Anharmonic vibrational calculations based on a QM/MM potential

Kiyoshi Yagi
kiyoshi.yagi@riken.jp

Theoretical Molecular Science Laboratory
RIKEN Cluster for Pioneering Research

2019/11/10

- In this guide, I will illustrate how to combine MakePES and GENESIS to generate anharmonic potential energy surface (PES) for complex systems using QM/MM, and how to calculate the vibrational spectrum using sindo.

- MakePES and sindo are command line based programs. This guide assumes that you are familiar with basic commands in UNIX. Shell scripts are given for Bourne Shell (bash).

- This guide also assumes that you have installed the program. Change “/path/to” to your installation directory when you see a command like this,

```
sindo_jar=/path/to/sindo-4.0/jar
java –cp ‘$sindo_jar/*' RunMakePES
```

- This sample is based on a QM/MM tutorial of GENESIS,

  https://www.r-ccs.riken.jp/labs/cbrt/tutorials2019/tutorial-16-1/

  where QM/MM calculations are carried out for an alanine tripeptide (Ala$_3$) in water solvent. It is recommended to go through this tutorial (and other GENESIS tutorials) before starting this guide.

Contents of Sample Files

The following folders are included in this sample:

toppar	Force field parameters
2_setup/snapshot50.crd, pdb, psf	Setup files
4_qmm-min/qmmm_min.rst	Optimized geometry
5_qmmm-vib/qmmm_vib.minfo	Normal modes
6_mkqff	QFF generation
7_mkgrid1	1MR-Grid generation
8_mrpes	2MR-MRPES generation
9_anharm	Vibrational calculations

Files from the GENESIS tutorial

This sample is based on a QM/MM tutorial of GENESIS,

https://www.r-ccs.riken.jp/labs/cbrt/tutorials2019/tutorial-16-1/

where QM/MM calculations are carried out for an alanine tripeptide (Ala_3) in water solvent. The first residue is treated by QM (B3LYP-D3/cc-pVDZ) and the others by MM (CHARMM36m, TIP3P). The following files are taken from the tutorial.

2_setup/snapshot50.psf, pdb, crd

- The files prepared by qmmm_generator
- Ala_3 + 1999 water molecules (6039 atoms)

4_qmmm-min/qmmm_min.rst

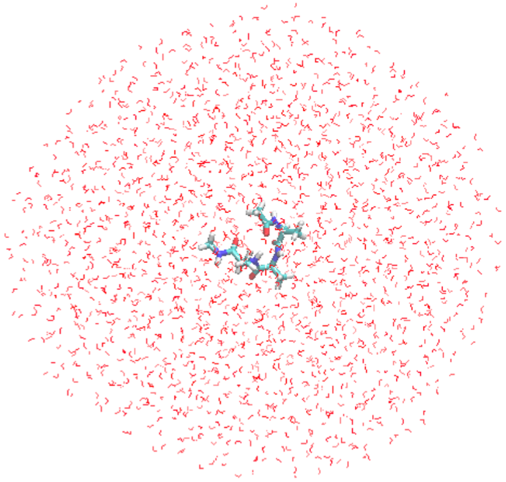
- GENESIS restart file containing the QM/MM optimized structure

5_qmmm-vib/qmmm_vib.minfo

- A minfo file containing the normal modes and harmonic frequencies

toppar

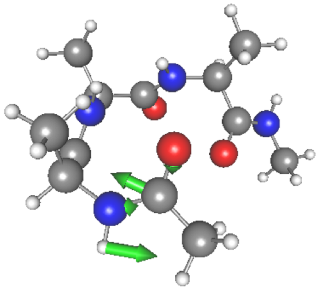
- CHARMM 36m force field for Ala and TIP3P for water



2_setup/ snapshot50.pdb

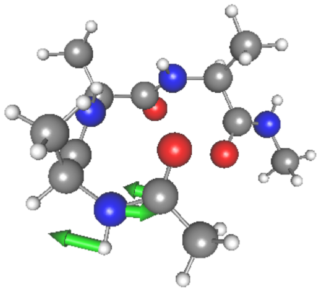
5_qmmm-vib

Before anharmonic calculations, let’s review what we‘ve got from the harmonic normal mode analysis. In the GENESIS tutorial, we obtained partial normal modes of an amide group at the c-terminal. Open 5_qmmm-vib/qmmm_vib.minfo using JSindo. The amide modes looks as follow:



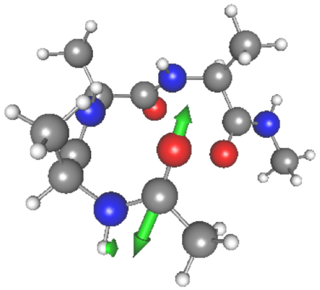
Amide III
(CN str + CNH bend)

1315.0 cm^{-1}



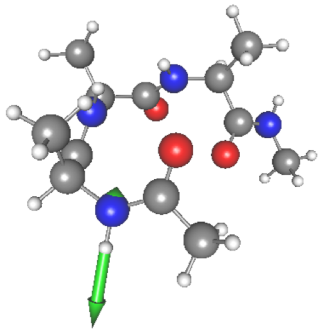
Amide II
(CN str + CNH bend)

1576.5 cm^{-1}



Amide I
(CO str)

1704.2 cm^{-1}



Amide A
(NH str)

3487.4 cm^{-1}

We will use these modes (mode 9 -12) in the following anharmonic calculations.

6_mkqff

Proceed to 6_mkqff to find input files to generate quartic force field (QFF) for Ala$_3$,

```
> cd 6_mkqff
> ls
gaussian.com     makePES.xml      runGau.sh*       script.sh*
log/             qmmm_mkqff.inp
```

We first run MakePES in generic mode. makePES.xml looks as follow:

makePES.xml

```
<makePES>
  <minfoFile   value="../5_qmmm-vib/qmmm_vib.minfo" />
  <MR        value="3" />
  <activemode   value="9-12" />
  <qchem>
      <program   value=”generic" />
      <title     value="B3LYP-D3/cc-pVDZ" />
      <xyzfile   value="makeQFF" />
  </qchem>
   <qff>
    <stepsize  value="0.5" />
    <ndifftype value="grad"/>
    <mopfile   value="prop_no_1.mop" />
  </qff>
</makePES>
```

specify the minfo file

3MR-QFF

set the Amide modes to active

the name of xyz file

set to gradient

the name of mop file

Running the program creates makeQFF.xyz,

```
sindo_jar=/path/to/sindo-4.0/jar
java –cp ‘$sindo_jar/*' RunMakePES –f makePES.xml >& makePES.out1
```

makeQFF.xyz is written in the usual xyz format,

makeQFF.xyz

```
 4              The number of atoms
mkqff-eq        name of the first point
 CY     -4.0454035711     6.0518032658    -1.0542278638
 OY     -3.6124748284     6.0503734927     0.1129067982
                          xyz coordinates
  N     -3.2838770225     5.8998076018    -2.1586664849
 HN     -3.7438296522     6.0476346883    -3.0591784947
 4
mkqff8-0        name of the second point
 CY     -4.0042961166     6.0463168527    -1.0649806269
                          xyz coordinates
 OY     -3.6218945397     6.0507865924     0.0984053428
```

Note that the coordinates are printed only for the atoms active in the vibrational analysis, that is, the amide group.

```
> grep -c mkqff makeQFF.xyz
41
```

Next, we run GENESIS to caluclate the energy and gradient at the grid points. In the input file, [VIBRATION] section looks as follow:

qmmm_mkqff.inp

```
[VIBRATION]
runmode            = QFF
nreplica           = 4
vibatm_select_index = 2
gridfile           = makeQFF.xyz

[SELECTION]
group1  = atomno:1-14
group2  = atomno:5-8
```

