MOMAP

Tutorial 02

Phosphorescence Spectrum Calculation

Molecular Material Property Prediction Package

Version 2019

September, 2019

MOMAP Tutorial 02

Version 2019 edited by:

Dr. Qikai Li

Dr. Yingli Niu

Ms. Lihui Yan

MOMAP Tutorial

- Phosphorescence Spectrum Calculation

Ir(PPy)₃ is able to emit light efficiently from the excited triplet state, due to the enhancement of intersystem crossing (ISC) induced by the Ir atom. The ISC efficiency is a result of mixing of singlet and triplet states, so that the normally forbidden singlet-triplet transitions become allowed.

MOMAP is able to simulate the phosphorescence 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 phosphorescence 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 of Phosphorescence Spectrum are as follows:
1. Gaussian calculations
2. Obtain EDMA, EDME, and Hso (by using Dalton)
3. Vibration analysis *etc*.
4. Phosphorescence spectrum calculation

Contents

Gaussian Calculations

Optimization calculation on ground state (S$_0$)

Once the initial geometry is obtained, we have to find the optimized S$_0$ ground state geometry. The route section is set as **#p opt b3lyp/gen pseudo=read**.

The initial geometry gaussian S$_0$ input file (`Irppys-s0-opt.com`) is shown as follows:

```
%chk=Irppy3-s0-opt.chk
%mem=32GB
%nprocl=4
%nprocs=16
#p opt b3lyp/gen pseudo=read

Ir Complex No.1 - S0 Opt

0 1
  Ir            -0.00026100    0.00081800   -0.02546400
  C              1.23475200    1.22563100   -1.06555400
  C              3.17477000    2.89735700   -2.32701500
  C              0.92442700    1.99391300   -2.20485100
  C              2.57228300    1.32514600   -0.57850700
  C              3.52102800    2.15413000   -1.20673700
  ......
  C              4.16509200    0.43765200    1.21020600
  H              5.33396800   -0.43571700    2.78791400
  H              3.39236400   -1.78733400    3.66167400
  H              1.19605700   -1.58570100    2.48734100
  H              4.98966700    1.01859900    0.81497000

C H N 0
6-31G**
****
Ir 0
LANL2DZ
****

Ir 0
LANL2DZ
```

We use g09 or g16 to do the geometry optimization.

Optimization calculation on lowest triplet excited state (T$_1$)

With the optimized S$_0$ geometry at hand, we can start optimizing T$_1$ geometry using the optimized S$_0$ geometry as the initial structure. The route section is set as **#p opt b3lyp/gen pseudo=read**.

Again, the initial gaussian T_1 input file (`Irppy3-t1-opt.com`) is shown as follows:

```
%chk=Irppy3-t1-opt.chk
%mem=32GB
%nprocl=1
%nprocs=16
#p opt b3lyp/gen pseudo=read

Ir Complex No.1 - T1 opt

0 3
  Ir              0.00043600    0.00028600    0.02544800
  C               0.20235100    1.72880300    1.06370900
  C               0.32555100    4.28900800    2.32011100
  C               1.00022100    1.95958100    2.20157500
  C              -0.54479300    2.84229300    0.57608300
  ......
  H              -3.51910800    1.55565400   -3.64762700
  H              -1.99610700   -0.04366300   -2.47809500
  H              -2.28620800    4.55115900   -0.81265400

C H N 0
6-31G**
****
Ir 0
LANL2DZ
****

Ir 0
LANL2DZ
```

Or

```
%chk=Irppy3-t1-opt.chk
%mem=32GB
%nprocl=1
%nprocs=16
#p opt td b3lyp/gen pseudo=read

Ir Complex No.1 - T1 opt

0 1
  Ir              0.00043600    0.00028600    0.02544800
  C               0.20235100    1.72880300    1.06370900
......
```

