MOMAP

Tutorial 03

Fluorescence Spectrum Calculation

Herzber-Teller effect

Version 2019

September, 2019

MOMAP Tutorial 03

Edited by:

Dr. Qikai Li

Dr. Yingli Niu

Ms. Lihui Yan

Last modified on Sep 28, 2024 by:

Dr. Qikai Li

Released by Hongzhiwei Technology (Shanghai) Co., Ltd
and Z.G. Shuai Group
The information in this document applies to version 2019 of MOMAP

MOMAP Tutorial

- Fluorescence Spectrum Calculation

Porphine or porphin is a organic chemical compound with formula $C_{20}H_{14}N_4$. The molecule consists of four pyrrole-like rings joined by four methine (=CH−) groups to form a larger macrocycle ring, which makes it the simplest of the tetrapyrroles. It is an aromatic and heterocyclic compound, solid at room temperature.

MOMAP is able to simulate fluorescence spectrum and calculate the corresponding radiative decay rate constant based on the TVCORF_SPEC and TVSPEC_SPEC subprograms. The TVCORF_SPEC subprogram is used to calculate thermal vibration correlation function (TVCF), while the TVSPEC_SPEC subprogram is used to simulate fluorescence spectrum.

To begin the TVCORF_SPEC and TVSPEC_SPEC calculations, we need the evc results. The evc calculation can use outputs from other QC programs, such as Gaussian, TURBOMOLE, ChemShell, Dalton, MOLPRO, DFTB and MOPAC *etc*. It can also read data from the output files, including vibrational frequencies and force constant matrix, and calculate normal mode displacement, Huang-Rhys factor, reorganization energy and Duschinsky rotation matrix between initial and final electronic states under both internal coordinate and Cartesian coordinate.

The basic steps involved in the calculations are as follows:

1. Gaussian calculations
2. Vibration analysis *etc*.
3. Fluorescence spectrum calculation

Contents

Gaussian Calculations

Calculation on ground state (S_0)

Once the initial geometry is obtained, we have to find the optimized S_0 geometry. The route section is set as **`#p opt freq b3lyp/6-31g*`**, which indicates an optimization calculation at `B3LYP/6-31G*` level.

The initial geometry gaussian S_0 input file (`porphine-s0.com`) is as follows:

```
%chk=porphine-s0.chk
%mem=32GB
%nprocl=4
%nprocs=16
#p opt freq b3lyp/6-31g(d)

opt-porphine-s0

0 1
 N                2.01760000   -2.03040000    0.00000000
 N                2.01120000   -4.36830000    0.00000000
 N                4.36180000   -2.03680000    0.00000000
 N                4.33573828   -4.45908404   -0.00000000
 C                0.65970000   -2.03040000    0.00000000
 C                2.01760000   -0.68530000    0.00000000
 C                0.65970000   -4.36830000    0.00000000
 C                4.36180000   -0.68530000    0.00000000
 C                2.01120000   -5.72610000    0.00000000
 C                5.71970000   -2.03680000    0.00000000
 C                4.34263459   -5.72608147    0.00000000
 C                5.70087915   -4.18727747    0.00000000
 C                0.00000000   -3.19610000    0.00000000
 C                3.18330000   -0.00640000    0.00000000
 C                3.17050000   -6.40510000    0.00000000
 C                6.37940000   -3.19610000    0.00000000
 C                0.00000000   -0.85190000    0.00000000
 C                0.85190000   -0.00640000    0.00000000
 C                0.00000000   -5.52760000    0.00000000
 C                5.52760000    0.00000000    0.00000000
 C                0.85190000   -6.39230000    0.00000000
 C                6.39230000   -0.87110000    0.00000000
 C                5.52120000   -6.39230000    0.00000000
 C                6.37940000   -5.54680000    0.00000000
 H                2.72470678   -2.73750678    0.00000000
 H                3.60232364   -3.73361018    0.00000000
 H               -1.06999924   -3.19482666    0.00000000
 H                3.18076601    1.06359700    0.00000000
 H                3.16668309   -7.47509319    0.00000000
 H                7.44939236   -3.19205755    0.00000000
 H               -1.05961692   -0.70319904    0.00000000
 H                0.71624112    1.05496547    0.00000000
 H               -1.06093024   -5.66662166    0.00000000
 H                5.66506569    1.06113297    0.00000000
 H                0.70721085   -7.45247218    0.00000000
 H                7.45338318   -0.73325047    0.00000000
 H                5.66051034   -7.45319237    0.00000000
 H                7.43956765   -5.69152234    0.00000000
```

We use g09 or g16 to do the geometry optimization.

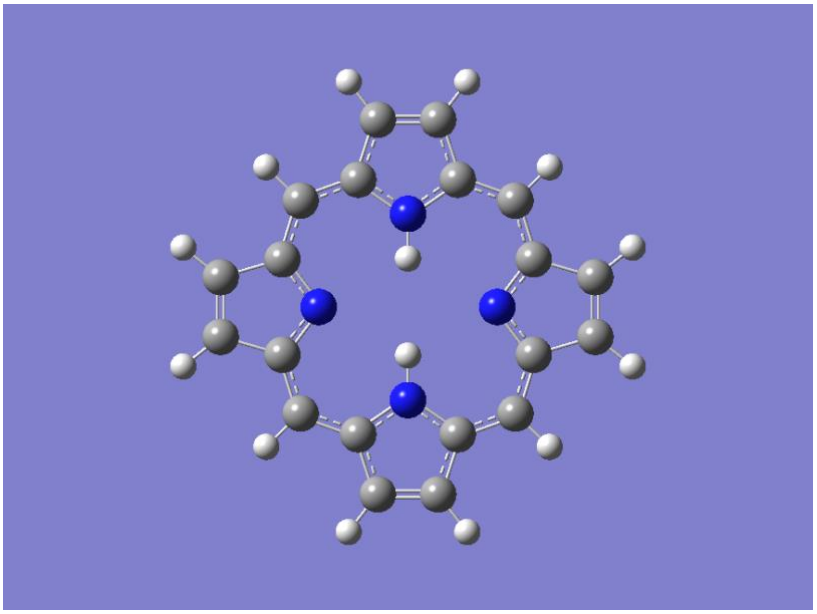


Fig. 1 Optimized S0 geometry

Calculation on lowest singlet excited state (S_1)

Now the optimized S_0 geometry is obtained, we can start optimizing S_1 geometry using the optimized S_0 geometry as the initial structure. The route section is set as **#p td opt freq b3lyp/6-31g***, which indicates an optimization calculation at B3LYP/6-31G* level using the TDDFT method.

