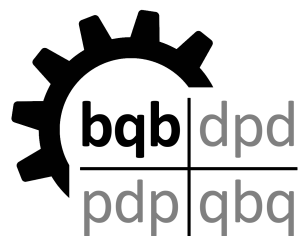
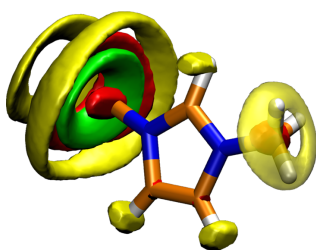
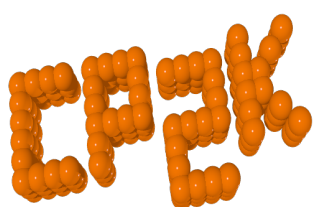


Computing Bulk Phase Vibrational Spectra^{*} with CP2k and TRAVIS (2018)

^{*} IR, Raman, VCD, ROA

Tutorial & User's Guide

Martin Brehm & Martin Thomas, 2018



<https://www.cp2k.org>

<http://www.travis-analyzer.de>

<https://brehm-research.de>

Document Version: 22.10.2018

Contents

1	Introduction	1
2	Download and Setup the Software	3
3	How to Cite	5
4	Workflow	
4.1	Prepare a Simulation Cell	7
4.2	Compute the AIMD Trajectory	9
4.3	Write the Electron Density Data	13
4.4	Compute the Electromagnetic Properties	17
4.5	Compute the Spectra	19
5	Advanced Topics	
5.1	Wannier Centers vs. Voronoi Integration	23
A	Appendix	
A.1	Initial Configuration	31
A.2	AIMD Massive Pre-Equilibration	36
A.3	AIMD Non-Massive Pre-Equilibration	38
A.4	AIMD Production Run	40
A.5	Total Electron Density Computation	42
A.6	Compute Properties for IR, Raman, VCD, and ROA Spectrum	44
A.7	Compute Properties for IR & Raman Spectrum	45
A.8	Compute IR, Raman, VCD, and ROA Spectrum	46
A.9	Compute IR & Raman Spectrum	47
B	Literature	48

1. Introduction

This guide explains how to compute vibrational spectra of bulk phase systems based on *ab initio* molecular dynamics (AIMD) simulations. This sentence already requires some clarification:

- In this document, the term “bulk phase system” refers to liquid or solid systems with periodic boundary conditions. Spectra of interfaces or adsorbed surface layers can be obtained by using a surface slab as periodic simulation cell. Gas phase systems (*i. e.*, molecules or clusters embedded in vacuum) can of course also be handled, but there are probably other methods which suit better in such cases.
- The term “vibrational spectra” stands for the set of infrared (IR), Raman, vibrational circular dichroism (VCD), and Raman optical activity (ROA) spectra of the system. The latter two are the chiral variants of the first two techniques. All four spectra can be computed in one run, using the same methodology. Of course it is not always desirable to compute all four spectra. For example, it would not make sense to compute the ROA and VCD spectra of a system which does not contain chiral molecules.

Some more general information:

- The approach presented here relies on AIMD, and therefore includes “automatic” conformer sampling. As we are **not** working in the static-harmonic approximation, the spectra computed here include some anharmonic effects, such as overtones and combination bands (this statement is sometimes contested by other scientists; however, it has been shown in literature¹ that these effects appear in spectra from MD as soon as the potential energy surface is non-harmonic, which is the case in AIMD for all modes). Furthermore, line shapes are often well reproduced, including effects such as line broadening due to hydrogen bonding and other solvent influences. For some example spectra, see the literature.²⁻⁶
- All steps in this guide are performed by free software. You will not have to buy any license. However, please make sure to cite the articles which correspond to the programs or modules if you use or publish any results obtained by this approach – see chapter 3.
- You will require a GNU/Linux computer to run all the calculations (any other Unix-like system might probably also work). As the whole system is treated quantum chemically, the computations are quite demanding. However, you don’t need a large compute cluster. We recommend to use at least 64 CPU cores for average-sized systems. On 64 cores, it will take around 3 weeks to compute the set of IR, Raman, VCD, and ROA spectra for one given system. If you only need IR, it will be somewhat faster. Compact desktop systems with 64 CPU cores (four CPU sockets) are available for less than 10 000 € today—this should be affordable for almost every working group at a university. If you are planning to buy such a system, make sure to order at least 4 GiB of RAM per CPU core.

1. Introduction

As an addition to this guide, you may want to look at the presentation slides of a talk which M.B. was giving at a CP2k workshop in Paderborn (August 2018). There, some of the scientific background information is described and illustrated. You find a PDF file with all the slides on the following website:

<https://brehm-research.de/spectroscopy>

If you have questions on this tutorial, please feel free to use the contact link on the above website. However, if the questions are CP2k specific, please contact the CP2k Google group instead to get support. Neither M.B. nor M.T. are part of the CP2k developer team.

Impressum

This tutorial was created by Martin Brehm using \LaTeX .

It shall be only used for teaching and educational purposes.

The fonts used are Bitstream Charter and Bitstream Vera Sans.

2. Download and Setup the Software

As mentioned in the introduction, you should install the software listed below on a GNU/Linux machine (probably any other Unix-like machine might also work). If you only have a Windows computer, use a free virtual machine with GNU/Linux as a guest system (*we can recommend Oracle VirtualBox*). Don't bother with compiling CP2k on Cygwin or even a native Windows build... It would probably take you many extra hours to get it to run with MPI parallelization.

CP2k

CP2k⁷ is a powerful free software package for running *ab initio* molecular dynamics (AIMD) simulations of periodic bulk phase systems (among many other features). You can find the source code as well as some tutorials (*including this document*) and a reference manual on the homepage:

<https://www.cp2k.org>

However, as CP2k is a large package and requires many external libraries (MPI, BLAS, LAPACK, ScaLAPACK, FFTW, ...), it is not a trivial task to compile an optimized parallel CP2k executable (although it has become significantly easier during the last 10 years (-:)). If your institute does not offer a pre-compiled CP2k executable, you can try to follow the steps in this guide:

<https://www.cp2k.org/howto:compile>

If you compiled your own CP2k executable, make sure to run the CP2k test suite ("regtests"). Many test jobs will be executed, and the results will be compared to the correct reference results. This step is absolutely vital! If something goes wrong with compiling, it can easily happen that the jobs seem to run fine but the results are wrong.

If you encounter some persistent problems with your CP2k setup and are out of ideas, you may contact the CP2k Google group, where users and developers offer help:

<https://groups.google.com/d/forum/cp2k>

After you succeeded in obtaining a working CP2k executable, make sure to put it to your system search path.

It is not so important to have the most recent CP2k version running. The features we will use for computing spectra are implemented and stable since many years...

TRAVIS

TRAVIS⁸ is a free software package which helps you to analyze and visualize your simulation trajectories (therefore the name: **T**rajectory **A**nalyzer and **V**isualizer). It offers many different functions (one- and multi-dimensional histograms, mean square displacement, lifetimes and autocorrelation factors, structure factors, order parameters, ...). Here, we are only going to consider the spectroscopic functions. You can find the source code as well as a quickstart manual and some additional material (*such as this document*) on the homepage:

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

Luckily, compiling TRAVIS is supposed to be easy. It does not require any external libraries, only a working C++ compiler (and GNU Make). We recommend the GCC compiler suite, which should be present on all modern GNU/Linux systems. Just download the latest source code version, extract the .tar.gz archive, enter the directory, and type “make”. After some minutes of compiling, an executable file should have appeared in the “exe” sub-directory. That’s all. Make sure to put this executable into your system search path.

Please use the TRAVIS version from October 2018 (or more recent, if there is any) for this tutorial.

bqbtool

The bqb file format⁹ is a storage format for the lossless compression of simulation trajectories and volumetric data such as electron density Cube files. The corresponding software which compresses and decompresses trajectories is the bqbtool. As we are going to work with very large volumetric trajectories in this guide, it will be very helpful to compress those. TRAVIS can directly read and analyze the compressed files in bqb format. You can download the bqbtool together with a user manual on the homepage:

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

Similar to TRAVIS, compiling the bqbtool should be easy, as it does not require any external libraries. Just download the latest source code version, extract the .tar.gz archive, enter the directory, and type “make”. After some minutes of compiling, an executable file should have appeared in the “exe” sub-directory. Make sure to put this executable into your system search path.

3. How to Cite

As all programs used in this guide are free software, you will not have to pay any license fees to compute spectra. The only thing which the software developers would like in return are your citations, which are very much valued. This chapter gives a list of articles which you should cite if you compute spectra by the approach described here. This may seem like a lot of citations. However, if any piece of work would be missing, the whole computation would not be possible. Please be fair and cite them all.

CP2k

The CP2k program package:⁷

- J. Hutter, M. Iannuzzi, F. Schiffmann, and J. VandeVondele: “CP2k: Atomistic Simulations of Condensed Matter Systems”, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, **2014**, 4, pp. 15–25.

The Quickstep algorithm:¹⁰

- J. VandeVondele, M. Krack, F. Mohamed, M. Parrinello, T. Chassaing, and J. Hutter: “Quickstep: Fast and Accurate Density Functional Calculations using a Mixed Gaussian and Plane Waves Approach”, *Comput. Phys. Commun.*, **2005**, 167, pp. 103–128.

The orbital transformation (“OT”) method:¹¹

- J. VandeVondele and J. Hutter: “An Efficient Orbital Transformation Method for Electronic Structure Calculations”, *J. Chem. Phys.*, **2003**, 118, pp. 4365–4369.

MOLOPT basis sets:¹²

- J. VandeVondele and J. Hutter: “Gaussian Basis Sets for Accurate Calculations on Molecular Systems in Gas and Condensed Phases”, *J. Chem. Phys.*, **2007**, 127, p. 114105.

GTH pseudopotentials:^{13,14}

- S. Goedecker, M. Teter, and J. Hutter: “Separable Dual-Space Gaussian Pseudopotentials”, *Phys. Rev. B*, **1996**, 54, pp. 1703–1710.
- C. Hartwigsen, S. Goedecker, and J. Hutter: “Relativistic Separable Dual-Space Gaussian Pseudopotentials from H to Rn”, *Phys. Rev. B*, **1998**, 58, pp. 3641–3662.

BLYP exchange–correlation functional (if you used it):^{15,16}

- A. Becke: “Density-Functional Exchange-Energy Approximation with Correct Asymptotic Behavior”, *Phys. Rev. A*, **1988**, 38, pp. 3098–3100.
- C. Lee, W. Yang, and R. Parr: “Development of the Colle-Salvetti Correlation-Energy Formula into a Functional of the Electron Density”, *Phys. Rev. B*, **1988**, 37, pp. 785–789.

3. How to Cite

Stefan Grimme's D3 dispersion correction (if you used it):¹⁷

- S. Grimme, J. Antony, S. Ehrlich, and S. Krieg: "A Consistent and Accurate ab initio Parametrization of Density Functional Dispersion Correction (DFT-D) for the 94 Elements H-Pu", *J. Chem. Phys.*, **2010**, *132*, p. 154104.

Nosé-Hoover thermostat (if you used it):^{18,19}

- S. Nosé: "A Unified Formulation of the Constant Temperature Molecular Dynamics Methods", *J. Chem. Phys.*, **1984**, *81*, pp. 511–519.
- S. Nosé: "A Molecular Dynamics Method for Simulations in the Canonical Ensemble", *Mol. Phys.*, **1984**, *52*, pp. 255–268.

TRAVIS / bqbttool

The TRAVIS main article:⁸

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

Computing vibrational spectra with TRAVIS:^{2,4}

- M. Thomas, M. Brehm, R. Fligg, P. Vöhringer, and B. Kirchner: "Computing Vibrational Spectra from ab initio Molecular Dynamics", *Phys. Chem. Chem. Phys.*, **2013**, *15*, pp. 6608–6622.
- M. Thomas, M. Brehm, and B. Kirchner: "Voronoi Dipole Moments for the Simulation of Bulk Phase Vibrational Spectra", *Phys. Chem. Chem. Phys.*, **2015**, *17*, pp. 3207–3213.

Trajectory compression in the bqbt format (if you used it):⁹

- M. Brehm and M. Thomas: "An Efficient Lossless Compression Algorithm for Trajectories of Atom Positions and Volumetric Data", *J. Chem. Inf. Model.*, **2018**, (in press), doi 10.1021/acs.jcim.8b00501.

In case you computed VCD or ROA spectra:⁵

- M. Thomas and B. Kirchner: "Classical Magnetic Dipole Moments for the Simulation of Vibrational Circular Dichroism by ab initio Molecular Dynamics", *J. Phys. Chem. Lett.*, **2016**, *7* (3), pp. 509–513.

In case you computed ROA spectra:⁶

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

If you used normal mode analysis:³

- M. Thomas, M. Brehm, O. Hollóczki, Z. Kelemen, L. Nyulászi, T. Pasinszki, and B. Kirchner: "Simulating the Vibrational Spectra of Ionic Liquid Systems: 1-Ethyl-3-methylimidazolium Acetate and its Mixtures", *J. Chem. Phys.*, **2014**, *141* (2), p. 024510.

