Azulene is an organic compound and an isomer of naphthalene. Whereas naphthalene is colorless, azulene is dark blue. Two terpenoids, vetivazulene (4,8-dimethyl-2-isopropylazulene) and guaiazulene (1,4-dimethyl-7-isopropylazulene), that feature the azulene skeleton are found in nature as constituents of pigments in mushrooms, guaiac wood oil, and some marine invertebrates.

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.

Thus, the basic steps involved in the calculations are as follows:
1. Gaussian calculations
2. Vibration analysis etc.
3. Fluorescence spectrum calculation

Contents

Gaussian Calculations .......................................................................................................................................................... 1
   Optimization calculation on ground state ($S_0$) ........................................................................................................... 1
   Optimization calculation on lowest singlet excited state ($S_1$) ................................................................................. 2
   Calculate non-adiabatic coupling matrix element (NACME) ..................................................................................... 3
Vibration Analysis .............................................................................................................................................................. 5
Adiabatic Excitation Energy ........................................................................................................................................... 7
Electronic Transition Dipole ........................................................................................................................................... 8
Fluorescence Spectrum Calculation ............................................................................................................................. 10
   Nonradiative rate $k_c$ .................................................................................................................................................. 10
   Radiative rate $k_r$ ....................................................................................................................................................... 13
Sum-over-states Approach ............................................................................................................................................. 17
Verify Convergence of Correlation Function ............................................................................................................. 20
Gaussian Calculations

Optimization 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 b3lyp/6-31g$^*$, which indicates an optimization calculation at B3LYP/6-31G* level.

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

```plaintext
%chk=azulene-s0.chk
%mem=4GB
%nprocl=1
%nprocs=8
#p opt freq B3LYP/6-31G*

azulene-s0 optimization

0 1
C   2.01378743  -1.48849852  0.00000000
C   2.28995141   -0.11795315  0.00000000
C   1.39185815   0.95357383  0.00000000
C   0.78413689  -2.15418449  0.00000000
C   0.00000000   0.93285810  0.00000000
C  -0.50398383  -1.61056958  0.00000000
C  -0.89316505   -0.27406276  0.00000000
H   2.88919252  -2.13621797  0.00000000
H   3.34387207   0.15083266  0.00000000
H   1.84191311   1.94635990  0.00000000
H   0.83658347  -3.24058384  0.00000000
H  -1.32037398  -2.33298523  0.00000000
C  -0.84567310   2.05536637  0.00000000
H  -0.51364908   3.08694089  0.00000000
C  -2.17758707   1.61062710  0.00000000
H  -3.04994479   2.25593917  0.00000000
C  -2.21339978   0.20656494  0.00000000
H  -3.10314368  -0.41207657  0.00000000
```

We use g09 or g16 to do the geometry optimization.

![Fig. 1 Optimized $S_0$ geometry](image_url)
Optimization calculation on lowest singlet excited state ($S_1$)

With the optimized $S_0$ geometry at hand, 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 b3lyp/6-31g*`, which indicates an optimization calculation at B3LYP/6-31G* level using the TDDFT method.

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

```
%chk=azulene-s1.chk
%mem=4GB
%nprocl=1
%nprocs=8
#p opt freq td B3LYP/6-31G*

azulene-s1 optimization
0 1
C   2.01378700  -1.48849900  0.00000000
C   2.28995100  -0.11795300  0.00000000
C   1.39185800   0.95357400  0.00000000
C   0.78413700  -2.15418400  0.00000000
C   0.00000000   0.93285800  0.00000000
C   -0.50398400  -1.61066000  0.00000000
C   -0.89316500  -0.27406300  0.00000000
H   2.88919300  -2.13621800  0.00000000
H   3.34387200   0.15083300  0.00000000
H   1.84191300   1.94636000  0.00000000
H   0.83658300  -3.24058400  0.00000000
H  -1.32037400  -2.33298500  0.00000000
C  -0.84567300   2.05536600  0.00000000
H  -0.51364900   3.08694100  0.00000000
C  -2.17758700   1.61062700  0.00000000
H  -3.04994500   2.25593900  0.00000000
C  -2.21340000   0.20656500  0.00000000
H  -3.10314400  -0.41207700  0.00000000
```

Again, use g09 or g16 to do the $S_1$ optimization.
Calculate non-adiabatic coupling matrix element (NACME)

Unlike rad_FL calculation, The NACME should be obtained before performing a nonrad calculation.

After finding the optimized S1 geometry, we can calculate NACME at this geometry.

