Hands-On Session

Calculation of Excited States with CASSCF in OpenMolcas

1 What this exercise is about

- Calculating properties of excited states with CASSCF and MS-CASPT2 methods
- Characterizing excited states by excitation energy, oscillator strength, and wave function character
- Using the OpenMolcas program package

2 Introduction

Excited states play a significant role in photochemistry. Calculation of excited states and their properties requires different methods than for ground state chemistry. For the excited states, the approximation of only one Slater determinant is virtually always invalid. Therefore, the excited state is usually described as a linear combination of different excited configurations.

There are in principle different approaches for excited states:

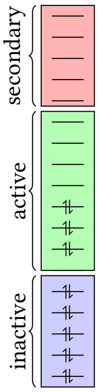
- **ΔEnergy approach:** Solve the electronic Schrödinger equation at least twice, obtaining two different wave functions and two energies. The excitation energy is then simply the energy difference.
  - + With the wave functions, all kinds of properties can be computed,
  - + Treats ground state and excited state equivalently,
  - – Only works for CI-type methods (not for DFT, MPn, CC),
  - – The different wave functions need to be made orthogonal.
- **Response approaches:** Simulate how the ground state density responds to a time-dependent electric field of different frequencies ω. The frequency where the response function $R(\omega)$ shows a pole (i.e., a singularity) corresponds to an excitation energy $\Delta E = \hbar\omega$.
  - + Works for many ground state methods (e.g., DFT),
  - – Does not directly produce excited-state wave functions,
  - – Does not treat ground state and excited state equivalently (can lead to problems when ΔE becomes very small).
- **Propagator methods:** Derive some operator that acts on a ground state wave function and produces an excited-state wave function. Derive a matrix representation and obtain eigenvalues that are the excitation energies.
  - + Works for many ground state methods (e.g., CC, MPn),
  - – Does not treat ground state and excited state equivalently (can lead to problems when ΔE becomes very small).

In this exercise, we will be using the *state-averaged complete active space SCF (SA-CASSCF)* and the *multi-state complete active space second-order perturbation theory (MS-CASPT2)* methods (both are ΔEnergy approaches) to compute excitation energies of acrolein.

2.1 SA-CASSCF and MS-CASPT2

SA-CASSCF is an example of the ΔEnergy approach. Each electronic state is described as a linear combination of all determinants that can be created by excitation within the active space. During the calculation, the linear combination expansion coefficients in front of all determinants are variationally optimized. Additionally, the orbitals (that are used to construct the determinants) are also variationally optimized. The special feature of the *state-averaged* CASSCF method is that the same orbitals are used for all electronic states—this automatically makes the wave functions orthogonal. However, in this way the orbitals cannot be variationally optimized for each state, and instead are computed to minimize the average energy of all states (this is the origin of the term *state-averaging*). Because of this, changing the number of states in the calculation changes the energies of all other states.

To construct the determinants that make up the wave functions, a complete active space is usually used. These determinants correspond to all possible electron distributions in the active space orbitals (i.e., full CI in the active space), whereas the inactive orbitals are always doubly occupied and secondary orbitals are always empty. The selection of the active space is usually carried out before the calculation according to chemical intuition (chemical thinking is required to decide whether a particular orbital might be involved in the process one wants to describe). Typically, a reasonable choice is to take the active orbitals to be some of the highest occupied and some of the lowest unoccupied molecular orbitals from a Hartree-Fock calculation. Orbitals involved in the excited state description should be included in any case. A common notation is CASSCF(n,m), indicating n electrons being distributed in m active orbitals.



CAS

Since CASSCF lacks the description of electron correlation outside the active space and also optimizes the average energy over several states, its excitation energies are rarely very accurate. This is especially notable in molecules with different kinds of states (e.g., $n\pi^*$ vs. $\pi\pi^*$), where one kind of states could be accurate and the other one not. A useful approach to improve the CASSCF energies is the CASPT2 method, which adds a second-order perturbation theory estimate of dynamical correlation energy to each of the CASSCF energies. Note that CASPT2 is not directly MPn applied to CASSCF, because for multi-configurational wave functions, defining the zero-order Hamiltonian and perturbation operators is much more complex than for Hartree–Fock. This also explains why there are different flavors of multi-reference perturbation theory, e.g., CASPT2, NEVPT2, MRPT2, QDPT2, …

An important additional feature in CASPT2 calculations is the so-called multi-state treatment (MS-CASPT2). This is important if at CASSCF level two energies are very similar, but the PT2 treatment gives very different correction energies so that the two states are shifted away from each other. For example, without an MS or (XMS) treatment, CASPT2 potential energy surfaces may not show the right intersection topology.