Again, use g09 or g16 to do the T_1 optimization. Here, the first option for T1 optimization is recommended, as it takes much less time.
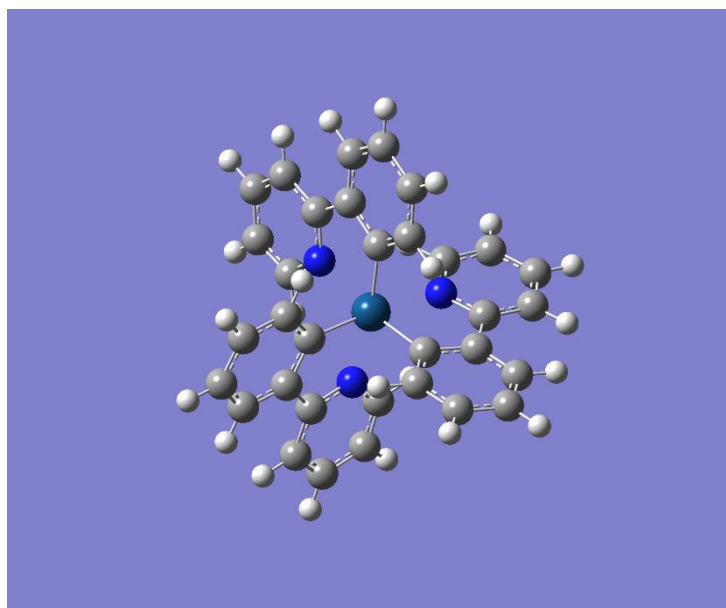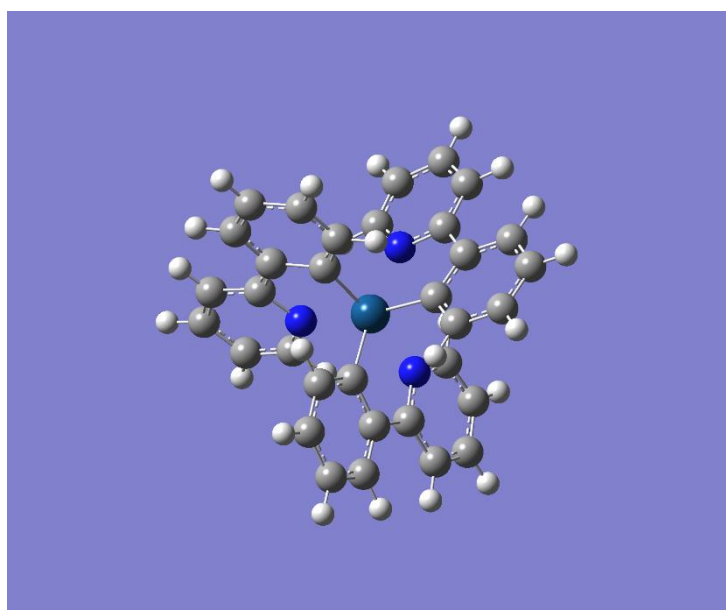
Fig. 1 Optimized S_0 geometry

Fig. 2 Optimized T_1 geometry

TIPS: To obtain the optimized ground state geometry, use **Gaussview** to open `Irppy3-s0-opt.log` file, and save as `Irppy3-t1-opt.com`. Then, modify the first and last few lines of `Irppy3-t1-opt.com`.

Frequecncy calculation on ground state (S$_0$)

The gaussian S$_0$ input file (`Irppy3-s0-freq.com`) is shown as follows:

```
%chk=Irppy3-s0-freq.chk
%mem=32GB
%nprocl=1
%nprocs=16
#p freq b3lyp/gen pseudo=read

Ir Complex No.1 - S0 freq

0 1
  Ir             0.00043600    0.00028600    0.02544800
  C              0.20235100    1.72880300    1.06370900
  C              0.32555100    4.28900800    2.32011100
  C              1.00022100    1.95958100    2.20157500
  C             -0.54479300    2.84229300    0.57608300
  C             -0.47534300    4.10152400    1.20174700
  C              1.06299300    3.20800700    2.81873400
  H              1.57895800    1.13987100    2.61491300
  ......
  N             -1.33525800    1.32844300   -1.09053800
  C             -1.39357600    2.59757600   -0.59325500
  C             -2.23463700    3.54335400   -1.20684000
  H             -3.65253800    3.91910500   -2.77737000
  H             -3.51910800    1.55565400   -3.64762700
  H             -1.99610700   -0.04366300   -2.47809500
  H             -2.28620800    4.55115900   -0.81265400

C H N 0
6-31G**
****
Ir 0
LANL2DZ
****

Ir 0
LANL2DZ
```

We use g09 or g16 to do the S$_0$ frequency calculation.

