The ORCA MD Module *(Feb 2020)*

Martin Brehm
Martin-Luther-Universität Halle–Wittenberg
https://brehm-research.de/

This is a very short tutorial on the ORCA molecular dynamics module. More details can be found in the corresponding ORCA manual chapter *(see „Detailed Input --> Ab initio Molecular Dynamics Simulations")*. The manual of the ORCA MD module can also be found on

`https://brehm-research.de/orcamd`

For this tutorial, we will not submit jobs to the queueing system to run. We will rather directly log in to a compute node and call our programs there, which is much more convenient for this kind of task. With the following command, you start an interactive SLURM job on a compute node *(the last letter is a lowercase „L")*. If you already have an open interactive session *(e.g., from the CP2k spectroscopy tutorial)*, just continue to use it, don't open another one.

```
srun -A hpc-lco-usrtr --pty bash -l
```

We have to load the ORCA module, and we have to tell ORCA to use only one CPU core per process *(i.e., no OpenMP)*:

```
module load winterschool2020/orca
export OMP_NUM_THREADS=1
```

Before beginning with the actual tutorial, we copy the prepared directory structure with the input files to the home directory:

```
cp -r ~/Desktop/Winter_School/Tutorials/TRAVIS/ORCA ~
cd ~/ORCA
```

First, we run a simulation of an $Al(H_2O)_6^{3+}$ cation together with an ammonia molecule in vacuum. It is the simulation which you can see visualized on my website:

`https://brehm-research.de/orcamd`

Enter the directory, have a look at the input file („`input.inp`") you find there, and start the calculation:

```
cd 1_aluminum
/cm/shared/apps/pc2/ORCA/orca_4_2_1_linux_x86-64_openmpi314/orca input.inp
cd ..
```

The calculation runs only on one CPU core, and therefore is quite slow *(around 15 seconds per step)*. We can accelerate such a simulation by using more CPU cores. This is simply achieved by adding, e.g., „PAL16" to the simple command line in the ORCA input. We run this as second exercise:

```
cd 2_aluminum_parallel
/cm/shared/apps/pc2/ORCA/orca_4_2_1_linux_x86-64_openmpi314/orca input.inp
cd ..
```

The input file just runs for 150 steps *(around 15 minutes)*. Afterwards, you can visualize the resulting trajectory with VMD. Don't do this from within the interactive SLURM job on the compute node. Open a new tab in your terminal, navigate to the directory, and start VMD there:

```
cd ~/ORCA/2_aluminum_parallel
vmd trajectory.xyz
```

As a third example, we will use a semiempirical method (PM3) to run a MD of a quite large system – hexaphenylbenzene :-) A more reasonable choice would be Grimme's xTB, but this ORCA module is currently not installed on the compute cluster. Semiempirics is very fast. On one single CPU core, each MD step of this system *(72 atoms)* takes less than 1 second!

```
cd 3_hpb_pm3
/cm/shared/apps/pc2/ORCA/orca_4_2_1_linux_x86-64_openmpi314/orca input.inp
cd ..
```

Again, you should visualize the resulting trajectory with VMD *(in a new local terminal, not on the compute node)*:

```
cd ~/ORCA/3_hpb_xtb  vmd trajectory.xyz
```

Finally, we will execute the SANscript example program which I have shown in my talk. Go to the directory, have a look at the input file, try to guess what the program might do/print, and then execute it via ORCA.

```
cd 4_sanscript
/cm/shared/apps/pc2/ORCA/orca_4_2_1_linux_x86-64_openmpi314/orca input.inp
cd ..
```

If you are interested in more details on the SANscript language *(which is still in development)*, you can visit the following page:

`https://brehm-research.de/sanscript`

If you have questions on the ORCA MD module, the ORCA Forum is a good place to ask:

`https://orcaforum.kofo.mpg.de`

Computing IR / Raman / VCD / ROA Spectra with CP2k and TRAVIS *(Feb 2020)*

Martin Brehm
Martin-Luther-Universität Halle–Wittenberg
`https://brehm-research.de/`

0. Introduction

In this exercise, we will compute the full set of vibrational spectra (IR, Raman, VCD, ROA) from a CP2k AIMD simulation of one (*R*)-propylene oxide molecule in vacuum. Normally, this takes several days/weeks of computer time. Here, we have only a few hours and CPU cores per user. Some severe simplications had to be introduced to the computational setup, so that the calculations will finish in time. This is only to get a hand on the workflow; the spectra computed here will be of no scientific use (sorry). If you want „real“ spectra, feel free to redo all steps with adequate parameters *(as I have shown in my talk)*. You find some nicely predicted bulk phase spectra in some of our articles in the literature *(see chapter 6 in this document)*.

