# Users’ guide to FSindo

Kiyoshi Yagi
kiyoshi.yagi@riken.jp

Theoretical Molecular Science Laboratory
RIKEN Cluster for Pioneering Research

2022/03/15

- In this guide, I will illustrate how to use “sindo” to perform VSCF, VCI, VMP2, and VQDPT2 calculations.

- “sindo” is a command line based program. This guide assumes that you are familiar with basic commands in UNIX. Shell scripts are given for Bourne Shell (bash).

- This sample also assumes that you have sourced “sindovars.sh” and setup the path to “sindo”,

```
. /path/to/sindo/sindovars.sh
sindo < vci.inp > vci.out
```

where “/path/to/sindo” indicates your installation directory, for example,

```
/path/to/sindo -> /home/yagi/pgm/sindo-4.0_220312
```

- In this guide, we will use the potential energy surface (PES) that are already pre-computed. To see how to generate the PES data, look into the usage of the MakePES program.

# Contents

1. Basic usage (1.water)
    - 1.1. VSCF
    - 1.2. VCI
    - 1.3. VMP2
    - 1.4. VQDPT2
    - 1.5. Summary

2. Infrared spectrum of $H_2CO$ (2.h2co)

3. Optimized coordinate VSCF (3.ethylene)

4. References

Appendix: List of all options

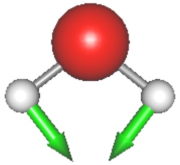
Sample files are found in sindo-4.0/doc/FSindo/sample_FSindo

# 1. Basic usage

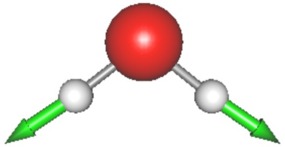
In this section, I will demonstrate the basic usage of sindo with a water molecule as an example. Proceed to 1.water,

```
> cd 1.water
> ls
h2o.minfo  q0.pot  q2.pot    q3.pot    q3q2.pot …
```

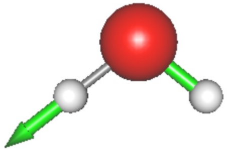
h2o.minfo includes the equilibrium geometry, harmonic frequencies, and vibrational displacement vectors. They can be visualized by JSindo.



Q1
HOH bending



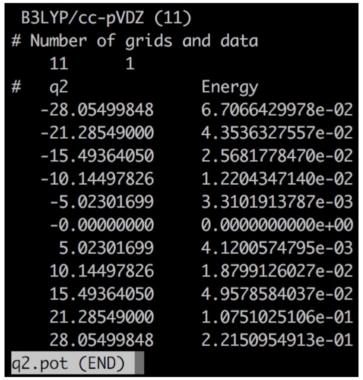
Q2
Sym. OH stretching



Q3
Asym. OH stretching

*.pot includes the information of the grid potential.

| | |
|---|---|
| eq. | q0.pot |
| 1MR | q1.pot, q2.pot, q3.pot |
| 2MR | q2q1.pot, q3q1.pot, q3q2.pot |
| 3MR | q3q2q1.pot |



```
 B3LYP/cc-pVDZ (11)
# Number of grids and data
    11       1
#   q2                 Energy
   -28.05499848     6.7066429978e-02
   -21.28549000     4.3536327557e-02
   -15.49364050     2.5681778470e-02
   -10.14497826     1.2204347140e-02
    -5.02301699     3.3101913787e-03
    -0.00000000     0.0000000000e+00
     5.02301699     4.1200574795e-03
    10.14497826     1.8799126027e-02
    15.49364050     4.9578584037e-02
    21.28549000     1.0751025106e-01
    28.05499848     2.2150954913e-01
q2.pot (END)
```

# 1-1. Vibrational self-consistent field method

vscf.inp is an input file to carry out VSCF calculations.

```
#--- [  INPUT PARAMETER  ]
&mol minfofile='h2o.minfo' /          input minfo file
&sys maxmem=10 /                      max memory in MB
```

```
#--- [  VIB  ]
&vib  MR=3  vmaxAll=10  vscf=.t. /
```

- MR : Mode coupling order of the PES
- vmaxAll : Max num of quanta of harmonic oscillator basis sets for all modes.
- vscf : true invokes VSCF.

```
#--- [ TARGET STATES ]
&states  fund=.t. /        Targets all fundamental levels.
```

```
#--- [  VSCF  ]
&vscf  Maxitr=20  Ethresh=1.D-03 /
```

- Maxitr : Max iteration for VSCF
- Ethresh : Threshold of convergence in $cm^{-1}$.

The options are given in a free format using the namelist of Fortran, that is, you can put space and new lines as you like as long as the options are defined in a section between &name … /. For example,

```
#--- [  VIB  ]
&vib  MR=3  vmaxAll=10  vscf=.t. /
```

and

```
#--- [  VIB  ]
&vib
  MR=3
  vmaxAll=10
  vscf=.t.
/
```

are the same, but

```
#--- [  VIB  ]
&vib  MR=3  vmaxAll=10 /
vscf=.t.
```

is different because “vscf = .t.” is out of &vib … /.

Run the job by,

```
> sindo < vscf.inp > vscf.out
```

The output, vscf.out, looks as follow:

```
>> BASIS FUNCTIONS
   MODE :      1       2       3
   MAXV :     10      10      10
   FREQ :  1659.40  3752.61  3853.49
```

Setting of HO basis sets. Frequencies are taken from the minfo file.

```
>> POTENTIAL

   [  OPTIONS  ]

     MR        =       3
     MCS_CUTOFF = 0.10E-03
     MCS_GRID   = -.10E+01
     POTDIR     = ./

     1MR-PEF

     o GRID PEF

      MODE=   1, GRID=  11    B3LYP/cc-pVDZ (11)
      MODE=   2, GRID=  11    B3LYP/cc-pVDZ (11)
      MODE=   3, GRID=  11    B3LYP/cc-pVDZ (11)
```

The information of the PES is printed. pot files in the current directory are read by the program. It's a good practice to check if the PES is specified in the way you intended.

1MR grid PES with 11 grid points.

```
    2MR-PEF

    o GRID PEF

     MODE=   2   1, GRID=  11  11    B3LYP/cc-pVDZ (11)
     MODE=   3   1, GRID=  11  11    B3LYP/cc-pVDZ (11)
     MODE=   3   2, GRID=  11  11    B3LYP/cc-pVDZ (11)


    3MR-PEF

    o GRID PEF

     MODE=   3   2   1, GRID=  11  11  11    B3LYP/cc-pVDZ (11)
```

2MR grid PES with 11 grid points.

3MR grid PES with 11 grid points.

```
o INITIAL GUESS FROM (CONTRACTED) HARMONIC OSCILLATOR

 -- (ITERATION) -------- (EOLD) -------- (ENEW) ----- (DELTA E) --
               1         4687.67         4589.42     -0.983D+02
               2         4589.42         4586.72     -0.270D+01
               3         4586.72         4586.64     -0.809D-01
               4         4586.64         4586.64     -0.254D-02
               5         4586.64         4586.64     -0.800D-04
 ------------------------------------------------------------------
    E(VSCF)=    4586.63552176
```

VSCF iteration

Zero-point Energy

```
>> VIRTUAL VSCF ENERGIES

   o VSCF STATES         TOTAL ENERGY       E-E0
      1_1                6184.36878830      1597.73326654
      2_1                8220.27988057      3633.64435881
      3_1                8388.85466071      3802.21913894
```

Virtual VSCF energies for the fundamental levels.