> **TIPS**: Copy `Irppy3-t1-opt.com` to `Irppy3-s0-freq.com`, then, modify the first few lines of `Irppy3-s0-freq.com` to suit for S$_0$ frequency calculation.

Frequecncy calculation on lowest triplet excited state (T₁)

The gaussian T₁ input file (`Irppy3-t1-freq.com`) is as follows:

```
%chk=Irppy3-t1-freq.chk
%mem=32GB
%nprocl=4
%nprocs=16
#p freq b3lyp/gen pseudo=read

Ir Complex No.1 – T1 freq

0 3
  Ir             -0.01639400    0.00293500    0.05369300
  C              -0.49053500   -1.69965800    1.03808600
  C              -1.03578300   -4.22859100    2.22923000
  C              -1.33931300   -1.82209600    2.15439300
  C               0.08514300   -2.90172200    0.53285900
  C              -0.19369100   -4.14580000    1.12866200
  C              -1.60735000   -3.05769200    2.74067000
  H              -1.78775300   -0.92837100    2.57535000
  ......
  N               1.13309800   -1.50308700   -1.09078500
  C               0.97867800   -2.77250100   -0.62156900
  C               1.66682200   -3.82909800   -1.24466900
  H               3.03338800   -4.39374900   -2.80336300
  H               3.30179200   -2.02181900   -3.61545900
  H               2.03081900   -0.22386100   -2.43415800
  H               1.54633100   -4.84073100   -0.87668900

C H N O
6-31G**
****
Ir 0
LANL2DZ
****

Ir 0
LANL2DZ
```

Again, we use g09 or g16 to do the T₁ frequency calculation.

> **TIPS**: To obtain the optimized excited state geometry, use **Gaussview** to open `Irppy3-t1-opt.log` file, and save as `Irppy3-t1-freq.com`. Then, modify the first and last few lines of `Irppy3-t1-freq.com`.

Now, all the Gaussian related calculations are done.

In the following calculations, the gaussian *.fchk files are needed, 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 Irppy3-s0-freq.chk
$ formchk Irppy3-t1-freq.chk
```

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 `momap.inp` for an 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)  = "Irppy3-s0-freq.log"      # log file of ground state
  ffreq(2)  = "Irppy3-t1-freq.log"      # log file of excited state
/
```

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/Irppy3-s0-freq.fchk
  ../gaussian/Irppy3-s0-freq.log
  ../gaussian/Irppy3-t1-freq.fchk
  ../gaussian/Irppy3-t1-freq.log

  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/Irppy3-s0-freq.fchk
  ../gaussian/Irppy3-s0-freq.log
  ../gaussian/Irppy3-t1-freq.fchk
  ../gaussian/Irppy3-t1-freq.log
```

to this `evc` work directory.

A `run` file is also created, and 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
```

The result files are shown as follows:

```
[evc]$ ls
evc.cart.abs  evc.dx.x.com  Irppy3-s0-freq.fchk  momap.inp      run
evc.cart.dat  evc.dx.x.xyz  Irppy3-s0-freq.log   momap.inp-new
evc.dint.abs  evc.out       Irppy3-t1-freq.fchk  nodefile
evc.dint.dat  evc.vib1.xyz  Irppy3-t1-freq.log   README
evc.dx.v.xyz  evc.vib2.xyz  log                  ref
```

➢ **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.nac**: Projection of NACME to normal modes.
➢ **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

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.

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 `Irppy3-s0-freq.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 Irppy3-s0-freq.log | grep "SCF Done"
```

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

```
SCF Done:  E(RB3LYP) =  -1541.04579988    A.U.
```

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

E_{gs} = -1541.04579990 a.u.

From the T$_1$ Gaussian log file, locate the last line with "SCF Done" in the output `Irppy3-t1-opt.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 Irppy3-t1-opt.log | grep "SCF Done"
```

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

```
SCF Done:  E(UB3LYP) =  -1540.95161701    A.U.
```

Then, we have the single point energy E_{es} at the optimized S$_1$ geometry:

E_{es} = -1540.95161701 a.u.