4. Workflow

This chapter will give a step-by-step description on how to compute vibrational spectra. We will do this at the example of a simulation of liquid (*R*)-propylene oxide which has been published before,⁶ but feel free to simulate the system of your personal interest.

All the input files are placed in the Appendix of this document. **Don't** copy/paste the data from the Appendix of this PDF document. You will lose all indentation, because PDF does not store space/tab characters. You can find a link to an archive file with all input files from the Appendix on

<https://brehm-research.de/spectroscopy>

4.1. Prepare a Simulation Cell

After you decided which system you are planning to investigate, you need to choose a system size for the simulation. This is always a compromise—small systems will lead to poor sampling of the phase space, and large systems will require large amounts of computational resources. We will give a few guidelines here:

- For organic molecules, the system size should be in the range of 300 to 1000 atoms in the simulation cell. For small molecules (*e. g.*, methanol or (*R*)-propylene oxide), 300 atoms will be fine. For larger molecules, you will need a slightly larger system to get better sampling.
- If you want to compute a spectrum of a solute molecule in solvent, don't use an excessively large solvent box. The spectrum is influenced most by the first two solvation layers around the solute. A large number of solvent molecules will only slow down your calculation.
- The approach presented here is currently limited to molecular substances. We did most of our testing with disordered liquid phase systems, where the method gives good results. Organic crystals are probably also going to work, as well as some macro-molecules such as small peptides in solution. However, you currently can't compute the spectrum of a simple inorganic solid (such as a metal or a sodium chloride crystal). That being said, you might still use a metal or sodium chloride surface in your simulation box to adsorb an organic molecule. You can obtain the spectrum of the organic molecule under the influence of the surface. You just can't yet compute spectra of the inorganic parts themselves. We are working to lift this limitation soon.

After you decided on a system and a reasonable system size, you need to create the initial configuration of the periodic box. For solid crystalline systems (*e. g.*, organic crystals), it is a good choice to start from an experimental crystal structure. One problem here could be the pressure:

4. Workflow

As the electron structure method will certainly yield a different equilibrium cell vector than the experimentally determined one, you will probably run your simulation under very high (or low) pressure if you simply use the experimental cell vector. This might impact the spectrum. You might want to run a cell optimization with CP2k to relax the unit cell before you start the actual AIMD run. Please refer to the CP2k manual.

For liquid phase systems and solutions, this step is even more involved. For many liquids (including “simple” ones such as water*), it is known that they need quite a long time to equilibrate. However, AIMD simulations only run for several dozens of picoseconds, which is by far too short to equilibrate any liquid phase system. Therefore, it is mandatory to run a force field simulation for pre-equilibration of the liquid. To the best of our knowledge, there is no true black-box solution here... There are several free force field molecular dynamics codes available (we recommend LAMMPS, but Gromacs is also good). There also exist good general-purpose all-atom force fields, *e. g.*, OPLS-AA and AMBER. However, the details are out of the scope of this guide. Just use your favorite force field simulation code together with a suitable force field for your specific system to pre-equilibrate your initial configuration. You will only need one snapshot (typically the last frame of the force field simulation) as a starting configuration.

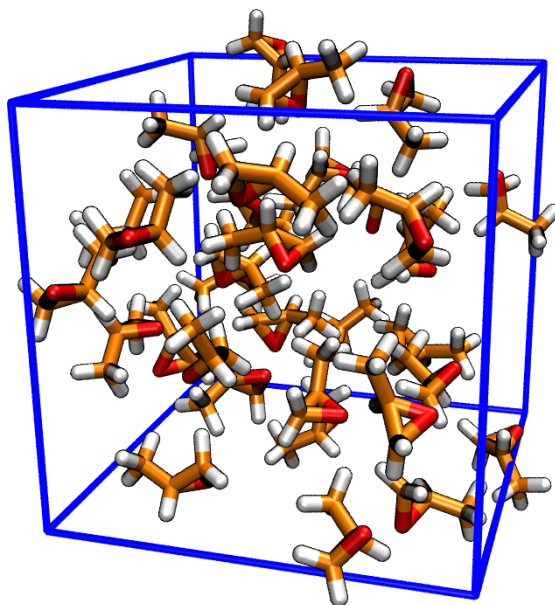


Figure 1: The liquid (*R*)-propylene oxide simulation cell⁶ which we will use in this tutorial (*see Appendix A.1*).

Save the initial configuration in XMol XYZ file format – this is a simple text format which contains one line per atom. Each line starts with the element label of the atom, followed by the three Cartesian atom coordinates in Angstrom. An XYZ file starts with two header lines, where the first one contains the total atom count, and the second one is a comment line. Please cut away these

*certainly one of the most complex liquids ever encountered

4. Workflow

two lines. Your text file shall contain only the lines with the element labels and atom coordinates in Angstrom. Of course, you also need to remember the cell vector which you used / found during pre-equilibration. It is not important that all coordinates are inside the primary image of the periodic cell. They will be wrapped automatically by CP2k.

In this tutorial, we will use the liquid (*R*)-propylene oxide simulation cell which we recently used to compute the first bulk phase ROA spectrum,⁶ see Figure 1. It contains 32 molecules of (*R*)-propylene oxide at a density of 0.816 g cm^{-3} , and was pre-equilibrated at 300 K with LAMMPS. The text file with the initial configuration can be found in Appendix A.1. The simulation cell is cubic, with all three edges having a length of 15.5835 Angstrom. **Don't** copy/paste the data from the Appendix of this PDF document. You will lose all indentation, because PDF does not store space/tab characters. You can find a link to an archive file with all input files from the Appendix on

<https://brehm-research.de/spectroscopy>

4.2. Compute the AIMD Trajectory

With the pre-equilibrated initial configuration from the last step, you can now start the actual AIMD runs. Before the production run, you should perform two additional short equilibration runs (one with massive thermostats and one with global thermostating). This will be described below. First, we would like to give some general remarks on the choice of the computational methods and algorithms:

- You need to decide for an electron structure method to describe the potential energy surface of the system. Density functional theory (DFT), and in particular generalized gradient approximation (GGA) functionals of DFT, offer a very good compromise between computational time and accuracy. In this tutorial, we use the BLYP functional.^{15,16} Feel free to use other functionals (another popular choice is PBE). You can even use hybrid functionals if you can stand the increased computational effort. However, this is not discussed in this tutorial.
- CP2k uses atom-centered Gaussian basis sets and plane waves simultaneously (as described in the Quickstep article¹⁰). You need to decide for a basis set. We recommend the MOLOPT basis sets, as these are specifically optimized for good performance in bulk phase systems and a relatively small basis set superposition error (BSSE). In this tutorial, basis functions from the “DZVP-MOLOPT-SR-GTH” set are used, which have proven to work well for the computation of spectra. You can go up to TZVP instead, but the change in results is often only minor. Apart from the atom-centered basis, you need to decide on a plane wave cutoff. The values we use in this guide (“CUTOFF 350, REL_CUTOFF 40”) are good starting points. A tutorial on how to converge these cutoffs can be found on the CP2k homepage.
- CP2k uses pseudopotentials to describe the core electrons of the atoms (this is a requirement in most plane wave electron structure codes). The number after the letter “q” in the name of

4. Workflow

the pseudopotential corresponds to the number of electrons which are explicitly present (not in the pseudopotential). This depends, of course, on the element, and also needs to match the basis set. Have a look into the files “BASIS_MOLOPT” and “POTENTIAL” in the CP2k data directory to find matching combinations.

- If your system contains hydrogen atoms, we generally recommend to use an integration timestep of 0.5 fs. This is in contrast to the recommendation of 1.0 fs, which is often used in force field MD simulations with flexible hydrogen atoms. Apart from the bad conservation of total energy this would introduce, there is also another difference in AIMD: Smaller time steps lead to smaller changes in the wave function after each step, and the SCF procedure will need less cycles to converge. Therefore, a given number of steps with $\Delta t = 1.0$ fs will be not much faster than double the number of steps with $\Delta t = 0.5$ fs. If your system does not contain hydrogen atoms, you can go to a larger time step. As a rule of thumb, the required time step approximately scales with the square root of the mass of the lightest element present in your system. If your lightest element is, *e. g.*, carbon, you might try a time step of $\Delta t = 0.5 \text{ fs} \cdot \sqrt{12} \approx 1.7 \text{ fs}$.
- The settings in the input file given below are designed for well-converging systems of organic molecules. If you have problems with SCF convergence, try to replace the “&SCF” section with the following block (at the cost of higher computational time):

```
&SCF
  SCF_GUESS ATOMIC
  MAX_SCF 45
  &OT
    PRECONDITIONER FULL_SINGLE_INVERSE
    MINIMIZER CG
    LINESEARCH 3PNT
  &END
  EPS_SCF 1.0E-6
  &OUTER_SCF
    MAX_SCF 20
    EPS_SCF 1.0E-6
  &END
  &PRINT
    &RESTART
      &EACH
        MD 0
      &END EACH
    &END
  &END
&END SCF
```

- If your system contains semiconducting or metallic parts (*i. e.*, if the band gap in any region of the system is lower than several eV), the OT method will no longer reliably work. In these cases, you need to switch to a classical diagonalization/mixing approach. Please replace the “&SCF” section with the following input. You will need to adapt the “ADDED_MOS” value to the system size to ensure that no warning message occurs. You also might need to adapt the “ALPHA” parameter (lower values are more stable but converge slower).

4. Workflow

```
&SCF
SCF_GUESS ATOMIC
MAX_SCF 50
&MIXING
METHOD BROYDEN_MIXING
ALPHA 0.7
&END
&SMEAR
METHOD FERMI_DIRAC
ELECTRONIC_TEMPERATURE 500
&END SMEAR
ADDED_MOS 10 # adapt this until warning disappears
EPS_SCF 1.0E-6
&PRINT
&RESTART
&EACH
MD 0
&END EACH
&END
&END
&END SCF
```

A brief description of some of the keywords in the input file can be found in the presentation slides of a talk which M.B. was giving at a CP2k workshop in Paderborn (August 2018), starting from slide 160. You find a PDF file with all the slides on the following website:

<https://brehm-research.de/spectroscopy>

Massive Equilibration

Now you can setup the first AIMD run. Use the CP2k input file given in Appendix A.2. Make sure that the “@INCLUDE” command points to your initial configuration file which you have created in the last step. For every element which is present in your simulation cell, add a corresponding “&KIND” section with information on the basis set and pseudopotential that shall be used for this element. Have a look into the files “BASIS_MOLOPT” and “POTENTIAL” in the CP2k data directory to find possible candidates. Also, make sure to adapt the input file as described above if your system has problems with SCF convergence or contains metallic/semiconducting parts. You need to carry over all these modifications to all other CP2k input files which you will use for the next steps below.

After your input file is ready, start the first equilibration phase. Depending on your software setup, this will involve a command similar to

```
mpirun -np 16 cp2k.popt input.txt > output.txt
```

where 16 is the number of CPU cores to use for this run. This first equilibration will run with massive thermostats for 2000 time steps (1.0 ps).

Global Equilibration

After this has finished, the second equilibration phase will follow, this time using global thermostating. You will need to perform some changes in your CP2k input file (add a “&EXT_RESTART” section with the correct restart file name and “RESTART_THERMOSTAT .FALSE.” flag, remove the “MASSIVE” keyword, change thermostat time constant from 10 to 100 fs, switch the “SCF_GUESS” from “ATOMIC” to “RESTART”, change the “STEPS” value). You can find the modified input file in Appendix A.3. As described above, make sure to carry over all changes which you introduced to the first CP2k input file above.

The global equilibration will run for 5000 time steps (2.5 ps). After this, you are ready to start the production run.

Production Run

In this part, we will start the AIMD production run of the system. Here, the question might arise why we only compute the position trajectory now, and subsequently re-traverse this file to obtain the volumetric electron density data along the trajectory (see section 4.3). This has several advantages:

- If the full AIMD trajectory is already present, we can trivially parallelize the computation of the electron density data (and subsequent compression of the files) by splitting the trajectory into parts.
- Separating the production run from the electron density computation allows to use different electron structure methods for both tasks. You could, *e. g.*, use a method based on perturbation theory (such as MP2) to perform the production run. Perturbative methods do not possess well-defined total electron densities... Subsequently, you could re-traverse the trajectory to compute the volumetric electron density data by some other method which produces total electron density. The production run decides on the dynamics of your system, which determines all frequencies in your spectrum. The electron density run, on the other hand, determines the spectral intensities. This enables to use different levels of theory for frequencies and intensities, which adds some degree of flexibility.
- If you want to compute Raman or ROA spectra, you need to re-traverse the AIMD trajectory anyway for the calculations with external electric field.

However, if you are not convinced by these arguments, please feel free to modify this tutorial and write out volumetric electron density data already in the production run.

For the production run, again, you need to perform some modifications on the CP2k input file (remove the “RESTART_THERMOSTAT .FALSE.” flag and change the “STEPS” value). You find the corresponding CP2k input file in Appendix A.4. For well-converged bulk phase spectra, you will require around 60 000 time steps (30 ps), depending on the system size. However, you do not need to perform the full production run in one go. On computer systems with queuing system,

4. Workflow

the maximum run time of jobs is often limited to a few days, which would require to split the production run into several parts. This can be easily done by reducing the “STEPS” value in the input file, and simply re-executing the same input file multiple times, until the desired total step number is reached. Due to the “&EXT_RESTART” section, CP2k will always restart the calculation at the point where it stopped in the last run.

