Hands-On Session

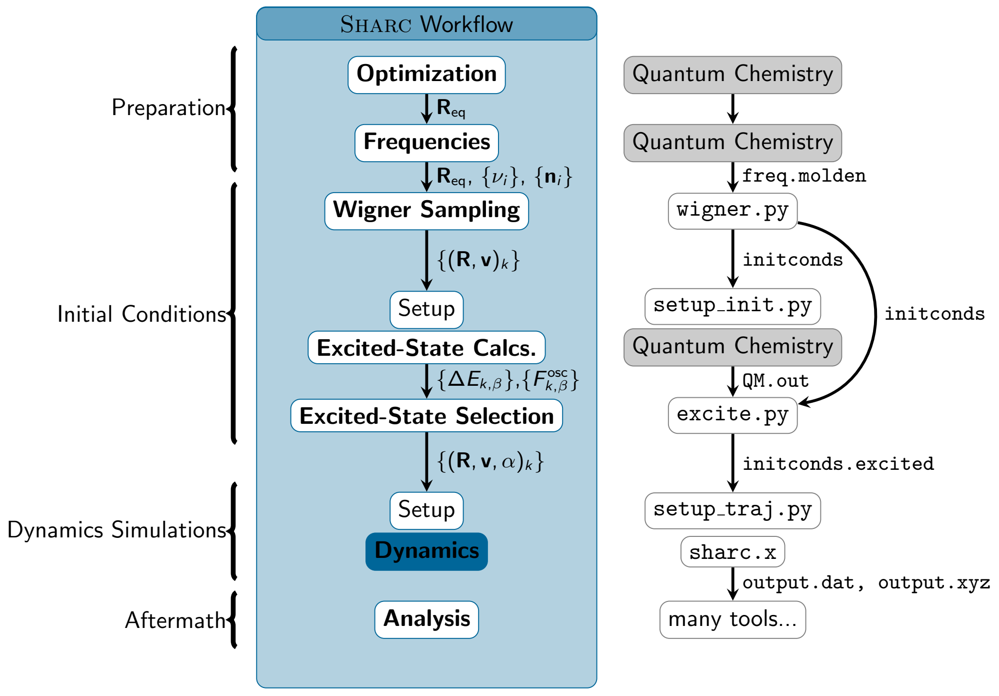
Preparation of Surface Hopping Trajectories with SHARC and OpenMolcas

1 What this exercise is about

- Using SHARC tools to perform several setup tasks to prepare surface hopping simulations.
- Performing an optimization and frequency calculation.
- Generating initial structures from a Wigner distribution.
- Performing vertical excitation calculations for all initial structures.
- Generating an absorption spectrum.
- Selecting initial excited states.
- Generating the input files for the surface hopping simulations.

2 Introduction

Preparing SHARC simulations (or any other nonadiabatic dynamics/surface hopping simulations) requires a couple of steps that are very important, because they decide what kind of physical processes (or what kind of experiment) one is simulating.



Here, so far SHARC was mostly used to simulate the photo-induced dynamics of cold, isolated, relatively rigid molecules in the gas phase. Such simulations can be conveniently set up by first generating initial geometries and velocities from the Wigner distribution of the ground state vibrational state, which can be described by a multi-dimensional harmonic oscillator. Hence, the first workflow steps (see scheme) are to perform an optimization and frequency calculation for the target molecule. The results of this calculation fully determine the harmonic oscillator, and one can directly perform the Wigner sampling (using `wigner.py`), which produces an `initconds` file with sets of geometries and velocities.

The second part of the workflow is to decide which is the initial excited state after photo-excitation. For this, one first needs to know the vertical excitation energies for all initial geometries, which can be found by single point calculations. These can be automatically set up with `setup_init.py` and read out with `excite.py`. The latter program then adds all excited-state information to the `initconds` file, producing an `initconds.excited` file. From this file, one can generate an absorption spectrum, decide which states or excitation range one is interested in, and then can use `excite.py` to select which states of which initial geometries are going to be used to launch trajectories.

In the third workflow step, one then sets up the trajectories. This involves making several methodological choices, like length of simulation, time step, algorithms for different steps (wave function propagation, hopping probabilities, decoherence correction, kinetic energy adjustment, etc.), output options, quantum chemistry settings, or parallelization. These steps are done with `setup_traj.py` and produce a directory tree with all relevant input files for the swarm of simulations.

Today, we will only work until this step, because the next step (running the trajectories) would take too long to finish within the scope of this workshop. If you want to continue, you can check the official SHARC tutorial, which this task sheet is based on.

3 Description of the model system

The task of the tutorial is to simulate the excited-state dynamics of the methaniminium cation ($CH_2NH_2{}^+$) after excitation to the bright excited state. The employed quantum chemistry method will be CASSCF(6,4)/cc-pVDZ, using the OpenMolcas program. This level of theory is not sufficient for a serious scientific investigation, but we only want to showcase the preparation steps anyways.

In the following, an overview over the level of theory and the initial geometry are given:

Ab initio level of theory for $CH_2NH_2^+$.

Charge	+1
Program	Molcas
Method	SA-CASSCF(6,4)
Basis set	cc-pVDZ
Number of states	4 Singlets and 3 Triplets

```
6
Starting  geometry  for  CH2NH2+
C +0.000000 +0.0 +0.000000
N +0.000000 +0.0 +1.350000
H +0.943102 +0.0 -0.544500
H +0.943102 +0.0 +1.879500
H -0.943102 +0.0 +1.879500
H -0.943102 +0.0 -0.544500
```

4 How to use the SHARC suite

Beyond the core dynamics program, the SHARC suite contains a number of Python scripts allowing to perform various types of setup and analysis tasks. There are two types of these scripts.

Non-interactive scripts can be controlled by command-line arguments and options. Every non-interactive script can be called with the command line option `-h` in order to get a description of the functionality and possible options.

Interactive scripts ask the user for information about the task to be conducted (using features like auto-complete and default values), and only perform the task after the input has been completed.

In the tutorial, the input dialogue of the interactive scripts is shown as in this example. **`Red bold text`** gives the input which the user has to type.

```
Type of calculation: 2
Frequency calculation? [True] <ENTER>

Geometry filename: [geom.xyz] (autocomplete enabled) g<TAB>
Geometry filename: [geom.xyz] (autocomplete enabled) geom.xyz

Enter atom indices: (range comprehension enabled) 1~4
```

During the interactive sessions, square brackets indicate that the question has a default answer, which can be used by just pressing ENTER. If filenames or directory paths need to be entered, auto-complete is active, which can be used by pressing TAB. If a list of integers (e.g., atom indices) needs to be entered, range comprehension is active, and ranges can be entered with the tilde symbol (e.g., `1~4` is equivalent to `1 2 3 4`). Upon completion, every interactive script produces a file `KEYSTROKES.<name>`, which contains all user input for the last run.

Please make sure before starting that `$SHARC` is set to the directory containing the SHARC scripts and executables.

It is also advisable to set `$MOLCAS` to the OPENMOLCAS main directory via the `set_openmolcas.sh` script from the previous hands-on session.