“m_1” means the 1st excited state of mode m, that is, the fundamental excitation of mode m. The virtual states are specified by the &state group.

VSCF calculations with the grid PES correspond to the direct VSCF method. See Ref. [1] for more details.

# 1-2. Vibrational configuration interaction method

vci.inp is an input file to carry out VCI calculations.

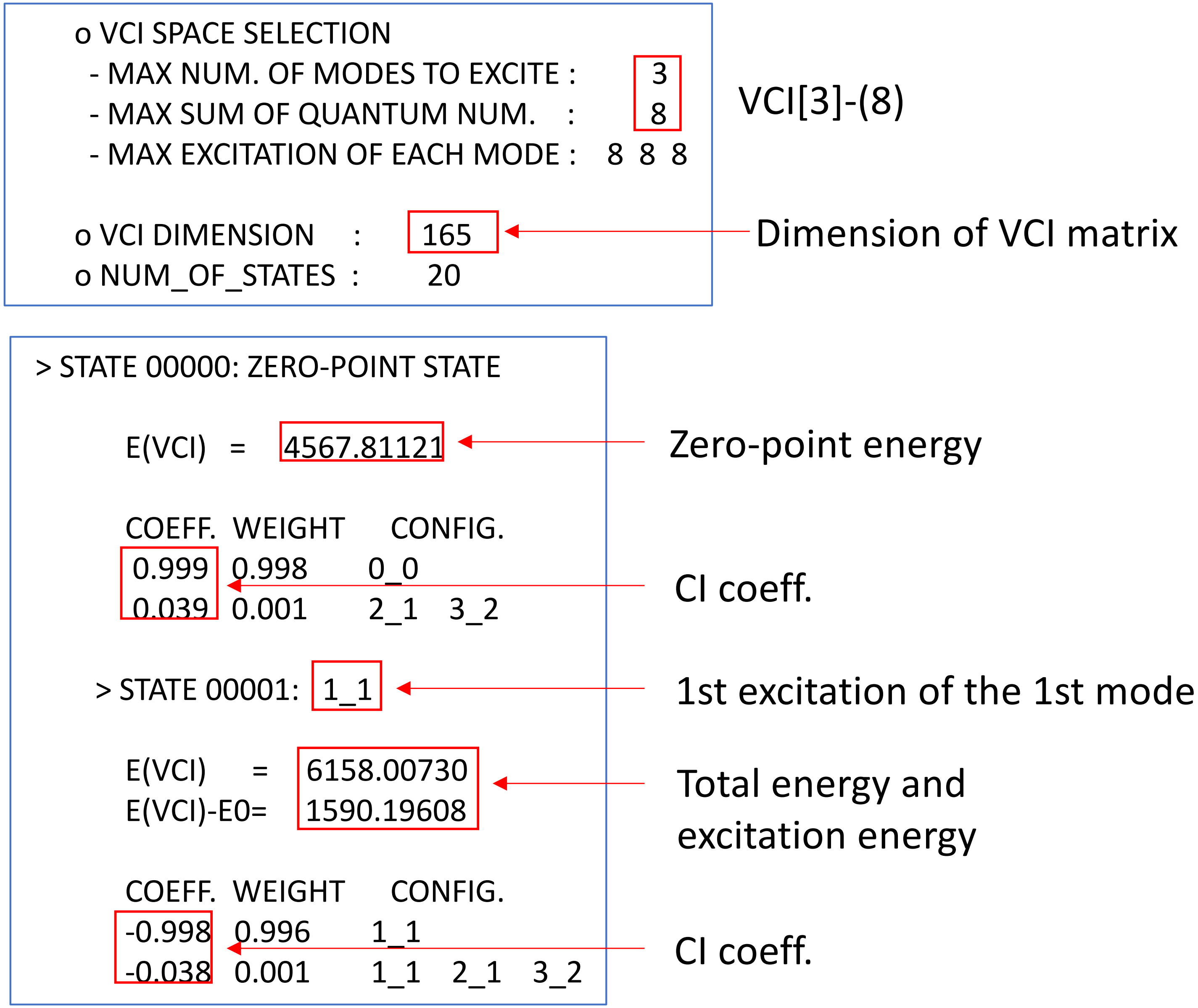
```
#--- [  VIB  ]
&vib  MR=3  vmaxAll=10  vscf=.t. vci=.t. /
                                 vscf = .t. and vci = .t. invokes VSCF/VCI.

#--- [  VCI  ]
&vci  nstate=20  nCUP=3  maxSum=8 /
          •    nstate          : Number of states to obtain.
          •    nCUP            : Max number of modes to excite.
          •    maxSum          : Max sum of quantum numbers to excite.
```

The level of VCI calculations is classified as VCI[n]-(m), where n is the maximum number of modes that are excited simultaneously, and m is the maximum number of quantum numbers excited. n and m are specified by nCUP and maxSum, respectively, in &vci group. In this example, nCUP=3 and maxSum=8, so that we carry out VCI[3]-(8).

Run the job by,

```
> sindo < vci.inp > vci.out
```

The output, vci.out, looks as follow:



```
      o VCI SPACE SELECTION
        - MAX NUM. OF MODES TO EXCITE :       3
        - MAX SUM OF QUANTUM NUM.    :        8
        - MAX EXCITATION OF EACH MODE :   8  8  8

      o VCI DIMENSION     :       165
      o NUM_OF_STATES  :        20
```

VCI[3]-(8)

Dimension of VCI matrix

```
  > STATE 00000: ZERO-POINT STATE

         E(VCI)   =    4567.81121

         COEFF.  WEIGHT     CONFIG.
          0.999  0.998       0_0
          0.039  0.001       2_1    3_2

       > STATE 00001:  1_1

         E(VCI)     =    6158.00730
         E(VCI)-E0=    1590.19608

         COEFF.  WEIGHT     CONFIG.
         -0.998  0.996       1_1
         -0.038  0.001       1_1    2_1    3_2
```

Zero-point energy

CI coeff.

1st excitation of the 1st mode

Total energy and
excitation energy

CI coeff.

# 1-3. Vibrational Møller-Plesset perturbation method

vmp2.inp is an input file to carry out VMP2 calculations.

```
#--- [  VIB  ]
&vib  MR=3  vmaxAll=10  vscf=.t. vpt=.t. /
```

vscf = .t. and vpt = .t. invokes VMP2.

```
#--- [ TARGET STATES ]
 &states fund=.t.
nstate=3
target_state(2,1)=2
target_state(2,2)=1, target_state(3,2)=1
target_state(3,3)=2
/
```

All fundamental levels  
+  
(020), (011), (002)

```
#--- [  VPT  ]
&vpt  maxSum=4 /
```

maxSum    : Max quantum numbers for excitation.

The level of excitation is controled by selecting VSCF configurations according to the maximum difference in quantum numbers (n) with respect to target states. The method is designated as VMP2-(n). See Ref. [2] for more details. In the input, n is specified by maxSum.

Note that &vpt also has nCUP, which restricts the number of modes to excite simultaneously. The default is nCUP = MR.  Usually, you don‘t need to change this number.