There is one more important points to consider here: CP2k always appends the position trajectory data to a file named “xxx-pos-1.xyz”, where “xxx” stands for the project name which you have specified in the input file. For the subsequent parts of the tutorial, you will need the trajectory file of the production run only, without the parts from the two equilibration runs. Therefore, you will need to rename/delete/move away the trajectory file before starting the production run.

After you finished this part of the tutorial, you have obtained a trajectory file from the production run in XYZ file format. You can visualize this file with the VMD program package to ensure that no unexpected events happened in your simulation.

4.3. Write the Electron Density Data

In this part of the tutorial, we will compute the volumetric electron density along the production run trajectory. Depending on which spectra you want to compute, there exist several cases here:

1. If you **only** want to compute the IR and VCD spectrum, it is sufficient to compute electron densities along the production run trajectory. If you also want to compute the Raman or ROA spectrum, you need to perform three additional computations with external electric fields applied along the X, Y, and Z direction. The former one will be termed as the “field-free run”, the latter ones will be referred to as the “field runs”.
2. If you **only** want to compute the IR or Raman spectrum, it is sufficient to compute the electron density every 4.0 fs (*i. e.*, every 8^{th} step with $\Delta t = 0.5$ fs). If you also want to compute the VCD or ROA spectrum, you need to compute the electron density in each simulation step.

Volumetric electron densities are huge data sets and lead to large file sizes. CP2k writes the total electron density in Gaussian Cube file format. For the system considered in this tutorial, the grid resolution is $160 \times 160 \times 160$, which leads to a file size of 52 MiB per frame. If we want to compute the full set of spectra, we need to store the electron density for all 60 000 frames, and in addition also with the three external electric field directions. This leads to a total raw data amount of $52 \text{ MiB} \times 60\,000 \times 4 = 11.9 \text{ TiB}$! To avoid this, it is very important to cut down the computation into small parts, and individually compress/process the resulting files instead of simply accumulating them.

Luckily, we already have the full trajectory file of the production run, and can split up the computation into many smaller blocks. CP2k offers a so-called “reference trajectory” operating mode,

4. Workflow

where it simply reads in an existing trajectory file, and re-computes the electron structure of each snapshot. This is what we will do in this section. CP2k offers to select the first and last frame which shall be considered from the reference trajectory, such that all individual instances of CP2k can work on the full production run trajectory, and only these numbers in the input file need to be adapted. You can run all these individual blocks in parallel (*in separate directories!*) if you have enough CPU cores or compute nodes.

You need to decide on a block length for the individual processing blocks of the production run trajectory. In this tutorial, we will use a block length of 2000 frames. Then, each individual electron density Cube file will have a size of 102 GiB, which is still acceptable. There is another important issue: If you also want to compute VCD or ROA spectra, the blocks into which you cut your trajectory need to have an **overlap of exactly 2 frames!** This is because the differential equation for the electric current involves the derivative of electron density with respect to simulation time, which is computed via central finite differences, such that you will “lose” two frames in every block.

In the CP2k input, the keywords “FIRST_SNAPSHOT” and “LAST_SNAPSHOT” refer to the first and last frame that shall be considered from the reference trajectory (counting starts at 1). “STRIDE” specifies to take only every n -th frame within this range. “STEPS” is the total number of frames to be considered. The numbers need to match. If you, e. g., specify “FIRST_SNAPSHOT 100”, “LAST_SNAPSHOT 130”, and “STRIDE 5”, then you need to use “STEPS 7” – otherwise, an error message will appear.

This might all seem a little confusing at first sight. For the case of this tutorial, we present two worked out examples here:

Example 1: IR, Raman, VCD, ROA

If you want the full set of IR, Raman, VCD, and ROA spectra, you need to use a stride of 1, and need to ensure an overlap of 2 frames between the blocks. Your first input file would use the values

```
&MD
  ENSEMBLE REFTRAJ
  STEPS 2002
  &REFTRAJ
    EVAL_ENERGY_FORCES
    FIRST_SNAPSHOT 1
    LAST_SNAPSHOT 2002
    STRIDE 1
    TRAJ_FILE_NAME Prop0x-pos-1.xyz
  &END REFTRAJ
&END MD
```


4. Workflow

while your second input file would use

```
&MD
  ENSEMBLE REFTRAJ
  STEPS 2002
  &REFTRAJ
    EVAL_ENERGY_FORCES
    FIRST_SNAPSHOT 2001
    LAST_SNAPSHOT 4002
    STRIDE 1
    TRAJ_FILE_NAME Prop0x-pos-1.xyz
  &END REFTRAJ
&END MD
```

and so on. You will need 30 such input files per electric field direction to cover the full production trajectory of 60 000 frames. If you only want IR and VCD spectra, you are done. If you also want Raman or ROA spectra, you need to repeat this process for each of the three external electric field directions.

Example 2: Only IR and Raman

If you only need the IR and Raman spectra, you can use a stride of 8, and do not need any overlap between the blocks. Your first input file would use the values

```
&MD
  ENSEMBLE REFTRAJ
  STEPS 2000
  &REFTRAJ
    EVAL_ENERGY_FORCES
    FIRST_SNAPSHOT 1
    LAST_SNAPSHOT 15993 # 1 + 1999*8
    STRIDE 8
    TRAJ_FILE_NAME Prop0x-pos-1.xyz
  &END REFTRAJ
&END MD
```

while your second input file would use

```
&MD
  ENSEMBLE REFTRAJ
  STEPS 2000
  &REFTRAJ
    EVAL_ENERGY_FORCES
    FIRST_SNAPSHOT 16001
    LAST_SNAPSHOT 31993 # 16001 + 1999*8
    STRIDE 8
    TRAJ_FILE_NAME Prop0x-pos-1.xyz
  &END REFTRAJ
&END MD
```

and so on. You will only need 4 such input files to cover the whole trajectory length of 60 000 frames. If you only want IR spectra, you are done. If you also want Raman spectra, you need to repeat this process for each of the three external electric field directions.

4. Workflow

You can find an exemplary input file for electron density calculation in Appendix A.5. You need to perform several modifications to the input:

- First, you need to decide on an electron structure method which you would like to use to compute the electron densities (often, this will be the same method you have used to run the production trajectory). Make sure to transfer all system-specific modifications (basis sets, convergence problems, etc.) also to this file.
- As just described, you need to insert the correct numbers in the “&MD” section of each input file. Make sure that “TRAJ_FILE_NAME” points to your production trajectory file. As you need to enter the correct numbers for all parts of your computation, you should probably use a small script which prepares all the input files.
- The “&PERIODIC_EFIELD” section controls the external electric field which you will need to apply for Raman and ROA spectra. The intensity is specified in atomic units. In the tutorial, we use a value of $5 \cdot 10^{-3}$ a.u. = $2.57 \cdot 10^9$ V m⁻¹, which we found to be a good compromise. Too small values lead to noisy spectra, whereas too large values might leave the linear regime of electronic polarizability. The “POLARISATION” keyword specifies the vector of the field direction. To perform the field runs in X, Y, and Z direction, the respective vector component should be 1.0, and the remaining two components should be zero. To perform the field-free run, please remove the whole “&PERIODIC_EFIELD” section.
- The “&PRINT” section within &SCF controls the output of the electron density Cube files. It is important to have “STRIDE 1 1 1” here to write the full grid. As even for small parts of the trajectory the electron densities will still lead to large file sizes, make sure that your specified “FILENAME” is located on a file system which is both fast and large. We recommend the local scratch file system of the compute node (in optimal case even a SSD). In particular, you should not write this file to a network mounted directory, as this would put very heavy loads onto your network...

Each of the individual block calculations will produce an electron density Cube file (in this tutorial, each such files contains 2000 or 2002 frames and has a size of 102 GiB). You now need to decide how to proceed with these files. There are two options:

- **Preferred Method:** If you want to keep the electron density data (to do further analyses afterwards, or as an insurance if something goes wrong during the computation), you can losslessly compress the Cube files to the bqb format (means “**B**inary **C**ube”) by using the bqbtool.⁹ For $\Delta t = 0.5$ fs, a typical compression ratio of 35:1 is reached. This means that the whole set of electron density data will be reduced from 11.9 TiB (see above) to just 350 GiB, which is quite acceptable. As mentioned, the compression is lossless, so you do not lose any data. The bqbtool is not parallelized (*i. e.*, runs on only one CPU core) and takes around 2 hours to compress a Cube file with 2000 frames (for the grid dimensions of this tutorial). As TRAVIS can directly work with the compressed files in bqb format, you will not need to

4. Workflow

decompress them later. After each CP2k run, you can directly compress the Cube file by an additional line in your input script, and then delete the original Cube file. Use the following command to compress a Cube file with bqbtool:

```
bqbtool compress voltraj input.cube output.bqb
```

- **On-the-fly Method:** If you do not want to use the electron density data later on, you can decide to directly compute the electric and magnetic moments from the Cube files of each block calculation, and then delete the Cube files. This can be done via a scripted call to TRAVIS after each block calculation. The details are described in section 4.4. This will save some time, because the compression of the volumetric data is no longer required. However, this saving is almost negligible, because the largest fraction of computational time is due to the AIMD simulations and electron density calculations. Please be aware that you will have lost everything if some problems or errors occur in the calculation of the electromagnetic properties. Furthermore, you will not be able to perform other analyses on the electron density data later.

4.4. Compute the Electromagnetic Properties

In this section, we will cover the process of computing the electric and magnetic moments for each atom from volumetric electron density data. This computation is performed with TRAVIS⁸ by using our Voronoi integration approach.⁴ The advantages of the Voronoi integration scheme over using Wannier centers are discussed in section 5.1.

The computation of electromagnetic properties is performed separately for each of the block calculations from the last section. If you decided to use the preferred method (compression to bqb file format), you will have to run one TRAVIS instance for each of your bqb files. If you used the on-the-fly approach instead, you should run one TRAVIS instance for each Cube file directly after the CP2k calculation has finished, and then delete the Cube file. The output of each such TRAVIS run will be an .emp file (ElectroMagnetic Properties), which contains the electronic charge, dipole vector, and quadrupole tensor for each atom in each frame as computed from the Voronoi integration.⁴ If you want to compute VCD or ROA spectra, the file additionally contains the magnetic dipole moment per atom, obtained by solving a partial differential equation for the electric current.⁵ The file name will always be “properties.emp” – make sure to run each instance in an individual directory, or the files will be overwritten.

The Voronoi integration is fast, it takes less than one second per volumetric frame. The computation of the electromagnetic properties from a 2000 frame Cube (or bqb) file therefore takes less than one hour on one CPU core (TRAVIS is not parallelized). If you want to compute VCD or ROA spectra, you need to solve the partial differential equation for the electric current, which is slower (around 20 seconds per frame). In this case, it will take around 12 hours to obtain the .emp file from a 2000 frame Cube/bqb file. However, as this process can be trivially parallelized, and each

4. Workflow

TRAVIS instance only occupies one CPU core, you can use all your available cores to speed up this step.

TRAVIS is an interactive program and asks questions on the command line. However, you can also supply an input file which contains all the answers to the questions. For automated batch processing of many files, at it shall be done here, this is clearly the preferred way. The lines starting with an exclamation mark in the input file are comment lines which are ignored by TRAVIS (*e. g.*, the question which was asked). The lines without exclamation marks are the actual answers. You can find some relevant TRAVIS input files in the Appendix. However, it regularly happens that questions are modified or added in TRAVIS. Therefore, we recommend not to just use the input files. Instead, please run TRAVIS one time in interactive mode (without input file) and answer all the questions as given in the tutorial input file (where you see all question/answer pairs). TRAVIS will automatically create an input file by the name “input.txt”, which you can then re-use for batch processing. This ensures that all the answers are in the correct positions.

To open a .bqb or .cube file with TRAVIS in interactive mode, please use

```
travis -p file.bqb
```

To run TRAVIS in batch mode instead (if you already prepared your input file), use

```
travis -p file.bqb -i input.txt
```

The example TRAVIS input file to compute electromagnetic properties for the full set of IR, Raman, VCD, and ROA spectra (including differential equation for electric current) can be found in Appendix A.6. The example input to compute the properties for IR and Raman only (without electric currents) follows in Appendix A.7. The values in square brackets after the questions are the default answers. If no answer was given to a question (*i. e.*, empty line), then the default answer was chosen.

Please note that you have to specify the correct cell vector from the CP2k runs when TRAVIS asks for it. If you used a non-orthorhombic cell, you have to activate the “advanced mode” in the beginning of the TRAVIS run by answering “yes” to the first question. Furthermore, you have to specify the core charges for your elements. As CP2k uses pseudopotentials, these core charges are **not** equivalent to the atomic numbers (*except for hydrogen and helium*), but rather correspond to the number of electrons explicitly included for each element. This is the number after the “q” letter in the pseudopotential name (*see above*).