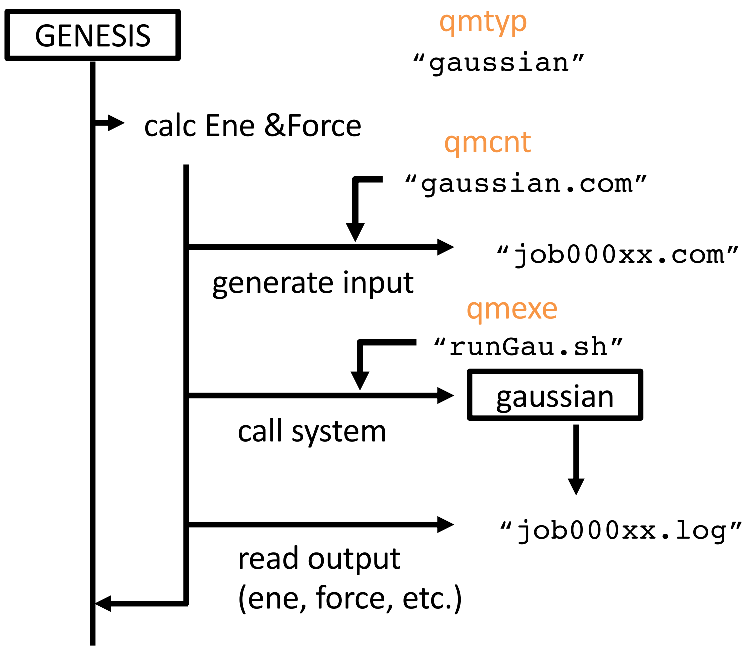
- runmode Set to QFF.
- nreplica # of MPI processes to parallelize over the grid points.
- vibatm_select_index Select VIB atoms through [SELECTION] section. Note that the VIB atoms must match with the atoms in the gridfile.
- gridfile The name of xyz file.

The input for GENESIS also has [QM/MM] section with the following options:

qmmm_mkqff.inp

```
[QMMM]
qmtyp            = gaussian
qmcnt            = gaussian.com
qmexe            = runGau.sh
qmatm_select_index  = 1
workdir          = qmmm_qff
basename         = job
qmsave_period      = 10
qmmaxtrial       = 1
exclude_charge     = group

[SELECTION]
group1  = atomno:1-14
group2  = atomno:5-8
```



GENESIS

calc Ene &Force

generate input

call system

read output
(ene, force, etc.)

qmtyp
“gaussian”

qmcnt
“gaussian.com”

“job000xx.com”

qmexe
“runGau.sh”

gaussian

“job000xx.log”

In this sample, we use Gaussian for the QM program. qmcnt and qmexe are a template input a and an execution script file, respectively. qmatm_select_index specifies the QM region to be atomno:1-14, that is, the first residue of Ala$_3$.

The template file looks as follow. #coordinate#, #charge#, and #elec_field# are replaced by the program. Note that we need to set a “force” option to obtain the gradient.

gaussian.com

```
%chk=gaussian.chk
...
Charge Force Prop=(Field,Read) pop=mk

Gaussian run for QMMM in genesis

0 1
#coordinate#
#charge#
#elec_field#
```

The name of chk file is fixed. Don’t change!

These keys are replaced by GENESIS.

The first part of runGau.sh is as follow. Make sure that the path to Gaussian and a local scratch directory are set correctly.

runGau.sh

```
# --- Set the path for Gaussian ---
export g09root=/path/to/gaussian
...
# --- Set the path for a scratch folder ---
scratch=/scr/$USER
```

Now, we run GENESIS. The following command invokes 4 MPI processes using 8 thread. The number of MPI processes must match with “nreplica” in [VIBRATION] section of the input.

```
> export  QM_NUM_THREADS=8
> export OMP_NUM_THREADS=8
> mpirun -np 4 --map-by node:pe=8 atdyn qmmm_mkqff.inp >&
qmmm_mkqff.out
```

You will see the following message in the output if the jobs are running,

```
      Compute energy at grid points: minfo files created in [ minfo.files ]
        Done for          mkqff8-0 :   replicaID =     3
        Done for          mkqff8-2 :   replicaID =     1
        Done for          mkqff-eq :   replicaID =     2
        Done for          mkqff8-1 :   replicaID =     4
```

and the results are stored in a “minfo.files” directory,

```
> ls minfo.files
mkqff-eq.minfo      mkqff10_9-0.minfo   mkqff11_10-1.minfo
mkqff10-0.minfo     mkqff10_9-1.minfo   mkqff11_10-2.minfo
...
```

When the GENESIS job is done, we run MakePES again,

```
sindo_jar=/path/to/sindo-4.0/jar
java –cp ‘$sindo_jar/*' RunMakePES –f makePES.xml >& makePES.out2
```

Then, it will create a mop file, where the information of QFF coefficients are included.