Thus, you should do this before starting with the exercise:

```
(base) user@node: ~> export SHARC=/projects/academic/cyberwksp21/Software/sharc2.1.1/sharc-master/bin/
(base) user@node: ~> export SCRADIR=/panfs/panfs.cbls.ccr.buffalo.edu/scratch/grp-cyberwksp21/$USER
```

Note that in principle one should be able to reproduce all results in this tutorial, i.e., if you closely follow the given steps then you should see exactly the same output (and the same figures). The only exception to this is that the results might be different if you employed a different compiler to compile `sharc.x` (this tutorial uses `gfortran 4.8.5`) or if you use a different OPENMOLCAS version (this tutorial uses OPENMOLCAS v21.10).

Note that it is recommended that for sections in which the user is especially interested, they should refer to the corresponding sections in the SHARC Manual.

5 Workflow

5.1 Optimization and Frequency calculation

The first general step of a dynamics simulation is the setup of the initial conditions. Here, we will sample initial conditions randomly from the ground state Wigner distribution.

In order to carry out this step, we need to prepare a Molden file containing the results of a frequency calculation for the ground state. In general, the user is free to calculate the frequencies and normal modes with any quantum chemistry software and any method he sees fit, as long as a Molden file can be produced. However, usually it is advisable to calculate the frequencies at the same level of theory as the dynamics calculation. For the methaniminium cation in our example, we will do the frequency calculation with the quantum chemistry method specified above: (SA(3)-CASSCF(6,4)/cc-pVDZ).

Create an empty directory. Prepare a geometry file called `geom.xyz` containing the geometry given above.

```
(base) user@node: ~> mkdir Opt_Freq
(base) user@node: ~> cd Opt_Freq
(base) user@node: ~> vi geom.xyz
```

The Sharc suite comes with an input generator for Molcas, which produces input for single-point calculations, optimizations and frequency calculations. It can be invoked with

```
(base) user@node: ~> python2 $SHARC/molcas_input.py
```

The script is interactive. Start the script and prepare an optimization plus frequency calculation on SA-CASSCF level. The script can also generate a Bash-script to instantly launch the Molcas calculation.

```
  ================================================================================
||                                                                                ||
||                           MOLCAS Input file generator                          ||
||                                                                                ||
||                              Author: Sebastian Mai                             ||
||                                                                                ||
||                                   Version:2.1                                  ||
||                                    01.09.19                                    ||
||                                                                                ||
  ================================================================================


This script allows to quickly create MOLCAS input files for single-points calculations
on the SA-CASSCF and (MS-)CASPT2 levels of theory.
It also generates MOLCAS.template files to be used with the SHARC-MOLCAS Interface.

---------------------Type of calculation--------------------

This script generates input for the following types of calculations:
  1       Single point calculations (RASSCF, CASPT2)
  2       Optimizations & Frequency calculations (RASSCF, CASPT2)
  3       MOLCAS.template file for SHARC dynamics (SA-CASSCF)
Please enter the number corresponding to the type of calculation.

Type of calculation: 2                     # Anything after # is a comment
Frequency calculation? [True] <ENTER>      # Opt + Freq

--------------------------Geometry--------------------------
```

```
Please specify the geometry file (xyz format, Angstroms):
Geometry filename: [geom.xyz] (autocomplete enabled) <ENTER>    # use default "geom.xyz"
Number of atoms: 6
Nuclear charge: 17

Enter the total (net) molecular charge:
Charge: [0] +1        # For cation
Number of electrons: 16


-----------------------Level of theory----------------------

Supported by this script are:
  1       RASSCF
  2       CASPT2 (Only numerical gradients)

Level of theory: 1

Please enter the basis set.
Common available basis sets:
  Pople:     6-31G**, 6-311G, 6-31+G, 6-31G(d,p), ...
  Dunning:   cc-pVXZ, aug-cc-pVXZ, cc-pVXZ-DK, ...
  ANO:       ANO-S-vdzp, ANO-L, ANO-RCC
Basis set: cc-pVDZ
Use Cholesky decomposition? [False] <ENTER>
Douglas-Kroll scalar-relativistic integrals? [True] no

-----------------------CASSCF Settings----------------------

Number of active electrons: 6
Number of active orbitals: 4
Please enter the number of states for state-averaging as a list of integers
e.g. 3 0 2 for three singlets, zero doublets and two triplets.
Number of states: [0 0 0 0 0 0 0 0] 4  0  3    # same as for dynamics
Accepted number of states: 4 0 3

Please specify the state to optimize
e.g. 3 2 for the second triplet state.
Root: [1 1] <ENTER>    # singlet ground state
Optimization: Only performing one RASSCF for Singlets.  # triplets ignored in optimization of S0
Accepted number of states: 4 0 0

----------------------Further Settings----------------------

#########################Full input#########################    # this output is for debugging

ltype            1
soc              False
cholesky         False
DK               False
basis            cc-pVDZ
cas.norb         4
maxmult          3
cas.nact         6
ctype            2
nelec            16
masslist         [['C', 12.0],
                  ['N', 14.00307],
                  ['H', 1.00782],
                  ['H', 1.00782],
                  ['H', 1.00782],
                  ['H', 1.00782]]
charge           1
geom             [['C', 0.0, 0.0, 0.0],
                  ['N', 0.0, 0.0, 1.3],
                  ['H', 0.943102, 0.0, -0.5445],
                  ['H', 0.943102, 0.0, 1.8795],
                  ['H', -0.943102, 0.0, 1.8795],
                  ['H', -0.943102, 0.0, -0.5445]]
natom            6
ncharge          17
freq             True
opt.root         [1, 1]
cas.nstates      [4, 0, 0]

Writing input to MOLCAS.input

Runscript? [True] false

Finished

************************** WARNINGS **************************
*                                                            *
*                                                            *
**************************************************************
```

In order to execute the generated OpenMolcas job, we use the same run script as in the previous hands-on:

```
#!/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
```

Save it as `run.sh` and execute it via the SLURM queueing system:

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

The calculation can take around 10 min.

The job will produce the files `MOLCAS.input.log`, `MOLCAS.freq.molden` and `MOLCAS.RasOrb`. The first file contains (among other things) the ground state minimum energy (should be −94.412945 Hartree) and the vibrational wavenumbers. Check for any imaginary frequencies. There should be an output block like this in the output file (search for `Numerical differentiation is finished!` to find it quickly, and note that Molcas also prints isotope-shifted frequencies which can be easily confused with the non-shifted results):

```
 Numerical differentiation is finished!

Observe that the harmonic oscillator analysis is only valid at stationary points!

Note that rotational and translational degrees have been automatically removed.


Harmonic frequencies in cm-1

IR Intensities in km/mol
                       1         2         3         4         5         6

    Frequency:       973.56   1022.45   1213.37   1312.70   1418.38   1537.70

    Intensity:    1.712E+02 2.257E+00 3.301E-03 2.653E+01 7.532E+01 5.631E-02
    Red. mass:      1.29032   1.03757   1.00784   1.29646   1.48037   1.23910

    C1         x    0.00045  -0.03547  -0.00005  -0.00038   0.12239  -0.00006
    C1         y    0.03961   0.00107  -0.00117  -0.14023  -0.00029  -0.00007
     ⋮          ⋮          ⋮          ⋮         ⋮          ⋮         ⋮          ⋮
```

The `MOLCAS.RasOrb` file contains the CASSCF orbital coefficients and can be used to provide starting orbitals for subsequent calculations. The `MOLCAS.freq.molden` is needed in the next step.