From the above obtained ground state S$_0$ and excited state T$_1$ energies, we can obtain the adiabatic excitation energy E_{ad}:

E_{ad} = E_{es} - E_{gs} = [(-1540.95161701) - (-1541.04579988)] a.u

= 0.0941829 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 and Hso Calculation

To calculate the EDMA, EDME and Hso for Irppy3, we need to do the calculaitons on both the S_0 and T_1 states. Here, **we use DALTON to do the calculations**. As we already have had the Gaussian input and output files, we just need to convert the Gaussian files to the DALTON .mol input files, which can be done easily by a MOMAP python script called **`gjf2mol.py`**.

Create a directory `dalton`, enter into that directory, and copy the `Irppy3-s0-freq.com` and `Irppy3-t1-freq.com` to the directory.

```
$ mkdir dalton; cd dalton
$ cp ../gaussian/Irppy3-??-freq.com .
$ ./gjf2mol.py Irppy3-s0-freq.com
$ ./gjf2mol.py Irppy3-t1-freq.com
```

```
[dalton]$ ls
gjf2mol.py          Irppy3-s0-freq.mol  Irppy3-t1-freq.mol
Irppy3-s0-freq.com  Irppy3-t1-freq.com  soc.dal
```

From the directory, we can see two mol files are created. The contents of a mol file are shown as follows:

```
[dalton/edma]$ cat Irppy3-s0-freq.mol
ATOMBASIS
Generated by MOMAP
through gjf2mol.py
Atomtypes=4 Charge=0 Angstrom NoSymmetry
Charge=7.0 Atoms=3 Basis=6-31G**
N     0.57150200   -1.79394800   -1.09124700
N    -1.83890300    0.40222600   -1.09208100
N     1.26831900    1.39188900   -1.09116400
Charge=77.0 Atoms=1 Basis=lanl2dz_ecp   ECP=lanl2dz_ecp
Ir    0.00008200    0.00019800    0.02644000
Charge=6.0 Atoms=33 Basis=6-31G**
C     1.73548000   -0.12432800    1.06566000
C     4.29816300   -0.13125900    2.32321300
C     2.00197900   -0.91143300    2.20328500
C     2.81438400    0.67205200    0.57816900
……
Charge=1.0 Atoms=24 Basis=6-31G**
H     1.20947900   -1.52774300    2.61532200
H     4.88837500    1.27092000    0.82177100
……
H     3.75799300    3.81417100   -2.78288300
H     1.40350100    3.57558900   -3.65497700
H    -0.12942700    1.98827000   -2.48227300
H     4.44652500    2.48280900   -0.81331700
```

As the DALTON defaults to use **Bohr radius** as length unit, so we set it to **Angstrom** in this case. Also remember to make modification to a line related to **Ir**, that is, change to "**`Basis=lanl2dz_ecp ECP=lanl2dz_ecp`**".

In addition to the molecule file, we also need DALTON control files for EDMA, EDME, and SOC calculations, for example:

The contents of DALTON control file for **EDMA** and **EDME** calculation are shown as follows:

```
[dalton]$ cat edma.dal
**DALTON INPUT
.DIRECT
.PARALLEL
.RUN RESPONSE                      ! Run integrals, wave function, and response
**INTEGRALS
.SPIN-ORBIT                        ! Calculate SOC 1e, 2e integrals
**WAVE FUNCTIONS
.DFT
 B3LYP
*SCF INPUT                         ! HF input
.MAX DIIS ITERATIONS
 300
**RESPONSE                         ! Input for the response module
*QUADRATIC                         ! Calculate quadratic response
.ECPHOS                            ! Specifies a phosphorescence calculation
.APROP
XDIPLEN
.APROP
YDIPLEN
.APROP
ZDIPLEN
.BPROP
X1SPNSCA
.BPROP
Y1SPNSCA
.BPROP
Z1SPNSCA
.ISPABC
 0 1 1
.ROOTS
 1                                 ! Triplet
**END OF INPUT
```

Notes:

.APROP, .BPROP Specify the operators A and B, respectively. The line following this option should be the label of the operator as it appears in the file AOPROPER. These two keywords can be repeated for different properties.

XDIPLEN, YDIPLEN, ZDIPLEN   Direction of the applied electric field

.ISPABC   Spin symmetry of excitation operators associated with physical operators A (ISPINA) and B (ISPINB), and the excited states specified with .ROOTS (ISPINC): "0" for singlet and "1" for triplet. Default is "0,0,0", i.e. all of singlet spin symmetry.