prop_no_1.mop

```
SCALING FREQUENCIES N_FRQS=12
8.2309152545488990000000e-04
1.3157446189111168000000e-03
…
DALTON_FOR_MIDAS B3LYP-D3/cc-pVDZ
-3.7152377587419910000000e-05    9
 2.9957574984083245000000e-03    9    9
-4.3574279250144276000000e-05    9    9    9
 8.7827684324674090000000e-06    9    9    9    9
 7.7816868733945820000000e-06   10
 3.5915650302996167000000e-03   10   10
…
```

7_mkgrid1

Proceed to 7_mkgrid1 to find input files to generate 1MR-Grid PES for Ala_3,

```
> cd 7_mkgrid1
```

We again run MakePES in generic mode, but now for Grid PES. makePES.xml looks as follow:

makePES.xml

```
<makePES>
  <minfoFile   value="../5_qmmm-vib/qmmm_vib.minfo" />
  <MR          value=”1" />               1MR-Grid
  <activemode  value="9-12" />
  <dipole      value="true" />            calculate dipole mement surfaces
  <qchem>
      <program  value=”generic" />
      <title    value="B3LYP-D3/cc-pVDZ" />
      <xyzfile  value="makeGrid" />   the name of a xyz file
  </qchem>
  <grid>
    <ngrid  value="11" />          the number of grid points in each coordinate
    <fullmc value="true"/>         generate all 1MR terms
  </grid>
</makePES>
```

Running the program creates makeGrid.xyz,

```
sindo_jar=/path/to/sindo-4.0/jar
java –cp ‘$sindo_jar/*' RunMakePES –f makePES.xml >& makePES.out1
```

makeGrid.xyz is written in the usual xyz format,

makeGrid.xyz

```
 4                The number of atoms
mkg-eq            The name of the first point
 CY     -4.0454035711      6.0518032658     -1.0542278638
 OY     -3.6124748284      6.0503734927      0.1129067982
                           xyz coordinates
  N     -3.2838770225      5.8998076018     -2.1586664849
 HN     -3.7438296522      6.0476346883     -3.0591784947
 4
mkg-q9-11-0      The name of the second point
 CY     -3.9448692389      6.0383854347     -1.0805253292
                           xyz coordinates
 OY     -3.6355121192      6.0513837888      0.0774413538
```

Again, the coordinates are printed only for the atoms active in the vibrational analysis, that is, the amide group. You can count the number of grid points by,

```
> grep -c mkg makeGrid.xyz
41
```

Next, we run GENESIS to caluclate the energy at the grid points. In the input file, [VIBRATION] section looks as follow:

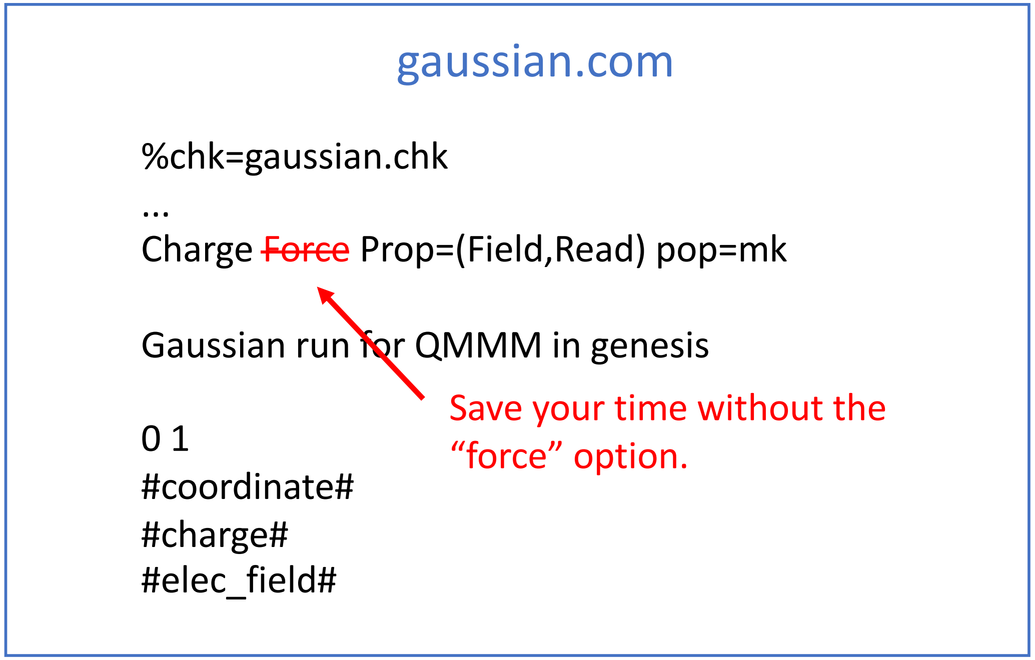
qmmm_mkgrid.inp

```
[VIBRATION]
runmode           = GRID
nreplica          = 4
vibatm_select_index = 2
gridfile          = makeGrid.xyz
datafile          = makeGrid.dat

[SELECTION]
group1  = atomno:1-14
group2  = atomno:5-8
```