As the Molcas calculation produces also several other output files, it is recommended that for the following steps you switch to a different directory:

```
(base) user@node: ~> mkdir ../Tutorial/
(base) user@node: ~> cp MOLCAS.freq.molden MOLCAS.RasOrb MOLCAS.input ../Tutorial/
(base) user@node: ~> cd ../Tutorial/
```

In some cases, OpenMolcas will not correctly write the element symbols into the `MOLCAS.freq.molden` file. In this case, you have to manually fix this problem by editing the file. The corrected part should look like this:

```
 [FR-COORD]
C  3.586620339811672E-010 -4.467640534501845E-005  6.656341381004939E-002
N  9.032741221459604E-010 -1.320316512501896E-005   2.47204540793567
H   1.77787413660691       3.134877222312748E-005 -0.944096197088421
H   1.62627742241765      -3.575897848586808E-006   3.47314129753096
H  -1.62627742254346       2.401178687595057E-005   3.47314129463728
H  -1.77787413774303       6.094909219504652E-006 -0.944096195829943
```

Questions:

1. What would you do if the molecule has multiple different minima in the ground state?
2. Do you think you need to check the active orbitals in the optimization calculation?

5.2 Sampling initial Conditions from a Wigner distribution

In the next step, the initial coordinates and velocities for the trajectories have to be generated. Here, this task is acomplished by sampling randomly from the Wigner distribution of the ground state nuclear wavefunction (in the harmonic approximation), which can be calculated from the vibrational frequencies and normal modes. This task is performed by the non-interactive script `wigner.py`, which can be executed by typing

```
(base) user@node: ~> python2 $SHARC/wigner.py -n 20 -r 16661 MOLCAS.freq.molden
```

The `-n` option is necessary to specify the number of initial conditions to be generated. Here, we generate 20 initial conditions. In order to obtain different results for each participant, please use your own random number seed with the `-r` option. Please remember this number.

The output should look like this:

```
Initial condition generation started...
INPUT  file                  = "MOLCAS.freq.molden"
OUTPUT file                  = "initconds"
Number of geometries         = 20
Random number generator seed = 16661
Temperature                  = 0.000000

***************************************************    # MOLCAS does not write translations and
WARNING: Less than 3*N_atom normal modes extracted!    # rotations to MOLCAS.freq.molden.
***************************************************    # You can ignore this warning.


Starting normal mode format determination...
Final format specifier: 2 [cartesian (Molpro, Molcas)]
Multiple possible flags have been identified:
  gaussian-type (Gaussian, Turbomole, Q-Chem, ADF, Orca)
  cartesian (Molpro, Molcas)
The most likely assumption is cartesian (Molpro, Molcas) coordinates.    # Correctly identified Molcas!
These have been used in the creation of inital conditions.

You can override this behavior by setting the -f [int] flag in the command line:
  1     gaussian-type (Gaussian, Turbomole, Q-Chem, ADF, Orca)
  2     cartesian (Molpro, Molcas)
  3     columbus-type (Columbus)
  4     mass-weighted


Geometry:
 C   6.0  -0.00000000   0.00002330   0.06656344  12.00000000
 N   7.0  -0.00000000   0.00000709   2.47204541  14.00307400
 H   1.0   1.77787414  -0.00000655  -0.94409618   1.00782500
 H   1.0   1.62627744  -0.00000895   3.47314127   1.00782500
 H   1.0  -1.62627744  -0.00000187   3.47314127   1.00782500
 H   1.0  -1.77787414  -0.00001303  -0.94409618   1.00782500
Assumed Isotopes: H-1 C-12 N-14
Isotopes with * are pure isotopes.

Frequencies (cm^-1) used in the calculation:
   1     973.5577
   2    1022.4479
   3    1213.3713
   4    1312.7017
   5    1418.3754
   6    1537.6966
   7    1679.0753
   8    1882.2763
   9    3329.9070
  10    3463.7137
  11    3679.7697
  12    3763.9964

Sampling initial conditions
Progress: [==================================================] 100%
```

The results of the sampling are written to the file `initconds`. This file contains all necessary information (equilibrium geometry, plus 20 sets of randomly sampled geometries with corresponding velocities) for subsequent steps.

Questions:

1. The Wigner distribution sampling only represents the molecular ground state vibrations only well if the molecule has only one reachable minimum and is reasonably rigid. Can you imagine other methods of generating initial conditions (geometries and velocities) that could be an alternative if Wigner sampling does not work?

5.3 Setting up the initial energy calculations

Besides the initial geometries and velocities, it is necessary to determine (for each initial geometry) the initial excited state from where the dynamics commences. In order to find the initial states after instantaneous vertical excitation, it is necessary to obtain the excitation energies and oscillator strengths for all initial geometries. These calculations can be setup using the script `setup_init.py`.

Note that it is also possible to simply specify the initial state for each initial condition manually; in this case, it is not necessary to use `setup_init.py` to prepare vertical excitation calculations.

5.3.1 Molcas input template

The computations which are setup by `setup_init.py` utilize the Sharc interfaces to call the respective quantum chemistry program (Molcas here). The Sharc-Molcas interface requires a template file which specifies the level of theory. Thus, before proceeding to setup the initial calculations, we need to prepare the Molcas template file.

Again, launch the Molcas input generator:

```
(base) user@node: ~> python2 $SHARC/molcas_input.py
```

Prepare a template file for the Sharc-Molcas interface.

```
  ================================================================================
 ||                                                                              ||
 ||                          MOLCAS Input file generator                         ||
 ||                                                                              ||
 ||                             Author: Sebastian Mai                            ||
 ||                                                                              ||
 ||                                  Version:2.1                                 ||
 ||                                   01.09.19                                   ||
 ||                                                                              ||
  ================================================================================


This script allows to quickly create MOLCAS input files for single-points calculations
on the SA-CASSCF and (MS-)CASPT2 levels of theory.
It also generates MOLCAS.template files to be used with the SHARC-MOLCAS Interface.

---------------------Type of calculation--------------------

This script generates input for the following types of calculations:
  1       Single point calculations (RASSCF, CASPT2)
  2       Optimizations & Frequency calculations (RASSCF, CASPT2)
  3       MOLCAS.template file for SHARC dynamics (SA-CASSCF)
Please enter the number corresponding to the type of calculation.

Type of calculation: 3        # MOLCAS.template generation

-------------------------Geometry-------------------------

No geometry necessary for MOLCAS.template generation

Number of electrons:  [16] <ENTER>       # If MOLCAS.input is in the same directory,
                                         # some of the following input is auto-detected.
```

```
-----------------------Level of theory----------------------

Supported by this script are:
  1       RASSCF
  2       CASPT2

Level of theory: 1

Please enter the basis set.
Common available basis sets:
  Pople:     6-31G**, 6-311G, 6-31+G, 6-31G(d,p), ...    (Not available)
  Dunning:   cc-pVXZ, aug-cc-pVXZ, cc-pVXZ-DK, ...
  ANO:       ANO-S-vdzp, ANO-L, ANO-RCC
Basis set: [cc-pVDZ] <ENTER>
Use Cholesky decomposition? [False] <ENTER>
Douglas-Kroll scalar-relativistic integrals? [True] no

-----------------------CASSCF Settings----------------------

Number of active electrons: [6] <ENTER>
Number of active orbitals: [4] <ENTER>
Please enter the number of states for state-averaging as a list of integers
e.g. 3 0 2 for three singlets, zero doublets and two triplets.
Number of states: [4] 4  0  3       # Override suggestion to include triplet states.
Accepted number of states: 4 0 3

----------------------Further Settings----------------------


#########################Full input#########################

ltype            1
soc              False
cholesky         False
DK               False
basis            cc-pVDZ
cas.norb         4
maxmult          3
cas.nact         6
ctype            3
cas.nstates      [4, 0, 3]
nelec            16
geom             None
freq             False

Writing input to MOLCAS.template

Finished

************************** WARNINGS **************************
*                                                            *
*                                                            *
**************************************************************
```

This will create a file called `MOLCAS.template`, which is needed for the following setup steps.