The initial gaussian S_1 input file (porphine-s1.com) is as follows:

```
%chk=porphine-s1.chk
%mem=32GB
%nprocl=4
%nprocs=16
#p td opt freq b3lyp/6-31g(d)

opt-porphine-s1

0 1
 N                 0.00016100   -2.11714500    0.00000000
 N                 2.02941000   -0.00021700    0.00000000
 N                -2.02930500    0.00013800    0.00000000
 N                -0.00017500    2.11713800    0.00000000
 C                 1.13015400   -2.89626300    0.00000000
 C                -1.13004200   -2.89587500    0.00000000
 C                 2.85552900   -1.08516900    0.00000000
 C                -2.85547700   -1.08473700    0.00000000
 C                 2.85509100    1.08441700    0.00000000
 C                -2.85507800    1.08479400    0.00000000
 C                 1.13015400    2.89593200    0.00000000
 C                -1.13027800    2.89613600    0.00000000
 C                 2.44125900   -2.42217000    0.00000000
 C                -2.44111000   -2.42173100    0.00000000
 C                 2.44086700    2.42186600    0.00000000
 C                -2.44102300    2.42217400    0.00000000
 C                 0.68600000   -4.26037000    0.00000000
 C                -0.68632300   -4.26013600    0.00000000
 C                 4.25808500   -0.67810100    0.00000000
 C                -4.25798700   -0.67784300    0.00000000
 C                 4.25789200    0.67789400    0.00000000
 C                -4.25790500    0.67815100    0.00000000
 C                 0.68599900    4.26043200    0.00000000
 C                -0.68586900    4.26061300    0.00000000
 H                 0.00057000   -1.10169100    0.00000000
 H                -0.00064500    1.10167800    0.00000000
 H                 3.21982300   -3.17967100    0.00000000
 H                -3.21964400   -3.17926500    0.00000000
 H                 3.21969900    3.17907900    0.00000000
 H                -3.21984800    3.17936700    0.00000000
 H                 1.34702200   -5.11705700    0.00000000
 H                -1.34768400   -5.11656600    0.00000000
 H                 5.10635600   -1.35193800    0.00000000
 H                -5.10625200   -1.35169400    0.00000000
 H                 5.10596000    1.35197600    0.00000000
 H                -5.10606600    1.35212700    0.00000000
 H                 1.34737700    5.11686400    0.00000000
 H                -1.34692600    5.11729600    0.00000000
```

Again, we use g09 or g16 to do the S_1 optimization.

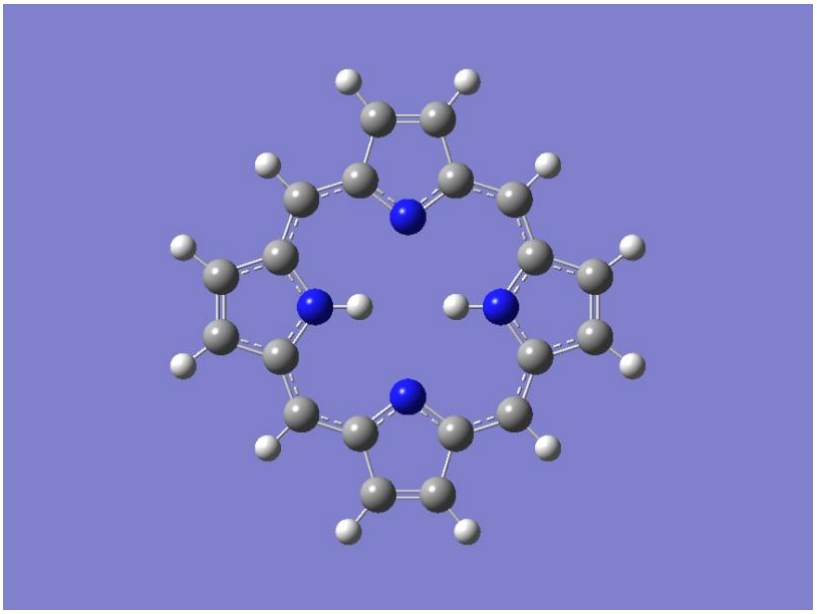


Fig. 2 Optimized S_1 geometry

TIPS: To obtain the optimized ground state geometry, use **Gaussview** to open `porphine-s0.log` file, and save as `porphine-s1.com`. Then, modify the first few lines of `porphine-s1.com` to suit for S_1 optimization.