- runmode   Set to GRID.
- nreplica   # of MPI processes to parallelize over the grid points.
- vibatm_select_index   Select VIB atoms through [SELECTION] section.
  Note that the VIB atoms must match with the atoms in the gridfile.
- gridfile   The name of xyz file.
- datafile   The name of dat file.

The [QMMM] section is the same as before. The template file looks almost the same, but now we don’t need a “force” option.



gaussian.com

%chk=gaussian.chk
...
Charge ~~Force~~ Prop=(Field,Read) pop=mk

Gaussian run for QMMM in genesis

0 1
#coordinate#
#charge#
#elec_field#

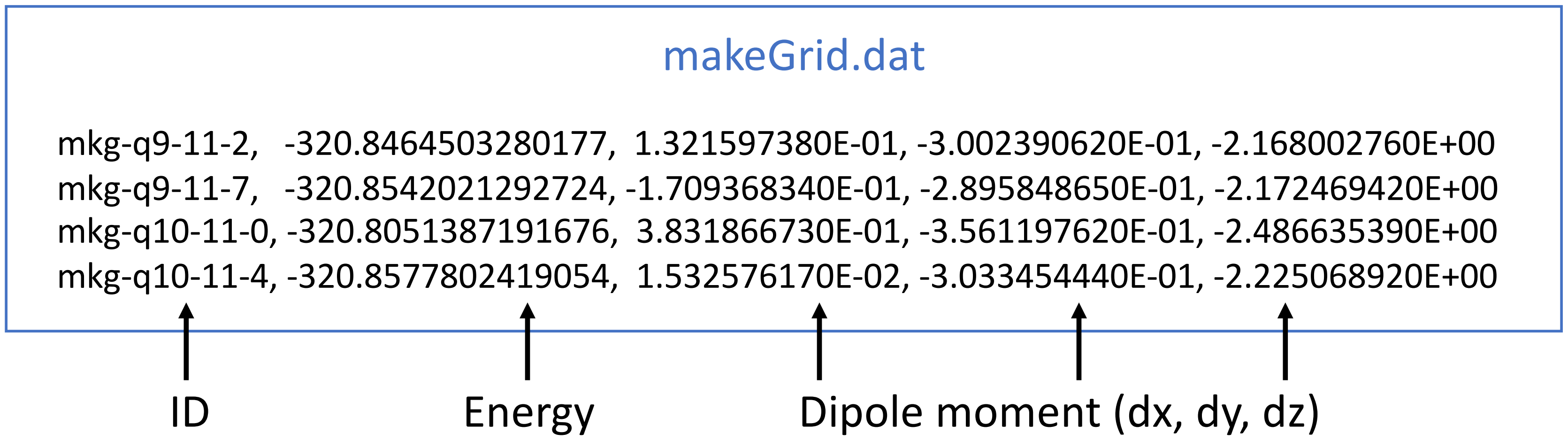
Save your time without the “force” option.

NOTE: In fact, we don’t use the field (the gradient for MM atoms), so “Prop=(Field, Read)” is not needed either. However, this option must be present in the input due to a restriction of the current program. This will be fixed in the near future.