5.3.2 Setup of vertical excitation calculations

With the necessary files (`initconds`, `MOLCAS.template`, `MOLCAS.RasOrb`) available, the script `setup_init.py` can be launched. Note that this script creates a large number of subdirectories in the directory where the script is run, so it is advisable to run the setup in a dedicated directory:

```
(base) user@node: ~> mkdir init/
(base) user@node: ~> cd init/
(base) user@node: ~> python2 $SHARC/setup_init.py
```

This script is also interactive.

```
  ================================================================================
||                                                                                ||
||                   Setup initial conditions for SHARC dynamics                  ||
||                                                                                ||
||                              Author: Sebastian Mai                             ||
||                                                                                ||
||                                   Version:2.1                                  ||
||                                    01.09.19                                    ||
||                                                                                ||
  ================================================================================


This script automatizes the setup of excited-state calculations for initial conditions
for SHARC dynamics.

-------------------Initial conditions file------------------


If you do not have an initial conditions file, prepare one with wigner.py!

Please enter the filename of the initial conditions file.
Initial conditions filename: [initconds] (autocomplete enabled) ../initconds

File "../initconds" contains 20 initial conditions.
Number of atoms is 6

-----------------Range of initial conditions----------------

Please enter the range of initial conditions for which an excited-state calculation should be
performed as two integers separated by space.
Initial condition range: [1 20] 1  10    # not all initial conditions need to be calculated

Script will use initial conditions 1 to 10 (10 in total).

----------------------Number of states----------------------

Please enter the number of states as a list of integers
e.g. 3 0 3 for three singlets, zero doublets and three triplets.
Number of states: 4  0  3     # this could differ from the number of SA roots in MOLCAS.template

Number of states: [4, 0, 3]
Total number of states: 13

-----------Choose the quantum chemistry interface-----------

Please specify the quantum chemistry interface (enter any of the following numbers):
```

```
1       MOLPRO (only CASSCF)
2       COLUMBUS (CASSCF, RASSCF and MRCISD), using SEWARD integrals
3       Analytical PESs
4       MOLCAS (CASSCF, CASPT2, MS-CASPT2)
5       ADF (DFT, TD-DFT)
6       TURBOMOLE (ricc2 with CC2 and ADC(2))
7       LVC Hamiltonian
8       GAUSSIAN (DFT, TD-DFT)
9       ORCA (DFT, TD-DFT, HF, CIS)
10      BAGEL (CASSCF, CASPT2, (X)MS-CASPT2)

Interface number: 4

-----------------Spin-orbit couplings (SOCs)----------------

Do you want to compute spin-orbit couplings?

Spin-Orbit calculation? [True] <ENTER>     # not required, but included anyways.
Will calculate spin-orbit matrix.


----------------Overlaps to reference states----------------

Do you want to compute the overlaps between the states at the equilibrium geometry
and the states at the initial condition geometries?
Reference overlaps? [False] yes    # not required, but included anyways.


  ================================================================================
||                            MOLCAS Interface setup                            ||
  ================================================================================



-----------------------Path to MOLCAS-----------------------


Please specify path to MOLCAS directory (SHELL variables and ~ can be used, will be expanded
when interface is started).

Path to MOLCAS: [$MOLCAS/] (autocomplete enabled) <ENTER>

----------------------Scratch directory---------------------

Please specify an appropriate scratch directory. This will be used to temporally store the integrals.
The scratch directory will be deleted after the calculation. Remember that this script cannot check
whether the path is valid, since you may run the calculations on a different machine.
The path will not be expanded by this script.
Path to scratch directory: (autocomplete enabled) $SCRADIR/Tutorial/Init

-----------------MOLCAS input template file-----------------

Please specify the path to the MOLCAS.template file. This file must contain the following settings:

basis <Basis set>
ras2 <Number of active orbitals>
nactel <Number of active electrons>
inactive <Number of doubly occupied orbitals>
roots <Number of roots for state-averaging>
```

```
The MOLCAS interface will generate the appropriate MOLCAS input automatically.

Template filename: (autocomplete enabled) ../MOLCAS.template

---------------Initial wavefunction: MO Guess---------------

Please specify the path to a MOLCAS JobIph file containing suitable starting MOs for the CASSCF
calculation. Please note that this script cannot check whether the wavefunction file and the
Input template are consistent!

Do you have initial wavefunction files for Singlet, Triplet? [True] <ENTER>
JobIph files (1) or RasOrb files (2)? 2
Initial wavefunction file for Singlets: [MOLCAS.1.RasOrb.init] (autocomplete enabled) MOLCAS.RasOrb
Initial wavefunction file for Triplets: [MOLCAS.3.RasOrb.init] (autocomplete enabled) MOLCAS.RasOrb

-------------------MOLCAS Ressource usage-------------------

Please specify the amount of memory available to MOLCAS (in MB). For calculations including moderately-
sized CASSCF calculations and less than 150 basis functions, around 2000 MB should be sufficient.

MOLCAS memory: [1000] 500
Please specify the number of CPUs to be used by EACH calculation.

Number of CPUs: [1] <ENTER>

  ================================================================================
||                                 Run mode setup                                 ||
  ================================================================================



-------------------------Run script------------------------

This script can generate the run scripts for each initial condition in two modes:

  - In the first mode, the calculation is run in subdirectories of the current directory.

  - In the second mode, the input files are transferred to another directory (e.g. a
  local scratch directory), the calculation is run there, results are copied back and
  the temporary directory is deleted. Note that this temporary directory is not the
  same as the scratchdir employed by the interfaces.

Note that in any case this script will setup the input subdirectories in the current working directory.

Do you want to use mode 1
(actually perform the calculations in subdirectories of:
/user/mai/Documents/NewSHARC/SHARC_2.0/TUTORIAL/2_full/Tutorial/init)

Calculate here? [True] <ENTER>     # this is now the default

----------------------Submission script---------------------

During the setup, a script for running all initial conditions sequentially in batch mode is generated.
Additionally, a queue submission script can be generated for all initial conditions.

Generate submission script? [False] <ENTER>


#########################Full input#########################
```

```
molcas.ncpu                1
initf                      <open file '../initconds', mode 'r' at 0xb65d20>
soc                        True
molcas                     $MOLCAS/
ninit                      20
molcas.guess               {1: '../MOLCAS.RasOrb', 3: '../MOLCAS.RasOrb'}
needed                     []
molcas.template            ../MOLCAS.template
here                       True
states                     [4, 0, 3]
interface                  4
molcas.mem                 500
scratchdir                 $SCRADIR/Tutorial/Init
irange                     [1, 10]
natom                      6
qsub                       False
molcas.jobiph_or_rasorb    2
nstates                    13
cwd                        /user/mai/Documents/NewSHARC/SHARC_2.0/TUTORIAL/2_full/Tutorial/init
refov                      True

Do you want to setup the specified calculations? [True] <ENTER>


  ==============================================================================
||                           Setting up directories...                          ||
  ==============================================================================$


Progress: [==================================================] 100%
```

The script will create directories ICOND_00001/, ICOND_00002/, ... for each initial condition (and ICOND_00000/ for the equilibrium geometry), with the corresponding inputs for the interface and a Bash runscript.