At the end of this section, you will have obtained a sequence of .emp files (or if Raman/ROA is required, four such sequences for the one field-free and the three field runs), which contain the electromagnetic properties of each atom in each frame. You finally need to concatenate the files within each sequence. Technically, the .emp files are also bqb files (the bqb format can store many kinds of data). The bqb format offers a flexible method for concatenation of several files. There is

4. Workflow

the concept of a so-called “list file”. It is a simple text file with the file extension “.blist”, which contains the character string “BLIST” in its first line (completely uppercase, no spaces before or afterwards). In each subsequent line, there follows the file name of a “normal” bqb (or emp) file. You can both use absolute and relative path names here. Relative path names are relative to the position of the .blist file. This list file can be used transparently as if it were a normal bqb file by all programs which support the bqb format. For more details on bqb list files, please refer to the user manual of the bqbtool:

<https://brehm-research.de/bqb>

At the end of this section, we will give a short example of how to concatenate .emp files. Imagine you were performing the computation of electromagnetic properties in the four sub-directories “part1”, “part2”, “part3”, and “part4”. In the parent directory, you create a text file with the name “properties.blist” and the following content:

```
BLIST
part1/properties.emp
part2/properties.emp
part3/properties.emp
part4/properties.emp
```

The newly created text file now behaves as if it would be a long .emp file with the content of all four individual .emp files in the specified order (*make sure to not mix up the order*).

4.5. Compute the Spectra

At this point of the tutorial, you have one (if only IR/VCD is required) or four (if also ROA/Raman is required) .blist files which contain links to the individual .emp files with the electromagnetic properties. The last step is to compute the spectra out of this data with TRAVIS. Fortunately, this is the fastest and easiest part. It will only take around ten minutes (depending on system size and simulation length).

As this task needs to be executed only once, there is no reason to use the batch mode of TRAVIS. Just start TRAVIS with the field-free .blist file in interactive mode, and answer the questions as found in the example input files below:

```
travis -p properties.blist
```

You can find an example TRAVIS input file for the computation of the full set of IR, Raman, VCD, and ROA spectra in Appendix A.8. If only IR and Raman spectra shall be computed, and the calculation of electromagnetic properties was performed every 4.0 fs, a corresponding input can be found in Appendix A.9. As explained above, the values in square brackets after the questions are

4. Workflow

the default values. If no value is given as an answer, the default value was chosen.

There are a few things to note here:

- Always start TRAVIS with the field-free trajectory. If you want to compute Raman or ROA spectra, TRAVIS will ask you for the paths to the `.blist` files with applied electric field during the run. Make sure to enter the correct relative or absolute path to the files. Don't mix up the field directions.
- As you can see, TRAVIS asks which molecule types (and which individual molecules of each type) you want to include into the computation of the spectrum. This is a very powerful feature. If you simulated a mixture, you can decide to compute and plot a spectrum which shows only one component of the mixture. This helps to assign spectral features to individual components of such a system. Or if you simulated a molecule which is adsorbed on a metal surface, you will probably want to compute the spectrum of the molecule only, omitting contributions from the surface.
- TRAVIS asks for the resolution of the correlation functions. This is a tradeoff – low resolution leads to blurred spectra, while high resolution values lead to very noisy spectra. A good compromise is to use a value in the range of 2.0 ps. Please note that this number has to be entered in time steps rather than physical units of time. For a trajectory stride of 0.5 fs in the `.emp` file (*i. e.*, every time step processed), we recommend a correlation resolution of 4096. If the `.emp` trajectory stride is 4.0 fs (*i. e.*, every 8th frame processed), a resolution of 512 is a good choice. These are also the default values in both cases, and both values correspond to approximately 2.0 ps of physical time. We chose powers of 2 to slightly accelerate the Fourier transformations. The spectra which we recently published in literature⁶ have been computed with these default values. Please feel free to experiment with higher or lower resolutions. As this step is not so time-consuming, it is always worth trying it out.
- If you requested Raman or ROA spectra, you need to enter the electric field strength that was used for the field runs (in atomic units, as specified in the CP2k input file). It is important that this number matches the actual number you used in your CP2k simulations. Otherwise, the polarizabilities from finite differences will be wrong.
- In the case of Raman and ROA spectra, you obtain several spectra (such as “spectrum_roa_90deg_ortho”) for each observation. These correspond to different measurement setups (scattering angle, etc.). Please refer to the literature.^{2,6}

To close this chapter, we present the Raman and ROA spectra which we computed from the (*R*)-propylene oxide system discussed here and recently published in literature.⁶

Figure 2 shows the computed Raman spectrum of the tutorial system together with an experimental spectrum.²⁰ It is visible that the peak frequencies do not exactly match. This is a well-known problem from DFT-based AIMD simulations, and not related to our approach for computing spectra.

4. Workflow

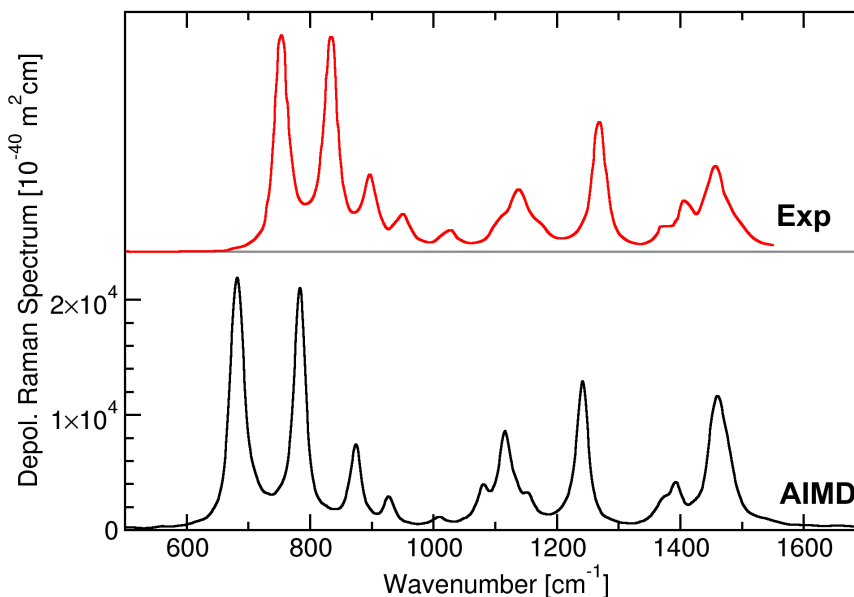


Figure 2: Raman spectrum of liquid (*R*)-propylene oxide as computed in this tutorial and published in literature,⁶ together with experimental spectrum from literature.²⁰

The frequencies would match a lot better if MP2 dynamics would have been used instead of DFT (however, we could not afford that). The good point about that is that there exists a systematic and canonical route for improvement if more computer time becomes available.

Apart from this limitation, the overall agreement between simulation and experiment is very good. We would like to stress that this is a pure *ab initio* spectrum, without any empirical corrections (frequency shift, line broadening, etc.) to better match the experiment.

In Figure 3, the computed ROA spectrum is presented together with an experimental spectrum.²¹ The peak shift discussed for the Raman spectrum is obviously still present (it is a “feature” of the underlying simulation, and not related to the computation of the spectrum). Apart from that, also here the overall agreement between computed and experimental spectrum is very good, including the signs and magnitudes of the bands. Also here, we note again that this is a pure *ab initio* spectrum, without any empirical corrections (frequency shift, line broadening, etc.) to better match the experiment. In ROA spectroscopy, the signs are often used to determine the absolute configuration of an unknown compound. We conclude that the quality of the computed spectrum is by far sufficient to determine the absolute configuration of the compound under investigation.

4. Workflow

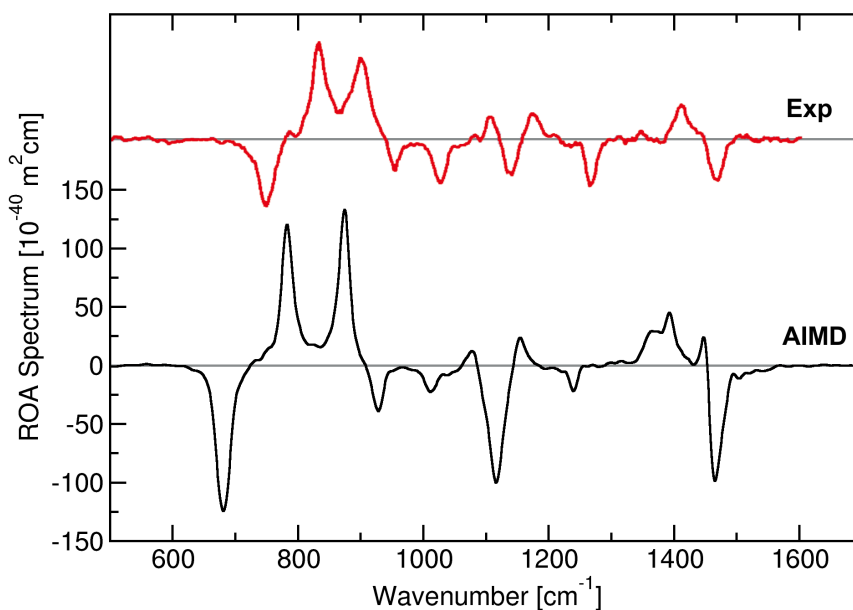


Figure 3: ROA spectrum of liquid (*R*)-propylene oxide as computed in this tutorial and published in literature,⁶ together with experimental spectrum from literature.²¹

The spectrum shown in Figure 3 was the first predicted ROA spectrum of a periodic bulk phase system that appeared in literature. However, with the availability of free software tools and tutorials on how to compute such spectra, we hope that there soon will be a lot more in literature from all parts of the world :-). If you closely followed the tutorial, you will hopefully have obtained similar spectra at this point.

5. Advanced Topics

5.1. Wannier Centers vs. Voronoi Integration

Most of the bulk phase spectra in literature which have been computed from *ab initio* molecular dynamics simulations are based on Wannier localization.²² This method offers a simple way to compute molecular dipole moments by assigning Wannier centers to molecules, and also works for systems with periodic boundary conditions. However, Wannier localization also comes with several disadvantages, which shall be discussed here. Many parts of the discussion are based on our article from 2015.⁴

Computational Time

To perform Wannier localization, it is required to find a unitary transformation of the molecular orbitals which minimizes the spread (*defined by some functional, often simply the variance*) of the molecular orbitals. For large systems with many thousand occupied molecular orbitals, this is tedious. A widely used standard method to find such a transformation is to use Jacobi diagonalization, an iterative algorithm which (hopefully) converges to the solution. This is also implemented in CP2k. However, CP2k uses a modern algorithm by default, which is called the “Crazy Angle algorithm”, and typically converges much faster than the Jacobi diagonalization.

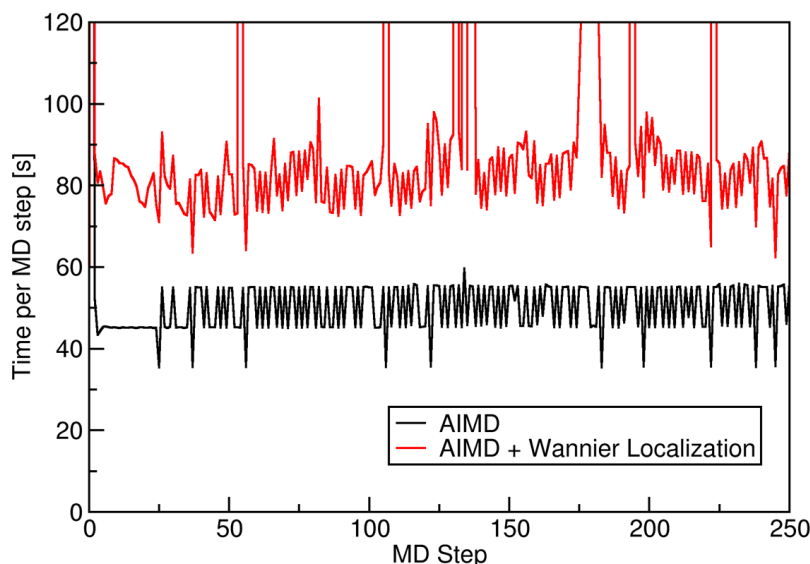


Figure 4: Frame times of a 936 atom bulk phase simulation with (red) and without (black) Wannier localization.

Let’s look at an example calculation now. We consider a bulk phase ionic liquid simulation that

5. Advanced Topics

has been investigated in literature before.²³ The system contains 36 ion pairs of [EMIm][OAc] (1-ethyl-3-methylimidazolium acetate), which corresponds to 936 atoms, in a cubic cell of 21.21 Angstrom edge length. There are 2500 electrons and 8100 basis functions (DZVP-MOLOPT-SR). This is quite a standard system size for AIMD – one simulation time step takes less than one minute on a single 16-core node.

Figure 4 shows the frame times of this simulation in the first 250 steps. The black curve depicts the frame times of a standard AIMD without Wannier localization. The average frame time is 47.9 s, and the variance is rather small – all as expected. The red curve depicts the frame times with Wannier localization switched on. The difference between the black and the red curve is therefore the time which the Wannier localization requires. It can be seen that the frame times are around 30 seconds higher now, which is the expected overhead of the Wannier localization. However, there are some points where the red curve exceeds the ordinate axis scale.

