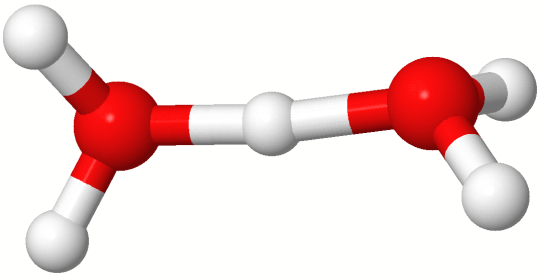
Hands-on Summer School: Electronic Structure Theory for Materials and (Bio)molecules

Los Angeles, CA, July 21 – August 1, 2014



Tutorial IV: *Ab initio* Molecular Dynamics Manuscript for Exercise Problems

Prepared by Mariana Rossi and Luca M. Ghiringhelli

Introduction

With this tutorial, we aim at introducing you to state-of-the-art techniques for sampling the potential energy surface of a system at constant energy (microcanonical ensemble) and at constant temperature (canonical ensemble). The technique we adopt is molecular dynamics (MD), that can be thought as old as modern physics itself, since we will *propagate* in time a system of particles, given initial positions and momenta, by *numerically integrate* Newton's equations of motion for the system.
Indeed the original formulation of Newton's second law is [1]: "*Lex II: Mutationem motus proportionalem esse vi motrici impressae, et fieri secundum lineam rectam qua vis illa imprimitur.*", in modern English, "Law II: The change of momentum of a body is proportional to the impulse impressed on the body, and happens along the straight line on which that impulse is impressed." In symbols:

$$\int_{\Delta t} \boldsymbol{F} \mathrm{d}t = \Delta \boldsymbol{p} \tag{1}$$

Where the vector $\boldsymbol{F}$ is the force acting on a body, $\boldsymbol{p}$ is its momentum, Δt is the time along which the force is applied. The whole l.h.s. expression is called *impulse*.
All the necessary theoretical tools for understanding this tutorial are given in the lecture of Luca Ghiringhelli and Mariana Rossi about molecular dynamics and path integral molecular dynamics.

The kind of MD we are interested in is Born-Oppenheimer MD, where nuclei are treated as classical particles and the forces between them are evaluated by solving the (ground-state) electronic structure problem for a given number of electrons and the nuclei at given positions. In this tutorial, the electronic structure problem is solved via Density Functional Theory. The tutorial contains two exercises plus one optional exercise; all of them adopts as guinea-pig the gas-phase (isolated) $H_5O_2^+$, which some of you may recognize as the Zundel (cat)ion. This system is one of the (protonated) water clusters that are thought to be the "building blocks" of liquid water. [2].
The first and introductory exercise poses the problem of choosing the right settings for obtaining a stable and reliable molecular dynamics trajectory.
The second exercise lures you into the core of the matter, i.e. the choice of a thermostat for simulating the canonical ensemble, i.e., the contact between your system of interest and a thermal reservoir. Many traditional thermostats are implemented in FHI-aims, but for this tutorial we focus on the most successful traditional one, the Nosé-Hoover thermostat, and two modern and increasingly famous ones, the stochastic velocity-rescaling and the colored-noise thermostats. The latter opens the possibility for accurate inclusion of quantum statistics for nuclei, whereas the usual MD machinery treat them as classical particles.
The third (optional) exercise shows you how to evaluate the Infra-Red spectrum of your molecule / complex after a (long) MD trajectory.

All the useful files for this tutorial (together with solutions) are located in:
`$HANDSON/tutorial_4`
All the scripts for this tutorial are located, as usual, in:
`$HANDSON/hands-on-2014-tutorials/tutorial_4/scripts`

The microcanonical ensemble

Exercise 1: setting the time step

In this first exercise, we will investigate the effects of the time step size for the integration of the equations of motion in a microcanonical molecular dynamics simulation. For a better illustration, we will not only consider the $H_5O_2^+$ cluster, but also its heavier, deuterated counterpart $D_5O_2^+$ [1]. In order to speed up the exercise, each group will simulate either $H_5O_2^+$ or $D_5O_2^+$.

Geometry files, containing both $H_5O_2^+$ and $D_5O_2^+$ are already provided in the folder `exercise_1`. (all files for exercise x are located in the folder named `exercise_`x).

In order to substitute hydrogen with deuterium, all you have to do is to put deuterium's mass in the `control.in` you are going to use:
`mass 2.014102`
Furthermore the `species` label should be changed such that in the `geometry.in` file the same label is appearing. In this case, the species label could be `D`, already matching the provided `geometry.in`:
`species D`.
The suggested choice of the molecule will be made clear (announced) by the tutors at hand.

You will notice that this file also contains velocities assigned to each atom, so that the Molecular Dynamics run will not use a random initialization. These velocities come from a previous equilibration of the molecule at ∼300K. You will see how such a thermalization is done in Exercise 2.