3 CASSCF Calculation with OpenMolcas

For the CASSCF and MS-CASPT2 calculation, we will use the OpenMolcas program package. Other programs like Orca or Gaussian can also run CASSCF calculations, but the OpenMolcas code is very robust and, in many cases, very fast, because it is primarily dedicated to such calculations.

3.1 OpenMolcas setup

In order to setup your shell for a simple OpenMolcas calculation, you can source the setup script provided:

```
(base) user@node: ~>
source /projects/academic/cyberwksp21/Students/smai/Instructors_material/set_openmolcas.sh
```

This sets all required environment variables in your current shell session, so in principle you could run OpenMolcas interactively after sourcing.

However, we will not run interactive calculations, because that would overburden the login servers of the cluster. Instead, we will submit computations to the queueing system SLURM, for which we will prepare a corresponding run script (see below). In this run script, we will source the above file to enable the calculations.

3.2 Task 1: Initial SCF calculation

The first step of our calculation is to do a simple HF computation. This provides us with initial orbitals that we can inspect to choose a sensible active space.

The initial geometry is:

```
8
Acrolein
C +0.002992 +0.715503 0.
C +1.212185 +1.295708 0.
C -0.145417 -0.744285 0.
O -1.220818 -1.310673 0.
H +2.116473 +0.699958 0.
H +1.329536 +2.369530 0.
H -0.910572 +1.293685 0.
H +0.790044 -1.323086 0.
```

Paste the geometry into a file called `acrolein_opt.xyz`.

Next, the input for this SCF calculation is:

```
&GATEWAY
Coord=acrolein_opt.xyz
Group=nosym
Basis=ANO-R1
&SEWARD
&SCF
```

Save the above input file as `MOLCAS_SCF.in`.

Before we actually run the job, some short notes on OpenMolcas input files. These are procedural and command-driven, i.e., you need to write “a little program” where you call the individual sub-steps of your calculation in the correct order. This input style even allows to use conditionals, loops, and file operations (like copy, link, remove), allowing you to do complicated workflows with multiple calculations in one file. However, in a simple input file, you just sequentially start the relevant subprograms (modules), indicated with `&`. The input for these modules is provided separately. As the input shows, we will run the `GATEWAY` program, the `SEWARD` program and then, finally `SCF` program. The `GATEWAY` program prepares initial input for all the subsequent programs. The `SEWARD` program calculates the one- and two-electron integrals. `SCF` runs a Hartree-Fock calculation, which, as mentioned before, we need for the initial orbital set.

Note that as basis set we use one of OpenMolcas’ new ANO-R basis set family[1] has been used. These are relativistic all-electron basis sets that are available for all elements from H to Rn in four quality levels from ANO-R0 (corresponds to single-zeta for most elements) to ANO-R3 (quadruple-zeta). They are optimized for accuracy and compactness in mind and can be recommended for most work done in OpenMolcas.

[1] Zobel, Widmark, Veryazov: *JCTC* **16**, 278–294 (2020).

In order to execute the calculation, prepare a simple submit script:

```
#!/bin/bash

source /projects/academic/cyberwksp21/Students/smai/Instructors_material/set_openmolcas.sh
cd $SLURM_SUBMIT_DIR
export WorkDir=$(pwd)/scratch/
pymolcas ${1} &> ${1}.log
```

You can call it `run.sh`. The script can be sent to the cluster queueing system by:

```
(base) user@node: ~> sbatch -A cyberwksp21 -p valhalla -q valhalla -c 1 run.sh MOLCAS_SCF.in
```

Here, we specify that we use the account `cyberwksp21`, the partition and quality of service `valhalla`, and one CPU core. We also specify that we are submitting the script `run.sh` and we provide one argument `MOLCAS_SCF.in` to this script. You can reuse the script to run other input files, if you simply change the last argument.