The route section is set as the following line:

```
#p td b3lyp/6-31g(d) prop=(fitcharge,field) iop(6/22=-4, 6/29=1, 6/30=0, 6/17=2)
```

The initial gaussian nacme input file (azulene-nacme.com) is as follows:

```
%chk=azulene-nacme.chk
%mem=4GB
%nprocl=1
%nprocs=8
%p td B3LYP/6-31G* prop=(fitcharge,field) iop(6/22=-4,6/29=1,6/30=0,6/17=2) nosymm

azulene excited state nacme calculation

0 1
C  2.00781300 -1.54567700 0.00000000
C  2.29335100 -0.16097300 0.00000000
C  1.43666200  0.92263700 0.00000000
C  0.74217000 -2.17592100 0.00000000
C  0.00000000  0.87899800 0.00000000
C -0.52457900  1.62505800 0.00000000
C -0.85006100 -0.22497100 0.00000000
H  2.86853900 -2.20838800 0.00000000
H  3.35493800  0.08599800 0.00000000
H  1.88767400  1.91181300 0.00000000
H  0.77544500 -3.26524800 0.00000000
H -1.36541000 -2.31420500 0.00000000
H -0.84651000  2.06953900 0.00000000
H -0.46545900  3.08512900 0.00000000
C -2.19214700  1.68790200 0.00000000
H -3.04949000  2.34818800 0.00000000
C -2.21739500  0.28941600 0.00000000
H -3.10206800 -0.33823600 0.00000000
```

Again, use g09 or g16 to do the NACME 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 azulene-s0.chk
$ formchk azulene-s1.chk
$ formchk azulene-nacme.chk

**TIPS:** To obtain the optimized excited state geometry for NACME calculation, use **Gaussview** to open azulen-s1.log file, and save as azulene-nacme.com. Then, modify the first few lines of azulene-nacme.com to suit for NACME 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 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) = "azulene-s0.log"  # log file of ground state
ffreq(2) = "azulene-s1.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/azulene-s0.fchk
../gaussian/azulene-s0.log
../gaussian/azulene-s1.fchk
../gaussian/azulene-s1.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/azulene-s0.fchk
../gaussian/azulene-s0.log
../gaussian/azulene-s1.fchk
../gaussian/azulene-s1.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
```
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.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.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 (viewed 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 azulene-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:

\[
\text{\$ cat azulene-s0.log | grep "SCF Done"}
\]

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

SCF Done: \( E(\text{RB3LYP}) = -385.838172128 \text{ A.U.} \)

Thus, we have the energy \( E_{sp} \) at optimized ground state geometry:

\( E_{sp} = -385.838172128 \text{ a.u.} \)

From the \( S_1 \) Gaussian log file, locate the last line with "Total Energy, \( E(\text{TD-HF/TD-KS}) \)" in the output azulene-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:

\[
\text{\$ cat azulene-s1.log | grep "Total Energy, \( E(\text{TD-HF/TD-KS}) \)"}
\]

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

Total Energy, \( E(\text{TD-HF/TD-KS}) = -385.763080213 \)

Then, we have the single point energy \( E_s \) at the optimized \( S_1 \) geometry:

\( E_s = -385.763080213 \text{ a.u.} \)

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

\[
E_{ad} = E_{es} - E_{gs} = [(-385.763080213) - (-385.838172128)] \text{ a.u}
\]

\[
= 0.075092 \text{ a.u}
\]

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

To calculate the spectrum by using the sum-over-states approach, we need the electronic transition dipole data.

The Gaussian log file for the optimized S1 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 azulene-s1.log file with vim, for example, search the string “transition electric dipole moments”, the first match is shown as follows:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Dip. S.</th>
<th>Osc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>state 1</td>
<td>0.2169</td>
<td>0.2932</td>
<td>-0.0000</td>
</tr>
<tr>
<td>state 2</td>
<td>0.1270</td>
<td>-0.0933</td>
<td>-0.0000</td>
</tr>
<tr>
<td>state 3</td>
<td>0.4091</td>
<td>0.5420</td>
<td>-0.0000</td>
</tr>
</tbody>
</table>

Focus on the “Dip. S.” column, this is the Dipole Square of the calculated Electronic Transition Dipole Absorption (EDMA), take note the data of the first excited state, i.e., 0.1330, this is the value of the expected EDMA. Thus, we have:

\[
\mu_{\text{trans}} = \sqrt{\mu^2_{\text{trans}}} = \sqrt{0.1330} \text{ a.u.} = 0.36469 \text{ a.u.} \\
= 0.36469 \text{ a.u.} \times 2.5417 \text{ Debye/a.u.} \\
= 0.92694 \text{ Debye}
\]

This is the value of parameter EDMA needed in our momap.inp file.

If the Linux command vim is used, press SHIFT + N, the search jumps to the last occurrence of “transition electric dipole moments”, shown as follows:
Again, focus on the "Dip. S." column, this is the Dipole Square of the calculated Electronic Transition Dipole Emission (EDME), take note the data of the first excited state, i.e., 0.0649, this is the value of the expected EDME. Thus, we have:

\[ \mu_{\text{trans}} = \sqrt{\mu_{\text{trans}}^2} = \sqrt{0.0649} \text{ a.u.} = 0.254755 \text{ a.u.} \]

\[ = 0.254755 \text{ a.u.} \times 2.5417 \text{ Debye/a.u.} \]

\[ = 0.64751 \text{ Debye} \]

Again, this is the value of parameter EDME needed in our momap.inp file.

**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.
Fluorescence Spectrum Calculation

Nonradiative rate $k_c$

Create a directory $kic$ and go to that directory, in this directory, we further create two directories evc and kic.

To start the calculation, you need a *.dat file, a MOMAP control file, and optionally a 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.

To begin with, the first step is to do an evc calculation. Note we also need the non-adiabatic coupling matrix element (NACME) calculation log file, that is, azulene-nacme.log, to do the nonradiative rate calculation.

Go to the directory evc, copy the following gaussian files from upper directory:

```
  ../../gaussian/azulene-s0.fchk
  ../../gaussian/azulene-s0.log
  ../../gaussian/azulene-s1.fchk
  ../../gaussian/azulene-s1.log
  ../../gaussian/azulene-nacme.log
```

to this work directory.

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

```plaintext
[kic/evc] cat momap.inp
do_evc = 1

&evc
  ffreq(1) = "azulene-s0.log"  # log file of ground state
  ffreq(2) = "azulene-s1.log"  # log file of excited state
  fnacme = "azulene-nacme.log"  # log file of NACME
/
```

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

```plaintext
#!/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
```

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

For example, we can use the Linux command cat and grep to do the job:

```bash
$ cat evc.cart.dat evc.dint.dat | grep "Total reorganization energy"
```

Total reorganization energy (cm⁻¹): 3390.305348 3453.436666
Total reorganization energy (cm⁻¹): 3412.711425 3449.528917

As can be seen, the energy difference is indeed rather small.

Once the evc calculation is done, we then go to the kic directory.

Copy the following evc files from upper directory:

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

to this kic work directory.

Create a momap.inp with its contents as follows:

```
[kic/kic]$ cat momap.inp
do_ic_tvcf_ft = 1  # toggle internal conversion correlation function, 1 or 0
do_ic_tvcf_spec = 1  # toggle internal conversion spectrum, 1 or 0
&ic_tvcf
  DUSHIN = .t.  # toggle Duschinsky rotation effect, t. or f.
  Temp = 300 K  # temperature
  tmax = 1000 fs  # integral interval of correlation function
  dt = 1 fs  # integration timestep of correlation function
  Ead = 0.075092 au  # adiabatic excitation energy difference between two states
  DSMFile = "evc.cart.dat"  # input dushin file
  CoulFile = "evc.cart.nac"  # input nacme info file
  Emmax = 0.3 au  # upper bound of spectrum frequency
  logFile = "ic.tvcf.log"  # output file for logging
  FtFile = "ic.tvcf.ft.dat"  # output file for correlation function info
  FoFile = "ic.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:

```
[kic/kic]$ ls
evc.cart.dat  ic.tvcf.fo.dat  ic.tvcf.log  momap.inp  README  run
evc.cart.nac  ic.tvcf.ft.dat  ic.tvcf.ft.gnu  log  nodefile  ref
```

The Internal conversion (IC) rate constant can be found at the end of ic.tvcf.log file. The relationship between IC rate constant and energy gap can be obtained from ic.tvcf.fo.dat file.

<table>
<thead>
<tr>
<th>Filename</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ic.tvcf.fo.dat</td>
<td>Output file for spectrum function</td>
</tr>
<tr>
<td>ic.tvcf.ft.dat</td>
<td>Output file for correlation function</td>
</tr>
<tr>
<td>ic.tvcf.log</td>
<td>Output file for logging</td>
</tr>
</tbody>
</table>

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](image.png)
Once the correlation function is known to be converged, we can obtain the nonradiative rate at the end of the ic.tvcf.log file. From the file, we can obtain the internal conversion radiative rate $k_{ic}$ for azulene molecule is $1.92466768 \times 10^{10} \text{s}^{-1}$, as shown below.