Instructions

- Copy the geometry file corresponding to the molecule you will calculate to `geometry.in`.
- First, build an input file for FHI-aims using the LDA (**pw-lda**) functional, no spin polarization (**spin none**), and appending the "**light**" standards for the species (O and D or H depending on your `geometry.in`, D species default can be found in this exercise's folder). Please, take few moments to understand the differences between the hydrogen and the deuterium species default. Please refer to the manual for the exact syntax of these flags.

  Do not add any flags that we do not mention! They are not needed and might hinder the performance of the calculation.
- Start a 0.15 ps MD run in the microcanonical ensemble, using a 0.0005 ps (Δt = 0.5 fs) time step (flags `MD_run` and `MD_time_step` respectively), with the following "accurate" self-consistency convergence criteria:

  ```
  sc_accuracy_rho 1E-5
  sc_accuracy_eev 1E-4
  sc_accuracy_etot 1E-6
  sc_accuracy_forces 5E-4
  ```

  If you are curious, after finishing these exercises you may want to play with these criteria and see how the outcome is affected.

[1] The deuterium atom has one proton and one neutron in its nucleus, thus about twice as heavy as hydrogen.

In order to start the run type:

`mpirun -np 4 $HANDSON/bin/aims.ipi.x > out &`

- Please note that the executable is different from the one used in the other tutorials. This is actually needed only for running exercise 3, while all the other exercises could be run also with the usual executable, `aims.x`.
- The & puts the run in the background, so that the output file is created, but the terminal is free for other use.
- If you anyway would like to have a dynamic view of what happens in your output, after starting the simulation you can type:

  `tail -f out`

- ATTENTION: do not start another FHI-aims run simultaneously. That would slow down BOTH calculations considerably.

- When it is done, use the "aims_MD_eval.pl" script to analyze your run by typing in the terminal:

  `perl aims_MD_eval.pl "FHI-aims-output-file" > "script-output-file"`

- Plot the total energy (fifth column of the script output file) vs. the simulation time (first column of the script output file) with `xmgrace` [2]. For a short guide on how to plot files with multiple columns via `xmgrace`, see the Appendix to this tutorial. How would you describe what you see? Was this somewhat predictable?
- Increase the time step (Δt) of the MD simulation to 0.001 ps. Run FHI-aims (total of 0.15 ps) and keep the output.
- Increase the time step to 0.003 ps and run the simulation again, redirecting the output to another file.
- Together with the plot for $\Delta t = 0.0005$ ps, plot the total energy vs. simulation time in `xmgrace` for $\Delta t = 0.001$ ps and $\Delta t = 0.003$ ps.

  How do the energy fluctuations develop? Do you notice something strange happening for the $\Delta t = 0.003$ ps run?

From the practical point of view, a larger time step is desirable, since it allows to assess longer trajectories in shorter computational times. Notice, however, that the $\Delta t = 0.003$ ps simulation diverges for $H_5O_2^+$. In fact, the molecule dissociates. You can inspect the dynamics of the molecule by running

`create_xyz_movie.pl "FHI-aims-output-file" > "script-output-file".xyz`

on your output and opening it in VMD (or Jmol, or Molden, whichever you prefer). The reason for the dissociation is that the integrator is unable to deal with these "big" time steps. This integrator uses a simple Verlet algorithm [5], where the error in the trajectory goes with Δt^4. If you are simulating $D_5O_2^+$, the $\Delta t = 0.003$ ps simulation does not diverge, although the energy fluctuations become very large. You can also look at the dynamics of this molecule in VMD.

Although such large energy fluctuations would be already not accurate enough for a production run with this molecule, the fact that it does not explode illustrates an important

[2] If you are more confortable with, e.g., `gnuplot`, please feel free to use it instead!

point: the largest Δt that can be used in a particular integration algorithm depends on the highest vibrational frequency of the system. Since the D atoms, being heavier, have a larger vibrational period (do you understand why this is obvious?), the used Δt can also be larger.

Timing: ∼30 minutes total

The canonical ensemble

Exercise 2: Testing thermostats

Most "real-life" experiments cannot be done in a situation where the energy is explicitly kept constant, but where other quantities like the average temperature or pressure are maintained instead. As you may know, an ensemble where the temperature is kept constant is the a canonical ensemble. In order to simulate this ensemble, the system has to be coupled to a heat bath. From a statistical mechanics point of view, the average kinetic energy in the canonical ensemble follows the equipartition theorem, which says that it is equally distributed on the various degrees of freedom of the system. Therefore, the momenta $\mathbf{p} = m\mathbf{v}$ follow the Maxwell-Boltzmann (MB) distribution:

$$P(|\mathbf{p}|) = \left(\frac{\beta}{2\pi m}\right)^{3/2} \exp\left(-\beta|\mathbf{p}|^2/(2m)\right). \tag{2}$$

The instantaneous temperature $\bar{T}$ is given by the relation $\bar{T} = \frac{2\langle K\rangle}{3k_B} = \frac{\sum_i^N |\mathbf{p}_i|^2/m_i}{3Nk_B}$, where $\langle K\rangle$ is the average kinetic energy of the system, m_i is the mass of atom i and N is the number of atoms. This means that the temperature is not constant but can (and should) fluctuate around the average value. The theoretical standard deviation is $\sigma^2 = \frac{2T^2}{3N}$.