Apart from CP2k, we will use the TRAVIS program package. It is freely available in the internet on

```
http://www.travis-analyzer.de/
```

All features of TRAVIS used in this tutorial are included in the public version, so you can repeat everything at your institute or at home if you are interested.

An in-depth written form of this tutorial *(with more details and explanations)* can be found here:

```
https://brehm-research.de/spectroscopy
```

There are also links pointing to this tutorial on the TRAVIS page as well as on the CP2k website *(see „HOWTOs“)*.

For this tutorial, we will not submit jobs to the queueing system to run. We will rather directly log in to a compute node and call our programs there, which is much more convenient for this kind of task. With the following command, you start an interactive SLURM job on a compute node *(the last letter is a lowercase „L“)*. If you already have an open interactive session *(e.g., from the ORCA MD tutorial)*, just continue to use it, don't open another one.

```
srun -A hpc-lco-usrtr --pty bash -l
```

As a first part of today's exercise, we will compile TRAVIS from the source code. This is a very easy task, as TRAVIS does not require any external libraries or dependencies (just a C++ compiler). First, create a „`travis`“ subdirectory in your home and enter it:

```
mkdir ~/travis
cd ~/travis
```

Copy the TRAVIS source code *(directly as downloaded from the TRAVIS homepage)* to this directory and unpack it:

```
cp ~/Desktop/Winter_School/Tutorials/TRAVIS/travis-src-190101.tar.gz .
gunzip travis-src-190101.tar.gz
tar -xvf travis-src-190101.tar
```

Now simply enter the directory, unload any modules *(to use the system C++ compiler)*, and type „make“. TRAVIS will be compiled *(ignore the warning messages which might appear)*. You will find the resulting executable in the „exe“ subdirectory.

```
cd travis-src-190101
module reset
make
```

For being able to call TRAVIS without the full path, add the directory to the system search path:

```
export PATH=~/travis/travis-src-190101/exe:$PATH
```

Now, you can load the module for CP2k *(it is important to do this after compiling TRAVIS)*:

```
module load winterschool2020/cp2k
```

Apart from that, we have to tell CP2k to use only one CPU core per process (i.e., no OpenMP):

```
export OMP_NUM_THREADS=1
```

Finally, before beginning with the actual tutorial, we copy the prepared directory structure with the input files to the home directory:

```
cp -r ~/Desktop/Winter_School/Tutorials/TRAVIS/Spectroscopy ~
cd ~/Spectroscopy
```

1. Computing the Trajectory

In the „Spectroscopy“ directory which we just copied and entered, you will find several sub-directories (`traj, nofield, exp, eyp, ezp, spectra`) inside of this directory. We will start by computing the AIMD trajectory.

```
cd traj
```

Have a look at the CP2k input file there (`propox.inp`) and convince yourself that it runs a standard AIMD simulation *(with very sloppy computational parameters; though)*. Then, run the calculation:

```
mpirun -np 16 cp2k.psmp propox.inp | tee log.out
```

It will run for a few inutes. Then, you will find a file „`propox-pos-1.xyz`“, which contains the trajectory of the system.