The setup script has automatically created run scripts for all the subdirectories. However, we need to add to these run scripts the sourcing of the OpenMolcas environment variables. This can be accomplished like this (write everything on one line):

```
user@node: ~> for i in ICOND*; do sed -i
  '2i source /projects/academic/cyberwksp21/Students/smai/Instructors_material/set_openmolcas.sh'
  $i/run.sh; done
```

This loop will add the sourcing as the second line to each of the run scripts. Then the jobs are in principle prepared. Because we requested reference overlaps, we must finish ICOND_00000 before launching any of the other jobs. Hence, run the first calculation:

```
user@node: ~> cd ICOND_00000
user@node: ~> sbatch -A cyberwksp21 -p valhalla -q valhalla -c 1 run.sh
```

Then wait the approximately 30 seconds for completion. If successful, the directory should now contain a file called QM.out. Then go back and launch all other jobs (again, write the loop as one line):

```
user@node: ~> cd ..
user@node: ~> for i in {00001..00010}; do cd ICOND_$i;
  sbatch -A cyberwksp21 -p valhalla -q valhalla -c 1 run.sh
  cd -; done
```

After the calculations are finished, each subdirectory should contain a file called `QM.out` holding the Hamiltonian and transition dipole moment matrices.

Questions:

1. Why should you always provide initial CASSCF orbitals for these vertical excitation calculations?

5.4 Selection of initial excited states

In the next step, the results of the excited-state calculations have to be read, converted to excitation energies and oscillator strengths, and the brightest initial conditions selected for the dynamics simulation. These tasks can be accomplished using

```
(base) user@node: ~> python2 $SHARC/excite.py
```

This script is interactive. Per default, during the run the script reads the ground state equilibrium energy from `ICOND_00000/QM.out`, if this file exists. Otherwise, the script asks the user to enter the ground states equilibrium energy.

```
  ================================================================================
||                                                                                ||
||                       Excite initial conditions for SHARC                      ||
||                                                                                ||
||                              Author: Sebastian Mai                             ||
||                                                                                ||
||                                   Version:2.1                                  ||
||                                    01.09.19                                    ||
||                                                                                ||
  ================================================================================


This script automatizes to read-out the results of initial excited-state calculations for SHARC.
It calculates oscillator strength (in MCH and diagonal basis) and stochastically determines whether
a trajectory is bright or not.

-------------------Initial conditions file------------------


If you do not have an initial conditions file, prepare one with wigner.py!

Please enter the filename of the initial conditions file.
Initial conditions filename: [initconds] (autocomplete enabled) ../initconds

File "../initconds" contains 20 initial conditions.
Number of atoms is 6

----------------Generate excited state lists----------------

Using the following options, excited state lists can be added to the initial conditions:

1       Generate a list of dummy states
2       Read excited-state information from ab initio calculations (from setup_init.py)

How should the excited-state lists be generated? [2] <ENTER>     # Read from ICOND_*/
Please enter the path to the directory containing the ICOND subdirectories.
Path to ICOND directories: (autocomplete enabled) .            # "." is the current directory

/user/mai/Documents/NewSHARC/SHARC_2.0/TUTORIAL/2_full/Tutorial/init
Directory contains 11 subdirectories.
There are more initial conditions in ../initconds.

----------------Excited-state representation----------------

This script can calculate the excited-state energies and oscillator strengths in two representations.
These representations are:
```

```
- MCH representation: Only the diagonal elements of the Hamiltonian are taken into account.
The states are the spin-free states as calculated in the quantum chemistry code.
This option is usually sufficient for systems with small SOC (below 300 cm^-1).
- diagonal representation: The Hamiltonian including spin-orbit coupling is diagonalized.
The states are spin-corrected, fully adiabatic. Note that for this the excited-state calculations
have to include spin-orbit couplings. This is usually not necessary for systems with small SOC.

Do you want to use the diagonal representation (yes=diag, no=MCH)? no


----------------------Reference energy----------------------

Reference energy read from file
/user/mai/Documents/NewSHARC/SHARC_2.0/TUTORIAL/2_full/Tutorial/init/ICOND_00000/QM.out
E_ref= -94.412945540000      # automatically read from ICOND_00000/QM.out


-------------------Excited-state selection------------------

Using the following options, the excited states can be flagged as valid initial states for dynamics:

1       Unselect all initial states
2       Provide a list of desired initial states
3       Simulate delta-pulse excitation based on excitation energies and oscillator strengths

How should the excited states be flagged? [3] <ENTER>


----------------------Excitation window---------------------

Enter the energy window for exciting the trajectories.
Range (eV): [0.0 10.0] 10  11

Script will allow excitations only between 10.000000 eV and 11.000000 eV.

----------------------Considered states---------------------

From which state should the excitation originate (for computation of excitation energies
and oscillator strength)?
Lower state for excitation? [1] <ENTER>
#State  Mult    M_s     Quant    # Here the states are listed.
1       1       +0.0    1
2       1       +0.0    2
3       1       +0.0    3
4       1       +0.0    4
5       3       -1.0    1
6       3       -1.0    2
7       3       -1.0    3
8       3       +0.0    1
9       3       +0.0    2
10      3       +0.0    3
11      3       +1.0    1
12      3       +1.0    2
13      3       +1.0    3

Do you want to include all states in the selection? [True] <ENTER>

---------------------Random number seed---------------------
```

```
Please enter a random number generator seed (type "!" to initialize the RNG from the system time).
RNG Seed:  [!] 1234



#########################Full input#########################

initf                      <open file '../initconds', mode 'r' at 0x2b98150>
eharm                      0.0
ninit                      20
diag                       False
erange                     [0.3307437794202775, 0.4409917058937033]
allowed                    set([])
excite                     3
repr                       MCH
states                     [4, 0, 3]
ion                        False
iconddir                   /user/mai/Documents/NewSHARC/SHARC_2.0/TUTORIAL/2_full/Tutorial/init
make_list                  False
eref                       -94.41294554
ncond                      11
natom                      6
read_QMout                 True
initstate                  0
diabatize                  False
gen_list                   2

Do you want to continue? [True] <ENTER>

Reading initial condition file ....
  Progress: [==================================================] 100%
Number of initial conditions in file:          20

Reading QM.out data ...
  Progress: [==================================================] 100%
Number of initial conditions with QM.out:      10

Selecting initial states ...
  Progress: [==================================================] 100%
Number of initial states:                       9

Number of initial conditions excited:
State   Selected   InRange   Total
    1          0         0      10
    2          0         2      10
    3          9        10      10     # we can setup 9 trajectories from state 3 (S2)
    4          0         6      10
    5          0         0      10
    6          0         0      10
    7          0        10      10
    8          0         0      10
    9          0         0      10
   10          0        10      10
   11          0         0      10
   12          0         0      10
   13          0        10      10
Writing output to ../initconds.excited ...
```

`excite.py` will generate a new file called `initconds.excited`, which contains all information from the `initconds` file, as well as information about the ground state equilibrium energy, the state representation and the excited states for each initial condition. This file is necessary in order to calculate absorption spectra and to setup trajectories.