Here we apply three schemes which simulate a thermostat in MD. Below is a short summary of the thermostats we will use in this tutorial:

1. Extended Lagrangian approach: the Nosé-Hoover thermostat [5, 6].
   Equations of motion derived from the Lagrangian of the system conserve the total energy of the system. One can write an *extended* Lagrangian, by adding fictitious degrees of freedom, such that the overall total energy is conserved but the atomic subsystem can span ensembles other than microcanonical. With the Nosé-Hoover Lagrangian, the atomic subsystem samples the canonical ensemble. The equations of motion of the Nose-Hoover thermostat are:

$$\dot{\mathbf{r}}_i = \mathbf{p}_i/m_i \tag{3}$$

$$\dot{\mathbf{p}}_i = -\frac{\partial \mathscr{U}\left(\mathbf{r}^N\right)}{\partial \mathbf{r}_i} - \frac{\Pi \mathbf{p}_i}{Q} \tag{4}$$

$$\dot{\eta} = \frac{\Pi}{Q} \tag{5}$$

$$\dot{\Pi} = \left(\sum_i \frac{\mathbf{p}_i^2}{m_i} - \frac{g}{\beta}\right) \tag{6}$$

   where g is the number of degrees of freedom of the system, $\mathscr{U}$ is the potential energy, Q the "thermostat mass", and $\mathbf{p}_i$ and m_i the momenta and masses of the ith particle of the system, respectively. The conjugated momentum Π of the extra coordinate η acts as a fluctuating drag parameter to the atomic subsystem. The conserved energy associated to the equations of motion is:

$$\mathscr{E} = \sum_i \frac{\mathbf{p}_i^2}{2m_i} + \mathscr{U}\left(\mathbf{r}^N\right) + \frac{1}{2}\frac{\Pi^2}{Q} + g\frac{\eta}{\beta} \tag{7}$$

2. Stochastic velocity-rescaling thermostat (Bussi-Donadio-Parrinello)[7].
   In this algorithm, a deviation of the instantaneous kinetic energy is corrected in the following way:

$$dK = \left[\overline{K} - K(t)\right] \frac{dt}{\tau} + 2\sqrt{\frac{K(t)\overline{K}}{N_f \tau}}\xi(t) \tag{8}$$

   where $\overline{K}$ is the target kinetic energy, $K(t) = p^2(t)/2m$ is the instantaneous kinetic energy, τ is the relaxation time of the thermostat, N_f is the number of degrees of freedom, and ξ is a white noise term (the derivative of a Wiener process [3]) that obeys $\langle \xi(t)\xi(t')\rangle = \delta(t-t')$.

   In practice the trajectory is first propagated for one time step with e.g. a velocity-verlet integrator and the new velocities are calculated as usual. Then, the new kinetic energy K is evaluated and the velocities are rescaled by a factor α such that:

$$\begin{aligned} \alpha^2 &= e^{-\Delta t/\tau} + \frac{\overline{K}}{N_f K}\left(1 - e^{-\Delta t/\tau}\right)\left(R_1^2 + \sum_{i=2}^{N_f} R_i^2\right) \\ &+ 2e^{-\Delta t/2\tau}\sqrt{\frac{\overline{K}}{N_f K}\left(1 - e^{-\Delta t/\tau}\right)} R_1 \end{aligned}$$

   where the R_i's are independent random numbers from a Gaussian distribution with unitary variance[4].

   For this thermostat a conserved pseudo-Hamiltonian $\tilde{H}(t)$ can be defined:

$$\tilde{H}(t) = H(t) - \int_0^t \left(\overline{K} - K(t')\right)\frac{dt'}{\tau} - 2\int_0^t \sqrt{\frac{K(t')\overline{K}}{N_f \tau}}\xi(t')$$

   where $H(t)$ is the total energy of the atomic system.

   The Bussi-Donadio-Parrinello thermostat yields the correct distribution of K, does not have ergodicity problems, does not perturb the dynamics, and its accuracy and efficiency is rather independent of τ.

3. Nuclear quantum effects: the colored noise thermostat.

   The colored noise thermostat is an extension of a Langevin thermostat; indeed it is also called Generalized Langevin Equation (GLE) thermostat. The classical Langevin thermostat is expressed through the following differential equation for the momentum (here in one dimension, without loss of generality):

$$\dot{p}(t) = -\gamma p(t) + \sqrt{2m\gamma T}\xi(t) \tag{9}$$

   where γ is a (friction) parameter and $\xi(t)$ is a stochastic variable distribute as a Gaussian white noise as above.