<table>
<thead>
<tr>
<th>degauss</th>
<th>FWHM w</th>
<th>FWHM t</th>
<th>lifetime</th>
<th>$6k_{ic}(s^{-1})$</th>
<th>$7\log(k_{ic})$</th>
<th>$8\text{time(ps)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-219474.63 cm$^{-1}$</td>
<td>-0.36E+05 s$^{-1}$ (Hz)</td>
<td>-0.10E+01 au</td>
<td>7.58975944E-02</td>
<td>2.64362959E+00</td>
<td>1.64780663E+04</td>
<td>6.96867326E+02</td>
</tr>
<tr>
<td>1.92466768E+10</td>
<td>1.02643558E+01</td>
<td>51.95702146</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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 calculation, 1 or 0
&spec_tvcf
  DUSHIN       = .t.        # toggle Duschinsky rotation effect, t. or f.
  Temp        = 300 K     # temperature
  tmax        = 1000 fs   # integration time
  dt          = 1 fs      # integration timestep
  Ead         = 0.075092 au # adiabatic excitation energy
  EDMN        = 0.92694 debye # electronic dipole moment of absorption (GS)
  EDME        = 0.64751 debye # electronic dipole moment of emission (ES)
  FreqScale   = 1.0       # frequency scaling factor
  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:

```
```

13
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:

```bash
$ ./run
```

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

```
[kic/kic]$ ls
 evc.cart.dat  README  spec.tvcf.dat  spec.tvcf.log
             log      spec.tvcf.ps
 momap.inp    run       spec.tvcf.spec
 nodefile     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, the plot script is shown as follows:

```
[sumstat]$ 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 [5000:30000]
set yrange [0:1.15]
set output "spec.tvcf.spec.ps"
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:

```bash
$ gnuplot *.gnu
$ ps2png *.eps
$ display *.png
```
Fig. 4 Distribution of time vs real part of a converged correlation function

Fig. 5 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;
    $filo = "$fili.png";
  } else {  # otoh, omit ps suffix
    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.
Sum-over-states Approach

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

..../gaussian/azulene-s0.fchk
..../gaussian/azulene-s0.log
..../gaussian/azulene-s1.fchk
..../gaussian/azulene-s1.log
../evc/evc.cart.dat
to a work directory, say, sumstat.

Create a momap.inp with its contents as follows:

```
[sumstat]$ cat momap.inp

dospec_sums = 1 # if use sum-over-states approach, 1 or 0

&spec_sums
    DSFile = "evc.cart.dat" # input evc file
    Ead = 0.075092 au # adiabatic excitation energy
    dipole_abs = 0.92694 debye # Electronic Transition Dipole Absorption
    dipole_emi = 0.64751 debye # Electronic Transition Dipole Emission
    maxvib = 10 # maximum vibration quantum number
    if_cal_ic = .t. # if do internal conversion analysis, t or f.
    promode = 24 # promotion mode (internal conversion)
    FC_eps_abs = 0.1 # eps of Franck-Condon factor (absorption)
    FC_eps_emi = 0.1 # eps of Franck-Condon factor (emission)
    FC_eps_ic = 0.1 # eps of Franck-Condon factor (internal conversion)
    FreqScale = 1.0 # frequency scaling factor
    FreqEPS = 0.01 # eps of frequency
    Seps = 0.00001 # eps of Huang-Rhys coupling constant
    eps = 0.00 #
    debug = .false. #
    FWHM = 500 cm-1 # broadening factor, full width at half maximum
    blocksize = 1000 #
    testpoints = 1000 #
    TEST = .f. #
    flog = "spec.sums.log" # output log file
    reduce_eps = 0.001 #
```

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 &
```

Then, perform the calculation by running the script file:

```
$ ./run
```
Finally, the result files are shown as follows:

<table>
<thead>
<tr>
<th>Filename</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>spec.sums.abs.dirac.dat</td>
<td>Absorption spectrum and vibrational transition quantum numbers</td>
</tr>
<tr>
<td>spec.sums.emi.dirac.dat</td>
<td>Emission spectrum and vibrational transition quantum numbers</td>
</tr>
<tr>
<td>spec.sums.ic.dirac.dat</td>
<td>Internal conversion and vibrational transition quantum numbers with the set promode as promotion mode</td>
</tr>
<tr>
<td>spec.sums.log</td>
<td>Logging file for sums-over-states aproach</td>
</tr>
<tr>
<td>spec.sums.spec.dat</td>
<td>Absorption and emission spectrum</td>
</tr>
</tbody>
</table>

Plot the data from file `spec.sums.spec.dat` by using columns 4, 7, and 13, in Linux, we can use Gnuplot to do the plotting, the plot script is shown as follows:

```
[sumstat]$ cat spec.sums.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 yrange [0:1.15]
set output "spec.sums.spec.eps"
plot "spec.sums.spec.dat" u 4:7 t "Absorption" w l lw 3 lt 1, \
"" u 4:13 t "Emission" w l lw 3 lt 2
```

Then use the following commands to generate the graph:

```bash
$ gnuplot *.gnu
$ ps2png *.eps
$ display *.png
```
Fig. 6 Absorption and emission spectrum by using sum-over-states approach
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:

```plaintext
[sunstat]$ 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:

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

![Fig. 7 Distribution of time vs real part of a converged correlation function](image.png)