If you later want to do another selection (with a different excitation window or with the exclusion of some states), you can tell `excite.py` to read from `initconds.excited`, instead of reading all `QM.out` files again.

From the `initconds.excited` file, also absorption spectra can be generated, see section 5.5. If you do not want to compute a spectrum and directly go to the trajectory setup, go to section 5.6.

Questions:

1. What would you do if you need more than 9 trajectories?

5.5 Absorption spectra from initial conditions file

The content of the file `initconds.excited` can be used to generate absorption spectra which go beyond the Condon approximation. The spectrum is the sum of the spectra of each initial condition, which is a line spectrum of the excitation energies versus the oscillator strengths. A Gaussian (or Lorentzian, or Log-normal) convolution of the line spectra can be done as well.

The calculation of convoluted or line spectra is carried out by `spectrum.py`.

Call the script by

```
(base) user@node: ~> cd ..
(base) user@node: ~> python2 $SHARC/spectrum.py -o spectrum.out -e 9 12 initconds.excited
```

Using command-line options, it is possible to calculate only spectra for part of the initial condition set, to change the size and limits of the energy grid (here we plot from 9 eV to 12 eV) and to influence the line shape (Gaussian vs. Lorentzian vs. Log-normal, as well as FWHM). With the `-l` option a line spectrum is produced, and with the `-D` option a density-of-states spectrum is produced.

The program also writes some information about the calculation to the screen:

```
Number of grid points: 500
Energy range: 9.000 to 12.000 eV
Lineshape: Gaussian (FWHM=0.100 eV)
Number of initial conditions: 20
Reference energy   -94.4129455400
Representation: MCH
Reading initial conditions 1 to 20

Number of states: 13
Number of initial conditions with excited-state information (per state):
10 10 10 10 10 10 10 10 10 10 10 10 10

Progress: [==================================================] 100%

Maximum of the absorption spectrum: 0.959090

Output spectrum written to "spectrum.out".
```

The results can be easily plotted using Gnuplot. Just give the corresponding command-line flag and then call Gnuplot:

```
(base) user@node: ~> python2 $SHARC/spectrum.py -o spectrum.out -e 9 12
                     --gnuplot spectrum.gp initconds.excited
(base) user@node: ~> gnuplot spectrum.gp
```

In Figure 1 an example of the result of this convolution is shown. Note that on CASSCF level of theory, the excitation energy of $CH_2NH_2^+$ is reproduced quite badly and the number of initial conditions is too low to reliably sample the ground state Wigner distribution.

Questions:

1. Do you think there is any significance in the band structure of the spectrum shown in Figure 1?
2. What could you do in order to improve the quality of the simulated spectrum?

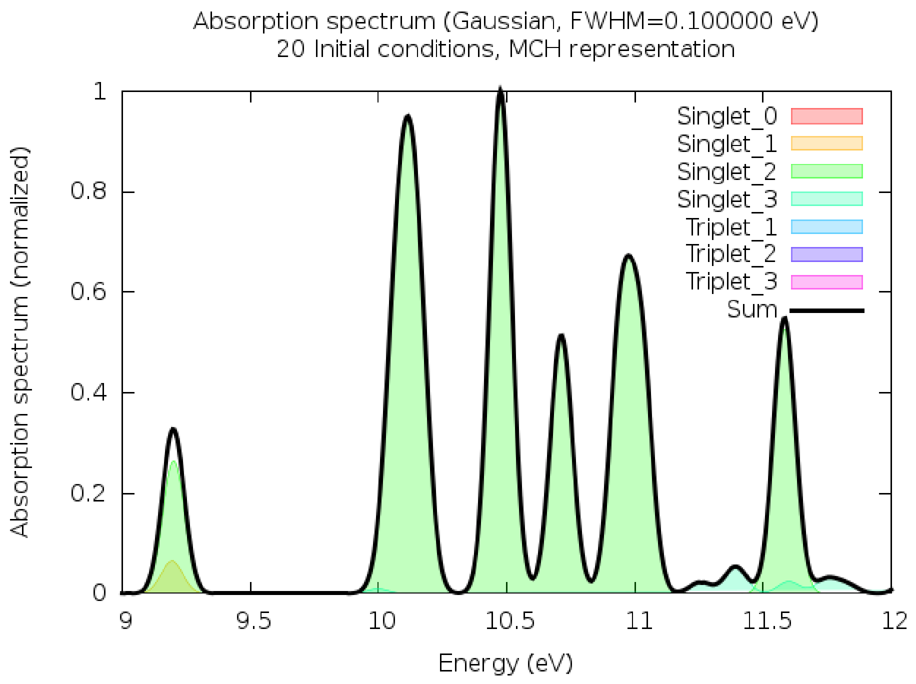


Figure 1: Absorption spectrum based on 10 initial conditions (Since there are 20 initial conditions in `initconds.excited`, the title lists 20 instead of 10).

5.6 Setting up dynamics simulations

The last preparatory step towards dynamics simulations consists naturally in setting up the SHARC input files and run scripts. The interactive script `setup_traj.py` takes care of this step. Run the script in the directory where the trajectories should be set up. For this, create a new directory:

```
(base) user@node: ~> mkdir traj
(base) user@node: ~> cd traj
```