Run GENESIS as before,

```
> export  QM_NUM_THREADS=8
> export OMP_NUM_THREADS=8
> mpirun -np 4 --map-by node:pe=8 atdyn qmmm_mkgrid.inp >&
qmmm_mkgrid.out
```

The results are written to makeGrid.dat,



makeGrid.dat

```
mkg-q9-11-2,   -320.8464503280177,  1.321597380E-01, -3.002390620E-01, -2.168002760E+00
mkg-q9-11-7,   -320.8542021292724, -1.709368340E-01, -2.895848650E-01, -2.172469420E+00
mkg-q10-11-0, -320.8051387191676,  3.831866730E-01, -3.561197620E-01, -2.486635390E+00
mkg-q10-11-4, -320.8577802419054,  1.532576170E-02, -3.033454440E-01, -2.225068920E+00
```

When the GENESIS job is done, we run MakePES again,

```
sindo_jar=/path/to/sindo-4.0/jar
java –cp ‘$sindo_jar/*' RunMakePES –f makePES.xml >& makePES.out2
```

which will create pot and dipole files,

```
> ls *.pot
eq.pot    q10.pot   q11.pot   q12.pot   q9.pot
> ls *.dipole
eq.dipole   q10.dipole  q11.dipole  q12.dipole  q9.dipole
```

8_mrpes

Proceed to 8_mrpes. Here, we will generate 2MR-Grid PES for Ala_3 by a multiresolution method. We first copy the QFF and 1MR-Grid files to current directory.

```
> cd 8_mrpes
> cp ../6_mkqff/prop_no_1.mop .
> cp ../7_mkgrid1/*pot ./
> cp ../7_mkgrid1/*dipole ./
```

We run MakePES in generic mode. makePES.xml looks as follow:

makePES.xml

```
<makePES>
  <minfoFile   value="../5_qmmm-vib/qmmm_vib.minfo" />
  <MR         value=”2" />          2MR-Grid
  ...
  <grid>
    <ngrid  value="9" />    the number of grid points in each coordinate
    <mcstrength value="10"/>
    <mopfile    value="prop_no_1.mop"/>
  </grid>
</makePES>
```

generate all 2MR terms if mode coupling strength (MCS) is larger than 10.0. MCS is evaluated from QFF coefficients in a mopfile.

Running the program creates makeGrid.xyz,

```
sindo_jar=/path/to/sindo-4.0/jar
java –cp ‘$sindo_jar/*' RunMakePES –f makePES.xml >& makePES.out1
```

In the output, you will find a message,

makePES.out1

```
Setup MakeGrid module

  o Setup MCS: Read QFF Data via ../6_mkqff/prop_no_1.mop ...  [OK]
  o ngrid =    9
  o 1MR Grid:
     12 9 10
  o 2MR Grid:
     (9,12) (10,12)
```

which detects that Q12Q9 and Q12Q10 are coupled with MCS > 10.0. The grid points are written to makeGrid.xyz. You can count the number of grid points by,

```
> grep -c mkg makeGrid.xyz
128
```

The input file for GENESIS is the same as before. Run GENESIS by,

```
> export  QM_NUM_THREADS=8
> export OMP_NUM_THREADS=8
> mpirun -np 4 --map-by node:pe=8 atdyn qmmm_mrpes.inp >&
qmmm_mrpes.out
```

When the GENESIS job is done, we run MakePES again,

```
sindo_jar=/path/to/sindo-4.0/jar
java —cp ‘$sindo_jar/*' RunMakePES —f makePES.xml >& makePES.out2
```

which will create pot and dipole files,

```
> ls *.pot
eq.pot   q10.pot  q11.pot  q12.pot  q12q10.pot  q12q9.pot  q9.pot
> ls *.dipole
eq.dipole   q10.dipole  q11.dipole  q12.dipole  q12q10.dipole
q12q9.dipole   q9.dipole
```

9_anharm

Finally, we perform vibrational calculations. Proceed to 9_anharm to find the input files for sindo.

```
> cd 9_anharm
> ls
log/   runSindo.sh   vci.inp   vmp2.inp   vqdpt2.inp vci-IR.gpi
```

These input files run VCI[4]-(8), VMP2-(4), VQDPT2-(4). The calculation is invoked by a script, runSindo.sh:

runSindo.sh

```
> cat runSindo.sh
SINDO=/path/to/FSindo/bin/sindo        set the path to SINDO
export POTDIR=../8_mrpes

${SINDO} < vmp2.inp   > vmp2.out   2>&1
${SINDO} < vqdpt2.inp > vqdpt2.out 2>&1
${SINDO} < vci.inp    > vci.out    2>&1
```