2. Computing the Electron Densities

With the trajectory which we just computed, we now produce the electron density CUBE trajectories. We need to do this without external electric field, and with external field applied along the X, Y, and Z axes. This leads to four calculations in total. We will run them all simultaneously, each on eight CPU cores. The & sign and the end of the lines send these jobs to the background, so that you can continue to work in the console.

```
cd ..
cd nofield
nohup mpirun -np 8 cp2k.psmp propox.inp > log.out &
cd ..
cd exp
nohup mpirun -np 8 cp2k.psmp propox.inp > log.out &
cd ..
cd eyp
nohup mpirun -np 8 cp2k.psmp propox.inp > log.out &
cd ..
cd ezp
nohup mpirun -np 8 cp2k.psmp propox.inp > log.out &
cd ..
```

These calculations should run for less than 30 minutes. While they run, you can have a look to the „`.ener`“ files to see the progress. The calculations are finished when the .ener file has 1024 lines *(as we run 1024 simulation steps)*. We obtained one `.cube` file per trajectory, which contains the electron density on a grid in every simulation step.

Just for completeness, I added the writing out of Wannier centers to the four input files. I recommended in my talk to not perform Wannier localization. However, for such a small gas phase system, this is really quick, so it does not hurt.

3. Compressing the CUBE Trajectories

This optional step can help to drastically reduce the size of the electron density data. The routines to compress trajectories into BQB files can either be used directly with TRAVIS, or with a stand-alone tool called „`bqbtool`“, which you can find on my homepage (`https://brehm-research.de/bqb`). Here, we will stick with the former option.

```
cd nofield
nohup travis compress voltraj result.cube result.bqb > log2.out &
cd ..
cd exp
nohup travis compress voltraj result.cube result.bqb > log2.out &
cd ..
cd eyp
nohup travis compress voltraj result.cube result.bqb > log2.out &
cd ..
cd ezp
nohup travis compress voltraj result.cube result.bqb > log2.out &
cd ..
```

This will run for around 5 minutes *(for all four trajectories in parallel)*. You can monitor the progress in the „`travis.log`“ files. We obtained `.bqb` files, which are much smaller than the `.cube` files *(reduced from approx. 3 GiB to less than 100 MiB in our case!)*, but still contain the same amount of information *(lossless compression)*.

4. Performing Voronoi Integration and Solving Current PDE

Now, we need to obtain the atomic and molecular electromagnetic moments, which are required for the spectra. We use TRAVIS to solve the classical current PDE and to perform the Voronoi integration of the electron density. Normally, TRAVIS is an interactive program, which asks questions to the user; but in this case, I prepared some TRAVIS input files („`integrate.txt`“) to speed up the process.

```
cd nofield
nohup travis -p result.bqb -i integrate.txt > log3.out &
cd ..
cd exp
nohup travis -p result.bqb -i integrate.txt > log3.out &
cd ..
cd eyp
nohup travis -p result.bqb -i integrate.txt > log3.out &
cd ..
cd ezp
nohup travis -p result.bqb -i integrate.txt > log3.out &
cd ..
```

This will run for around 10 minutes *(for all four trajectories in parallel)*. You can monitor the progress in the „`travis.log`“ files. For each trajectory, we will obtain an electromagnetic property file („`properties.emp`“), which contains the electric dipole vector, quadrupole tensor, and magnetic dipole vector *(among other properties)* per atom.

Apart from the properties.emp file, you also obtained a „`properties.csv`“ text file, where you can have a look at all the electric and magnetic moments per atom :-)

5. Compute the Spectra

Now, we finally have all the data to compute the spectra. We need to supply the four `.emp` files from the last chapter. We start TRAVIS with the field-free property file:

```
cd spectra
travis ../nofield/properties.emp
```