Make sure that you have all required files (`initconds.excited`, `MOLCAS.template`, `MOLCAS.RasOrb`) in this directory. Then start the setup script:

```
(base) user@node: ~> $SHARC/setup_traj.py
```

```
  ================================================================================
||                                                                                ||
||                       Setup trajectories for SHARC dynamics                    ||
||                                                                                ||
||                     Author: Sebastian Mai, Philipp Marquetand                  ||
||                                                                                ||
||                                    Version:2.1                                 ||
||                                     01.09.19                                   ||
||                                                                                ||
  ================================================================================


This script automatizes the setup of the input files for SHARC dynamics.


  ================================================================================
||                                Initial conditions                              ||
  ================================================================================



This script reads the initial conditions (geometries, velocities, initial excited state)
from the initconds.excited files as provided by excite.py.

Please enter the filename of the initial conditions file.
Initial conditions filename: [initconds.excited] (autocomplete enabled) ../initconds.excited

File ../initconds.excited contains 20 initial conditions.
Number of atoms is 6
Reference energy -94.412945540000 a.u.
Excited states are in MCH representation.


Please enter the number of states as a list of integers
e.g. 3 0 3 for three singlets, zero doublets and three triplets.
Number of states: [4 0 3] <ENTER>        # a different number of states than in the
                                         # initial calculations could be used in the dynamics

Number of states: [4, 0, 3]
Total number of states: 13

Do you want all states to be active? [True] <ENTER>

Do you want to see the content of the initconds file? [True] <ENTER>
```

```
Number of initial conditions in file:         20
Contents of the initconds file:

Legend:
?       Geometry and Velocity
.       not selected
#       selected

State 1:                     # we only performed 10 calculations, hence the "?"
             10         20
              |          |
 0 | .......... ??????????   # no initial conditions selected in S0
State 2:
             10         20
              |          |
 0 | .......... ??????????   # no initial conditions selected in S1
State 3:
             10         20
              |          |
 0 | .######### ??????????   # initial conditions 2 to 10 are selected
State 4:
             10         20
              |          |
 0 | .......... ??????????   # no initial conditions selected in S3
State 5:
             10         20
              |          |
 0 | .......... ??????????   # no initial conditions selected in T1
State 6:
             10         20
              |          |
 0 | .......... ??????????   # no initial conditions selected in T2
State 7:
             10         20
              |          |
 0 | .......... ??????????   # no initial conditions selected in T3
State 8:
             10         20
              |          |
 0 | .......... ??????????
State 9:
             10         20
              |          |
 0 | .......... ??????????
State 10:
             10         20
              |          |
 0 | .......... ??????????
State 11:
             10         20
              |          |
 0 | .......... ??????????
State 12:
             10         20
              |          |
 0 | .......... ??????????
State 13:
             10         20
              |          |
 0 | .......... ??????????
Number of excited states and selections:
State    #InitCalc       #Selected
    1          10              0
    2          10              0
    3          10              9  # we can setup 9 trajectories starting in state 3 (S2)
    4          10              0
    5          10              0
    6          10              0
    7          10              0
    8          10              0
    9          10              0
   10          10              0
   11          10              0
   12          10              0
   13          10              0

Please enter a list specifying for which excited states trajectories should be set-up
e.g. 6 10 11 to select states 6, 10, and 11.
States to setup the dynamics: [3] (range comprehension enabled) 3

There can be 9 trajectories set up.

Please enter the index of the first initial condition in the initconds file to be setup.
Starting index: [1] <ENTER>

There can be 9 trajectories set up, starting in 1 states.

Please enter the total number of trajectories to setup.
Number of trajectories: [9] <ENTER>

Please enter a random number generator seed (type "!" to initialize the RNG from the system time).
RNG Seed:  [!] 1234


  ================================================================================
||                     Choose the quantum chemistry interface                     ||
  ================================================================================



Please specify the quantum chemistry interface (enter any of the following numbers):
1       MOLPRO (only CASSCF)
2       COLUMBUS (CASSCF, RASSCF and MRCISD), using SEWARD integrals
3       Analytical PESs
4       MOLCAS (CASSCF, CASPT2, MS-CASPT2)
5       ADF (DFT, TD-DFT)
6       TURBOMOLE (ricc2 with CC2 and ADC(2))
7       LVC Hamiltonian
8       GAUSSIAN (DFT, TD-DFT)

Interface number: 4

  ================================================================================
||                        Surface Hopping dynamics settings                       ||
  ================================================================================



----------------------Simulation time----------------------
```

```
Please enter the total simulation time.
Simulation time (fs): [1000.0] 100

Please enter the simulation timestep (0.5 fs recommended).
Simulation timestep (fs): [0.5] <ENTER>

Simulation will have 201 timesteps.

Please enter the number of substeps for propagation (25 recommended).
Nsubsteps: [25] <ENTER>

The trajectories can be prematurely terminated after they run for a certain time in the lowest state.
Do you want to prematurely terminate trajectories? [False] <ENTER>


----------------------Dynamics settings---------------------

Do you want to perform the dynamics in the diagonal representation (SHARC dynamics)
or in the MCH representation (regular surface hopping)?
SHARC dynamics? [True] <ENTER>
Do you want to include spin-orbit couplings in the dynamics?

Spin-Orbit calculation? [True] <ENTER>
Will calculate spin-orbit matrix.


Please choose the quantities to describe non-adiabatic effects between the states:
1       DDT     =  < a|d/dt|b >       Hammes-Schiffer-Tully scheme   (not available)
2       DDR     =  < a|d/dR|b >       Original Tully scheme          (not available)
3       overlap = < a(t0)|b(t) >      Local Diabatization scheme
Coupling number: [3] <ENTER>

For SHARC dynamics, the evaluation of the mixed gradients necessitates to calculate
non-adiabatic coupling vectors (Extra computational cost).
... but interface cannot provide non-adiabatic coupling vectors, turning option off.

During a surface hop, the kinetic energy has to be modified in order to conserve total energy.
There are several options to that:
1       Do not conserve total energy. Hops are never frustrated.
2       Adjust kinetic energy by rescaling the velocity vectors. Often sufficient.
3       Adjust kinetic energy only with the component of the velocity vector along
        the non-adiabatic coupling vector.        (not possible)
4       Adjust kinetic energy only with the component of the velocity vector along
        the gradient difference vector.
EkinCorrect: [2] <ENTER>

If a surface hop is refused (frustrated) due to insufficient energy, the velocity can either be
left unchanged or reflected:
1       Do not reflect at a frustrated hop.
2       Reflect the full velocity vector.
3       Reflect only the component of the velocity vector along the non-adiabatic coupling vector.
        (not possible)
4       Reflect only the component of the velocity vector along the gradient difference vector.
Reflect frustrated: [1] <ENTER>

Please choose a decoherence correction for the diagonal states:
1       No decoherence correction.
2       Energy-based decoherence scheme (Granucci, Persico, Zoccante).
3       Augmented fewest-switching surface hopping (Jain, Alguire, Subotnik).
```

```
Decoherence scheme: [2] <ENTER>

Please choose a surface hopping scheme for the diagonal states:
1       Surface hops off.
2       Standard SHARC surface hopping probabilities (Mai, Marquetand, Gonzalez).
3       Global flux surface hopping probabilities (Wang, Trivedi, Prezhdo).
Hopping scheme: [2] <ENTER>

Do you want to perform forced hops to the lowest state based on a energy gap criterion?
(Note that this ignores spin multiplicity)
Forced hops to ground state? [False] <ENTER>

Do you want to scale the energies and gradients?
Scaling? [False] <ENTER>

Do you want to damp the dynamics (Kinetic energy is reduced at each timestep by a factor)?
Damping? [False] <ENTER>

Do you want to use an atom mask for velocity rescaling or decoherence?
Atom masking? [False] <ENTER>

---------------Selection of Gradients and NACs--------------

In order to speed up calculations, SHARC is able to select which gradients and NAC vectors it has to
calculate at a certain timestep. The selection is based on the energy difference between the state
under consideration and the classical occupied state.

Select gradients? [False] yes    # this strongly speeds up the calculations

Please enter the energy difference threshold for the selection of gradients and non-adiabatic
couplings (in eV). (0.5 eV recommended, or even larger if SOC is strong in this system.)
Selection threshold (eV): [0.5] 0.05


-------------------------Laser file-------------------------

Do you want to include a laser field in the simulation? [False] <ENTER>

  ================================================================================
||                             MOLCAS Interface setup                             ||
  ================================================================================



-----------------------Path to MOLCAS-----------------------

Please specify path to MOLCAS directory (SHELL variables and ~ can be used,
will be expanded when interface is started).

Path to MOLCAS: [$MOLCAS/] (autocomplete enabled) <ENTER>

----------------------Scratch directory---------------------

Please specify an appropriate scratch directory. This will be used to temporally
store the integrals. The scratch directory will be deleted after the calculation.
Remember that this script cannot check whether the path is valid, since you may
run the calculations on a different machine. The path will not be expanded by this script.
Path to scratch directory: (autocomplete enabled) $SCRADIR/Tutorial/Traj/

-----------------MOLCAS input template file-----------------
```

```
Please specify the path to the MOLcas.template file. This file must contain the following settings:

basis <Basis set>
ras2 <Number of active orbitals>
nactel <Number of active electrons>
inactive <Number of doubly occupied orbitals>
roots <Number of roots for state-averaging>

The MOLCAS interface will generate the appropriate MOLCAS input automatically.

Template filename: (autocomplete enabled) ../MOLCAS.template

---------------Initial wavefunction: MO Guess---------------

Please specify the path to a MOLCAS JobIph file containing suitable starting MOs for the CASSCF
calculation. Please note that this script cannot check whether the wavefunction file and the
Input template are consistent!

Do you have initial wavefunction files for Singlet, Triplet? [True] <ENTER>
JobIph files (1) or RasOrb files (2)? 2
Initial wavefunction file for Singlets: [MOLCAS.1.RasOrb.init] (autocomplete enabled) ../MOLCAS.RasOrb
Initial wavefunction file for Triplets: [MOLCAS.3.RasOrb.init] (autocomplete enabled) ../MOLCAS.RasOrb
-------------------MOLCAS Ressource usage-------------------

Please specify the amount of memory available to MOLCAS (in MB). For calculations including moderately-
sized CASSCF calculations and less than 150 basis functions, around 2000 MB should be sufficient.

MOLCAS memory: 500
Please specify the number of CPUs to be used by EACH calculation.

Number of CPUs: 1

  ================================================================================
||                          Content of output.dat files                          ||
  ================================================================================


SHARC or PYSHARC can produce output in ASCII format (all features supported currently)
or in NetCDF format (more efficient file I/O, some features currently not supported).
Write output in NetCDF format? [False] <ENTER>

Do you want to write the gradients to the output.dat file ?
Write gradients? [False] <ENTER>

Do you want to write the non-adiabatic couplings (NACs) to the output.dat file ?
Write NACs? [False] <ENTER>

Do you want to write property matrices to the output.dat file  (e.g., Dyson norms)?
Write property matrices? [False] <ENTER>

Do you want to write property vectors to the output.dat file  (e.g., TheoDORE results)?
Write property vectors? [False] <ENTER>

Do you want to write the overlap matrix to the output.dat file ?
Write overlap matrix? [True] <ENTER>

Do you want to modify the output.dat writing stride?
Modify stride? [False] <ENTER>
```

```
  ================================================================================
||                                Run mode setup                                ||
  ================================================================================


------------------------Run script------------------------

This script can generate the run scripts for each trajectory in two modes:

  - In the first mode, the calculation is run in subdirectories of the current directory.

  - In the second mode, the input files are transferred to another directory (e.g. a local scratch
  directory), the calculation is run there, results are copied back and the temporary directory is
  deleted. Note that this temporary directory is not the same as the scratchdir employed by the interfaces.

Note that in any case this script will setup the input subdirectories in the current working directory.

In case of mode 1, the calculations will be run in:
/user/mai/Documents/NewSHARC/SHARC_2.1/TUTORIAL/traj

Use mode 1 (i.e., calculate here)? [True] <ENTER>

---------------------Submission script---------------------

During the setup, a script for running all initial conditions sequentially in batch mode is generated.
Additionally, a queue submission script can be generated for all initial conditions.

Generate submission script? [False] <ENTER>


#########################Full input#########################$

molcas                          $MOLCAS/
soc                             True
ntraj                           9
ninit                           20
molcas.guess                    {1: '../MOLCAS.RasOrb', 3: '../MOLCAS.RasOrb'}
write_NAC                       False
write_property1d                False
dipolegrad                      False
states                          [4, 0, 3]
molcas.jobiph_or_rasorb         2
kill                            False
qsub                            False
nstates                         13
show_content                    True
molcas.ncpu                     1
eharm                           0.0
n_issel                         [0, 0, 9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
nsubstep                        25
scratchdir                      $TMPDIR/Tutorial/Traj_WORK/
diag                            False
actstates                       [4, 0, 3]
ekincorrect                     2
atommaskarray                   []
needed                          []
eselect                         0.1
eref                            -94.41294554
damping                         False
```

```
molcas.mem                  500
firstindex                  1
cwd                         /user/mai/Documents/NewSHARC/SHARC_2.0/TUTORIAL/2_full/Tutorial/traj
write_property2d            False
surf                        diagonal
initf                       <open file '../initconds.excited', mode 'r' at 0x1714c90>
molcas.template             ../MOLCAS.template
dtstep                      0.5
repr                        MCH
phases_from_interface       False
reflect                     1
ion                         False
statemap                    {1: [1, 1, 0.0],
                             2: [1, 2, 0.0],
                             3: [1, 3, 0.0],
                             4: [1, 4, 0.0],
                             5: [3, 1, -1.0],
                             6: [3, 2, -1.0],
                             7: [3, 3, -1.0],
                             8: [3, 1, 0.0],
                             9: [3, 2, 0.0],
                             10: [3, 3, 0.0],
                             11: [3, 1, 1.0],
                             12: [3, 2, 1.0],
                             13: [3, 3, 1.0]}
here                        True
interface                   4
write_overlap               True
scaling                     False
laser                       False
coupling                    3
hopping                     sharc
sel_g                       True
tmax                        100.0
write_grad                  False
copydir                     /user/mai/Documents/NewSHARC/SHARC_2.0/TUTORIAL/2_full/Tutorial/traj
sel_t                       False
setupstates                 set([3])
printlevel                  2
natom                       6
gradcorrect                 False
decoherence                 ['edc', '0.1']
isactive                    [True, True, True, True, True, True, True, True, True, True, True, True, True]

Do you want to setup the specified calculations? [True] <ENTER>


  ================================================================================
||                           Setting up directories...                            ||
  ================================================================================


Progress: [==================================================] 100%

9 trajectories setup, last initial condition was 10 in state 3.
```