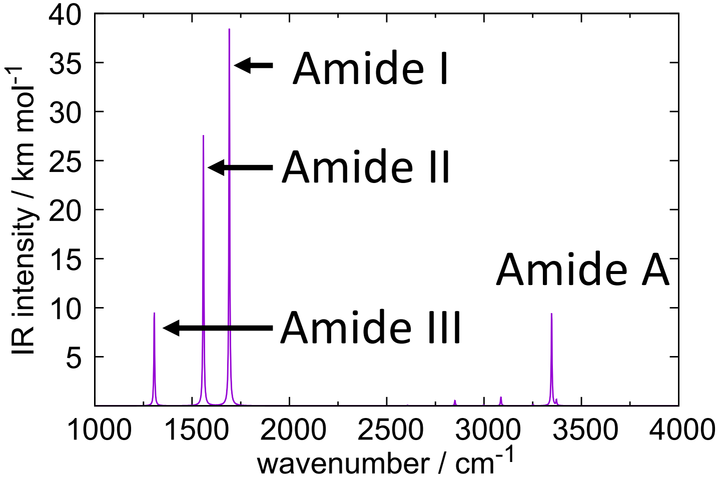
Note that we read the information of the PES from 8_mrpes (specified by POTDIR).

From the output file, we obtain a table like this:

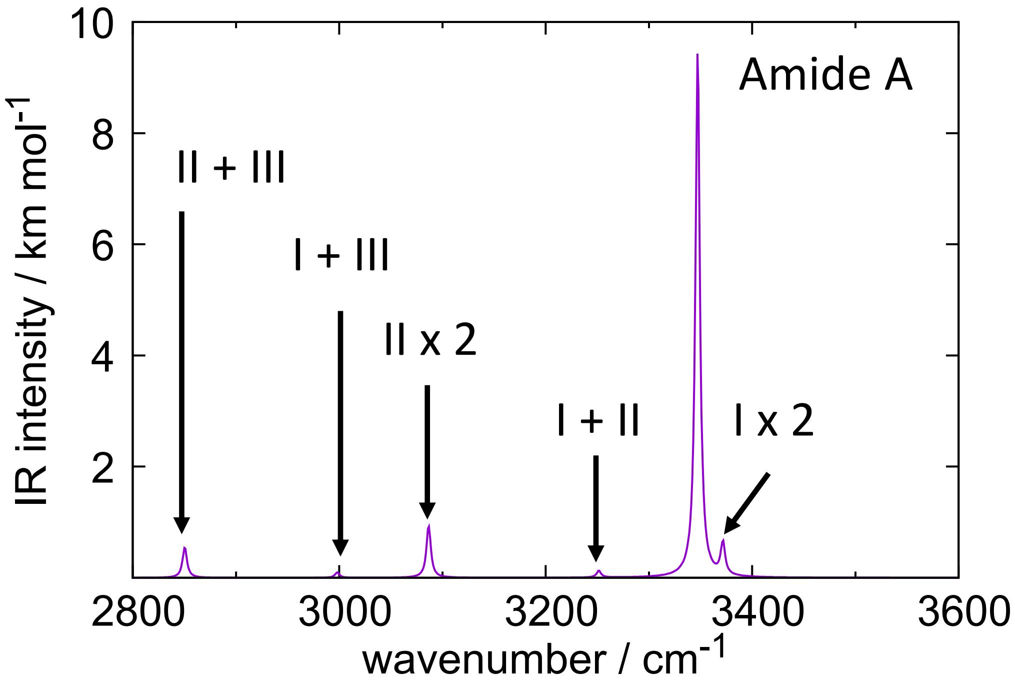
	HARM	VSCF	VMP2	VQDPT2	VCI
ZPE	4041.58	4016.35	4012.44	4012.44	4012.01
Amide III (9_1)	1314.99	1316.74	1307.31	1307.31	1305.71
Amide II (10_1)	1576.51	1572.46	1559.88	1559.88	1560.42
Amide I (11_1)	1704.22	1698.11	1691.15	1691.15	1691.25
Amide A (12_1)	3487.43	3305.84	3358.45	3347.85	3347.26

We may also plot the IR spectrum using gnuplot,

```
> gnuplot vci-IR.gpi
```



If we magnify the high frequency region, we see overtones and combination tones of the amide I, II, and III.


IR intensity / km mol-1
wavenumber / cm-1
Amide A
II + III
I + III
II x 2
I + II
I x 2