VMP2 is a state-specific approach, where we specify the states of interest in the input. In &state, “nstate” specifies the number of state, and an array,

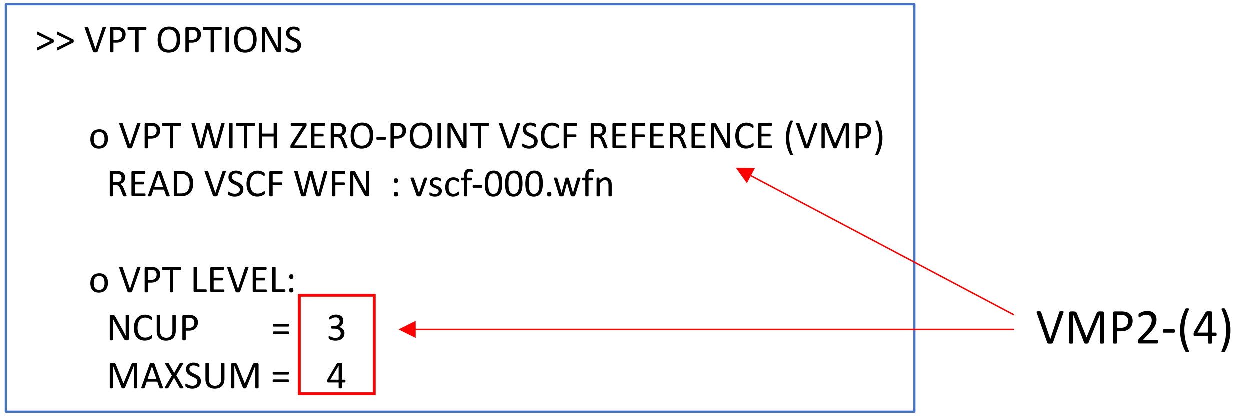
target_state(i,ns) = mi

specifies the quantum number (mi) of the i-th mode for the ns-th state. In addition, fund = .t. specifies all the fundamental excitations. In the example, we specify the fundamentals and overtone / combination states of OH stretching modes.

Run the job by,

```
> sindo < vmp2.inp > vmp2.out
```

The output, vmp2.out, looks as follow:



```
>> VPT OPTIONS

    o VPT WITH ZERO-POINT VSCF REFERENCE (VMP)
      READ VSCF WFN  : vscf-000.wfn

    o VPT LEVEL:
      NCUP      =  3
      MAXSUM =  4
```

VMP2-(4)

```
  o STATE 000:   ZERO-POINT STATE
    …
         E(VMP2)   =   4567.73900
```

VMP2 energy for the zero-point state.

```
  o STATE 001:   1_1
    …
         E(VMP2)      =   6159.84114
    …
         E(VMP2)-E0 =   1592.10214
```

VMP2 total energy and excitation energy for the fundamental of mode 1.

## 1-4. Vibrational quasi-degenerate perturbation theory

vqdpt2.inp is an input file to carry out VQDPT2 calculations.

```
#--- [  VIB  ]
&vib  MR=3  vmaxAll=10  vscf=.t. vqdpt=.t. /
```

vscf = .t. and vqdpt = .t. invokes VQDPT2.

```
&states
fund=.t.
nstate=3
target_state(2,1)=2
target_state(2,2)=1, target_state(3,2)=1
target_state(3,3)=2
/
```

All fundamental levels
+
(020), (011), (002)

```
#--- [  VPT  ]
&vqdpt  nGen=3 maxSum=4 /
```

- nGen : The number of iteration for generating the P space.
- maxSum : Max quantum numbers for excitation.

The level of excitation is controled by maxSum. nGen is the number of iteration to find VSCF configurations that are quasi-degereate to target states (P space). See Ref. [3] and [4] for more details.

Run the job by,

```
> sindo < vqdpt2.inp > vqdpt2.out
```

The output, vqdpt2.out, looks as follow:

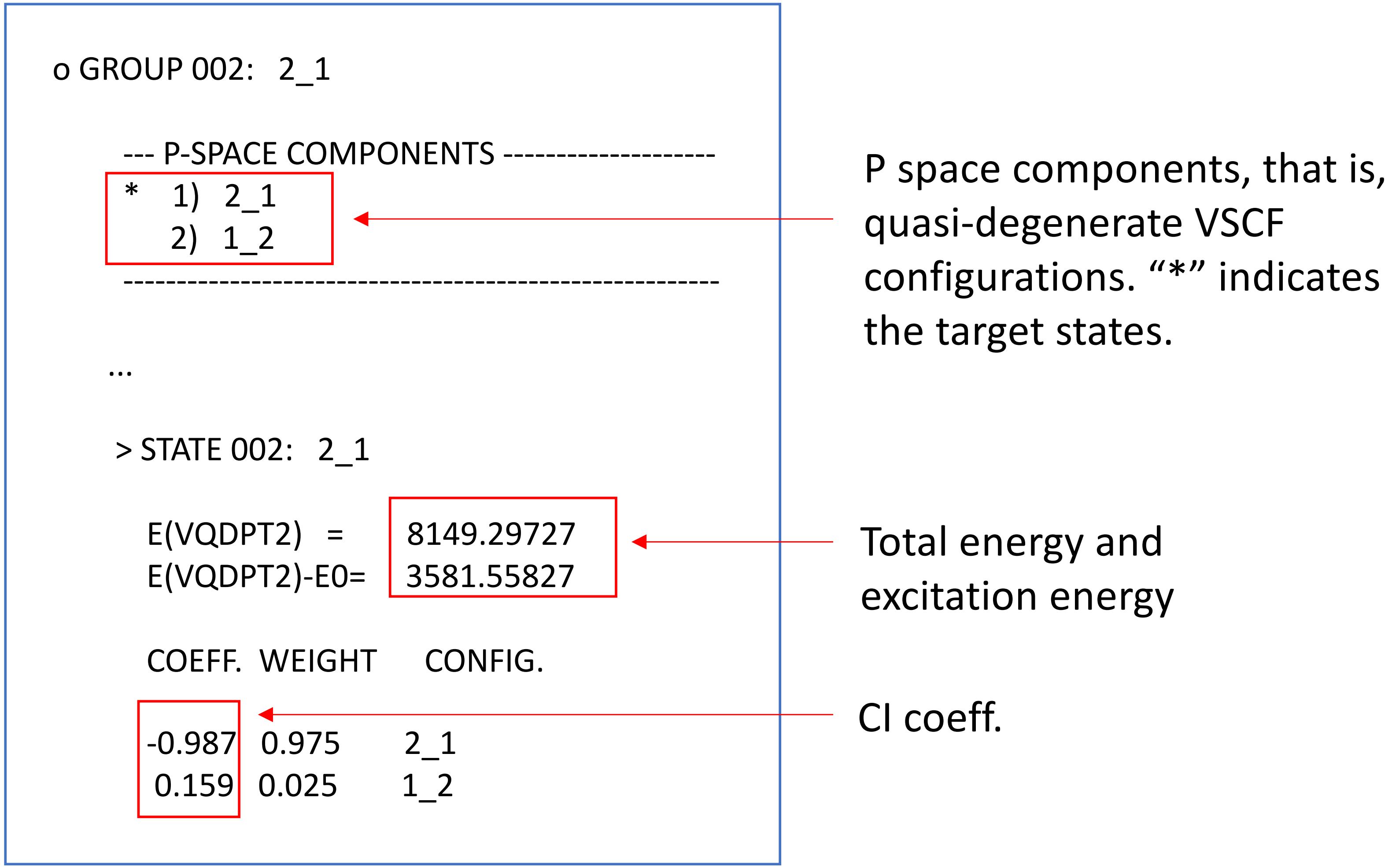
```
o P-SPACE CONSTRUCTION
     NGEN =    3
     THRESH_P0  = 0.50E+03
     THRESH_P1  = 0.10E+00
     THRESH_P2  = 0.50E-01
     THRESH_P3  = 0.90E+00
     P SET      =       0

    o Q-SPACE CONSTRUCTION
     NCUP   =    3
     MAXSUM =    4
```

Parameters for the P space.