   The Langevin thermostat is constructed via a Markovian (i.e. memoryless) stochastic differential equation. Its extension, which leads to the colored noise thermostat, is constructed via introducing auxiliary degrees of freedom $\mathbf{s}$ to the dynamics. These extra degrees of freedom model a Markovian process in higher dimensions, but give

[3] An example of a Wiener process $W(t)$ is the Brownian motion (you might have heard about it...). $W(t)$ has the following characteristics: $\xi(0) = 0$; $W(t)$ is continuous; the increments are independent and $W(t_2) - W(t_1)$ is a Gaussian with average 0 and $\sigma = t_2 - t_1$.

[4] Note that $\sum_{i=2}^{N_f} R_i^2$ can be drawn directly from a suitable Gamma distribution

rise to non-Markovian dynamics when the fictitious degrees of freedom are integrated out. The equations of motion are:

$$\dot{r} = p/m \tag{10}$$

$$\begin{pmatrix} \dot{p} \\ \dot{\mathbf{s}} \end{pmatrix} = \begin{pmatrix} -V'(R) \\ 0 \end{pmatrix} - \mathbf{A}_p \begin{pmatrix} p \\ \mathbf{s} \end{pmatrix} + \mathbf{B}_p \begin{pmatrix} \boldsymbol{\xi} \end{pmatrix}, \tag{11}$$

where $\boldsymbol{\xi}$ is an array of uncorrelated Gaussian noises, $V'(R)$ is the gradient of the potential and the $\mathbf{A}_p$ and $\mathbf{B}_p$ are matrices that obey the relation

$$\mathbf{A}_p\mathbf{C}_p + \mathbf{C}_p\mathbf{A}_p^T = \mathbf{B}_p\mathbf{B}_p^T, \tag{12}$$

where $\mathbf{C}_p$ is the covariance matrix defined as $\mathbf{C}_p = \langle (p, \mathbf{s})^T (p, \mathbf{s}) \rangle$. By integrating out the $\mathbf{s}$ degrees of freedom, one gets dynamics of a non-Markovian process in the physical variables, with the EOM given by

$$\dot{r} = p/m \tag{13}$$

$$\dot{p} = -\frac{\partial V}{\partial r} - \int_{-\infty}^{t} Q(t-\tau)p(\tau) + \zeta(t), \tag{14}$$

where $\zeta(t)$ is a correlated noise and $Q(t-\tau)$ is a frequency dependent memory kernel which depends on $\mathbf{A}_p$. The fluctuation-dissipation theorem (and canonical sampling) is obeyed if $\langle \zeta(t)\zeta(0) \rangle = k_B T Q(t)$. However, the FDT can be broken and one can enforce quantum statistic to some selected degrees of freedom (note: FDT is equivalent to equipartition, thus breaking FDT implies breaking equipartition). This is the way we will apply the thermostat in this exercise.

In the same spirit as the stochastic velocity rescaling thermostat, the colored noise thermostat, defines a conserved pseudo-Hamiltonian:

$$\tilde{H} = H - \sum_i \Delta K_i \tag{15}$$

where ΔK_i is the change in kinetic energy due to the action of the thermostat at the i-th time-step, and the sum is extended over the past trajectory.

Instructions

- Use the `control.in` (tight convergence settings) and `geometry.in` for $H_5O_2^+$ / $D_5O_2^+$ that are in Exercise 2 folder. You will only have to change the MD related flags, as explained below.
- You should use a time step of **0.001** ps and we will be simulating all systems at 300K. The keywords you should use are shown below. The general syntax of the thermostat flags is:

  `<MD command> <time> <thermostat name> <temperature> <parameter>`

  You should provide an educated guess for the value of each thermostat's parameter, following the explanations given above and the hints below. If you have time, try to play with these thermostat parameters afterwards.

1. Nosé-Hoover

   `MD_run 1.0 NVT_nose-hoover 300` Q

   Hint: It can be shown (and is actually a fun exercise) that the optimal mass for the Nosé-Hoover thermostat is $Q = 3Nk_BT/\omega^2$, as shown in Ref. [9]. Considering $H_5O_2^+$ ($D_5O_2^+$) has vibrational modes between roughly 100 and 4000 cm^{-1} (70 and 3000) cm^{-1}, which mass would you choose for the thermostat? More straightforwardly, the flag `MD_thermostat_units cm^-1` can be used to input the thermostat mass directly in cm^{-1}.
   (A control file `control.nose-hoover_programmed.in` with a reasonable value of Q is provided in `exercise_2/solution/`)