Calculate EDMA

After finding the optimized S_0 geometry, we can calculate EDMA at this geometry. The route section is set as:

```
#p td b3lyp/6-31g(d)
```

The gaussian EDMA input file (porphine-edma.com) is shown as follows:

```
%chk=porphine-edma.chk
%mem=32GB
%nprocl=1
%nprocs=16
#p td b3lyp/6-31g(d)

edma-porphine

0 1
 N                 0.00016100   -2.11714500    0.00000000
 N                 2.02941000   -0.00021700    0.00000000
 N                -2.02930500    0.00013800    0.00000000
 N                -0.00017500    2.11713800    0.00000000
 C                 1.13015400   -2.89626300    0.00000000
 C                -1.13004200   -2.89587500    0.00000000
 C                 2.85552900   -1.08516900    0.00000000
 C                -2.85547700   -1.08473700    0.00000000
 C                 2.85509100    1.08441700    0.00000000
 C                -2.85507800    1.08479400    0.00000000
 C                 1.13015400    2.89593200    0.00000000
 C                -1.13027800    2.89613600    0.00000000
 C                 2.44125900   -2.42217000    0.00000000
 C                -2.44111000   -2.42173100    0.00000000
 C                 2.44086700    2.42186600    0.00000000
 C                -2.44102300    2.42217400    0.00000000
 C                 0.68600000   -4.26037000    0.00000000
 C                -0.68632300   -4.26013600    0.00000000
 C                 4.25808500   -0.67810100    0.00000000
 C                -4.25798700   -0.67784300    0.00000000
 C                 4.25789200    0.67789400    0.00000000
 C                -4.25790500    0.67815100    0.00000000
 C                 0.68599900    4.26043200    0.00000000
 C                -0.68586900    4.26061300    0.00000000
 H                 0.00057000   -1.10169100    0.00000000
 H                -0.00064500    1.10167800    0.00000000
 H                 3.21982300   -3.17967100    0.00000000
 H                -3.21964400   -3.17926500    0.00000000
 H                 3.21969900    3.17907900    0.00000000
 H                -3.21984800    3.17936700    0.00000000
 H                 1.34702200   -5.11705700    0.00000000
 H                -1.34768400   -5.11656600    0.00000000
 H                 5.10635600   -1.35193800    0.00000000
 H                -5.10625200   -1.35169400    0.00000000
 H                 5.10596000    1.35197600    0.00000000
 H                -5.10606600    1.35212700    0.00000000
 H                 1.34737700    5.11686400    0.00000000
 H                -1.34692600    5.11729600    0.00000000
```

Again, we use g09 or g16 to do the EDMA calculation.

Now, all the Gaussian related calculations are done.

In the following calculations, we need the gaussian *.fchk files, we use the Gaussian built-in command **formchk** to generate the *.fchk file based on output *.chk. The *.fchk file contains readable force constant matrix information that is needed in dushin calculation.

```
$ formchk porphine-s0.chk
$ formchk porphine-s1.chk
$ formchk porphine-edma.chk
```

TIPS: To do EDMA calculation, use **Gaussview** to open `porphine-s0.log` file, and save as `porphine-edma.com`. Then, modify the first few lines of `porphine-edma.com` to suit for EDMA calculation.

Vibration Analysis

The evc calculation requires the basic information on initial and final electronic states. Thus, to begin an evc calculation, you need to designate the related file names in MOMAP input file (i.e., momap.inp).

For the Gaussian output files, you have to provide the corresponding .fchk files as well, as done in the last section. The numfreq-es.out file is obtained from the later Numfreq calculation.

The momap.inp for evc calculation is straightforward and is shown as follows:

```
[evc]$ cat momap.inp
do_evc      = 1                          # toggle dushin rotation effect, 1 or 0

&evc
  ffreq(1)   = "porphine-s0.log"         # log file of ground state
  ffreq(2)   = "porphine-s1.log"         # log file of excited state
  ffdipd     = "numfreq-es.out"          # derivative file of transition dipole moments from numfreq calc.
/
```

TIPS: In each directory, there exists a README file, just follow the instructions in README to carry out the operations. For example, the README in evc is shown as follows:

```
                      How to run MOMAP

1) Copy the following gaussian files from upper directory:

  ../gaussian/porphine-s0.fchk
  ../gaussian/porphine-s0.log
  ../gaussian/porphine-s1.fchk
  ../gaussian/porphine-s1.log
  ../numfreq/PES-0/numfreq-es.out

  to this directory.

2) Change momap.inp accordingly.

3) Run MOMAP to do the calculation by the following command:
   ./run
```

Copy the following gaussian output files from upper directory:

```
../gaussian/porphine-s0.fchk
../gaussian/porphine-s0.log
../gaussian/porphine-s1.fchk
../gaussian/porphine-s1.log
../numfreq/PES-0/numfreq-es.out
```

to this evc work directory.

A run file is also created, and is shown as follows:

```
#!/bin/sh

momap -input momap.inp -np 4
```

Users may modify the `run` file, for example, by changing the `np` option from 4 to 8, and perform the calculation by running the script file:

```
$ ./run
```

The result files are as follows:

```
[evc]$ ls
evc.cart.abs  evc.dint.dat  evc.out         nodefile            porphine-s1.fchk
evc.cart.dat  evc.dx.v.xyz  evc.vib1.xyz    numfreq-es.out      porphine-s1.log
evc.cart.dip  evc.dx.x.com  evc.vib2.xyz    porphine-s0.fchk    README
evc.dint.abs  evc.dx.x.xyz  momap.inp       porphine-s0.log     run
```

Note that if one uses the Gaussian g16 to do the calculations, the process can be greatly simplified without doing the time-consuming `Numfreq calculation` as mentioned later. As with Gaussian 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 the Gaussian g16 manual for more details.

Thus, one can simply add a line, e.g., `ftdipd = "porphine-s1.log"` in the momap.inp file.

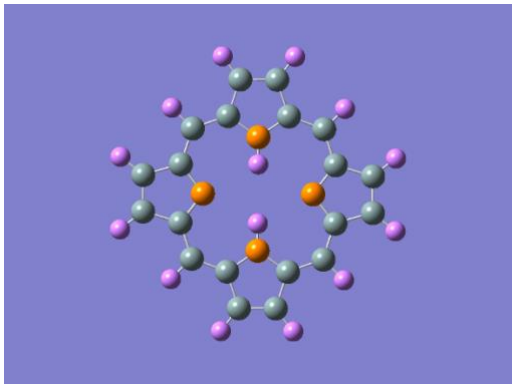
```
[evc]$ cat momap.inp
do_evc      = 1                         # toggle dushin rotation effect, 1 or 0

&evc
  ffreq(1)   = "porphine-s0.log"        # log file of ground state
  ffreq(2)   = "porphine-s1.log"        # log file of excited state
  ffdipd     = "porphine-s1.log"        # derivatives of transition dipole moments from g16 calc.
/
```

Then, one can do the evc calculation as usual.