Parameters for the Q space.

```
o STATE 000:   ZERO-POINT STATE
   …
       E(VMP2)   =   4567.73900
```

VQDPT2 energy for the zero-point state. Note that VQDPT2 coinsides with VMP2 if no degerate state exists. This is usually the case for the zero-point state.



```
o GROUP 002:   2_1

      --- P-SPACE COMPONENTS --------------------
      *    1)   2_1
           2)   1_2
      --------------------------------------------------------

     ...

     > STATE 002:   2_1

        E(VQDPT2)   =        8149.29727
        E(VQDPT2)-E0=        3581.55827

        COEFF.  WEIGHT      CONFIG.

        -0.987   0.975       2_1
         0.159   0.025       1_2
```

P space components, that is, quasi-degenerate VSCF configurations. "*" indicates the target states.

Total energy and excitation energy

CI coeff.

## 1-5. Summary

The results are summarized in a Table:

| | Harm | VCI[3]-(8) | VMP2-(4) | VQDPT2-(4) | Exp. |
|---|---|---|---|---|---|
| 1 | 1659.4 | 1590.2 | 1592.1 | 1592.1 | 1595 |
| 2 | 3752.6 | 3577.7 | 3581.7 | 3581.6 | 3652 |
| 3 | 3853.5 | 3657.8 | 3656.6 | 3656.6 | 3756 |
| 2x2 | 7505.2 | 7047.2 | 7062.2 | 7062.3 | 7201 |
| 2+3 | 7606.1 | 7083.9 | 7099.7 | 7092.1 | 7249 |
| 3x2 | 7707.0 | 7254.9 | 7245.8 | 7245.8 | 7445 |

Harmonic frequencies are significantly higher than the anharmonic ones. VCI, VMP2, VQDPT2 methods give similar numbers.

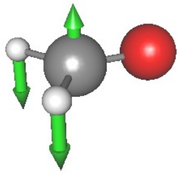
The agreement with experiment is worse compared to the reported data in Ref. [1]. Can you tell where the error comes from?

# 2. Infrared spectrum of $H_2CO$

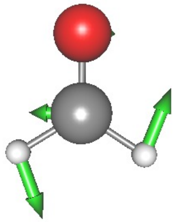
In this section, we will calculate the infrared spectrum of formaldehyde ($H_2CO$). Proceed to 2.h2co,

```
> cd 2.h2co
> ls
h2co-b3lyp-dz.minfo  pes_mrpes    runSindo.sh*    vqdpt2.inp
log/                 plotIR.gpi   vci.inp
```

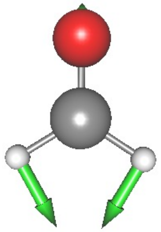
h2co-b3lyp-dz.minfo includes the normal coordinates:



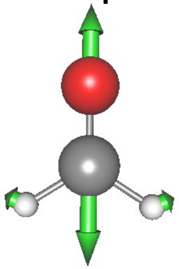
Q1
Out of plane



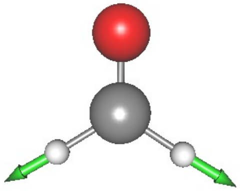
Q2
$CH_2$ rock



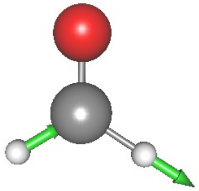
Q3
$CH_2$ bending



Q4
CO stretch



Q5
Sym. CH stretch



Q6
Asym. CH stretch

A multiresolution PES [5] is found in “pes_mrpes”:

```
> cd pes_mrpes
> ls *.pot
eq.pot       q2q1.pot    q4.pot  …
```

The PES data has been generated at the following level:

| | **Electronic Structure** | **Functional Form** |
|---|---|---|
| 1MR | CCSD(T)/cc-pVTZ | Grid-PES (11 points) |
| Strongly coupled terms (MCS>10) | CCSD(T)/cc-pVTZ | Grid-PES (9 points) |
| Weakly coupled terms (MCS>1) | B3LYP/cc-pVDZ | Grid-PES (7 points) |
| Other terms | B3LYP/cc-pVDZ | QFF |

The data of dipole moment surfaces (DMS) are also stored in this directory, which have been generated at the B3LYP/cc-pVDZ level.

These data were generated in Section 3 of Users’ guide to MakePES. See the guide for further details.

We will calculate the infrared (IR) spectrum of $H_2CO$ by VCI and VQDPT2 methods. The input files are vci.inp and vqdpt2.inp.

```
> ls *inp
vci.inp   vqdpt2.inp
```

vci.inp has the following options in addition to those for VCI:

```
 …
 &mrpes
  mopFile='prop_no_1.mop’
  mcs_cutoff=-1.0D-03
 /
```

The name of a mop file to read QFF data

A threshold value for MCS to cutoff the PES. A negative value turns off the cutoff.

```
&vib  MR=3  vmaxAll=10  vscf=.t.  vci=.t.  prpt=.t.  /
```

prpt=.t.: Invokes the calculation of properties.

```
…
&prpt  MR=3  vciprpt=.t.  infrared=.t. /
```

Invokes the calculation of IR spectrum for VCI wavefunction.

```
 &IRspectrum
  minOmega=800.0  maxOmega=4000.0
  delOmega=1.0  fwhm=5.0
 /
```

The IR spectrum for 800 - 4000 $\text{cm}^{-1}$ with a resolution of 1 $\text{cm}^{-1}$

Full width half maximum of Lorentz functions vqdpt2.inp has the same options except that the VQDPT wavefunction is specified in &prpt:

```
&prpt MR=3  vqdptprpt=.t.  infrared=.t. /
```

runSindo.sh is a script to run the job. “POTDIR” is an environment variable to tell the program where the PES and DMS data are located.

```
export POTDIR=./pes_mrpes

sindo < vqdpt2.inp > vqdpt2.out 2>&1
sindo < vci.inp       > vci.out    2>&1
```

We now run the job:

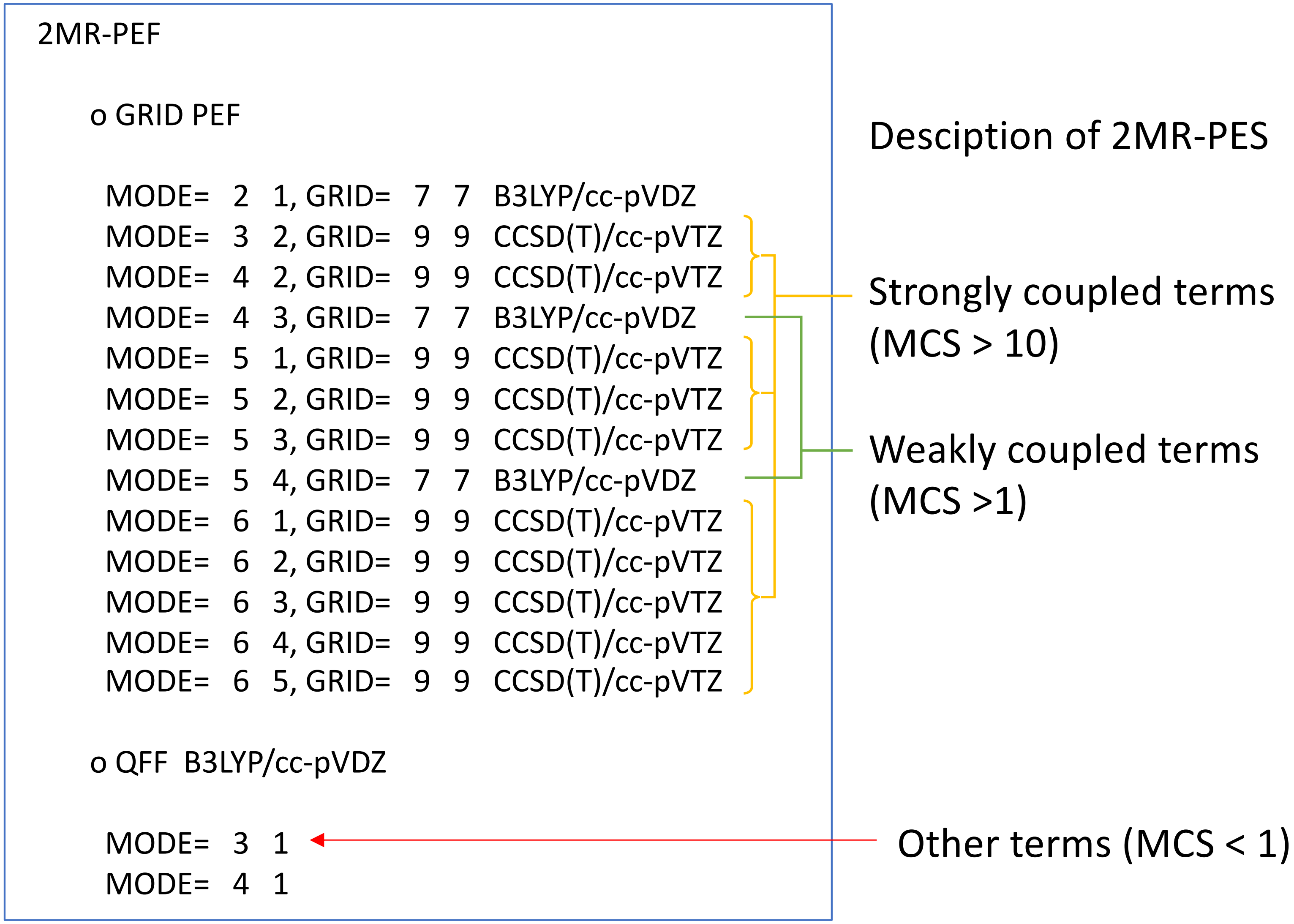
```
> ./runSindo.sh
```

Let’s first check the information of the PES. We find the same output for both vci.out and vqdpt2.out as follow:

```
>> POTENTIAL

  [ OPTIONS ]

    MR         =        3
    MCS_CUTOFF = -.10E-02
    MCS_GRID   = -.10E+01
    POTDIR     = ./pes_mrpes/
    MOPFILE    = prop_no_1.mop


    1MR-PEF

    o GRID PEF

     MODE=   1, GRID=  11   CCSD(T)/cc-pVTZ
     MODE=   2, GRID=  11   CCSD(T)/cc-pVTZ
     MODE=   3, GRID=  11   CCSD(T)/cc-pVTZ
     MODE=   4, GRID=  11   CCSD(T)/cc-pVTZ
     MODE=   5, GRID=  11   CCSD(T)/cc-pVTZ
     MODE=   6, GRID=  11   CCSD(T)/cc-pVTZ
```

The PES data is retrieved from “pes_mrpes”. The QFF data is set to “prop_no_1.mop”.

1MR grid PES with 11 grid points obtained at the CCSD(T)/cc-pVTZ level.



```
2MR-PEF

   o GRID PEF

     MODE=  2  1, GRID=  7  7  B3LYP/cc-pVDZ
     MODE=  3  2, GRID=  9  9  CCSD(T)/cc-pVTZ
     MODE=  4  2, GRID=  9  9  CCSD(T)/cc-pVTZ
     MODE=  4  3, GRID=  7  7  B3LYP/cc-pVDZ
     MODE=  5  1, GRID=  9  9  CCSD(T)/cc-pVTZ
     MODE=  5  2, GRID=  9  9  CCSD(T)/cc-pVTZ
     MODE=  5  3, GRID=  9  9  CCSD(T)/cc-pVTZ
     MODE=  5  4, GRID=  7  7  B3LYP/cc-pVDZ
     MODE=  6  1, GRID=  9  9  CCSD(T)/cc-pVTZ
     MODE=  6  2, GRID=  9  9  CCSD(T)/cc-pVTZ
     MODE=  6  3, GRID=  9  9  CCSD(T)/cc-pVTZ
     MODE=  6  4, GRID=  9  9  CCSD(T)/cc-pVTZ
     MODE=  6  5, GRID=  9  9  CCSD(T)/cc-pVTZ

   o QFF  B3LYP/cc-pVDZ

     MODE=  3  1
     MODE=  4  1
```

Desciption of 2MR-PES

Strongly coupled terms (MCS > 10)

Weakly coupled terms (MCS >1)

Other terms (MCS < 1)

```
3MR-PEF

o GRID PEF

 MODE=   5   4   3, GRID=   7   7   7   B3LYP/cc-pVDZ
 MODE=   6   3   2, GRID=   9   9   9   CCSD(T)/cc-pVTZ
 MODE=   6   4   2, GRID=   9   9   9   CCSD(T)/cc-pVTZ

o QFF  B3LYP/cc-pVDZ

 MODE=   3   2   1
 MODE=   4   2   1
 MODE=   4   3   1
 …
```

A similar description for 3MR-PES. Note that many terms are now classified to “others”.

Note that the labels “B3LYP/cc-pVDZ” and “CCSD(T)/cc-pVTZ” come from the input file for MakePES, specifically, the <title> keyword of <qchem>. It is quite important to give an appropriate name here. For example, if you name B3LYP for CCSD(T) jobs, you may get screwed up at this stage.

```
<qchem id="ene">
   <program      value="gaussian" />
   <dryrun        value="false"/>
   <removefiles value="true" />
   <title              value="CCSD(T)/cc-pVTZ" />
   <template      value="GaussianTemplate3" />
 </qchem>
```

We may find the energy levels of the fundamentals as before. The results are summarized in a Table:

| | Harm | VCI[3]-(6) | VQDPT2-(4) | Exp. |
|---|---|---|---|---|
| 1 | 1186.9 | 1162.4 | 1163.1 | 1167 |
| 2 | 1254.0 | 1248.7 | 1249.9 | 1249 |
| 3 | 1515.1 | 1507.5 | 1508.5 | 1500 |
| 4 | 1831.8 | 1750.3 | 1749.7 | 1746 |
| 5 | 2863.0 | 2767.0 | 2770.8 | 2782 |
| 6 | 2916.2 | 2834.7 | 2828.6 | 2843 |

Here, we see good agreement between experiment and theory (both VCI and VQDPT2), while the harmonic frequencies deviate from the experiment, in particular, for the high frequency modes (4, 5, 6).