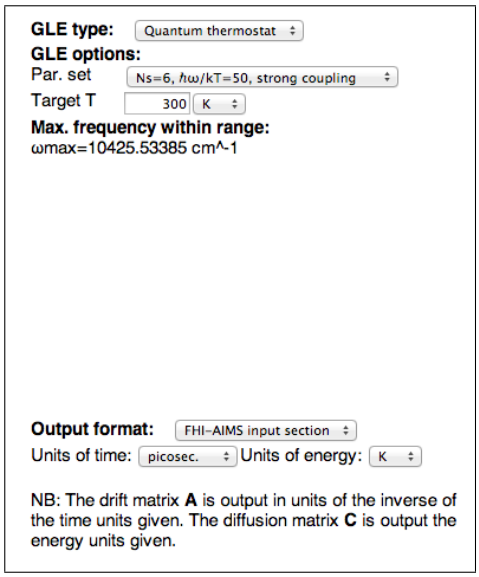
2. Bussi-Donadio-Parrinello (BDP)

   `MD_run 1.0 NVT_parrinello 300` τ

   Hint: Use τ (the unit is ps) equal to $\approx$ 20–50 time steps. Unless τ is too small with respect to the time step (1–5 time steps), the actual value of τ does not affect the performance of this thermostat; with a milder coupling, one observes a slower response to the imposed change of temperature, though.

3. Colored noise thermostat

   `MD_run 1.0 GLE_thermostat 300` n
   The parameters for this thermostat are quite complex. Here n refers only to the number of fictitious degrees of freedom. Choose n =6 in the line above. We will here use this thermostat to approximate nuclear quantum effects (we will not simulate a canonical ensemble), so that the matrices **A** and **C** (Eq. 11) will have to be given as an input. They can be generated at `https://epfl-cosmo.github.io/gle4md/`, in the section `INPUT` .
   Choose the parameters in the website like it is shown in the picture .



- While waiting for the simulations to complete, you are challenged to demonstrate Eq. 2 and calculate what would be temperature corresponding to a zero point energy ($\hbar\omega/2$) for a frequency $\omega_1 = 3000$ cm^{-1} and for $\omega_2 = 100$ cm^{-1}. Also show which frequency ω corresponds to a temperature of 300K.
- Calculate the instantaneous temperature of each atomic species for each thermostat you tried by using the following script:

  `python get_temperatures.py 'output name' O H`

  where 'O' and 'H' define the oxygen and hydrogen species. How do the different thermostats work in each case? Do the thermostats couple efficiently to the system? Can you understand the result for the quantum thermostat? (Hint: different vibrational frequencies involve different atoms and give rise to different zero point energies)
- After running the script "aims_MD_eval.pl" on the output, plot total energy (column 5) vs time step (column 1). Did you expect such a behavior? Now use the

script “aims_NH-BDP.pl” (works analogously to “aims_MD_eval.pl”) to extract the respective conserved quantity from the trajectory and plot it (sixth column) vs. the time steps (first column). What do you observe?

- The script “get_rdf_h5o2.py” writes the radial distribution function $g(r)$, decomposed into OH, OO, and HH pairs. Run the script on each output by typing:

  ```
  python get_rdf_h5o2.py 'output name'
  ```

  It will output three files, starting with ‘rdf_oo’, ‘rdf_oh’ and ‘rdf_hh’. The first column is the distance (in Ångstroms) and the second column the count. Plot these files for each thermostat you used and compare the results for each $g(r)$. In the next exercise we will calculate the same data with path integral molecular dynamics, so that we can compare! For which $g(r)$ of you see the larger differences between the classical and “quantum thermostat" distribution? Compare the $g(r)$ of $H_5O_2^+$ and $D_5O_2^+$ (the results for the cluster that you have not run are in `exercise_2/solutions/`). Could you predict the differences? Can you understand them?

- In the folder `exercise_2/solutions/` we provide the output files for 20ps long simulations (all three cases). You can compare with the simulations you have done.

Time estimation: ∼1h30m

Exercise 3: Path integral molecular dynamics and the i-PI code

In the previous exercise we already simulated nuclear quantum effects in an approximate way by using the "quantum thermostat" based on the generalized Langevin equations (GLE). Here we will use path integral molecular dynamics (PIMD) in the NVT ensemble in order to look at the same quantities as we did in Exercise 2 for $H_5O_2^+$ and $D_5O_2^+$. As was discussed in the lecture, PIMD is classical molecular dynamics in an extended space: Each quantum particle is mapped onto a ring polymer of classical particles, which consists of n repetitions of the original system connected by harmonic springs. This method thus increases the cost of the calculation by n. Since the harmonic forces of the springs are easily and analytically evaluated, the method is also trivially parallelizable.

Here we will use the i-PI [10] program in order to manage the calculation of the several replicas of the system, that will, in turn, be simulated by FHI-aims. The i-PI program works through a client-server architecture, where i-PI is the server that provides the evolution of the equations of motion that sample the desired ensemble, and the *ab initio* code (in this case FHI-aims) is the client that provides the calculation of interatomic forces. i-PI uses internet or unix sockets, that allow the replicas of the system to be simulated all in the same machine, in different nodes of one machine, or in different machines, as long as all the calculations can communicate with the server.