The contents of DALTON control file for **Hso** calculation are shown as follows:

```
[dalton]$ cat soc.dal
**DALTON INPUT
.DIRECT
.PARALLEL
.RUN RESPONSE
**INTEGRALS
.SPIN-ORBIT
**WAVE FUNCTIONS
.DFT
 B3LYP
*SCF INPUT
.MAX DIIS ITERATIONS
 300
**RESPONSE
.TRPFLG
*LINEAR
.SINGLE RESIDUE
.PROPRT
X1SPNSCA
.PROPRT
Y1SPNSCA
.PROPRT
Z1SPNSCA
.ISPABC
 0 1 1
.ROOTS
 1
```

The molecule file for **EDMA** calculation is **Irppy3-s0-freq.mol**, while the molecule file for **EDME** and **Hso** calculation is **Irppy3-t1-freq.mol**, as shown below:

```
[dalton]$ cat Irppy3-t1-freq.mol
ATOMBASIS
Generated by MOMAP
through gjf2mol.py
Atomtypes=4 Charge=0 Angstrom NoSymmetry
Charge=7.0 Atoms=3 Basis=6-31G**
N    -1.81693100   -0.04119500   -1.12840500
N     0.88745100    1.66140100   -1.08084500
N     1.01965700   -1.58551300   -1.08420700
Charge=77.0 Atoms=1 Basis=lanl2dz_ecp ECP=lanl2dz_ecp
Ir   -0.01622700    0.00540200    0.05635400
Charge=6.0 Atoms=33 Basis=6-31G**
C    -0.62604700   -1.65690400    1.03633100
C    -1.37514700   -4.13735100    2.21899300
C    -1.49005500   -1.71448700    2.14629900
……
Charge=1.0 Atoms=24 Basis=6-31G**
H    -1.87095200   -0.78930900    2.56583400
H    -0.14879700   -5.06093700    0.73220700
……
H     2.01637700   -0.37959700   -2.42547500
H     1.18202800   -4.94419400   -0.86229500
```

A sample DALTON **run** file for EDMA calculation is shown as follows:

```bash
#!/bin/bash

export PATH=/opt/dalton-2018.2:$PATH
export DALTON_TMPDIR=/tmp

module load dalton/2018.2

nohup dalton -mb 120 -noarch -nodelist hosts -N 12 edma Irppy3-s0-freq &> log &
```

The DALTON calculation results for **EDMA** are shown as follows:

```
Transition energy:  2.630 eV
   or   471.394 nm

 Length gauge / effective charge spin-orbit integrals:
 Partial rates (ECSO): X-polarization 6.37282E+05 Transition moment :  0.182
 Length gauge / effective charge spin-orbit integrals:
 Partial rates (ECSO): Y-polarization 6.40145E+05 Transition moment :  0.182
 Length gauge / effective charge spin-orbit integrals:
 Partial rates (ECSO): Z-polarization  5142.6     Transition moment :  1.631E-02

  Phosphorescence - length gauge / effective charge spin-orbit integrals:
 Oscillator strength (/2PI)     (ECSO)     6.800284E-04
 Dipole strength [a.u.]         (ECSO)     6.630809E-02
 Dipole strength E-40 [esu**2 cm**2]       4.283817E+03
 Total transition rate          (ECSO)     4.275233E+05 s-1
 Total phosphorescence lifetime (ECSO)     2.339054E-06 s
```

From the above figure, we can obtain the EDMA:

$$\sqrt{0.182^2 + 0.182^2 + 0.01631^2} \ \mathrm{a}u \ \times 2.5417 \ \mathrm{debye/au} = \mathbf{0.655512} \ \mathrm{debye}$$

Similarly, for the EDME calculation, the .dal control file is the same as the that of EDMA, but the .mol molecule file is **Irppy3-t1-freq.mol** instead.