After the output of energy levels, the program enters into PRPT module to calculate the infrared spectrum:

```
( ENTER PROPERTY MODULE )

  >> RUN OPTIONS
    MR    =     3
     ...
```

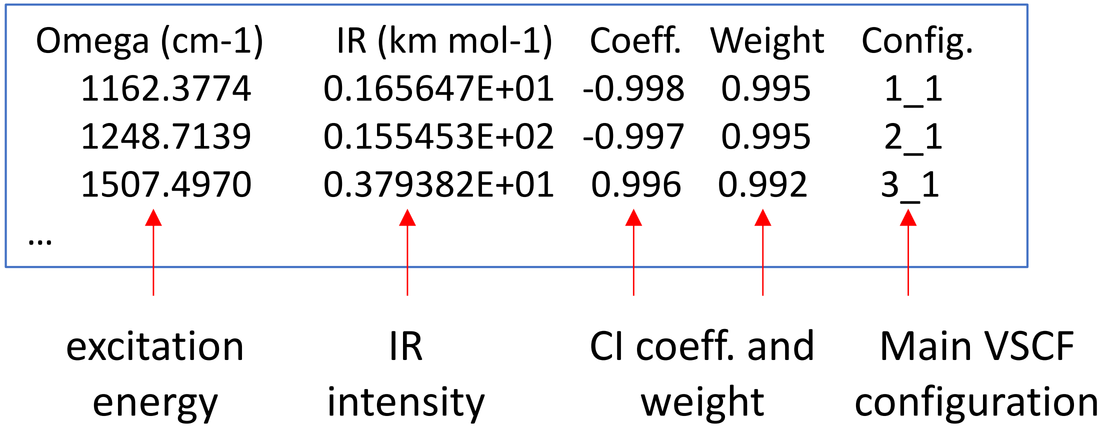
There is an output for the DMS, which is similar to the PES:

```
  >> PROPERTY SURFACE
      MR        =       3
      EXT       = .dipole
      POTDIR     = ./pes_mrpes/

      1MR-PRPT SURFACE

       MODE=   1, GRID=  11   B3LYP/cc-pVDZ
       MODE=   2, GRID=  11   B3LYP/cc-pVDZ
       MODE=   3, GRID=  11   B3LYP/cc-pVDZ
       ...
```

The results of IR calculations are written to *.data and *.spectrum,

```
> ls *data *spectrum
vci-IR.data      vci-IR.spectrum
vqdpt-IR.data    vqdpt-IR.spectrum
```

vci/vqdpt-IR.data look like follow:



```
 Omega (cm-1)     IR (km mol-1)   Coeff.  Weight   Config.
    1162.3774     0.165647E+01   -0.998   0.995     1_1
    1248.7139     0.155453E+02   -0.997   0.995     2_1
    1507.4970     0.379382E+01    0.996   0.992     3_1
…
```

excitation energy

IR intensity

CI coeff. and weight

Main VSCF configuration

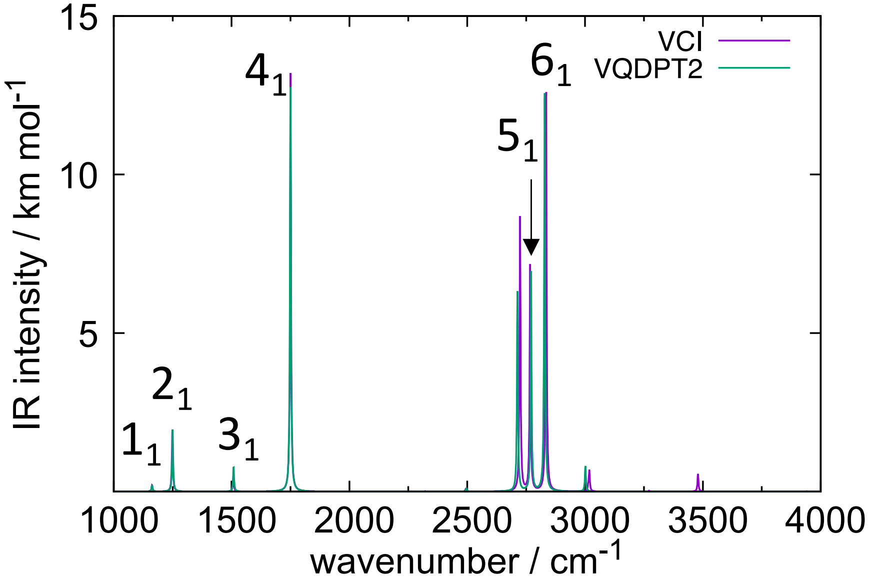
vci/vqdpt-IR.spectrum has the frequency and the IR intensity in the first and second column, respectively. You can make a plot using gnuplot by,

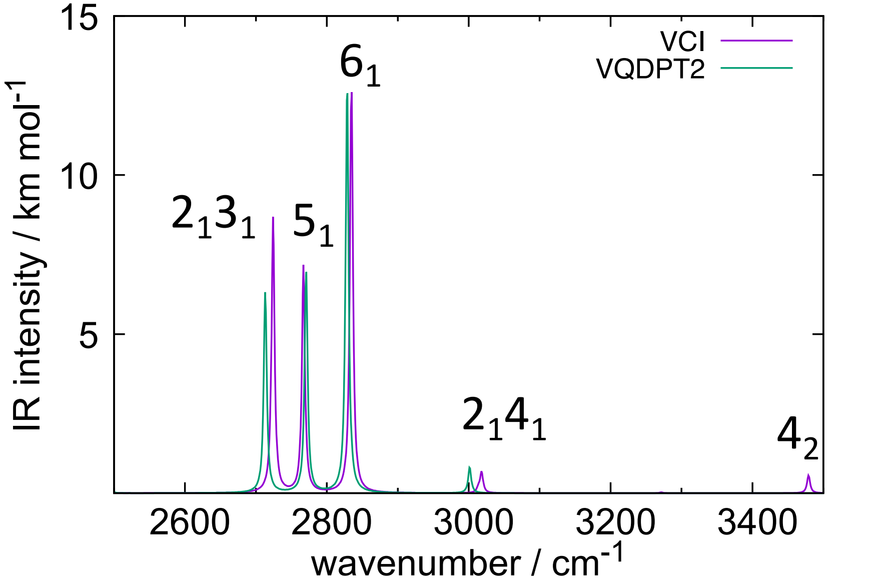
```
> gnuplot plotIR.gpi
```

Then, we obtain the spectrum in "plotIR.pdf". By looking at the peak positions in vci/vqdpt-IR.data, we can assign the fundamental bands.

Surprisingly or not, we see several more peaks in a high frequency range of the spectrum, most notably a strong peak around 2700 cm$^{-1}$.

"plotIR2.pdf" is a zoom up of a region, 2500 – 3500 cm$^{-1}$. The strong peak at 2724 cm$^{-1}$ comes from a combination of $2_1 3_1$, which is in Fermi resonance with $6_1$. $2_1 4_1$ and $4_2$ are also seen in the spectrum with much weaker intensity though.





