methods for which only energies are available. Please refer to the Freq keyword in Gaussian g16 manual for more details.

Thus, one can simply add a line, e.g., ftdipd = "porphine-s1.log" in momap.inp as shown below and skip the following time-consuming numfreq calculation altogather.

```
$ cat momap.inp
```

```
do_evc = 1
&evc
  ffreq(1) = "porphine-s0.log"
  ffreq(2) = "porphine-s1.log"
  ftdipd   = "porphine-s1.log"
/
```

Here we take ammonia as an example to show how to do the numfreq calculations with MOMAP. In this tutorial, we assume the gaussian QC package (g09 or g16) is properly installed and the scheduling system SLURM is used.

The three files, that is, a template gaussian input file, a control file, and a job script file, as shown below, are needed to carry out the numfreq calculations.

```
$ ls
```

```
ammonia.com  input  numfreq.slurm
```

Prepare a template gaussian input file

```
$ cat ammonia.com
```

```
%chk=ammonia.chk
%rwf=ammonia.rwf
%mem=10GB
%nprocs=8

#p td(root=1) b3lyp/3-21g geom=connectivity

Title Card Required

0 1
 N                  0.00000000   -0.00000000    0.07917814
```

```
 H                    0.00000000    1.14570194   -0.18474898
 H                   -0.99220698   -0.57285097   -0.18474898
 H                    0.99220698   -0.57285097   -0.18474898

 1 2 1.0 3 1.0 4 1.0
 2
 3
 4
```

Note: The `%nprocs` parameter in the generated gaussian input files will be adjusted automatically to agree with the value in the scheduling job script set by using `--ntasks-per-node` with SLURM, an example is shown as below:

```
$ cat PES-0/coord-0-00.com
```

```
%chk=coord-0-00.chk
%rwf=coord-0-00.rwf
%mem=10GB
%nprocs=20

#p td(root=1) b3lyp/3-21g geom=connectivity force nosymm

Title Card Required

0 1
N                  0.00000000   -0.00000000    0.07917814
H                  0.00000000    1.14570194   -0.18474898
H                 -0.99220698   -0.57285097   -0.18474898
H                  0.99220698   -0.57285097   -0.18474898

 1 2 1.0 3 1.0 4 1.0
 2
 3
 4
```

Prepare a control file

```
$ cat input
```

```
&control
  qctype = "gaussian"
  task   = "numfreq"
  fxyz   = "ammonia.com"
  symm   = .false.
  dx     = 0.01
/
```

Prepare a scheduling job script

```
$ cat numfreq.slurm
```

```
#!/bin/sh
```

```
#SBATCH --output=stdout.txt
#SBATCH --partition=short
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=20
#SBATCH --time=0-2:00:00

USER=`whoami`

module purge
module load anaconda/3
module load momap/2021B-mpich2
module load gaussian/g16.c01-avx2

# if the environment module is not available, we can use source to setup
# the running environments, please adjust to your specific situation!
#source /opt/MOMAP-2021B/env.sh
#export g16root=/export/home/gaussian/g16_C01_AVX2
#source $g16root/g16/bsd/g16.profile

#-------------------------------------------------------------------------------
export RUN_NUM_PPN=$SLURM_NPROCS
export RUN_QUEUE=$SLURM_JOB_PARTITION
export RUN_JOBNAME="numfreq"

#-------------------------------------------------------------------------------
jobname=${RUN_JOBNAME}
username=`whoami`

#-------------------------------------------------------------------------------
srun hostname -s | sort -n > nodefile

#echo "Starting Gaussian run at" `date`
#-------------------------------------------------------------------------------
qcpath=`which g16`
#-------------------------------------------------------------------------------
scratch=/scratch/$USER
program=numfreq.py
posfix=${RUN_JOBNAME}_${RUN_QUEUE}
#-------------------------------------------------------------------------------
# Do Gaussian PES scanning
#
#RUN_NUM_PPN=1
$program                              \
 -qctype           "gaussian"         \
 -qcpath           "$qcpath"          \
 -scratch          "$scratch"         \
 -nodefile         "nodefile"         \
 -ncore            "$RUN_NUM_PPN"     \
 -recalc           "no"               \
 -input            "input"            \
 -task             "numfreq"          \
 -lock             "lock_$posfix"     \
 -stop             "stop_$posfix"     \
 -test_rs          "no"               \
 -save_chk         "yes"              \
 -save_fchk        "yes"              \
 -search_chk       "yes"              \
 -coord            "yes"              \
 -scan             "yes"              \
 -energy           "yes"              \
 -ignore_log_err   "no"
```

```
#-----------------------------------------------------------------------------
echo "Finished Gaussian run at" `date`
```

The users may need to make some minor modifications to the above files as shown blow:

1. We assume `g16` is used in the job script, users may need to modify it according to one's specific situation.
2. The default control file of `numfreq.py` is `input`, users can change it by using the option `-input`.
3. We also need to designate the scratch location for the QC program, here we set it to `/scratch` in our computing cluster. In addition, if your computing nodes have enough memory, you may even set it to `/scratchmem` which is a symbolic link to `/dev/shm`.
4. The `numfreq.py` script uses the `nodefile` to judge if we can do parallel computing. Here we use the slurm job script to demand some nodes which are reserved as the node pool for the numfreq calculations.

Run the job

Once the above preparations are done, we can simply launch the slurm job to do the numfreq calculations.

```
$ sbatch numfreq.slurm
```

If the calculations are finished successfully, then the final result is put in file `PES-0/numfreq-es.out`.

Note: The example files with this tutorial are put under `$MOMAP_ROOT/tests/numfreq` with MOMAP 2021B.