The DALTON calculation results for **EDME** are shown as follows:

```
Transition energy:  2.153 eV
   or   575.832 nm

 Length gauge / effective charge spin-orbit integrals:
 Partial rates (ECSO): X-polarization 1.54617E+05 Transition moment :  0.121
 Length gauge / effective charge spin-orbit integrals:
 Partial rates (ECSO): Y-polarization  21916.     Transition moment :  4.544E-02
 Length gauge / effective charge spin-orbit integrals:
 Partial rates (ECSO): Z-polarization  4526.6     Transition moment :  2.065E-02

  Phosphorescence - length gauge / effective charge spin-orbit integrals:
 Oscillator strength (/2PI)     (ECSO)     1.432485E-04
 Dipole strength [a.u.]         (ECSO)     1.706244E-02
 Dipole strength E-40 [esu**2 cm**2]       1.102314E+03
 Total transition rate          (ECSO)     6.035318E+04 s-1
 Total phosphorescence lifetime (ECSO)     1.656914E-05 s
```

From the above figure, we can obtain the EDME:

$$\sqrt{0.121^2 + 0.04544^2 + 0.02065^2} \ \mathrm{a}u \ \times 2.5417 \ \mathrm{debye/au} = \mathbf{0.332683} \ \mathrm{debye}$$

The formula for calculating the **Hso** is shown as follows:[1]

$$\left\langle S_0 \left| \hat{H}^{SO} \right| T_1 \right\rangle = \sqrt{\left(\left| \left\langle S_0 \left| \hat{H}^{SO} \right| T_{1,x} \right\rangle \right|^2 + \left| \left\langle S_0 \left| \hat{H}^{SO} \right| T_{1,y} \right\rangle \right|^2 + \left| \left\langle S_0 \left| \hat{H}^{SO} \right| T_{1,z} \right\rangle \right|^2 \right) / 3}$$

The DALTON calculation results for Hso are shown as follows:

```
@ Excited state no:    1 in symmetry 1  ( A  ) - triplet excitation
  ----------------------------------------------------------------

@ Excitation energy :  7.91261834E-02 au
@                      2.1531330     eV;   17366.190     cm-1;   207.74576     kJ / mol

@ Total energy :       -1539.9969     au

@ Operator label: X1SPNSCA; Transition moment :  3.01887547E-04 au
@ Spin-orbit coupling constant (X1SPNSCA) :   66.256658     cm-1;   3.01887547E-04 au

@ Operator label: Y1SPNSCA; Transition moment : -1.28498669E-03 au
@ Spin-orbit coupling constant (Y1SPNSCA) :  -282.02198     cm-1;  -1.28498669E-03 au

@ Operator label: Z1SPNSCA; Transition moment :  4.34306537E-04 au
@ Spin-orbit coupling constant (Z1SPNSCA) :   95.319266     cm-1;   4.34306537E-04 au
```

From the figure, the Hso is obtained as:

$$\sqrt{\left(66.2567^2 + 282.022^2 + 95.3193^2 \right) / 3} \text{ cm}^{-1} = \textbf{176.08} \text{ cm}^{-1}$$

Once the EDMA, EDME, and Hso are obtained, we can continue to do the Phosphorescence Spectrum Calculation.

TIPS: The Gaussian .com file to DALTON .mol file conversion can be done by using the MOMAP python script **gif2mol.py**, while the EDMA, EDME and Hso values can be easily obtained by using the MOMAP python script **dalton_get_info.py** from the DALTON output files.

---

[1] Qian Peng, Qinghua Shi, Yingli Niu, Yuanping Yi,* Shaorui Sun, Wenqiang Li and Zhigang Shuai*, Understanding the efficiency drooping of the deep blue organometallic phosphors: a computational study of radiative and non-radiative decay rates for triplets. J. Mater. Chem. C, 2016, 4,6829-6838.

Phosphorescence Spectrum Calculation

Nonradiative rate k_{isc}

Create a directory `kisc` and go to that directory

To start the calculation, you need a `*.dat` file, a MOMAP control file, and an optional parallel control file. The `*.dat` file is obtained from the previous mentioned `evc` calculation. A MOMAP control file is used to control how `TVCORF_SPEC` and `TVSPEC_SPEC` subprograms behavior. An optional parallel control file is used to control how many computing processes will be used.

Copy the following `evc` files from upper directory:

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

to this `kisc` work directory.