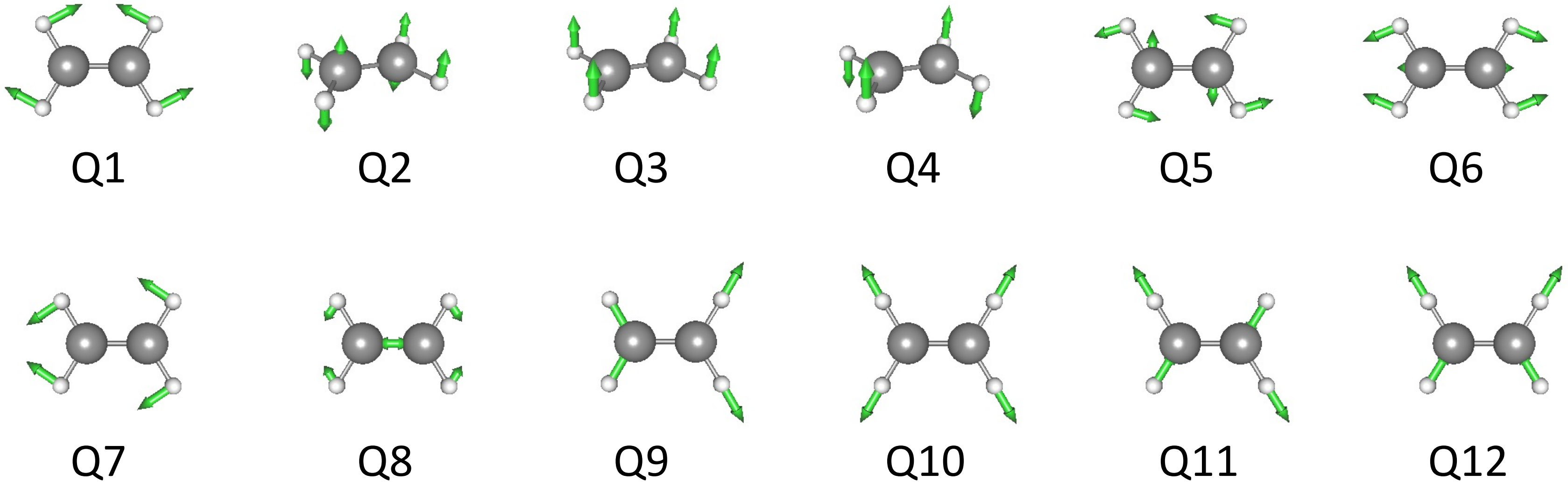
VCI and VQDPT2 agree well for the fundamentals. However, the agreement becomes worse for the combination and overtones; in particular, $4_2$ is not present in VQDPT2. This is because VQDPT2 is a state-specific scheme. The agreement will improve if we add those state to target states in &state.

# 3. Optimized coordinate VSCF

In this section, we carry out optimized coordinate VSCF (oc-VSCF) [6] for ethylene ($C_2H_4$). Proceed to 3.ethylene,

```
> cd 3.ethylene
> ls
eq-mp2dz.minfo ncvci.inp   ocvci.inp     ocvscf.inp  runSindo.sh*
log/         ncvqdpt2.inp   ocvqdpt2.inp  prop_no_1.mop
```

The normal coordinates of $C_2H_4$ looks like this:



In oc-VSCF, not only the one-mode function but also the coordinates are variationally optimized. For this purpose, the PES must be invariant to the coordinate transformation. The Taylor expansion PES fulfills this requirement. In SINDO, we can use a cubic force field (CFF) or a full quartic force field (4MR-QFF) for oc-VSCF calculations.

Here, we use a 4MR-QFF calculted at the MP2/cc-pVDZ level. “prop_no_1.mop” contains the QFF coefficients up to 4MR:

```
SCALING FREQUENCIES N_FRQS=12
3.7604565842336010000000e-03
4.2633781173158170000000e-03
4.4317087065806600000000e-03
…
-4.5104054238305110000000e-13   6   10   11   12
 6.5048942246367230000000e-05   7   10   11   12
 4.6335694968527480000000e-12   8   10   11   12
-1.1828490307864063000000e-03   9   10   11   12
```

4MR terms

Note that 4MR-QFF can be generated using MakePES by setting <MR> to 4 in the input. However, 4MR-QFF is far more costly than 3MR-QFF. If the calculation is prohibitive, we may use CFF which is a part of 3MR-QFF.

```
<makePES>
  <MR        value=”4" />
  …
</makePES>
```

ocvscf.inp is an input file to carry out oc-VSCF calculations:

```
...
#--- [  VIB  ]
&vib  MR=4 ocvscf=.t. vscf=.f. vci=.f. vpt=.f.  vmaxAll = 10 /
                 true invokes oc-VSCF.

#--- [  OCVSCF  ]
 &ocvscf
    mopfile='prop_no_1.mop'     the name of a mop file
    icff=0                      icff = 0 : QFF, 1 : CFF
  /
...
```

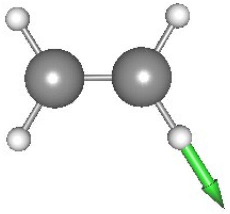
Run the program by,

```
> sindo < ocvscf.inp > ocvscf.out
```

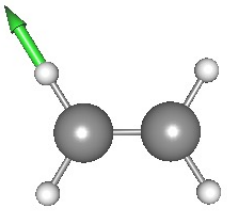
Upon successful convergence, we find in the output:

```
 > OPTIMIZATION CONVERGED!
 > END OF OPTIMIZA

   o TRANSFORMATION MATRIX WRITTEN TO   : [ u1.dat ]
   o FORCE CONSTANTS WRITTEN TO         : [ prop_no_1.mop_ocvscf ]
   o NEW COORDINATES WRITTEN TO         : [ eq-mp2dz_ocvscf.minfo ]
```

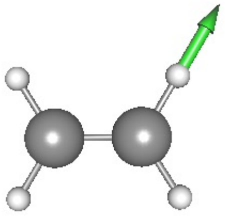
The optimized coordinates are written in “eq-mp2dz_ocvscf.minfo”. Modes 1 – 8 are almost the same. However, the CH stretching modes are drastically changed to local CH stretching modes.



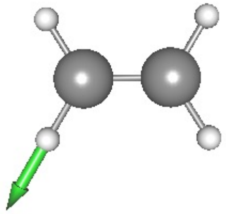
Q9



Q10



Q11



Q12

The QFF coefficients in terms of optimized coordinates are written in “prop_no_1.mop_ocvscf”.

“u1.dat” contains a transformation matrix from normal to optimized coordinates, though we don’t use this file here.

We now perform vibrational calculations using QFF in terms of both normal and optimized coordinates. ncvci.inp and ncvqdpt2.inp are input files for VCI and VQDPT2 calculations based on normal coordinates, and ocvci.inp and ocvqdpt2.inp are those based on optimized coordinates. The difference is the mopfile of &mrpes:

```
 &mrpes mopFile='prop_no_1.mop' mcs_cutoff=-1.0D-03 /
```

```
 &mrpes mopFile='prop_no_1.mop_ocvscf' mcs_cutoff=-1.0D-03 /
```

The level of VCI excitations is VCI[6]-(6). The parameters for VQDPT2 are maxSum=4 and nGen=1.

We run the program by,

```
> sindo < ncvqdpt2.inp > ncvqdpt2.out
> sindo < ocvqdpt2.inp > ocvqdpt2.out
> sindo < ncvci.inp    > ncvci.out
> sindo < ocvci.inp    > ocvci.out
```

Note that the VCI calculation may take more than 10 minitues, since the VCI space is large (VCI dimension = 18,562).

Here, we summarize the results of CH stretching modes:

| | nc-VQDPT2 | oc-VQDPT2 | nc-VCI[6]-(6) | oc-VCI[6]-(6) |
|---|---|---|---|---|
| 9 | 2996.6 | 3005.2 | 3008.5 | 3009.1 |
| 10 | 3063.9 | 3063.7 | 3069.6 | 3068.6 |
| 11 | 3121.5 | 3137.0 | 3140.2 | 3139.1 |
| 12 | 3139.2 | 3159.6 | 3161.6 | 3160.4 |

The results of nc-VCI and oc-VCI match within 2 cm$^{-1}$, indicating that the result is close to the exact solution, which is independent to the choice of coordinates. In constrast, nc- and oc-VQDPT2 results are not the same, because VQDPT2 is an approxiate, second-order perturbative approach. Nevertheless, oc-VQDPT2 is much closer to VCI than nc-VQDPT2. This result suggests that optimized coordinates give a faster convergence to the exact solution in post-VSCF calculations. See Ref. [4] and [6] for more discussion.