The quantum properties of the system converge with the number of beads. Typically the number of beads needed is of the order of $\hbar\omega_{max}/k_BT$, where T is the temperature being simulated and ω_{max} is the maximum frequency of vibration of the system.

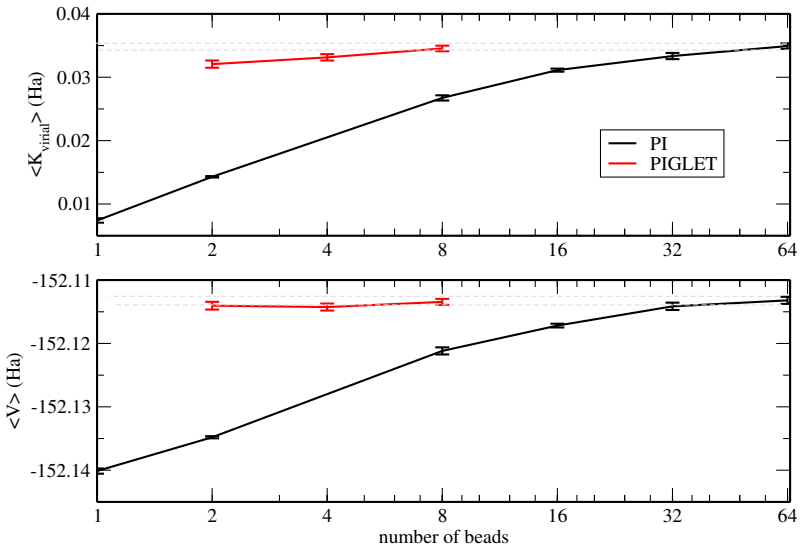


Figure 1: Convergence of the average virial estimator for kinetic energy $\langle K_{cv}\rangle$ and average potential energy $\langle V\rangle$ for the $H_5O_2^+$ molecule at 300K with path integral molecular dynamics.

One should always check the convergence of a property of interest before doing a PIMD calculation, but there will be no time for this in this exercise. Instead, in Figure 1 the average potential energy $\langle V\rangle$ and the average virial estimator for the kinetic energy $\langle K_{cv}\rangle$ are given for $H_5O_2^+$ at 300K with LDA (like in the previous exercises). The expressions for these quantities in the PIMD formalism for a system of n beads and N particles are

$$\langle V \rangle = \frac{1}{n} \sum_{i=1}^{n} \langle V(\mathbf{q}_i^1, ..., \mathbf{q}_i^N) \rangle \tag{16}$$

$$\langle K_{cv} \rangle = \frac{3Nk_BT}{2} + \frac{1}{2n} \sum_{j=1}^{N} \sum_{i=1}^{n} \langle (\mathbf{q}_i^j - \bar{\mathbf{q}}^j) \cdot \nabla V(\mathbf{q}_i^1, ..., \mathbf{q}_i^N) \rangle \tag{17}$$

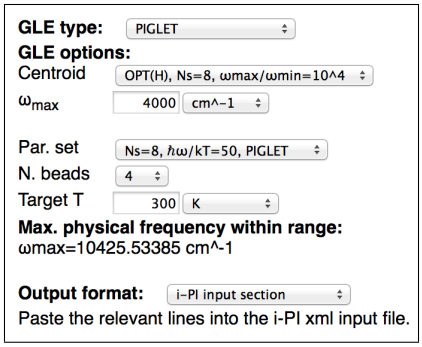
where $\bar{\mathbf{q}} = 1/n \sum_i \mathbf{q}_i$ is the position of the centroid. In Fig. 1, one sees that up to 64 beads (64 replicas of the system) are necessary to fully converge the quantities. This would be prohibitive for the use with first principles potential energy surfaces. One way to reduce the number of beads necessary to converge static (average) properties is to attach properly tuned thermostats based on the generalized langevin equation to the beads of the ring polymer, as proposed in Ref. [11]. This is what is shown in red in Fig. 1, where one achieves convergence already with 8 replicas. Due to the computational resources available, we will only be able to perform calculations with 4 beads.

- *Server:* An example of an input file for i-PI can be found inside the `exercise_3/` folder, called `input.xml`. It is an xml file, which is quite intuitive to learn. The input is not complete. Please take your time to understand the keywords that are there and consult the i-PI manual found in `$HANDSON/doc/manual.pdf`.

  Besides setting the number of beads (4), there are two other fields that should be completed in the file: The field address about the socket communication and the field of the thermostat.

  1. We will be using UNIX domain sockets here, since we will be running both server and client in the same machine. Just write in the `<address>` field a string of your choice, e.g. ipam-2014-h5o2 . In the input there is a commented block with an example of how it would look if we would use internet sockets communicating with the local host. If using in different machines or nodes, one would have to give a proper IP address at the "address" field.