The calculation should only take a few seconds. You can check your submitted job with `squeue -u $USER`. After the calculation has finished, there should be an output file called `MOLCAS_SCF.in.log` in the directory. If the calculation is successful, "`Happy landing!`" should appear at the end of this file. The total SCF energy should be about -190.91846 Hartree. Once the initial OpenMolcas calculation has been finished, OpenMolcas will write a bunch of output files. A `Molden` file `MOLCAS.scf.molden` will appear in the same directory as the input file that allows the visualization of the orbitals.

3.3 Task 2: Choice of the active space

Before proceeding to the CASSCF calculation, inspect the SCF orbitals visually to choose the desired active space and select the correct starting orbitals for the CASSCF calculation. Selection of the active space is a highly important step: these orbitals are used as a first guess in the CASSCF orbital optimisation step, and with a poor guess, a CASSCF calculation may take very long or end up with a wrong active space and yield physically meaningless results, despite apparently having converged.

The first step here is to visualize the SCF orbitals. For this, you need to install a suitable visualization software on your own computer, e.g., `gabedit`. Binaries for Windows and Linux can be downloaded from `http://gabedit.sourceforge.net/`. The rest of this document assumes that you use `gabedit` on your own computer.

Thus, download `MOLCAS.scf.molden` from the CCR cluster to your computer and visualize the orbitals. If you use `gabedit`, it is advisable to create an orbital slideshow (Display Geometry/Orbitals ⇒ Orbitals ⇒ Slideshow), since it will create a png file for every orbital in the slideshow, which you may later look at again without replotting the orbital. Visualize only the orbitals around the HOMO–LUMO gap, since it is unlikely that very low-lying occupied or very high-lying virtual orbitals are important for us.

We are going to perform a CASSCF(6,5) calculation, i.e., our active space will contain 5 orbitals with 6 electrons distributed in them. Since we want to calculate the $\pi\pi^*$ and the $n\pi^*$ excited states, we want to have n and π^* orbitals in our active space. In addition to the π and π^* orbitals, select also an n orbital. During the selection, note the numbers of the orbitals you have selected. Now it is time to enter the selection to the OpenMolcas input.

Questions:

1. It is also important to include π orbitals if we include π^* orbitals. Why?
2. Why are we interested in $\pi\pi^*$ and $n\pi^*$ excited states but not in $\sigma\pi^*$, $\pi\sigma^*$, $n\sigma^*$, or $\sigma\sigma^*$ states?
3. The π system of acrolein contains four atoms, each of which contributes a p orbital. How many π orbitals are we going to select? How many electrons do they contribute?

3.4 Task 3: Input the orbital selection

There are different possible ways to tell OpenMolcas which orbitals to use for the active space. Here, we will directly mark the active orbitals in the file that contains the SCF orbital coefficients.

Open the file `MOLCAS.ScfOrb`. This file contains the orbital coefficients in a format which OpenMolcas uses to read and write. At the end, the orbital occupancy is listed in the following format:

```
#INDEX
* 1234567890
0 iiiiiiiiii
1 iiiiisssss
2 ssssssssss
3 ssssssssss
```

This table indicates the orbital partitioning. Orbitals are counted row-wise (i.e., the first row lists orbitals 1–10, the second one 11–20 and so on; the numbers above and on the left act as a help to identify the index), and `i` stands for *inactive* and `s` for *secondary*. Since this is the result of an SCF calculation, we only have inactive (doubly occupied) and secondary (virtual) orbitals. To select the active space, replace the labels of the orbitals you have selected by a 2.[2] Note again that this is a critical step. If you do not identify the correct orbitals, then your calculation will most likely not produce the right results.

Save the `MOLCAS.ScfOrb` file and copy it to the `scratch` directory:

```
(base) user@node: ~> cp MOLCAS.ScfOrb scratch/
```

Then prepare the following OpenMolcas input:

```
>>>COPY FORCE MOLCAS.ScfOrb INPORB
&RASSCF
NActEl=6,0,0
Spin=1
LumOrb
TypeIndex
CIRoot=6,6,1
```

On the first line, we copy the just-prepared file to `INPORB`, so that OpenMolcas can find the starting orbitals. Then, the CASSCF calculation is invoked by the `RASSCF` command. The RASSCF program may be used to perform calculations with the RASSCF method, which is an extension of the CASSCF (i.e., CASSCF is a special case of RASSCF). The *NActEl* parameter specifies 6 active electrons, *Spin=1* instructs OpenMolcas to calculate singlet states. The keywords `LumOrb` and `TypeIndex` instruct OpenMolcas to read the starting orbitals including the orbital partitioning table from a text file (in our case, the `ScfOrb` file), and the *CIRoot* line enables the state-averaged CASSCF calculation over 6 roots (i.e., 6 states). Note that here we do not specify the number of active orbitals, because this is implicitly done in the orbital occupancy indices. In most other CASSCF calculations, one would explicitly specify the number of orbitals.