To have a closer look on this, we present the same plot with logarithmic ordinate scale in Figure 5. Now it can be seen that the frame time goes up to 6700 seconds in certain frames! A closer look to the log files reveals that these are exactly the cases where the Crazy angle algorithm does not converge. Jacobi diagonalization is used as a fallback algorithm then. And this can obviously take up to 2 hours per frame (where the AIMD step takes less than one minute!).

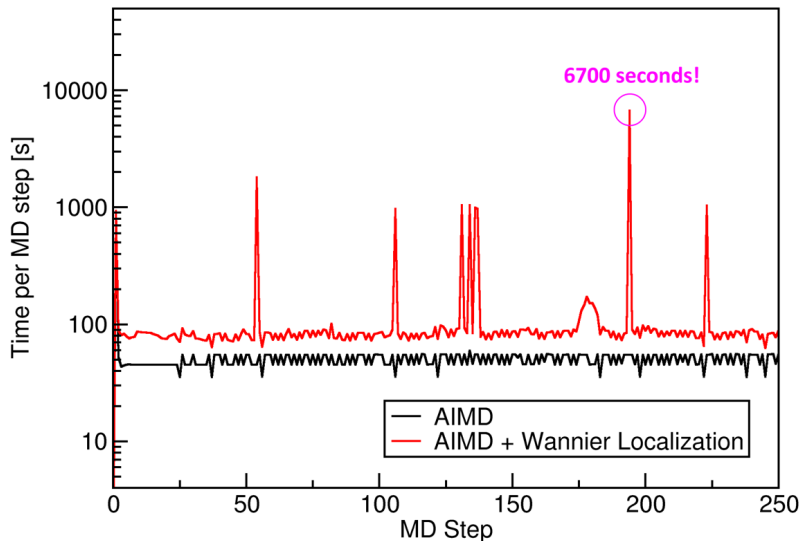


Figure 5: Frame times of a 936 atom bulk phase simulation with (red) and without (black) Wannier localization. Note the logarithmic ordinate scale.

To put it in a more statistical statement: The average frame time without Wannier localization is 47.9 s, whereas the average frame time with Wannier localization switched on is 139.3 s. This means that on average 91.4 s per frame are exclusively spent on the Wannier localization, which is 65% of the total computer time.

5. Advanced Topics

Even if the Crazy angle algorithm *would* converge in every frame, the time overhead would still be 30 s per frame for the localization, which still would be 39% of the total computer time. This is still not easy to accept.

To make matters worse, it even happens that the Jacobi fallback algorithm does not converge. In these frames, there will be no Wannier localization output written to trajectory. In other words, it happens that certain frames are simply missing from the trajectory, which is a very bad thing if you want to compute autocorrelation functions.

Artificial Bands

It is well known that Wannier localization localizes the electrons of aromatic rings into a single bond/double bond pattern. For a benzene molecule, this looks as shown in Figure 6, where the black spheres represent the Wannier centers. So far, nothing bad.



Figure 6: Wannier localization of an aromatic ring system leads to a single bond/double bond pattern.⁴

Some years ago, we computed and published an IR spectrum of a bulk phase simulation of liquid benzene.⁴ The IR spectrum which is based on Wannier localization (*dashed line*) and based on the total electron density (*solid line*) are shown in Figure 7. Both spectra look quite similar; however, there is an additional peak at around 1300 cm^{-1} in the Wannier-based spectrum that does not appear in the other spectrum, and also has no analogon in the experimental spectrum. It took us a while to figure out what happened here...

In benzene, there exists a normal mode which deforms it into cyclohexatriene. Along that normal mode, every second C–C bond is elongated, and the other three C–C bonds are shortened. This mode is visualized in Figure 8. From symmetry considerations, it is obvious that this normal mode does not change the total dipole moment of the molecule, and therefore is not visible in an infrared spectrum.

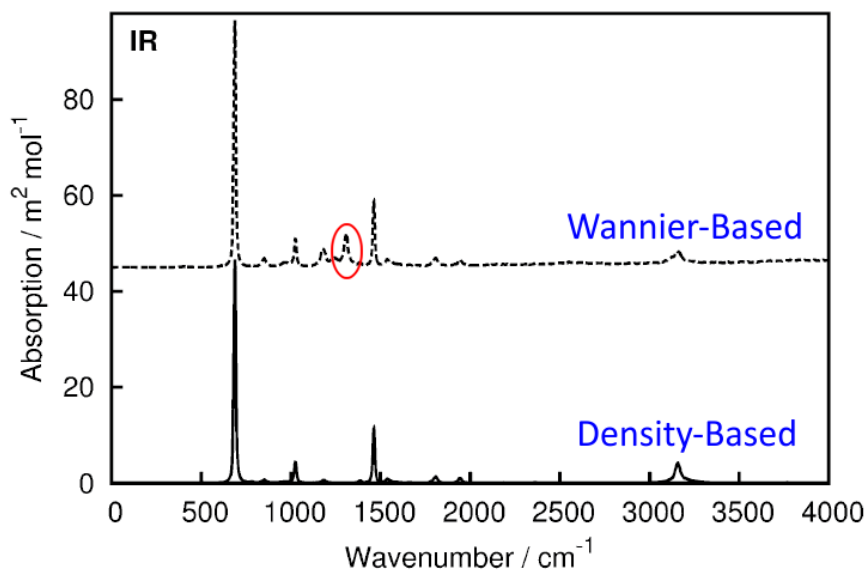


Figure 7: Computed IR spectra of a liquid benzene system, based on Wannier localization (*dashed line*) and on total electron density (*solid line*). Artificial peak from Wannier localization indicated by red ellipse.⁴

However, we found that the artificial peak in the Wannier-based infrared spectrum exactly matches the frequency of this mode. This means that in the Wannier-based formalism, there actually is an artificial change of molecular dipole along this mode. The reason is that the Wannier localization localizes the aromatic electrons into a single bond/double bond pattern (*see Figure 6*). The “double bonds” will of course preferably be localized to the positions where the C–C bonds are shorter on average. If the normal mode from Figure 8 is active, the single bond/double bond pattern of the Wannier localization flips with the frequency of the normal mode. This flipping should not lead to a change in molecular dipole moment in theory; however, due to slight charge transfer effects between molecules, it leads to small abrupt jumps in the molecular dipole moment. These jumps become prominently visible as artificial spectral feature, see red ellipse in Figure 7.

If such an effect already happens in one of the simplest cyclic aromatic systems, it can easily be imagined how many artificial peaks the Wannier localization will probably cause in more complicated aromatic molecules, such as naphthalene or anthracene.

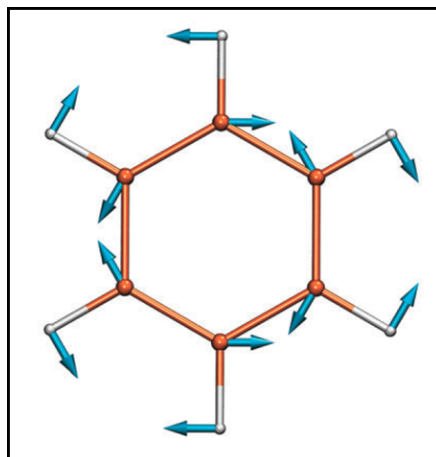


Figure 8: A normal mode of benzene, which deforms it into cyclohexatriene. This mode is not infrared active, as it does not change the total dipole moment (*obvious by symmetry considerations*).⁴

No Polarizabilities for Aromatic Systems

If one tries to compute Raman spectra of aromatic systems via the Wannier localization formalism, it becomes even worse (at least as long finite differences w.r.t electric field are used to compute the polarizability). When applying an external electric field to the single bond/double bond localization pattern of a benzene molecule, it is very probable that the single bond/double bond pattern will flip. As discussed above, this leads to a small artificial jump in the molecular dipole moment. Together with the finite differences scheme for the polarizability, this yields almost only noise as polarizability.

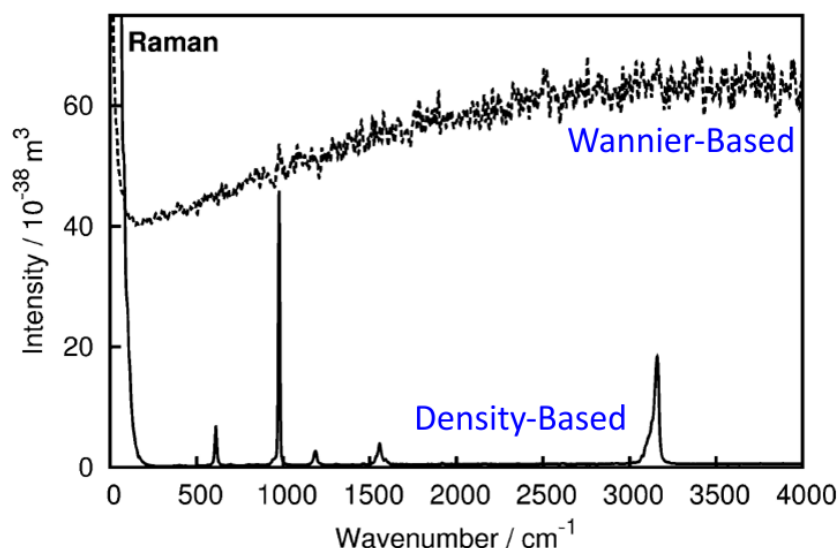


Figure 9: Computed Raman spectrum of a liquid benzene system, based on Wannier localization (*dashed line*) and on total electron density (*solid line*). The Wannier-based spectrum contains almost only noise.⁴

5. Advanced Topics

Figure 9 depicts computed Raman spectra from both the Wannier localization based approach (*dashed line*) and the total density based approach (*solid line*). While the latter one looks qualitatively right, the former one contains almost only noise, which is due to the flipping of the bond pattern that was just described. We conclude that finite differences of the electric field coupled to Wannier-based dipole moments cannot be used to compute Raman spectra of bulk phase systems at all. This claim is corroborated by the fact that we did not find a single such spectrum in literature. You can, of course, use some kind of perturbation theory instead to obtain the polarizability without electric field finite differences. However, this usually makes the whole method a lot more computationally expensive.

No Higher Multipoles

This is a very short argument. Wannier centers are good to define molecular dipole moments, but can't be used to obtain proper molecular quadrupole moments to the best of our knowledge. However, the computation of ROA spectra requires molecular quadrupole moments. This means that a Wannier-based approach will never be able to compute bulk phase ROA spectra (except for the case in which the quadrupole term is simply neglected for the spectrum...).

Voronoi Integration

Above, we discussed four issues with Wannier localization, which are all quite severe with respect to general applicability as a black-box method for spectroscopy. To overcome these problems, we suggested in 2015 to use a scheme which we called "Voronoi integration".⁴ It is based on a radical Voronoi tessellation²⁴ of the simulation cell, using all the atoms as Voronoi sites. In contrast to the standard Voronoi tessellation the radical version assigns radii to the sites which influence the position of the separating planes between adjacent sites. We recommend to simply use van-der-Waals atom radii, as this has shown to produce good spectra.⁴ However, a thorough investigation on the influence of the radii on the spectra is still ongoing.

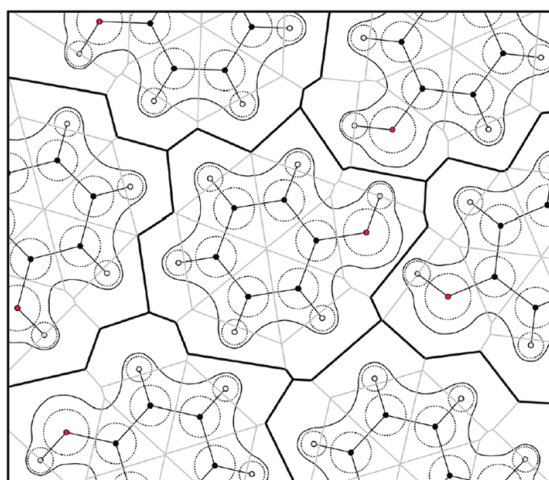


Figure 10: Schematic illustration of the radical Voronoi tessellation leading to atomic and molecular Voronoi cells at the example of phenol.⁴

5. Advanced Topics

The radical Voronoi tessellation yields Voronoi polyhedra for each atom. In the boundaries of each such polyhedron, we integrate the total electron density, and use the classical expressions for charge distributions to compute electric dipole and quadrupole moments. We sum up these moments for all atoms within a molecule (taking into account the gauge dependency, and using the molecular center of mass as new coordinate origin), and obtain the molecular dipole and quadrupole moments, which can be used to compute spectra. A schematic illustration is shown in Figure 10. Our implementation of the Voronoi integration in TRAVIS is very fast, it requires less than one second of CPU time for a typical bulk phase system as discussed above.