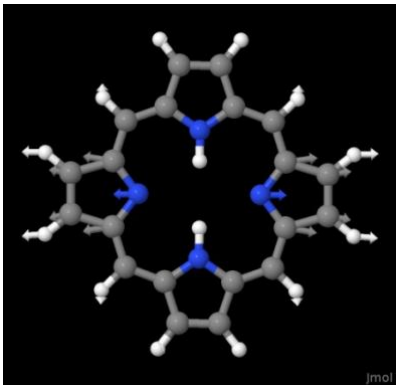
- **evc.cart.dat**: includes frequency, Huang－Rhys factor, and Duschinsky matrix (Cartesian coordinate system).
- **evc.dint.dat**: includes frequency, Huang－Rhys factor, and Duschinsky matrix (D solved by using internal coordinate system).
- **evc.cart.abs**: Duschinsky matrix file, used to plot 2D Duschinsky figure.
- **evc.cart.inp**: Projection of derivatives of transition dipoles to normal modes.
- **evc.dx.x.com**: Molecular overlapping figure of two electron states (viewed by using Gaussview)
- **evc.dx.x.xyz**: Molecular overlapping figure of two electron states (viewed by using Jmol)
- **evc.dx.v.xyz**: Molecular displacement vectors of two electron states (viewed by using Jmol)
- **evc.vib1.xyz**: Molecular vibrational vectors at ground state (viewed by using Jmol)
- **evc.vib2.xyz**: Molecular vibrational vectors at excited state (viewd by using Jmol)
- **evc.out**: evc log file
- **numfreq-es.out**: derivative file of transition dipole moments

Except for `ffreq(1)` and `ffreq(2)` parameters, the `evc` program also allows user to project reorganization energy onto the internal coordinate, to take account of isotope effect, and to configure many other advanced settings *etc.*, please refer to the **MOMAP User Guide** for details.

Please check the reorganization energy results between `evc.cart.dat` and `evc.dint.dat`. If the energy difference is small (< 1000 cm^{-1}), then use the results in `evc.cart.dat` to do the next calculations. However, if the energy difference is large, then use `evc.dint.dat` to do the next calculations.



evc.dx.x.com



evc.dx.v.xyz

Electron vibration coupling

Adiabatic Excitation Energy

Before we can calculate the Fluorescence Spectrum, we need to known the adiabatic excitation energy E_{ad}. The adiabatic excitation energy is the energy difference between the relaxed excited state energy and the ground state energy.

From the S_0 Gaussian log file, locate the last line with "`SCF Done`" in the output `porphine-s0.log` file in order to find the single point energy at the optimized S_0 geometry.

For example, you may use the following commands:

```
$ cat porphine-s0.log | grep "SCF Done"
```

In this example, the last line with "`SCF Done`" is like the following:

```
SCF Done:  E(RB3LYP) =  -989.551251544     A.U.
```

Thus, we have the energy E_{gs} at optimized ground state geometry:

E_{gs} = -989.551251544 a.u.

From the S_1 Gaussian log file, locate the last line with "`Total Energy, E(TD-HF/TD-KS)`" in the output `porphine-s1.log` file in order to find the single point energy at the optimized S_1 geometry.

For example, you may use the following commands:

```
$ cat porphine-s1.log | grep "Total Energy, E(TD-HF/TD-KS)"
```

In this example, the last line with "`Total Energy, E(TD-HF/TD-KS)`" is like the following:

```
Total Energy, E(TD-HF/TD-KS) =  -989.468129912
```

Then, we have the single point energy E_{es} at the optimized S_1 geometry:

E_{es} = -989.468129912 a.u.

From the above obtained ground state S_0 and excited state S_1 energies, we can obtain the adiabatic excitation energy E_{ad}: -

$E_{ad} = E_{es} - E_{gs}$ = [(-989.468129912) - (-989.551251544)] a.u

= **0.083122** a.u

TIPS: To find the energies, users may use **Gaussview** to open the Gaussian log file, from the menu item **Results** | **Summary** to obtain the value, which is valid for both the ground state and excited state.

Electronic Transition Dipole

The Gaussian log file for the optimized S_1 excited state has already included the Dipole Square of Electronic Transition Dipole Absorption (EDMA) and the Dipole Square of Electronic Transition Dipole Emission (EDME) information.

Open `porphine-s1.log` file with `vim`, for example, search the string "`transition electric dipole moments`", the first match is shown as follows:

```
1257  ********************************************************************
1258  Excited states from <AA,BB:AA,BB> singles matrix:
1259  ********************************************************************
1260
1261  1PDM for each excited state written to RWF  633
1262  Ground to excited state transition densities written to RWF  633
1263  Ground to excited state transition electric dipole moments (Au):
1264        state          X           Y           Z        Dip. S.      Osc.
1265          1         0.0006     -0.0053      0.0000      0.0000      0.0000
1266          2         0.0121     -0.0005      0.0000      0.0001      0.0000
1267          3        -0.0021     -2.2710      0.0000      5.1574      0.4217
1268  Ground to excited state transition velocity dipole moments (Au):
1269        state          X           Y           Z        Dip. S.      Osc.
1270          1        -0.0001     -0.0042      0.0000      0.0000      0.0001
1271          2        -0.0038      0.0000      0.0000      0.0000      0.0001
1272          3         0.0002      0.2593      0.0000      0.0672      0.3654
1273  Ground to excited state transition magnetic dipole moments (Au):
1274        state          X           Y           Z
                                                                1248,26       0%
```

Focus on the "`Dip. S.`" column, this is the Dipole Square of the calculated Electronic Transition Dipole Absorption (EDMA), take note the value of the first excited state, i.e., `0.0000`.