# 4. References

- Direct VSCF/VCI

  [1] Direct vibrational self-consistent field method: Applications to $H_2O$ and $H_2CO$,
  K. Yagi, T. Taketsugu, K. Hirao, and M. S. Gordon, J. Chem. Phys. **113**, 1005 (2000).
- VMP2-(n)

  [2] Efficient configuration selection scheme for vibrational second-order perturbation theory,
  K. Yagi, S. Hirata, and K. Hirao, J. Chem. Phys. **127**, 034111 (2007).
- VQDPT2

  [3] Vibrational quasi-degenerate perturbation theory: Applications to Fermi resonance in $CO_2$, $H_2CO$, and $C_6H_6$,
  K. Yagi, S. Hirata, and K. Hirao, Phys. Chem. Chem. Phys. **10**, 1781 (2008).

  [4] Vibational quasi-degenerate perturbation theory with optimized coordinates: Applications to ethylene and trans-1,3-butadine,
  K. Yagi and H. Otaki, J. Chem. Phys. **140**, 84113 (2014).
- MRPES

  [5] Multiresolution potential energy surfaces for vibrational state calculations,
  K. Yagi, S. Hirata, and K. Hirao, Thoer. Chem. Acc. **118**, 681 (2007).
- oc-VSCF

  [6] Optimized coordinates for anharmonic vibrational structure theories,
  K.Yagi, M. Keçeli, and S. Hirata, J. Chem. Phys. **137**, 204118 (2012).

# List of all Options

## &mol

Character(80) :: minfoFile

The name of the .minfo file, in which the information of molecule is written.

Integer :: Nat

The number of atoms

Real(8), dimension(Nat) :: Mass

The mass of each atoms (in atomic mass unit)

Real(8), dimension(3,Nat) :: x

The reference (equilibrium) geometry (in Angstrom)

Real(8), dimension(Nfree) :: omega

The frequencies for the HO basis sets (in cm−1)

Real(8), dimension(Nat*3,Nfree) :: L

The vibrational displacement vectors

**[Note]** ’minfoFile’ is mutually exclusive from others.

## &sys

Integer(8) :: Maxmem

Maximum size of memory (MB)

## &mrpes

Integer :: MR

Mode representation (MR=1-4)

Real(8) :: mcs_cutoff

Cutoff of QFF based on MCS in cm-1 (default = 1.d-04)

Logical :: au

The grid data in atomic unit (default = true)

Character(80) :: mopFile

The name of the mop file.

## &vib

Integer :: Nfree

Number of degrees of freedom (default = 3Nat - 6)

Integer :: MR

Mode representation (MR=1-4)

Integer, dimension(Nfree) :: vmax

Number of basis functions for each mode (default=10)

Integer :: vmaxALL

Number of basis functions for all modes (default=10)

Integer :: vmax base

same as vmaxALL

Logical :: vscf, ocvscf, vci, vpt, vqdpt

invoke vscf/ocvscf/vci/vpt/vqdpt

Logical :: prpt

invoke property calculation

Logical :: readBasis

read the basis functions from cho.basis

## &states

Integer :: Nstate

Number of states to calculate

Integer, dimension(Nfree,Nstate) :: target state

Labels of the target states

Logical :: fund

Compute fundamentals

## &vscf

Logical :: state specific

State specific VSCF if true (default = .false.)

Logical :: restart

Restart from vscf xxx.wfn (default = .false.)

Integer :: Maxitr

Maximum number of iteration (default = 10)

Real(8) :: Ethresh

Threshold of convergence (default = 1e-03 cm−1)

## &ocvscf

- Integer :: maxOptIter
  - Maximum number of iteration (default = 30)
- Real(8) :: ethresh
  - Threshold of the energy (default = 1e-06 cm−1)
- Real(8) :: gthresh
  - Threshold of the gradient (default = 1e-06 cm−1 rad−1)
- Integer :: pfit
  - Order of the Fourier fitting (default = 2)
- Character(80) :: mopFile
  - The name of the mopfile
- Character(80) :: u1File
  - The name of the file to write the transformation matrix (default = u1.dat)
- Integer :: icff
  - Switch on CFF when icff = 1 and QFF when icff = 0 (default = 0)
- Integer :: iscreen
  - Switch off/on pair selection when iscreen=0/1 (default = 1)
- Real(8) :: eta12thresh
  - Threshold value for the pair screening (default = 500 cm−1)

## &vci

Integer :: Nstate

  Number of states to calculate

Integer :: nCI

  Max CI dimension (cutoff based on the energy)

Integer(Nfree) :: maxEx

  Max quantum number to excite for each mode

Integer :: maxExALL

  Max quantum number to excite for all the modes

Integer :: maxSum

  Max sum of quantum number

Integer :: nCUP

  Max number of modes to excite

Logical :: geomAv

  If true, calculate vibrationally averaged geometry

Logical :: dump

  If true, dump the vci wavefunction to vci-w.wfn

Real(8) :: printWeight

  Print the configuration with the weight larger than this threshold

Logical :: readCIbasis

  If true, read CI basis from vci-w.wfn

Logical :: dumpHmat

  If true, write the VCI hamiltonian matrix

Logical :: noDiag

  If true, the diagonalization is skipped

## &vpt

Integer :: maxSum

Max sum of quantum number to excite (default = -1)

Integer :: maxEx

Max quantum number to excite (default = -1)

Integer :: nCUP

Max number of modes to excite (default = MR)

Real(8) :: thresh ene

Threshold energy to avoid divergence (default=1e-04 Hartree)

Logical :: dump

Dump the information to vmp-w.wfn

## &vqdpt

Integer :: nGen

The generation of P space (default=3)

Real(8) :: thresh p0

E0 pruning (default=500 cm−1)

Real(8) :: thresh p1

VPT based pruning (default=0.1)

Real(8) :: thresh p2

VCI pruning (default=0.05)

Real(8) :: thresh p3

VCI pruning (default=0.9)

Integer :: pset

Combine the p-space generated from several target states
=0 when the target states have an overlap (default)
=1 when the p-space components have an overlap

Integer :: maxSum

Max sum of quantum number to excite (default = -1)

Integer :: nCUP

Max number of modes to excite (default = MR)

Integer :: pqSum

P/Q interaction scheme

> 0 prune the interaction when $\lambda pq$ > maxSum (default)

< 0 full interaction

Integer :: vqdpt2 loop

=0 loop over q, then p, p’ (default)

=1 loop over p, then p’, q

Real(8) :: thresh ene

Threshold energy to avoid divergence (default=1e-04 Hartree)

Real(8) :: printWeight

Print the configuration with the weight larger than this threshold (default=0.001)

Logical :: dump

Dump the information to vqdpt-w.wfn (default=true)

## &prpt

Logical :: vscfprpt, vciprpt, vptprpt, vqdptprpt

Invoke property calculation for vscf, vci, vpt, vqpdt wavefuncion

Integer :: MR

Mode representation (default = 3)

Character :: extn(*)

The extension of the property files

Integer :: matrix(*)

= 0 calculate only the average

> 0 calculate the matrix

Logical :: infrared

If true, calculate the IR intensity.

## &prptvci

Integer :: Nstate

The number of states

## &IRspectrum

Real(8) :: minOmega, maxOmega

Min/Max value of the spectrum (default = 100 - 4000 cm−1)

Real(8) :: delOmega

Interval of the data (default = 1 cm−1)

Real(8) :: fwhm

Full-width half maximum of the Lorentz function for convolutions (default = 20 cm−1)

Real(8) :: cutoff

Cutoff of the band (default = -1 km mol−1)