In Figure 11, computed infrared (*upper panel*) and Raman (*lower panel*) spectra of liquid methanol are presented, based on Wannier localization (*dashed lines*) and Voronoi integration (*solid lines*). It can be seen that both approaches lead to very similar spectra. We conclude that for systems which are not problematic for the Wannier-based approach, both methods yield spectra of the same quality.⁴

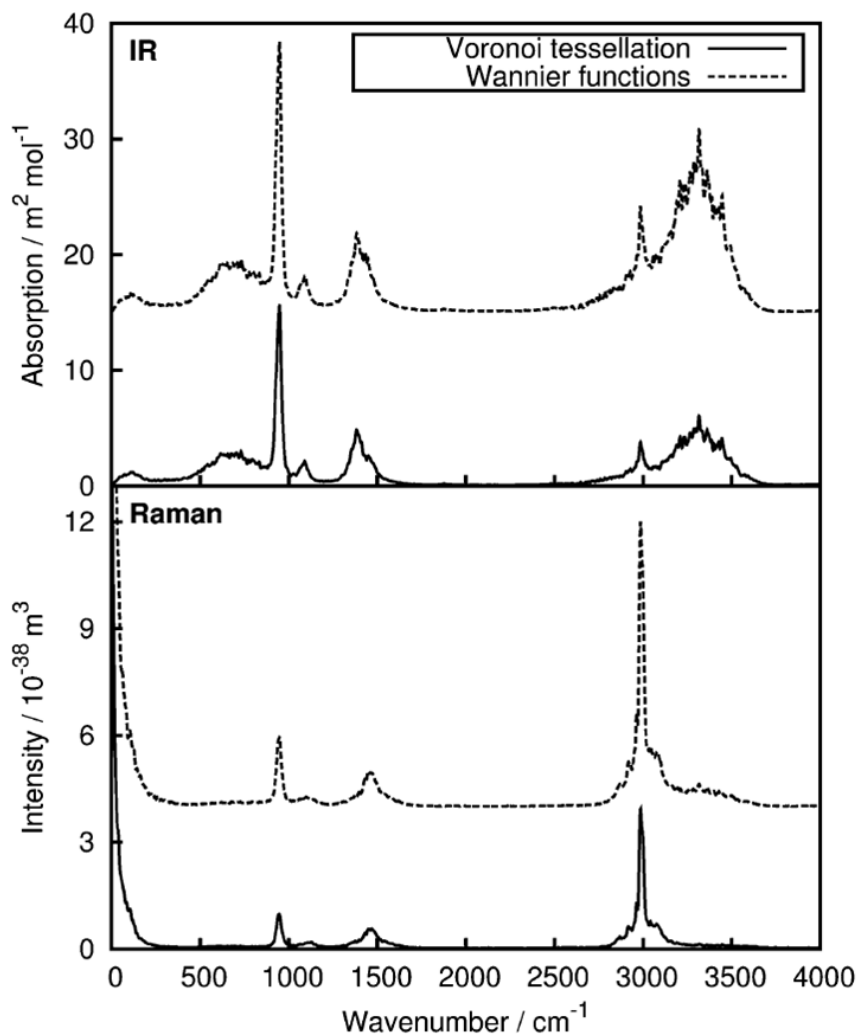


Figure 11: Computed infrared (*upper panel*) and Raman (*lower panel*) spectra of liquid methanol, based on Wannier localization (*dashed lines*) and Voronoi integration (*solid lines*).⁴

Conclusion

When comparing the Wannier localization based approach and the Voronoi integration based approach for computing bulk phase spectra, we come to the following conclusions:

- **Computational Time:** Wannier localization easily consumes half of the total computer time of the simulation for large systems, and is not guaranteed to converge at all. Voronoi integration is very fast (less than one second) and does not involve any iterative steps (no problems with convergence). By switching to Voronoi integration, a lot of computational time is saved.
- **Aromatic Systems:** We demonstrated that Wannier localization can lead to artificial bands in the infrared spectrum of aromatic systems.⁴ Raman spectra of aromatic systems cannot be computed based on Wannier localization. Voronoi integration, on the other hand, works fine for aromatic systems – both for infrared and Raman spectra.
- **Higher Multipole Moments:** Wannier localization can't yield higher multipole moments above electric dipole moment, which are required, *e. g.*, to compute ROA spectra. Voronoi integration can yield such multipole moments, and already has been successfully employed to compute bulk phase ROA spectra.⁶

Therefore, we focus only on the approach based on Voronoi integration in this tutorial. We cannot recommend the usage of the Wannier localization based formalism if bulk phase spectra shall be computed. However, if you insist on computing spectra from Wannier centers, the legacy modules for infrared and Raman spectra are still present in TRAVIS. They can be activated by entering “ir” or “raman” in the main function menu. For all new applications, we strongly recommend to use the new spectroscopy module (activated by entering “spec” in the main function menu), which can compute infrared, Raman, VCD, and ROA spectra in one pass.

A. Appendix

Don't copy/paste these inputs from the PDF file. You will lose indentation (PDF does not store space/tab characters). You can find a link to an archive file with all input files listed here on

<https://brehm-research.de/spectroscopy>

A.1. Initial Configuration

C	12.68275	13.74062	4.25993
C	13.36425	13.60423	5.64515
C	12.77375	15.13262	3.60356
H	12.03575	12.89945	3.98373
H	13.21955	12.64995	6.17170
H	14.00607	14.42730	6.00475
H	12.89855	14.99806	2.52848
H	13.65967	15.65698	4.04944
H	11.86365	15.67447	3.82368
O	13.96366	13.22895	4.39113
C	9.67815	3.70047	2.94346
C	8.28065	3.22864	2.91378
C	10.23645	4.34700	4.23257
H	10.40295	3.63150	2.11681
H	7.93485	2.89434	1.94317
H	7.55445	3.28709	3.76207
H	10.04165	3.67047	5.06233
H	9.86795	5.28645	4.48398
H	11.32305	4.38818	4.07262
O	9.39085	2.37844	3.17158
C	2.23615	11.69625	5.88035
C	1.87931	13.17735	6.03659
C	1.20275	10.71745	5.95949
H	3.28575	11.50005	5.91580
H	2.71135	13.83855	6.09445
H	0.82584	13.56805	6.07462
H	0.29825	11.26295	5.68372
H	1.19102	10.27565	6.99183
H	1.37865	9.94145	5.27779
O	1.91094	12.57565	4.81397
C	15.15888	7.13535	9.79003
C	15.59453	6.24265	10.96109
C	16.29276	7.86285	8.95064
H	14.19122	7.39125	9.44826
H	15.02085	6.04925	11.87459
H	16.54278	5.76055	11.10388
H	16.29630	7.63395	7.86190
H	17.29353	7.65995	9.29011
H	16.20721	8.93155	8.97678
O	15.10652	5.71415	9.75834
C	-0.22338	0.59880	1.09554
C	-0.81317	0.14203	-0.23307
C	1.01180	-0.15708	1.61783
H	-0.69223	1.42895	1.68957
H	-1.71674	0.64761	-0.55085
H	-0.42415	-0.67380	-0.91446
H	1.71281	0.55137	1.98826
H	1.49761	-0.85252	0.93008

A. Appendix

H	0.65345	-0.83030	2.36995
O	0.17351	1.14043	-0.13833
C	4.50479	7.62635	7.64935
C	4.14980	6.39735	8.44405
C	3.47362	8.32955	6.83315
H	5.57251	7.88105	7.68475
H	4.95514	5.80185	9.00585
H	3.17517	6.06155	8.54465
H	2.73529	7.66615	6.47415
H	3.89764	8.79325	5.96085
H	2.90364	9.09015	7.42035
O	4.14463	7.63275	8.96865
C	4.54205	15.13215	0.98518
C	5.33345	15.94905	2.03169
C	4.46295	15.52865	-0.49700
H	3.89685	14.37945	1.48374
H	5.28205	15.75275	3.11328
H	5.91655	16.82605	1.75338
H	3.78745	14.79635	-0.98595
H	5.46085	15.61095	-0.94088
H	4.01175	16.52205	-0.67915
O	5.89655	14.83475	1.39754
C	7.59360	13.18955	13.82644
C	8.21976	11.85125	14.23133
C	7.92130	13.67585	12.42623
H	7.16537	13.72495	14.59492
H	8.15189	11.58055	15.25767
H	8.73330	11.21035	13.47606
H	9.00146	13.78925	12.29257
H	7.54168	14.62945	12.19269
H	7.56886	12.99525	11.66811
O	6.91612	11.96245	13.69423
C	9.87015	11.98355	6.78715
C	9.67751	10.62225	7.46765
C	9.83051	13.30415	7.56295
H	9.99619	12.01335	5.75025
H	9.39415	9.88635	6.79575
H	9.86978	10.35365	8.53015
H	10.25574	13.36135	8.51795
H	8.89639	13.89520	7.52435
H	10.45691	13.98209	6.99875
O	10.96804	11.10015	7.17685
C	1.44541	11.41075	13.89808
C	0.40934	10.93573	12.86544
C	2.16363	12.73863	13.92963
H	1.86623	10.67101	14.59444
H	0.09078	9.91530	12.80490
H	0.01955	11.70372	12.21329
H	1.44934	13.52147	13.53405
H	3.05802	12.55174	13.39052
H	2.45684	13.03509	14.95780
O	0.02765	11.42702	14.11056
C	7.16042	4.01418	14.47435
C	6.10667	4.21682	15.56415
C	8.30064	3.01809	14.80405
H	7.24747	4.70654	13.63625
H	5.22919	4.85797	15.46135
H	6.30008	3.69024	16.54705
H	8.00263	2.06364	15.13925
H	8.97116	3.56218	15.52315
H	8.97805	2.85336	13.98215
O	5.93176	3.29922	14.53575
C	8.02859	6.71961	2.64145
C	6.55884	6.87387	3.00175
C	8.98530	7.88756	2.94975
H	8.39422	5.88451	2.02302
H	5.84224	6.13245	2.73275
H	6.20380	7.77097	3.47755

A. Appendix

H	8.88081	8.31753	3.92365
H	8.98985	8.67130	2.15500
H	10.05541	7.57656	2.94455
O	7.44885	6.14910	3.81485
C	8.93035	6.35030	7.65015
C	8.61005	7.00126	8.96055
C	8.50115	4.90531	7.34505
H	9.72595	6.71076	6.98005
H	9.19135	7.91524	9.23945
H	7.92835	6.54186	9.69145
H	9.30475	4.16578	7.15675
H	7.93515	5.04802	6.38595
H	7.78985	4.39744	8.00045
O	7.88075	7.19629	7.77485
C	12.65435	14.85275	10.86535
C	11.12125	15.07045	10.86465
C	13.33825	14.47725	12.13665
H	13.20365	15.04735	9.91415
H	10.47385	15.23685	9.91035
H	10.58815	15.34275	11.84535
H	14.05845	13.67065	11.86555
H	12.65525	14.04145	12.85725
H	13.86885	15.31715	12.53645
O	11.67795	13.76155	10.74375
C	1.47205	9.35834	1.70862
C	1.41731	8.30702	2.72178
C	2.03300	10.80250	1.99183
H	1.01168	9.15498	0.79107
H	1.12093	7.29441	2.45844
H	1.49100	8.48780	3.81747
H	1.29966	11.47701	2.27873
H	2.59487	11.20140	1.14762
H	2.74529	10.77059	2.79427
O	2.59306	8.55171	1.91949
C	10.88305	14.60828	0.48037
C	10.37465	15.92865	-0.17005
C	9.98735	13.74038	1.41685
H	11.91805	14.40336	0.25320
H	11.10995	16.56589	-0.62092
H	9.35635	16.26560	-0.20210
H	10.37065	13.57830	2.39685
H	9.77055	12.77185	0.89060
H	8.94035	14.15915	1.59608
O	10.38055	14.66527	-0.80059
C	0.62115	1.57555	5.86280
C	0.32845	2.40855	4.58629
C	1.03855	2.19575	7.17321
H	0.71375	0.48085	5.88490
H	0.29525	1.98385	3.54868
H	0.33295	3.50365	4.74133
H	2.08955	2.56975	7.18271
H	0.99655	1.41355	7.97582
H	0.37805	2.97795	7.55224
O	-0.69995	1.88595	5.44319
C	4.41612	3.72185	10.58225
C	5.23265	2.51535	10.97575
C	4.30340	4.80035	11.68745
H	3.85938	3.81275	9.66675
H	5.64096	1.80723	10.29075
H	5.37989	2.48565	12.09455
H	4.22632	5.75735	11.17055
H	5.14231	4.81395	12.42165
H	3.42092	4.80715	12.31835
O	5.67938	3.67085	10.26435
C	6.56575	-0.39365	7.15706
C	6.88475	1.01035	6.49253
C	6.50765	-0.33715	8.68283
H	6.42965	-1.39155	6.67097