Thus, we have EDMA = **0.0** Debye in this case, we will fix it later on.

If the Linux command `vim` is used, then press **`SHIFT + N`**, the search jumps to the last occurrence of "`transition electric dipole moments`", shown as follows:

```
190862  ********************************************************************
190863  Excited states from <AA,BB:AA,BB> singles matrix:
190864  ********************************************************************
190865
190866  1PDM for each excited state written to RWF  633
190867  Ground to excited state transition densities written to RWF  633
190868  Ground to excited state transition electric dipole moments (Au):
190869        state          X           Y           Z        Dip. S.      Osc.
190870          1        -0.1491      0.0001      0.0000      0.0222      0.0012
190871          2        -0.0001      0.0502      0.0000      0.0025      0.0001
190872          3         0.0000      0.0000      0.0000      0.0000      0.0000
190873  Ground to excited state transition velocity dipole moments (Au):
190874        state          X           Y           Z        Dip. S.      Osc.
190875          1         0.0068      0.0000      0.0000      0.0000      0.0004
190876          2         0.0000     -0.0070      0.0000      0.0000      0.0004
190877          3         0.0000      0.0000      0.0000      0.0000      0.0000
190878  Ground to excited state transition magnetic dipole moments (Au):
190879        state          X           Y           Z
190880          1         0.0000      0.0000      0.0000
190881          2         0.0000      0.0000     -0.0001
search hit TOP, continuing at BOTTOM                                  190868,26     98%
```

Again, focus on the "`Dip. S.`" column, this is the Dipole Square of the calculated Electronic Transition Dipole Emission (EDME), take note the value of the first excited state, i.e., `0.0649`. Thus, we have:

$$\mu_{trans} = \sqrt{\mu_{trans}^2} = \sqrt{0.0222} \text{ a.u.} = 0.148997 \text{ a.u.}$$

$$= 0.148997 \text{ a.u.} \times 2.5417 \text{ Debye/a.u.}$$

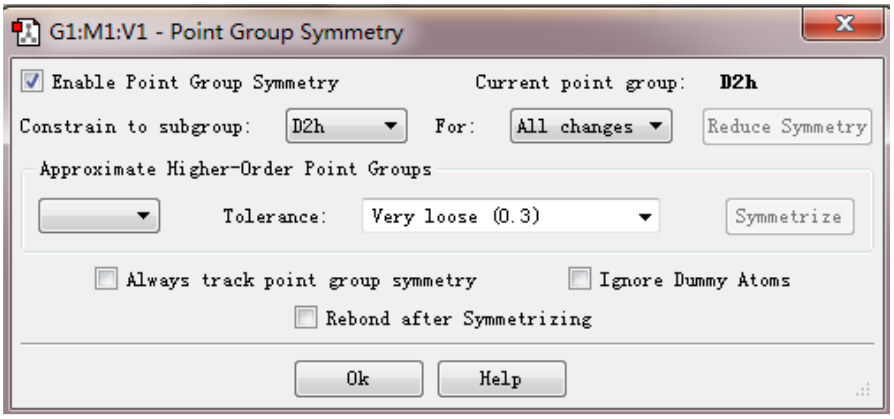
$$= 0.378705 \text{ Debye}$$

Thus, we have EDME = **0.378705** Debye.

TIPS: If the optimization and frequency calculations are separate, then the data of EDME and EDMA should be taken from the log file of excited state geometry optimization.

As can be seen, the obtained EDMA is 0, thus, we need to calculate the EDMA by other means, as shown in the following descriptions, the results are put in `porphine/gaussian-symm` directory.

We first assign symmetry to the ground state S_0 geomitry, then do optimization. Use Gaussview to open the initial S_0 configuration, click `Tools | Point Group...` to define symmetry, as shown in the following figure:



From Gaussview, save as the Gaussian input file `porphine-s0.com`, then do an S_0 optimization. When the optimization finishes, load `porphine-s0.log` into Gaussview and save the optimized geometry as `porphine-edma.com`. Next, do a **td** calculation to obtain the `porphine-edma.log`. From the file `porphine-edma.log`, search the string "`transition electric dipole moments`", we have:

```
 910  ********************************************************************
 911  Excited states from <AA,BB:AA,BB> singles matrix:
 912  ********************************************************************
 913
 914  1PDM for each excited state written to RWF  633
 915  Ground to excited state transition densities written to RWF  633
 916  Ground to excited state transition electric dipole moments (Au):
 917        state          X            Y            Z        Dip. S.      Osc.
 918          1         0.0000       0.0000       0.0086       0.0001     0.0000
 919          2         0.0000       0.0152       0.0000       0.0002     0.0000
 920          3         0.0000       0.0000       0.0000       0.0000     0.0000
 921  Ground to excited state transition velocity dipole moments (Au):
 922        state          X            Y            Z        Dip. S.      Osc.
 923          1         0.0000       0.0000       0.0039       0.0000     0.0001
 924          2         0.0000      -0.0041       0.0000       0.0000     0.0001
 925          3         0.0000       0.0000       0.0000       0.0000     0.0000
 926  Ground to excited state transition magnetic dipole moments (Au):
 927        state          X            Y            Z
 928          1         0.0000       0.0000       0.0000
/transition electric dipole moments                          916,26        60%
```

$$\mu_{trans} = \sqrt{\mu_{trans}^2} = \sqrt{0.0001}\ \text{a.u.} = 0.01\ \text{a.u.}$$

$$= 0.01\ \text{a.u.} \times 2.5417\ \text{Debye/a.u.}$$

$$= 0.025417\ \text{Debye}$$

Thus, we have EDMA = **0.025417** Debye.