After preparing the input, save it as a new file as e. g. `MOLCAS_CAS.in` and run the following command to perform the CASSCF calculation **in the same directory** where you performed the first OpenMolcas calculation:

```
(base) user@node: ~> sbatch -A cyberwksp21 -p valhalla -q valhalla -c 1 run.sh MOLCAS_CAS.in
```

[2] Technical detail: The reason behind the number 2 has to do with further partitioning of the active space in three subspaces in the RASSCF method, which is closely related to CASSCF. The second subspace in RASSCF corresponds to the active space in CASSCF.

Due to the `LumOrb` keyword, OpenMolcas will try to find an `INPORB` file which is to be used as an initial guess for the calculated orbitals.[3] The calculation should finished within less than a minute. If you have picked the correct starting orbitals, the calculation should converge to the expected result, with the energy of the lowest state being −190.96607 Hartree.

Questions:

1. Why is the energy of the lowest CASSCF state lower than the SCF energy?
2. What is the meaning of the other states in the CASSCF results?
3. What would you expect how the energy of the lowest CASSCF state when you increase the number of state-averaging roots?

3.5 Task 4: OpenMolcas Output

Once again, OpenMolcas will produce a `Molden` file, now called `MOLCAS.rasscf.molden`, which may be opened for orbital inspection with `gabedit` again. **This is an absolutely important check for any CASSCF calculation!** Very often the active space is not right, and orbitals have to be swapped and the calculation has to be repeated until the optimization has found the right active space. This is because the optimization wants to minimize the energy (at all cost), even if the lowest energy is obtained with active space orbitals that will not describe your chemical problem.

If the orbitals do not have the correct shape (i.e., are not n, π or π^* orbitals as we are expecting), inspect inactive and secondary orbitals to find the correct orbitals, edit the orbital assignment indices in `MOLCAS.RasOrb` (not `MOLCAS.ScfOrb`), modify the input file to restart from `MOLCAS.RasOrb`, and repeat the CASSCF calculation.

After ensuring the correctness of the active space, you may characterize the excited states wave functions. You do this by inspecting two kinds of information: (i) the orbital shapes by looking at the visualization, and (ii) the coefficients and configurations that contribute to a state. The latter information is written in the log file. An example output is shown here for the ground state:

```
printout of CI-coefficients larger than  0.05 for root  1
energy=    -190.966072
conf/sym  11111     Coeff  Weight
        1  22200  -0.89295 0.79735
        2  22ud0   0.28929 0.08369
        3  22020   0.11991 0.01438
```

Root 1 denotes the ground state, and other roots refer to excited singlet states. Here, root 1 has a 80% contribution from the configuration `22200`, where the first 3 active orbitals are doubly occupied and the two last ones are not occupied. For an excited state, the leading configuration might be `2u2d0`, indicating an excitation from the second to the fourth active orbital. In order to find out the state character, inspect the second and fourth active orbitals in the visualization; if you find that they are a π and a π^* orbital, then you know that the corresponding root is a $\pi\pi^*$ state.

[3] The approach of doing all calculations separately but in the same directory, which also serves as the working directory, is rather unusual. More commonly one provides only the needed files to the calculations via linking in the input file or combines all modules in one input file.

Questions:

1. What are the state characters of the states that you have computed?
2. What are their energies?
3. How do the energies compare to the experimental values[1] for acrolein (3.77 eV and 6.42 eV)? Which states should you compare against these values?

3.6 Task 5: MS-CASPT2 calculation

The CASSCF energies can be improved by running a MS-CASPT2 calculation. The input is:

```
&CASPT2
Multistate=6 1 2 3 4 5 6
```

The input is almost self-explanatory—it just runs the CASPT2 program and calculates 6 states with the multi-state method. Note that you can use a smaller number of states than in the SA-CASSCF run (e.g., to save computer time). Save the input as, e.g., MOLCAS_PT2.in and run OpenMolcas as above.