A. Appendix

H	7.18285	1.17675	5.45448
H	6.96555	1.84425	7.21803
H	5.49615	-0.18875	8.99460
H	7.19195	0.48245	9.04317
H	6.92615	-1.27035	9.03422
O	5.56945	0.56475	6.65475
C	6.46335	10.91535	3.40155
C	5.51605	11.85455	2.55415
C	6.43715	11.03835	4.93425
H	7.18675	10.33255	2.79185
H	5.57585	11.76315	1.42969
H	4.83465	12.52215	3.01755
H	7.35655	11.28935	5.43285
H	6.00715	10.07235	5.30685
H	5.80315	11.93545	5.11315
O	5.17135	10.54745	2.92155
C	2.42845	5.28075	0.55366
C	1.86328	3.84645	0.72127
C	1.89221	6.23065	-0.56547
H	3.38245	5.45635	1.08426
H	2.28555	3.11345	1.45447
H	1.12223	3.45015	0.07964
H	1.12375	6.87285	-0.10870
H	1.53266	5.65985	-1.39839
H	2.71215	6.89275	-0.90459
O	1.40231	4.92465	1.47982
C	11.33317	1.86495	8.11455
C	12.23864	2.43855	6.99795
C	9.85188	1.66045	7.85865
H	11.71890	1.61225	9.13985
H	13.28455	2.75115	7.22065
H	11.91515	2.59405	5.97055
H	9.68512	0.64395	8.19675
H	9.67067	1.79275	6.80105
H	9.16694	2.33195	8.33835
O	12.11967	1.08565	7.28225
C	1.08435	12.70055	9.43418
C	0.10365	13.82428	9.11585
C	2.50455	13.07545	9.58963
H	0.70415	11.69875	9.55934
H	-0.93425	13.55622	8.79668
H	0.19355	14.90682	9.24614
H	3.07185	12.52775	8.74979
H	3.07455	12.79115	10.51270
H	2.63435	14.19741	9.57401
O	0.40645	13.45294	10.45074
C	10.60145	4.84415	12.35115
C	11.93105	5.56705	12.50685
C	10.36795	4.05105	11.02995
H	9.63985	4.89355	12.95405
H	11.89395	6.40215	13.23765
H	12.98585	5.26065	12.07205
H	11.35165	3.76755	10.70075
H	9.83765	4.65295	10.29285
H	9.75475	3.17815	11.34575
O	11.60595	4.33095	13.15515
C	6.26095	8.16795	14.23616
C	6.13455	9.26715	15.40527
C	7.61715	7.97555	13.64719
H	5.43855	7.73335	13.78625
H	5.22115	9.64665	15.85544
H	7.02175	9.73025	15.76544
H	7.64435	7.48635	12.72159
H	8.22045	7.40065	14.35401
H	8.05445	8.93455	13.37749
O	6.07355	7.85075	15.62304
C	1.53185	1.61185	11.14145
C	2.36298	0.73413	12.14615

A. Appendix

C	0.05108	1.83966	11.35475
H	2.00717	2.00175	10.26375
H	3.29723	0.44167	11.92075
H	1.83107	0.26424	13.01335
H	-0.23563	1.64179	12.41405
H	-0.56844	1.22505	10.64295
H	-0.28336	2.87365	11.06385
O	2.18930	2.13670	12.26475
C	12.86157	10.57165	1.14707
C	11.75312	11.01565	0.17463
C	13.92294	11.68115	1.50408
H	12.94296	9.43985	1.52495
H	10.91565	10.44685	-0.21377
H	11.73859	12.10305	-0.13046
H	14.83722	11.56425	0.87469
H	14.41059	11.45095	2.46762
H	13.52263	12.64965	1.46161
O	11.57177	10.88915	1.57119
C	12.55463	10.18696	11.96995
C	12.67578	10.70007	10.52155
C	11.24218	9.60420	12.43455
H	13.28061	10.21295	12.75365
H	13.61365	11.13999	10.12805
H	11.84088	10.52650	9.85215
H	10.50939	10.13876	11.85995
H	11.17813	9.65750	13.53836
H	11.10890	8.57505	12.18465
O	12.94837	9.41506	10.86705
C	3.33520	4.06839	4.34785
C	3.80487	2.62558	4.45465
C	3.87444	5.08948	5.31615
H	2.60426	4.38295	3.62715
H	3.15007	1.90684	3.92845
H	4.56325	2.24489	5.11985
H	4.87808	4.97953	5.48565
H	3.27512	5.17708	6.24525
H	3.82518	6.11743	4.86695
O	4.48589	3.54499	3.60375
C	12.97070	7.73255	4.54112
C	12.89545	9.24915	4.73583
C	14.31883	7.09015	4.55397
H	12.11653	7.16705	4.20445
H	11.99992	9.72905	4.39949
H	13.69594	9.88795	5.23299
H	14.58825	6.77945	3.52106
H	14.12401	6.28245	5.21315
H	15.04290	7.84125	5.06184
O	12.62129	8.26895	5.79887
C	12.47833	5.64055	0.57742
C	11.30868	6.58785	0.44212
C	13.90685	6.21965	0.20872
H	12.46939	4.64605	0.95442
H	10.30109	6.24975	0.61318
H	11.42193	7.58785	0.06300
H	13.93225	7.31615	0.35421
H	14.20787	6.16095	-0.82887
H	14.79689	5.79035	0.78315
O	11.98429	6.44295	1.64201
C	5.73793	10.77853	9.76839
C	6.58357	10.74681	8.48756
C	6.00597	9.62940	10.84055
H	5.06920	11.59239	10.08412
H	6.39057	11.54587	7.74111
H	7.11065	9.83995	8.29420
H	5.19943	9.49648	11.55138
H	6.79923	9.82300	11.50769
H	6.12280	8.60859	10.37098
O	7.05907	11.31682	9.64598

A.2. AIMD Massive Pre-Equilibration

```

&FORCE_EVAL
  METHOD Quickstep
  &DFT
    BASIS_SET_FILE_NAME BASIS_MOLOPT
    POTENTIAL_FILE_NAME POTENTIAL

    &MGRID
      CUTOFF 350
      NGRIDS 4
      REL_CUTOFF 40
    &END MGRID

    &QS
      EPS_DEFAULT 1.0E-12
    &END QS
    &SCF
      SCF_GUESS ATOMIC
      MAX_SCF 15
      &OT
        PRECONDITIONER FULL_KINETIC
        MINIMIZER DIIS
      &END
      EPS_SCF 1.0E-6
      &OUTER_SCF
        MAX_SCF 20
        EPS_SCF 1.0E-6
      &END
      &PRINT
        &RESTART
          &EACH
            MD 0
          &END EACH
        &END
      &END
    &END SCF

    &XC
      &XC_FUNCTIONAL BLYP
      &END XC_FUNCTIONAL
      &XC_GRID
        XC_DERIV NN10_SMOOTH
        XC_SMOOTH_RHO NN10
      &END XC_GRID
      &vdW_POTENTIAL
        DISPERSION_FUNCTIONAL PAIR_POTENTIAL
        &PAIR_POTENTIAL
          TYPE DFTD3
          PARAMETER_FILE_NAME dftd3.dat
          REFERENCE_FUNCTIONAL BLYP
        &END PAIR_POTENTIAL
      &END vdW_POTENTIAL
    &END XC
  &END DFT

  &SUBSYS
    &CELL
      ABC 15.5835 15.5835 15.5835 # Adapt this
    &END CELL

    &COORD
      @INCLUDE a1.txt # Adapt this
    &END COORD

    &KIND C
      BASIS_SET DZVP-MOLOPT-SR-GTH

```

A. Appendix

```
POTENTIAL GTH-BLYP-q4
&END KIND

&KIND H
  BASIS_SET DZVP-MOLOPT-SR-GTH
  POTENTIAL GTH-BLYP-q1
&END KIND

&KIND O
  BASIS_SET DZVP-MOLOPT-SR-GTH
  POTENTIAL GTH-BLYP-q6
&END KIND

&END SUBSYS
&END FORCE_EVAL

&GLOBAL
  PROJECT Prop0x
  RUN_TYPE MD
  PRINT_LEVEL LOW
  FFTW_PLAN_TYPE EXHAUSTIVE
&END GLOBAL

&MOTION
  &MD
    ENSEMBLE NVT
    STEPS 2000
    TIMESTEP 0.5
    &THERMOSTAT
      TYPE NOSE
      REGION MASSIVE
      &NOSE
        TIMECON 10.00
      &END NOSE
    &END THERMOSTAT
    TEMPERATURE 300
  &END MD
  &PRINT
    &RESTART
      BACKUP_COPIES 0
      &EACH
        MD 1
      &END EACH
    &END RESTART
    &RESTART_HISTORY
      &EACH
        MD 0
      &END EACH
    &END RESTART_HISTORY
  &END PRINT
&END MOTION
```

A.3. AIMD Non-Massive Pre-Equilibration

```

&FORCE_EVAL
  METHOD Quickstep
  &DFT
    BASIS_SET_FILE_NAME BASIS_MOLOPT
    POTENTIAL_FILE_NAME POTENTIAL

    &MGRID
      CUTOFF 350
      NGRIDS 4
      REL_CUTOFF 40
    &END MGRID

    &QS
      EPS_DEFAULT 1.0E-12
    &END QS
    &SCF
      SCF_GUESS RESTART
      MAX_SCF 15
      &OT
        PRECONDITIONER FULL_KINETIC
        MINIMIZER DIIS
      &END
      EPS_SCF 1.0E-6
      &OUTER_SCF
        MAX_SCF 20
        EPS_SCF 1.0E-6
      &END
      &PRINT
        &RESTART
          &EACH
            MD 0
          &END EACH
        &END
      &END
    &END SCF

    &XC
      &XC_FUNCTIONAL BLYP
      &END XC_FUNCTIONAL
      &XC_GRID
        XC_DERIV NN10_SMOOTH
        XC_SMOOTH_RHO NN10
      &END XC_GRID
      &vdW_POTENTIAL
        DISPERSION_FUNCTIONAL PAIR_POTENTIAL
        &PAIR_POTENTIAL
          TYPE DFTD3
          PARAMETER_FILE_NAME dftd3.dat
          REFERENCE_FUNCTIONAL BLYP
        &END PAIR_POTENTIAL
      &END vdW_POTENTIAL
    &END XC
  &END DFT

  &SUBSYS
    &CELL
      ABC 15.5835 15.5835 15.5835 # Adapt this
    &END CELL

    &COORD
      @INCLUDE a1.txt # Adapt this
    &END COORD

    &KIND C
      BASIS_SET DZVP-MOLOPT-SR-GTH

```


A. Appendix

```
POTENTIAL GTH-BLYP-q4
&END KIND

&KIND H
  BASIS_SET DZVP-MOLOPT-SR-GTH
  POTENTIAL GTH-BLYP-q1
&END KIND

&KIND O
  BASIS_SET DZVP-MOLOPT-SR-GTH
  POTENTIAL GTH-BLYP-q6
&END KIND

&END SUBSYS
&END FORCE_EVAL

&GLOBAL
  PROJECT Prop0x
  RUN_TYPE MD
  PRINT_LEVEL LOW
  FFTW_PLAN_TYPE EXHAUSTIVE
&END GLOBAL

&MOTION
  &MD
    ENSEMBLE NVT
    STEPS 5000
    TIMESTEP 0.5
    &THERMOSTAT
      TYPE NOSE
      &NOSE
        TIMECON 100.00
      &END NOSE
    &END THERMOSTAT
    TEMPERATURE 300
  &END MD
  &PRINT
    &RESTART
      BACKUP_COPIES 0
    &EACH
      MD 1
    &END EACH
  &END RESTART
  &RESTART_HISTORY
    &EACH
      MD 0
    &END EACH
  &END RESTART_HISTORY
  &END PRINT
&END MOTION

&EXT_RESTART
  EXTERNAL_FILE Prop0x-1.restart # Needs to match project name above
  RESTART_THERMOSTAT .FALSE.
&END
```

A.4. AIMD Production Run

```

&FORCE_EVAL
  METHOD Quickstep
  &DFT
    BASIS_SET_FILE_NAME BASIS_MOLOPT
    POTENTIAL_FILE_NAME POTENTIAL

    &MGRID
      CUTOFF 350
      NGRIDS 4
      REL_CUTOFF 40
    &END MGRID

    &QS
      EPS_DEFAULT 1.0E-12
    &END QS
    &SCF
      SCF_GUESS RESTART
      MAX_SCF 15
      &OT
        PRECONDITIONER FULL_KINETIC
        MINIMIZER DIIS
      &END
      EPS_SCF 1.0E-6
      &OUTER_SCF
        MAX_SCF 20
        EPS_SCF 1.0E-6
      &END
      &PRINT
        &RESTART
        &EACH
          MD 0
        &END EACH
      &END
    &END SCF
  &END DFT

  &XC
    &XC_FUNCTIONAL BLYP
    &END XC_FUNCTIONAL
    &XC_GRID
      XC_DERIV NN10_SMOOTH
      XC_SMOOTH_RHO NN10
    &END XC_GRID
    &vdW_POTENTIAL
      DISPERSION_FUNCTIONAL PAIR_POTENTIAL
      &PAIR_POTENTIAL
        TYPE DFTD3
        PARAMETER_FILE_NAME dftd3.dat
        REFERENCE_FUNCTIONAL BLYP
      &END PAIR_POTENTIAL
    &END vdW_POTENTIAL
  &END XC
&END DFT

&SUBSYS
  &CELL
    ABC 15.5835 15.5835 15.5835 # Adapt this
  &END CELL

  &COORD
    @INCLUDE a1.txt # Adapt this
  &END COORD

  &KIND C
    BASIS_SET DZVP-MOLOPT-SR-GTH