Fluorescence Spectrum Calculation

Radiative rate k_r

Next, we create a directory `kr` and go to that directory.

Copy the following `evc` files from upper directory:

```
../evc/evc.cart.dat
../evc/evc.cart.dip
```

to this `kr` work directory.

Create a `momap.inp` with its contents as follows:

```
[kr]$ cat momap.inp
do_spec_tvcf_ft     = 1                     # toggle correlation function calculation, 1 or 0
do_spec_tvcf_spec   = 1                     # toggle fluorescence spectrum calcluation, 1 or 0

&spec_tvcf
  DUSHIN          = .t.                     # toggle Duschinsky rotation effect, .t. or .f.
  HERZ            = .t.                     # toggle Herzber-Teller effect, .t. or .f.
  Temp            = 300 K                   # temperature
  tmax            = 3000 fs                 # integration time
  dt              = 0.01   fs               # integration timestep
  Ead             = 0.083122 au             # adiabatic excitation energy
  EDMA            = 0.025417 debye          # electronic dipole moment of absorption (GS)
  EDME            = 0.378705 debye          # electronic dipole moment of emission (ES)
  DSFile          = "evc.cart.dat"          # input dushin file name
  DDplFile        = "evc.cart.dip"          # derivative file of transition dipole moments
  Emax            = 0.3 au                  # upper bound of spectrum frequency
  dE              = 0.00001 au              # output energy interval
  logFile         = "spec.tvcf.log"         # output file for logging
  FtFile          = "spec.tvcf.ft.dat"      # output file for correlation function info
  FoFile          = "spec.tvcf.fo.dat"      # output file for spectrum function info
  FoSFile         = "spec.tvcf.spec.dat"    # output file for spectrum info
/
```

Also create a `run` file and change it with execution attribute (e.g., `chmod a+rx run`), the `run` file is shown as follows:

```
#!/bin/sh

momap -input momap.inp -np 4  &> log &
```

Users may modify the `run` file, for example, by changing the `np` option from 4 to 8, and perform the calculation by running the script file:

```
$ ./run
```

When the calculation finishes, the result files are shown as follows:

```
[kr]$ ls
evc.cart.dat    nodefile    spec.tvcf.fo.dat    spec.tvcf.log
evc.cart.dip    README      spec.tvcf.ft.dat    spec.tvcf.spec.dat
momap.inp       run         spec.tvcf.ft.gnu    spec.tvcf.spec.gnu
```

The radiative decay rate constant can be found at the end of `spec.tvcf.log` file, while the fluorescence spectrum information can be obtained from `spec.tvcf.spec.dat`.

From the end of `spec.tvcf.log` file, we obtain the radiative rate is **1.57621496×10^{8}** s^{-1} by considering the Herzber-Teller effect, as shown below:

```
radiative rate      (0):     8.95372133E-13    3.70159136E+04 /s,    27015.41 ns
radiative rate      (1):    -8.99110573E-15   -3.71704659E+02 /s, ********** ns
radiative rate      (2):     3.81179529E-09    1.57584852E+08 /s,        6.35 ns
radiative rate (total):      3.81268167E-09    1.57621496E+08 /s,        6.34 ns
```

However, if the Herzber-Teller effect is not taken into account, the radiative rate would be **2.38508923×10^{5}** s^{-1}, as shown below:

```
 Vertical Energy      2       is:       18073.00400 cm-1
  Spectra overlap integral = I ( Energy^-1 )
  I^-1 =         0.00215032 Hartree =      471.94140365 cm-1 =       0.05851330 eV

radiative rate      (0):     5.76925496E-12    2.38508923E+05 /s,    4192.72 ns
```

Plot the data from file `spec.tvcf.spec.dat` by using columns 3, 5, and 6, in Linux, we can use **Gnuplot** to do the plotting, the plot script is shown as follows:

```
[kr]$ cat spec.tvcf.spec.gnu
reset
set nogrid
set lmargin 10
set pointsize 1.0
set encoding iso_8859_1
set term postscript eps enhanced color 20
set xlabel "Wave number, cm^{-1}" offset 0,0
set ylabel "Intensity, a.u." offset 0,0

set xtics nomirror
set ytics nomirror

set xrange [15500:19500]
set yrange [0:1.15]
set output "spec.tvcf.spec.eps"

plot  \
  "spec.tvcf.spec.dat"  u 3:5 t "Absorption" w l lw 3 lt 1, \
  ""  u 3:6 t "Emission" w l lw 3 lt 2
```

Then use the following commands to generate the correlation and spectrum plots:

```
$ gnuplot *.gnu
$ ps2png *.eps
```

Or if your gnuplot has terminal `pngcairo`, then you can generate the png files in one step:

```
$ gnuplot *.gnu-png
$ display *.png
```