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

```
[kisc]$ cat momap.inp
do_isc_tvcf_ft     = 1                 # toggle inter-system crossing correlation function, 1 or 0
do_isc_tvcf_spec   = 1                 # toggle inter-system crossing spectrum, 1 or 0

&ic_tvcf
  DUSHIN      = .t.                     # toggle Duschinsky rotation effect, .t. or .f.
  Temp        = 298 K                   # temperature
  tmax        = 1500 fs                 # integral interval of correlation function
  dt          = 1   fs                  # integration timestep of correlation function
  Ead         = 0.0941829 au            # adiabatic excitation energy difference between two states
  Hso         = 176.08 cm-1             # spin-orbital coupling
  DSFile      = "evc.cart.dat"          # input dushin file
  Emax        = 0.3 au                  # upper bound of spectrum frequency
  logFile     = "isc.tvcf.log"          # output file for logging
  FtFile      = "isc.tvcf.ft.dat"       # output file for correlation function info
  FoFile      = "isc.tvcf.fo.dat"       # output file for spectrum function 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:

```
[kisc]$ ls
evc.cart.dat      isc.tvcf.ft.dat isc.tvcf.ft.gnu   log       nodefile
isc.tvcf.fo.dat isc.tvcf.log     momap.inp         README    ref        run
```

The intersystem crossing (ISC) rate constant can be found at the end of `isc.tvcf.log` file. The relationship between ISC rate constant and energy gap can be obtained from `isc.tvcf.fo.dat` file.

Filename	Meaning
`isc.tvcf.fo.dat`	Output file for spectrum function
`isc.tvcf.ft.dat`	Output file for correlation function
`isc.tvcf.log`	Output file for logging

Then use the following commands to generate the correlation function plot to check for convergence:

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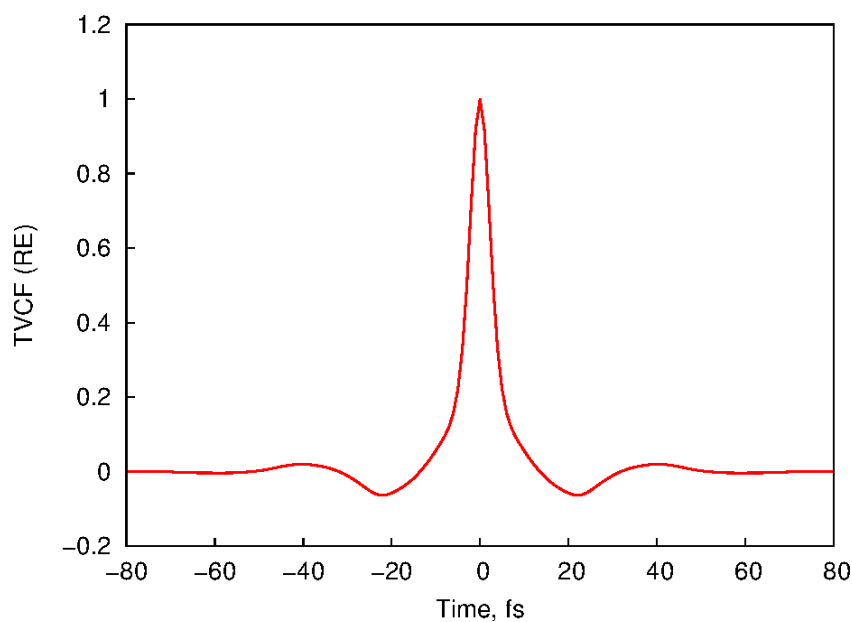

Fig. 3 Distribution of time *vs* real part of a converged correlation function

Once the correlation function is known to be converged, we can obtain the nonradiative rate at the end of the `isc.tvcf.log` file. From the file, we can obtain the intersystem crossing radiative rate k_{isc} for Irppy3 molecule is **1.69216792×10^8** s^{-1}. In addtion, the reverse intersystem crossing radiative rate is also given in the file.

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
```