```

A. Appendix

```
POTENTIAL GTH-BLYP-q4
&END KIND

&KIND H
  BASIS_SET DZVP-MOLOPT-SR-GTH
  POTENTIAL GTH-BLYP-q1
&END KIND

&KIND O
  BASIS_SET DZVP-MOLOPT-SR-GTH
  POTENTIAL GTH-BLYP-q6
&END KIND

&END SUBSYS
&END FORCE_EVAL

&GLOBAL
  PROJECT Prop0x
  RUN_TYPE MD
  PRINT_LEVEL LOW
  FFTW_PLAN_TYPE EXHAUSTIVE
&END GLOBAL

&MOTION
  &MD
    ENSEMBLE NVT
    STEPS 60000
    TIMESTEP 0.5
    &THERMOSTAT
      TYPE NOSE
      &NOSE
        TIMECON 100.00
      &END NOSE
    &END THERMOSTAT
    TEMPERATURE 300
  &END MD
  &PRINT
    &RESTART
      BACKUP_COPIES 0
    &EACH
      MD 1
    &END EACH
  &END RESTART
  &RESTART_HISTORY
    &EACH
      MD 0
    &END EACH
  &END RESTART_HISTORY
  &END PRINT
&END MOTION

&EXT_RESTART
  EXTERNAL_FILE Prop0x-1.restart # Needs to match project name above
&END
```

A.5. Total Electron Density Computation

```

&FORCE_EVAL
  METHOD Quickstep
  &DFT
    BASIS_SET_FILE_NAME BASIS_MOLOPT
    POTENTIAL_FILE_NAME POTENTIAL

    &MGRID
      CUTOFF 350
      NGRIDS 4
      REL_CUTOFF 40
    &END MGRID

    &QS
      EPS_DEFAULT 1.0E-12
    &END QS
    &SCF
      SCF_GUESS ATOMIC
      MAX_SCF 15
      &OT
        PRECONDITIONER FULL_KINETIC
        MINIMIZER DIIS
      &END
      EPS_SCF 1.0E-6
      &OUTER_SCF
        MAX_SCF 20
        EPS_SCF 1.0E-6
      &END
      &PRINT
        &RESTART
          &EACH
            MD 0
          &END EACH
        &END
      &END
    &END SCF

    &PERIODIC_EFIELD
      INTENSITY 5.0E-3
      POLARISATION 1.0 0.0 0.0
    &END PERIODIC_EFIELD

    &PRINT
      &E_DENSITY_CUBE
        STRIDE 1 1 1
        FILENAME =result.cube
        APPEND
      &END
    &END PRINT

    &XC
      &XC_FUNCTIONAL BLYP
      &END XC_FUNCTIONAL
      &XC_GRID
        XC_DERIV NN10_SMOOTH
        XC_SMOOTH_RHO NN10
      &END XC_GRID
    &END XC
  &END DFT

  &SUBSYS
    &CELL
      ABC 15.5835 15.5835 15.5835 # Adapt this
    &END CELL

  &COORD

```

A. Appendix

```
@INCLUDE a1.txt                # Adapt this
&END COORD

&KIND C
  BASIS_SET DZVP-MOLOPT-SR-GTH
  POTENTIAL GTH-BLYP-q4
&END KIND

&KIND H
  BASIS_SET DZVP-MOLOPT-SR-GTH
  POTENTIAL GTH-BLYP-q1
&END KIND

&KIND O
  BASIS_SET DZVP-MOLOPT-SR-GTH
  POTENTIAL GTH-BLYP-q6
&END KIND

&END SUBSYS
&END FORCE_EVAL

&GLOBAL
  PROJECT Prop0x
  RUN_TYPE MD
  PRINT_LEVEL LOW
  FFTW_PLAN_TYPE EXHAUSTIVE
&END GLOBAL

&MOTION
  &MD
    ENSEMBLE REFTRAJ
    STEPS 2002                # Adapt this
    &REFTRAJ
      EVAL_ENERGY_FORCES
      FIRST_SNAPSHOT 1        # Adapt this
      LAST_SNAPSHOT 2002     # Adapt this
      STRIDE 1                # Adapt this
      TRAJ_FILE_NAME Prop0x-pos-1.xyz
    &END REFTRAJ
  &END MD
  &PRINT
    &RESTART
      &EACH
        MD 0
      &END EACH
    &END RESTART
    &RESTART_HISTORY
      &EACH
        MD 0
      &END EACH
    &END RESTART_HISTORY
  &END PRINT
&END MOTION
```

A.6. Compute Properties for IR, Raman, VCD, and ROA Spectrum

```
! Use the advanced mode until the analysis selection menu (y/n)? [no]
! Are the 3 cell vectors of the same size (yes/no)? [yes]
! Enter length of cell vector in pm:
1558.35
! 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)? [yes]
! 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]
! Compute magnetic moments (time consuming, required for VCD and ROA) (y/n)? [yes]
! 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]
! Do not assign radii (0), use covalent radii (1), use Van-der-Waals radii (2)? [2]
! Relative convergence threshold for PDE solver: [0.001]
! Write CSV text file with atomic properties in addition to EMP file (y/n)? [no]
```

A.7. Compute Properties for IR & Raman Spectrum

```

! Use the advanced mode until the analysis selection menu (y/n)? [no]
! Are the 3 cell vectors of the same size (yes/no)? [yes]
! Enter length of cell vector in pm:
1558.35
! 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)? [yes]
! Enter the length of one trajectory time step in fs: [0.5]
4.0
! 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]
! Compute magnetic moments (time consuming, required for VCD and ROA) (y/n)? [yes]
n
! 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]
! Do not assign radii (0), use covalent radii (1), use Van-der-Waals radii (2)? [2]
! Write CSV text file with atomic properties in addition to EMP file (y/n)? [no]

```

A.8. Compute IR, Raman, VCD, and ROA Spectrum

```

! 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 (=depth) of the correlation functions (in time steps): [4096]
! Calculate spectrum up to which wave number (cm-1)? [5000.00 cm-1]
! Calculate scattering cross section for which laser wave number (cm-1)? [20000.0]
! Calculate scattering cross section for which temperature (K)? [350.0]
300
! Correct spectrum for a certain simulation temperature (y/n)? [yes]
! Enter the simulation temperature (K): [300.0]
! Add another observation (y/n)? [no]
! Use central differences (y) or one-sided differences (n) for polarizabilities? [no]
! Enter electric field strength used 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:
../field_x/properties.blist
! Enter data file name for field Y:
../field_y/properties.blist
! Enter data file name for field Z:
../field_z/properties.blist

```


A.9. Compute IR & Raman Spectrum

```

! 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]
4.0
! 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
! 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 (=depth) of the correlation functions (in time steps): [512]
! Calculate spectrum up to which wave number (cm-1)? [5000.00 cm-1]
! Calculate scattering cross section for which laser wave number (cm-1)? [20000.0]
! Calculate scattering cross section for which temperature (K)? [350.0]
300
! Correct spectrum for a certain simulation temperature (y/n)? [yes]
! Enter the simulation temperature (K): [300.0]
! Add another observation (y/n)? [no]
! Use central differences (y) or one-sided differences (n) for polarizabilities? [no]
! Enter electric field strength used 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:
../field_x/properties.blist
! Enter data file name for field Y:
../field_y/properties.blist
! Enter data file name for field Z:
../field_z/properties.blist

```

B. Literature

- [1] M. Thomas: *Theoretical Modeling of Vibrational Spectra in the Liquid Phase*, Springer Theses, **2017**.
- [2] M. Thomas, M. Brehm, R. Fligg, P. Vöhringer, and B. Kirchner: “Computing Vibrational Spectra from ab initio Molecular Dynamics”, *Phys. Chem. Chem. Phys.*, **2013**, *15*, pp. 6608–6622.
- [3] M. Thomas, M. Brehm, O. Hollóczki, Z. Kelemen, L. Nyulászi, T. Pasinszki, and B. Kirchner: “Simulating the Vibrational Spectra of Ionic Liquid Systems: 1-Ethyl-3-methylimidazolium Acetate and its Mixtures”, *J. Chem. Phys.*, **2014**, *141* (2), p. 024510.
- [4] M. Thomas, M. Brehm, and B. Kirchner: “Voronoi Dipole Moments for the Simulation of Bulk Phase Vibrational Spectra”, *Phys. Chem. Chem. Phys.*, **2015**, *17*, pp. 3207–3213.
- [5] M. Thomas and B. Kirchner: “Classical Magnetic Dipole Moments for the Simulation of Vibrational Circular Dichroism by ab initio Molecular Dynamics”, *J. Phys. Chem. Lett.*, **2016**, *7* (3), pp. 509–513.
- [6] M. Brehm and M. Thomas: “Computing Bulk Phase Raman Optical Activity Spectra from ab initio Molecular Dynamics Simulations”, *J. Phys. Chem. Lett.*, **2017**, *8* (14), pp. 3409–3414.
- [7] J. Hutter, M. Iannuzzi, F. Schiffmann, and J. VandeVondele: “CP2k: Atomistic Simulations of Condensed Matter Systems”, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, **2014**, *4*, pp. 15–25.
- [8] M. Brehm and B. Kirchner: “TRAVIS – A Free Analyzer and Visualizer for Monte Carlo and Molecular Dynamics Trajectories”, *J. Chem. Inf. Model.*, **2011**, *51*, pp. 2007–2023.
- [9] M. Brehm and M. Thomas: “An Efficient Lossless Compression Algorithm for Trajectories of Atom Positions and Volumetric Data”, *J. Chem. Inf. Model.*, **2018**, (in press), doi 10.1021/acs.jcim.8b00501.
- [10] J. VandeVondele, M. Krack, F. Mohamed, M. Parrinello, T. Chassaing, and J. Hutter: “Quickstep: Fast and Accurate Density Functional Calculations using a Mixed Gaussian and Plane Waves Approach”, *Comput. Phys. Commun.*, **2005**, *167*, pp. 103–128.
- [11] J. VandeVondele and J. Hutter: “An Efficient Orbital Transformation Method for Electronic Structure Calculations”, *J. Chem. Phys.*, **2003**, *118*, pp. 4365–4369.
- [12] J. VandeVondele and J. Hutter: “Gaussian Basis Sets for Accurate Calculations on Molecular Systems in Gas and Condensed Phases”, *J. Chem. Phys.*, **2007**, *127*, p. 114105.
- [13] S. Goedecker, M. Teter, and J. Hutter: “Separable Dual-Space Gaussian Pseudopotentials”, *Phys. Rev. B*, **1996**, *54*, pp. 1703–1710.

Bibliography

- [14] C. Hartwigsen, S. Goedecker, and J. Hutter: “Relativistic Separable Dual-Space Gaussian Pseudopotentials from H to Rn”, *Phys. Rev. B*, **1998**, 58, pp. 3641–3662.
- [15] A. Becke: “Density-Functional Exchange-Energy Approximation with Correct Asymptotic Behavior”, *Phys. Rev. A*, **1988**, 38, pp. 3098–3100.
- [16] C. Lee, W. Yang, and R. Parr: “Development of the Colle-Salvetti Correlation-Energy Formula into a Functional of the Electron Density”, *Phys. Rev. B*, **1988**, 37, pp. 785–789.
- [17] S. Grimme, J. Antony, S. Ehrlich, and S. Krieg: “A Consistent and Accurate ab initio Parametrization of Density Functional Dispersion Correction (DFT-D) for the 94 Elements H-Pu”, *J. Chem. Phys.*, **2010**, 132, p. 154104.
- [18] S. Nosé: “A Unified Formulation of the Constant Temperature Molecular Dynamics Methods”, *J. Chem. Phys.*, **1984**, 81, pp. 511–519.
- [19] S. Nosé: “A Molecular Dynamics Method for Simulations in the Canonical Ensemble”, *Mol. Phys.*, **1984**, 52, pp. 255–268.
- [20] P. L. Polavarapu, L. Hecht, and L. D. Barron: “Vibrational Raman Optical Activity in Substituted Oxiranes”, *J. Phys. Chem.*, **1993**, 97 (9), pp. 1793–1799.
- [21] J. Šebestík and P. Bouř: “Raman Optical Activity of Methyloxirane Gas and Liquid”, *J. Phys. Chem. Lett.*, **2011**, 2 (5), pp. 498–502.
- [22] G. H. Wannier: “The Structure of Electronic Excitation Levels in Insulating Crystals”, *Phys. Rev.*, **1937**, 52, pp. 191–197.
- [23] M. Brehm, H. Weber, A. Pensado, A. Stark, and B. Kirchner: “Proton Transfer and Polarity Changes in Ionic Liquid-Water Mixtures: A Perspective on Hydrogen Bonds from ab initio Molecular Dynamics at the Example of 1-Ethyl-3-Methylimidazolium Acetate-Water Mixtures - Part 1”, *Phys. Chem. Chem. Phys.*, **2012**, 14, pp. 5030–5044.
- [24] B. J. Gellatly and J. L. Finney: “Calculation of protein volumes: An alternative to the Voronoi procedure”, *J. Mol. Biol.*, **1982**, 161 (2), pp. 305 – 322.