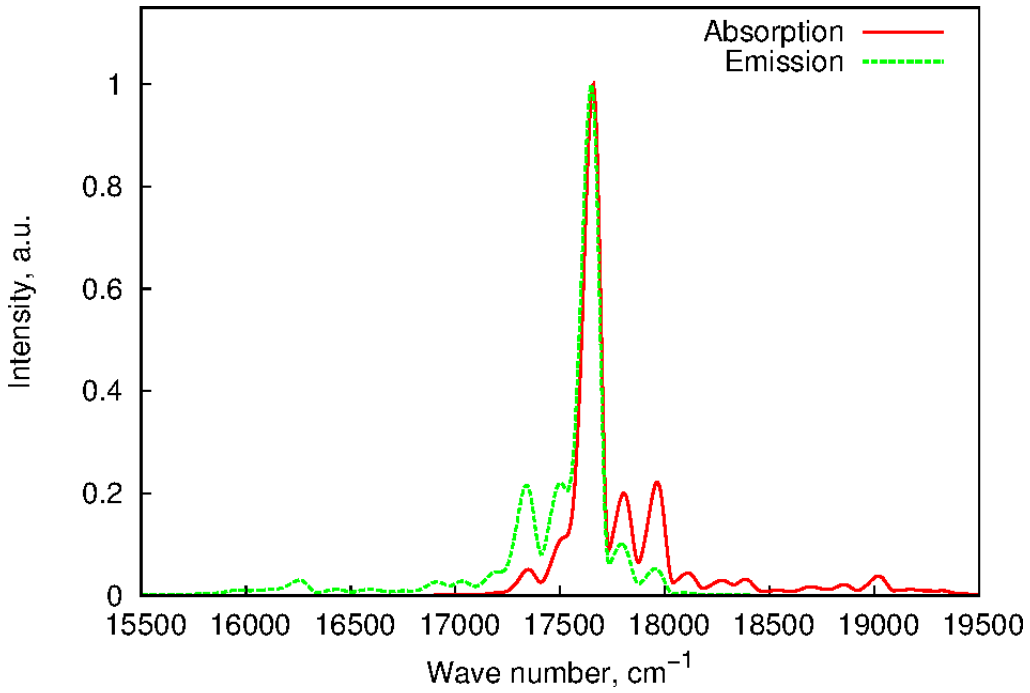


Fig. 4 Absorption and emission spectrum

The script ps2png is used to convert a .eps file to .png file, with its contents as follows:

```
$ cat  ~/bin/ps2png
#!/usr/bin/perl -w
#
# ps2png [resolution] file...
#
# Convert a postscript file to PNG, using the gs (GhostScript) command.   The
# resolution defaults to 200, which is a readable compromise for most screens.
# The files should be postscript files.  You can omit a .ps suffix and  we'll
# assume it.

# Author: John Chambers <jc@trillian.mit.edu>

$ENV{LD_LIBRARY_PATH} = '/usr/X11R6/lib/:/usr/eecs/lib:/usr/lib:/usr/lib/aout';

if ( @ARGV == 0 )
{
    print "Usage: ps2png [resolution] file...\n";
    exit $?;
}

if (($res = $ARGV[0]) =~ /^\d+$/) {shift @ARGV} else {$res = 200}

file: for $file (@ARGV) {
    if ($file =~ /(.*)\.(\w*ps)$/i) {
        $fili = $file;
        $filo = "$1.png";
    } else {
        if      (-f ($fili = "$file.ps" )) {$filo = "$file.png";
        } elsif (-f ($fili = "$file.eps")) {$filo = "$file.png";
        } elsif (-f ($fili = "$file.PS" )) {$filo = "$file.PNG";
        } else {
            print STDERR "Can't find postscript file for $file.\n";
            next file;
        }
    }
    system "gs -q -DNOPAUSE -sDEVICE=ppmraw -r$res -sOutputFile='|pnmcrop|pnmtopng > $filo' -- $fili";
    if ($?) {
        print STDERR "Conversion of \"$fili\" failed with exit status $?.\n";
        exit $?;
    }
}
```

TIPS: The ps2png script needs the pnmcrop and pnmtopng commands, which can be resolved by installing the netpbm packages:

```
$ yum install netpbm netpbm-progs        #  provide pnmcrop & pnmtopng etc.
```

Calculate numfreq

If the Herzber-Teller effect is to be considered, then we need to do `numfreq` calculation, the running directory is `porphine/numfreq`. The input files include: `porphine-s1.com`, `input`, and `nodefile`. Here, the `porphine-s1.com` file is the optimized S_1 geometry.

```
[numfreq]$ cat porphine-s1.com
%chk =porphine-s1.chk
%mem =32GB
%nprocl=4
%nprocs=16
#p td(root=1) b3lyp/6-31g(d)

numfreq-porphine-s1

0 1
  N               -2.11384400   -0.00001800    0.00000000
  N               -0.00006900   -2.05188400    0.00000000
  N                0.00006800    2.05188500    0.00000000
  N                2.11384300    0.00001700    0.00000000
  C               -2.89604400   -1.13283300    0.00000000
  C               -2.89618900    1.13269100    0.00000000
  C               -1.08901100   -2.88277400    0.00000000
  ……
  H                1.35313600   -5.12728300    0.00000000
  H                1.35301400    5.12739100    0.00000000
  H                5.11733400   -1.34781000    0.00000000
  H                5.11722500    1.34805600    0.00000000
```

Similar to the above calculations, first we copy the following files from upper directory:

`../gaussian/porphine-s1.com`

to the `numfreq` work directory.

Create a `momap.inp` with its contents as follows:

```
[numfreq]$ cat input
&control
  Qctype     = "gaussian"            # QC calculation method: Gaussian
  task       = "numfreq"             # computing task: numfreq
  fxyz       = "porphine-s1.com"     # molecular geometry file
  symm       = .false.               # consider symettry
  dx         = 0.01                  # can use the default value
/
```

In this example, we use PBS script to submit the jobs. If other job scheduling systems, for example, SLURM, LSF etc., are used, please use numfreq.pbs as a template, and make modifications accordingly. The keywords that need to be modified are: job name (e.g., -N), queue name (e.g., -q), number of nodes (e.g., node=3), processes per node (e.g., ppn=16), installation location of QC calculation software (e.g., Gaussian, qcpath="/home/gaussian/g09/g09"), installation location of MOMAP software (e.g., program="$HOME/MOMAP/bin/numfreq.sh") etc.