In the output, it is recommended to first check the Reference weight of each excited state, which should be all rather similar. If any of the reference weights is significantly lower than the others, then the CASPT2 calculation should be repeated with the addition of the line imaginary = 0.1. This introduces a so-called imaginary level shift in the calculation, which avoids the so-called intruder state problem, which occurs when the assumptions of the perturbational treatment are not met and leads to nonsensical results. Increasing this shift will reduce the intruder state problems, but at the cost of correlation energy; in the limit of an infinite shift, CASPT2 reproduces the CASSCF energies.

Note that in CASPT2 in OpenMolcas there is another kind of shift parameter, the so-called IPEA shift (ionization potential/electron affinity). This shift is intended to reduce systematic errors in CASPT2 for open-shell states and has a default value of 0.25 a.u. As shown in the literature, for small basis sets (double-zeta quality or worse), it is recommended to turn the IPEA shift off, by adding IPEA = 0 to the input file.

Once you have run the MS-CASPT2 calculation with imaginary and without IPEA shift, you can analyze the output. You can find the single-state CASPT2 energies (i.e., before the multi-state treatment) by searching for :: CASPT2 Root and the MS-CASPT2 energies with :: MS-CASPT2 Root. The MS-CASPT2 energy of the first state should be −191.51189 Hartree. Calculate again the excitation energies for the states. Do you see an improvement over the CASSCF results?

In general, the states calculated with MS-CASPT2 are mixtures of the original CASSCF states. Hence, to interpret the MS-CASPT2 results, you need to check the configurations of the states once again in the CASPT2 output file. These can be found after Mixed CI coefficients. You can also inspect the eigenvector matrix of the MS-CASPT2 treatment, indicated by Eigenvectors:, to identify how the CASSCF states have been mixed or whether they have changed order.

Questions:

1. How much correlation energies is CASPT2 recovering?
2. Is there a strong differential correlation effect (i.e., do different states recover different amounts of correlation energy)?
3. How much does the multi-state treatment change the results?

[1] Walsh, *Trans. Faraday Soc.* **41**, 498–505 (1945).

3.7 Task 6: Transition probabilities

The last step in a vertical excitation calculation is typically to compute the transition intensities from the ground state to the different excited states. This is particularly important to interpret absorption spectra.

For this task, we use the RASSI (restricted active space state interaction) program:

```
>> COPY FORCE MOLCAS.JobIph JOB001
&RASSI
NROFJOBIPHS
1 6
1 2 3 4 5 6

>> COPY FORCE MOLCAS.JobMix JOB001
&RASSI
NROFJOBIPHS
1 6
1 2 3 4 5 6
EJOB
```

In this way, you can directly run the analysis for both the CASSCF (first input block) and MS-CASPT2 (second block) states after each other. Save this as `MOLCAS_SI.in` and run OpenMolcas again.

Here one should note that the second calculation does not actually compute the transition probabilities for the full CASPT2 states, because those are never computed. Instead, the MS-CASPT2 procedure takes the original CASSCF wave functions and remixes them according to the CASPT2 results; hence, one uses wave functions of CASSCF quality, but adapted to match what CASPT2 would predict.

The most important sections of the RASSI output are marked by `SPIN-FREE ENERGIES` and `Dipole transition strengths (spin-free states)`. The first section contains the energies again, so that you can make sure that RASSI took the desired states. The output of the first half of the log file should match your CASSCF energies, the second half your MS-CASPT2 energies. RASSI also nicely prints the excitation energies in eV, so that you can directly read them. The second output section gives you the oscillator strengths, which tell you how bright a transition is. For acrolein, there should not be a transition from 1 to 2 in this table, as the $S_0 \longrightarrow S_1$ transition is forbidden. In general, a transition with oscillator strength of about 0.1 is a bright state that will show up in the absorption spectrum, whereas states of oscillator strength below 0.001 are probably not observable, except if they do not overlap with any other state. Note that you can ignore any transitions that do not start from state 1 if you are only interested in the steady-state absorption spectrum.

Questions:

1. Which is the brightest state of the ones you computed? What is its energy?
2. Which color should acrolein have according to your results?
3. Can you make any connection between the magnitude of oscillator strengths and state character ($n\pi^*$, $\pi\pi^*$)?