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.
  Temp         = 298 K             # temperature
  tmax         = 1500 fs           # integration time
  dt           = 1   fs            # integration timestep
  Ead          = 0.0941829 au      # adiabatic excitation energy
  EDMA         = 0.655512 debye    # electronic dipole moment of absorption (GS)
  EDME         = 0.332683 debye    # electronic dipole moment of emission (ES)
  DSFile       = "evc.cart.dat"    # input dushin file
  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    README              spec.tvcf.ft.dat    spec.tvcf.spec.gnu
log             ref                 spec.tvcf.ft.gnu
momap.inp       run                 spec.tvcf.log
nodefile        spec.tvcf.fo.dat    spec.tvcf.spec.dat
```

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`.

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, and 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 [10000:30000]
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
$ display *.png
```
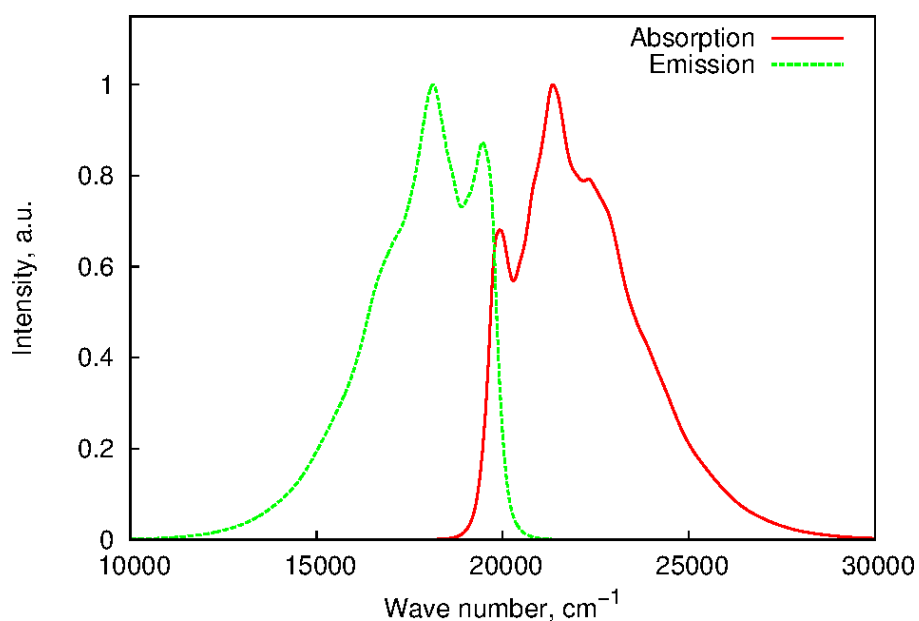

Fig. 5 Absorption and emission spectrum

17

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.
```

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 6 shows the distribution of a converged correlation function.

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

```
[sumstat]$ 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 [-80:80]
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:
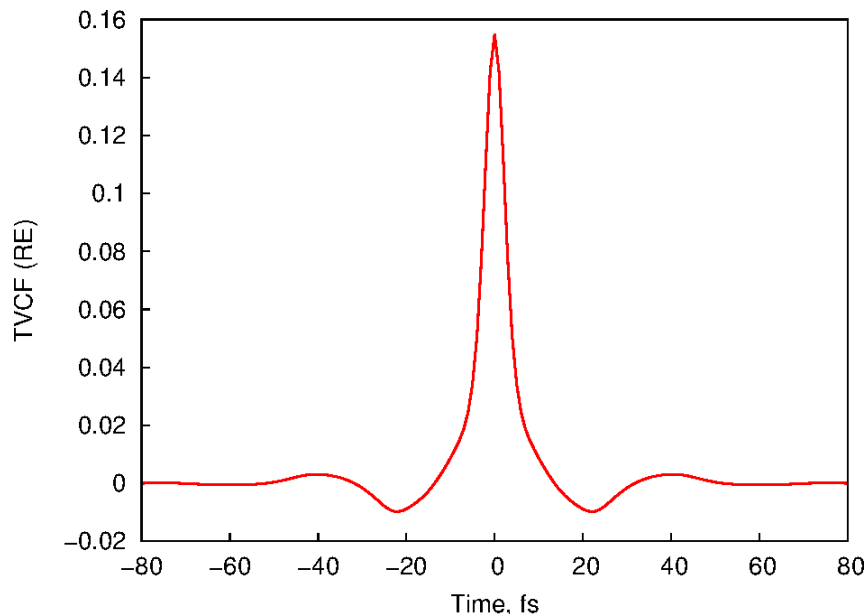
```
$ gnuplot *.gnu
$ ps2png *.eps
$ display *.png
```


Fig. 6 Distribution of time vs real part of a converged correlation function