  2. The thermostat parameters can be found in `https://epfl-cosmo.github.io/gle4md/`. Choose the parameters in the website like it is shown in the picture and paste it into the relevant section of the i-PI input file.



  More detailed instructions about the input file can be found in manual of i-PI, in Sec. 5.1.2.

- *Client:* The keyword to add to the `control.in` file that is in the `exercise_3` folder of FHI-aims is
  `use_pimd_wrapper hostaddress portnumber`
  where hostaddress should be substituted by the address (IP or name) of the server and portnumber by the number of the port with which it should communicate. Since we are using UNIX domain sockets, one can just put a dummy number on the `portnumber` field. In the `hostaddress` field write, for example, `UNIX:ipam-2014-h5o2`, i.e., the only constrain is that the string after the colon matches the `address` field in the i-PI input file.

- Launch the server by typing:

  ```
  python $HANDSON/ipi/i-pi input.xml > out &
  ```

- Launch four instances of FHI-aims by typing:

  ```
  for i in {1..4}; do ($HANDSON/bin/aims.ipi.x > aims${i}.out &) ; done
  ```

- When the simulation is done, run the script "get_rdf_h5o2_PI.py" to compute the radial distribution function $g(r)$ in the directory where the output files are, by typing:

  ```
  python get_rdf_h5o2_PI.py piglet-h5o2 4
  ```

  where the first field is the prefix of the position files of all the beads and the second field is the number of beads.

- Compare the $g(r)$ obtained here with the one from the previous exercises. For which species do you see most differences between the classical and the quantum case? Can you explain these results? (The simulation here may take a long time, but you can look at the rdf's while the simulation is running.)

- While the simulation is running take your time to read the introduction of the i-PI manual, which explains in detail the architecture of the code and its capabilities.

In case you like to get challenged by a statistical mechanics question during the waiting time of this exercise, you may try to demonstrate that the theoretical relative standard deviation of the temperature in the canonical ensemble is $\frac{\sigma^2}{T^2} = \frac{2}{3N}$.

Exercise 4 (optional): Harmonic vs. anharmonic vibrations

Vibrational spectroscopy is, nowadays, a very important tool for the characterization of molecules. Usually, the frequencies measured experimentally are compared to theoretical calculations in order to determine the geometry and electronic structure of the molecule. For this purpose, the most common approach is to relax the geometry of a molecule on the Born-Oppenheimer potential energy surface (PES) and then to perform a vibrational analysis in the the *rigid-rotor/harmonic-oscillator* approximation. There are a few problems in this approach: the inability to probe all representative conformations of the molecule and the inability to include anharmonic effects for particularly floppy vibrational modes. Furthermore, rotations along certain axis of symmetry in the molecule cannot be considered rigid rotors. This is important for the molecule studied here, which vibrates while it rotates and thus changes its moment of inertia.

It is possible to go beyond the harmonic approximation by "brute-force", but calculating the shape of the potential energy surface even for very few degrees of freedom is an amazingly demanding task.

One can overcome some of these drawbacks by performing a Molecular Dynamics simulation of the system in question. In the framework of Linear Response Theory, one can rewrite the Fermi Golden rule by means of the Fourier transform of the dipole moment time correlation function [12]:

$$I(\omega) = F(\omega) \int_{-\infty}^{\infty} dt\, \mathrm{e}^{i\omega t} \langle \vec{M}(t) \cdot \vec{M}(0) \rangle_{t_0} \tag{18}$$

In this formula, $I(\omega)$ is the intensity of the vibrations and $F(\omega)$ is a quantum corrector factor which must be multiplied with the classical line shape in order to reproduce the measured amplitudes [13, 14]. The angular brackets denote a statistical time average for the auto-correlation of the dipole moment of the molecule. Formula 18 will give all frequencies that are active in the IR range. Therefore, the whole IR spectrum of the molecule can be calculated within one MD run, since one can choose various t=0 to average the dipole auto-correlation over.

A similar relation can be found for the time average of the velocity auto correlation function:

$$\mathrm{VDOS}(\omega) = \sum_{i=1,N} \int_{-\infty}^{\infty} dt\, \mathrm{e}^{i\omega t} \langle \vec{v}_i(t) \cdot \vec{v}_i(0) \rangle_{t_0} \tag{19}$$

In this case, N is the number of atoms in the molecule and the quantities that are computed are all vibrational frequencies of the molecule, not only the ones active in the IR range (due to selection rules). The advantage is that this function allows to assign the frequencies to the displacements of the individual atoms [12].

For this exercise we will use the deuterated version of our molecule molecule, $D_5O_2^+$, since it allows a larger time-step and hence a longer simulation (in time). The instructions follow:

1. **Harmonic vibrations**

   - You should evaluate the harmonic vibrational frequencies of the molecule, in the same way you did for H_3O^+ in the first practical section (Tutorial 1) of this workshop. Fore reference, the solution is provided in:

`exercise_3/harmonic_vibs`.