The script creates for each of the initial states ("States to setup the dynamics") a directory called `<Mult>_<Num>`, e.g., `Singlet_2/`, which contains the input for all trajectories starting in that state. Each of these directories contains subdirectories named `TRAJ_00001/`, `TRAJ_00002/`, etc. Note that these numbers are not consecutive: if an initial condition has not been selected, the number will be missing. Each subdirectory contains the SHARC input (consisting of the files `input`, `geom`, and `veloc`), the directories `QM/` and `restart/`, and the run script for the trajectory, `run.sh`.

Questions:

1. At which workflow step would you need to restart if you want to simulate with a different temperature of your molecule?
2. At which workflow step would you need to restart if you want to simulate an experiment with a different excitation laser?
3. **Bonus**: Could you imagine what you would need to do to prepare simulations of collisions of two ground state molecules? What about if one of the molecules is excited?

5.7 Running the trajectories

This part is optional. Tomorrow in the workshop, you will be provided with a sizable swarm of trajectories. However, if you want, you can run the few trajectories that you have just set up and analyze them tomorrow, too.

Running works similarly to the initial condition calculations. First, we need to add to these run scripts the sourcing of the OpenMolcas environment variables. This can be accomplished like this (write everything on one line):

```
user@node: ~> for i in Singlet_2/TRAJ*; do sed -i
  '2i source /projects/academic/cyberwksp21/Students/smai/Instructors_material/set_openmolcas.sh'
  $i/run.sh; done
```

This loop will add the sourcing as the second line to each of the run scripts. Then the jobs are in principle prepared.

You can then launch all jobs (again, write the loop as one line):

```
user@node: ~> for i in Singlet_2/TRAJ*; do $i;
  sbatch -A cyberwksp21 -p valhalla -q valhalla -c 1 run.sh
  cd -; done
```

These calculations can be expected to take 3–4 hours to complete the 100 fs. You can thus launch them and then forget about them until tomorrow.