As mentioned above, TRAVIS is an interactive program and asks questions. Please give the following answers in that order *(and read the questions before)*. The lines starting with exclamation marks are just comments which indicate the question that was asked. If the answer line is empty, simply press enter; this will choose the default answer to that question, which is indicated in square brackets.

```
! Use the advanced mode until the analysis selection menu (y/n)? [no]

! Use these values (y) or enter different values (n)? [yes]

! Update cell geometry in every time step (y) or use fixed cell (n)? [yes]

! Create images of the structural formulas (y/n)? [no]

! Accept these molecules (y) or change something (n)? [yes]

! Which functions to compute (comma separated)?
spec
! Use the advanced mode for the main part (y/n)? [no]

! Use two-pass mode (y/n)? [yes]

! Is this a gathering run (y) or the analyzing run (n)? [no]

! Enter the length of one trajectory time step in fs: [0.5]

! In which time step to start processing the trajectory? [1]

! How many time steps to use (from this position on)? [all]

! Use every n-th time step from the trajectory? [1]

! Spectra to compute for this observation (comma separated):
ir,raman,vcd,roa
! Observe molecules of type C3H6O for this spectrum (y/n)? [yes]

! Which molecules of type C3H6O to observe (e.g. 1,5-7)? [all]

! Enter the resolution of the correlation functions: [511]
512
! Calculate spectrum up to which wave number (cm^-1)? [5000.00 cm^-1]

! Calculate scattering for which wave number (cm^-1)? [20000.0]

! Calculate scattering cross section for which temperature (K)? [350.0]

! Correct spectrum for a certain simulation temperature (y/n)? [yes]

! Enter the simulation temperature (K): [350.0]

! Add another observation (y/n)? [no]

! Use central finite differences (y) or one-sided differences (n)? [no]

! Enter electric field strength for field trajectories (in a.u.): [5.0E-3]

! Enter core charge for atom type C: [4.0]

! Enter core charge for atom type H: [1.0]

! Enter core charge for atom type O: [6.0]

! Enter data file name for field X:
../exp/properties.emp
! Enter data file name for field Y:
../eyp/properties.emp
! Enter data file name for field Z:
../ezp/properties.emp
```

After you answered all questions, TRAVIS will run a few seconds, and then you will have all spectra as .csv text files. You can plot these spectra with any plotting program of your choice (gnuplot, xmgrace, ...). The first column always contains the wave number in cm^{-1}, the second column contains the spectral intensity.

Have a look at the following five spectra:

```
power_spectrum.csv
spectrum_ir_C3H6O.csv
spectrum_raman_ortho_ct_C3H6O.csv
spectrum_vcd_ct_C3H6O.csv
spectrum_roa_90deg_ortho_ct_C3H6O.csv
```

As explained, they will have very low quality due to the limited computational resources and time in this exercise. Redo the whole procedure with longer simulations to obtain more realistic spectra *(see the written tutorial on my homepage for reasonable suggestions of system size and simulation time)*.

6. Literature – Please Cite!

"TRAVIS - A Free Analyzer and Visualizer for Monte Carlo and Molecular Dynamics Trajectories", M. Brehm, B. Kirchner; *J. Chem. Inf. Model.* **2011**, *51 (8)*, pp 2007-2023.

"An Efficient Lossless Compression Algorithm for Trajectories of Atom Positions and Volumetric Data", M. Brehm, M. Thomas; *J. Chem. Inf. Model.* **2018**, *submitted*.

"Computing vibrational spectra from ab initio molecular dynamics", M. Thomas, M. Brehm, R. Fligg, P. Voehringer, B. Kirchner; *Phys. Chem. Chem. Phys.* **2013**, *15*, pp 6608-6622.

"Simulating the vibrational spectra of ionic liquid systems: 1-Ethyl-3-methylimidazolium acetate and its mixtures", M. Thomas, M. Brehm, O. Holloczki, Z. Kelemen, L. Nyulaszi, T. Pasinszki, B. Kirchner; *J. Chem. Phys.* **2014**, *141*, 024510.

"Voronoi dipole moments for the simulation of bulk phase vibrational spectra", M. Thomas, M. Brehm, B. Kirchner; *Phys. Chem. Chem. Phys.* **2015**, *17*, pp 3207-3213.

"Classical Magnetic Dipole Moments for the Simulation of Vibrational Circular Dichroism by Ab Initio Molecular Dynamics", M. Thomas, B. Kirchner; *J. Phys. Chem. Lett.* **2016**, *7*, pp 509-513.

"Computing Bulk Phase Raman Optical Activity Spectra from ab initio Molecular Dynamics Simulations", M. Brehm, M. Thomas; *J. Phys. Chem. Lett.* **2017**, *8 (14)*, pp 3409-3414.

"Computing Bulk Phase Resonance Raman Spectra from ab Initio Molecular Dynamics and Real-Time TDDFT", M. Brehm, M. Thomas; *J. Chem. Theory Comput.* **2019**, *15 (7)*, 3901–3905.