The sample `pbs` file is shown as follows.

```
#!/bin/sh
#PBS -S /bin/bash
#PBS -N numfreq
#PBS -o stdout.txt
#PBS -j oe
#PBS -q work
#PBS -l nodes=1:ppn=16
#PBS -V

#-------------------------------------------------------------------------------
if [ -z $PBS_O_WORKDIR ]; then
    export PBS_O_WORKDIR=`pwd`
    export PBS_NODEFILE=$PBS_O_WORKDIR/nodefile
    export PBS_NUM_PPN=`cat $PBS_NODEFILE | wc -l`
    export PBS_JOBNAME="numfreq"
    export PBS_QUEUE="queue"
fi
#-------------------------------------------------------------------------------
jobname=${PBS_JOBNAME}
username=`whoami`
#-------------------------------------------------------------------------------
cd $PBS_O_WORKDIR
echo $PBS_NODEFILE > nodefile
echo "Starting Gaussian run at" `date`
#
echo
echo "PBS variables:"
echo
export | grep PBS
echo
#-------------------------------------------------------------------------------
# run numforce
# qcpath=`which dscf`
#-------------------------------------------------------------------------------
qcpath=`which g09`
#-------------------------------------------------------------------------------
scratch=/tmp/$USER
program=numfreq.sh
posfix=${PBS_JOBNAME}_${PBS_QUEUE}
#-------------------------------------------------------------------------------
# Do Gaussian PES scanning
#
#PBS_NUM_PPN=1
$program                            \
 -qctype            "gaussian"      \
 -qcpath            "$qcpath"       \
 -scratch           "$scratch"      \
 -nodefile          "nodefile"      \
 -ncore             "$PBS_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`
```

Then, perform the calculation by submitting the script file:

```
$ qsub numfreq.pbs
```

Finally, the result files are shown as follows:

```
[numfreq]$ tree ./
./
├── porphine-s1.com
├── input
├── nodefile
├── numfreq.pbs
└── PES-0
    ├── coord-00-00.com
    ├── coord-00-00.log
    ├── coord-01-dx.com
    ├── coord-01+dx.com
    ├── coord-01-dx.log
    ├── coord-01+dx.log
    ├── coord-01-dy.com
    ├── coord-01+dy.com
    ├── coord-01-dy.log
    ├── coord-01+dy.log
    ├── coord-01-dz.com
    ├── coord-01+dz.com
    ├── coord-01-dz.log
    ├── coord-01+dz.log
    ......
    ├── coord-18-dx.com
    ├── coord-18+dx.com
    ├── coord-18-dx.log
    ├── coord-18+dx.log
    ├── coord-18-dy.com
    ├── coord-18+dy.com
    ├── coord-18-dy.log
    ├── coord-18+dy.log
    ├── coord-18-dz.com
    ├── coord-18+dz.com
    ├── coord-18-dz.log
    ├── coord-18+dz.log
    ├── files.out
    ├── numfreq-es.out
    ├── vibp.xyz
    └── vib.xyz

1 directory, 226 files
```

As can be seen, a `PES-0` directory is created, and there also exists a `numfreq-es.out` file. Copy this `numfreq-es.out` file to your `evc` work directory. Modify the input file `momap.inp`, and add keywords

```
ftdipd = "numfreq-es.out"
```

into the `evc` block. Then change the keyword `Herz` to `.t.`

```
HERZ = .t.
```

After running the momap program, you will get `evc.cart.dip` file, which include the transition dipole in the first line and transition dipole moment derivitives for the other 3N-6 lines.

Verify Convergence of Correlation Function

Correlation function must be converged before obtaining any calculation results. To verify, plot a graph using the first 2 columns in `spec.tvcf.ft.dat`, which are time and real part of the correlation function (`TVCF_RE`). `TVCF_RE` should be very close to zero and stop oscillating before it reaches the integration time limit. Figure 7 shows the distribution of a converged correlation function.

The Gnuplot plot script for the figure is shown as follows:

```
[kr]$ cat spec.tvcf.ft.gnu
reset
set nogrid
set lmargin 10
set pointsize 1.0
set encoding iso_8859_1
set term postscript eps enhanced color 20
set xlabel "Time, fs" offset 0,0
set ylabel "TVCF (RE)" offset 0,0
set xtics nomirror
set ytics nomirror

set xrange [-1000:1000]
set output "spec.tvcf.ft.eps"
plot "spec.tvcf.ft.dat"  u 1:2 t "" w l lw 3 lt 1
```

Then use the following commands to generate the graph:

```
$ gnuplot *.gnu
$ ps2png *.eps
```

Or if your gnuplot has terminal `pngcairo`,

```
$ gnuplot *.gnu-png
$ display *.png
```

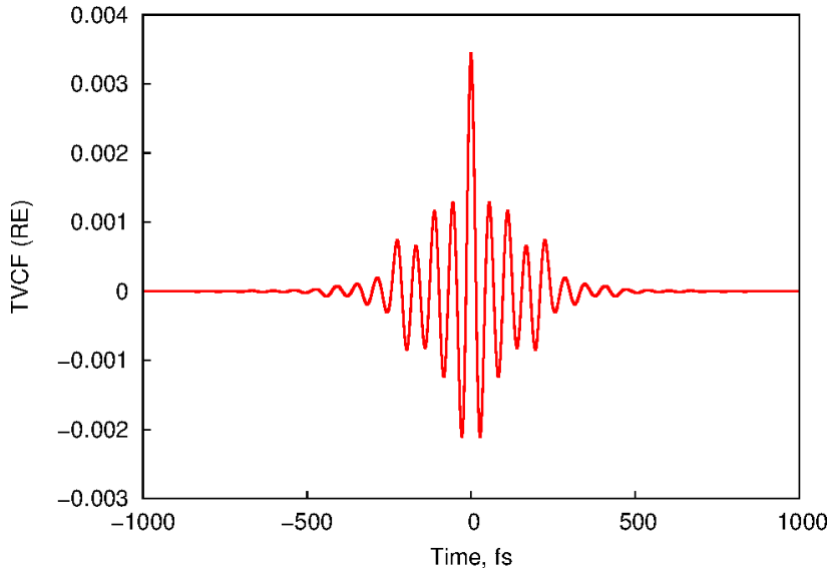


Fig. 7 Distribution of time vs real part of a converged correlation function