2. **Calculating the anharmonic vibrations**

   - We will simulate at $T = 300$K (you may try other temperatures afterwards) A geometry file called `geometry.in.300K`, which includes a thermalized geometry is provided in the exercise's folder.
   - Copy `geometry.in.300K` to `geometry.in`
   - Use the `control.in` file provided in this folder. There are some settings for the integration grids that are different, so that the simulation is computationally lighter.
   - We will now run a 4ps MD simulation in the canonical ensemble with the BDP thermostat. As has been discussed above, *this specific* thermostat does not perturb strongly the dynamics of the system. The dynamic quantities are preserved, and thus the autocorrelation functions can be reliably calculated. Note that this is not the case for other thermostats. The "traditional" way of performing this simulation would be to first thermalize the molecule at the desired temperature and then start an MD run in the microcanonical ensemble, where the dynamical quantities would be computed. In the case of few degrees of freedom, though, the microcanonical sampling would be wrong, because the distribution of velocities is rather different from a Maxwell-Boltzmann one. With increasing number of degrees of freedom (more than ≈ 50), the distribution of velocities in the microcanonical and canonical ensembles become indistinguishable.

     Your MD block should read:

     ```
     MD_run 4.0 NVT_parrinello 300 0.1

     MD_time_step 0.002
     ```

     This run takes $\sim$35 minutes.

   - **Dipole-dipole correlation function: IR spectrum.** For the analysis of this simulation, we will use the script `auto-correlate.py` in order to see the evolution of the auto correlation function and of the spectrum with the time of the run. Besides doing this analysis at the end of the simulation, you can do it after approximately 1ps, 2ps, and 3ps of simulation (or more often if you want). To check interactively how far the simulation is, you may write:

     ```
     grep "Updated atomic structure" "Output-File" | wc -l
     ```

     which gives the number of completed time steps.

     In order to run `auto-correlate.py`, you need to prepare an input file called '`control.autocorr.in`'. The flags are the following:

     - `path` *string*

       'string' is the name of the FHI-aims output

     - `choice` *string*

       'string' is the type of autocorrelation you want to calculate. Accepts either 'velocity' or 'dipole'

- `sampling_interval` *integer*

  'integer' is an integer number that defines the sampling interval (each $t = 0$) to calculate the autocorrelation. Recommended value is 1.

- `cutoff_ratio` *float*

  'float' is a float number in the interval $[0, 1]$ that defines the ratio of the tail of the autocorrelation function you wish to leave out in order to make the fourier transform. Recommended value is 0.1.

- `broadening` *float*

  'float' is a float number in units of wavenumbers (cm^{-1}) that defines the broadening of the Gaussian to be convoluted with the Fourier transform. Recommended value is 3.

An example of input file for this script is provided in the folder `exercise_3/anharmonic_vibs/`. Copy also the executable `"home_made_ft.x"` to the folder where you are performing this exercise. You can then run the script by typing:

```
python auto-correlate.py
```

Three files will be generated, namely:

- **autocorr.dat** contains 3 columns, the first being time in ps, the second being the autocorrelation function, and the third being the autocorrelation function times a window function that makes it go to zero on the edges. The window function is essential for reducing the noise in the Fourier transform.
- **raw_fourier_transform.dat** contains 2 columns, the first being wavenumbers in cm^{-1} and the second the intensities in arbitrary units
- **convoluted_fourier_transform.dat** contains 2 columns, the first being wavenumbers in cm^{-1} and the second the intensities in arbitrary units convoluted with a gaussian curve (width given in the input).

- Run first the python script with the '`dipole`' option in the `dipole-autocorr` folder. Remember to rename the outputs of the script so that they don't get overwritten. Visualize them in xmgrace and see how the autocorrelation function and the peaks of the Fourier transform evolve with time. What does the "long" time (average) correlation indicate? Did you expect the curve not to go to zero?
- **Watch the movie.** After extracting the xyz movie of the MD run, via the script `create_xyz_movie.pl`, you may also want to visualize the trajectory, e.g., with VMD. Checking the visualized trajectory, together with the plot of the total(-should-be-conserved)-energy, is generally an unmatched test to see whether something (and what, if the case) went wrong.
- Now that you have the anharmonic and the harmonic vibrational frequencies, try to plot them on top of one another to see the differences. The outputs are in arbitrary units, therefore you should scale one of the two spectra in order to compare them. Higher temperatures should show more anharmonic effects, while low temperatures should be closer to the harmonic result. Which peak shows more anharmonicity?

Timing: xxx

References

[1] I. Newton, *Philosophiæ Naturalis Principia Mathematica*, London (1687)

[2] D. Marx, M. E. Tuckerman, J. Hutter, and M. Parrinello Nature **397**, 601 (1999)

[3] J. Kolafa, Mol. Simul. **18**, 193 (1996)

[4] Kühne, Thomas D. *et al.*, Phys. Rev. Lett. **98**, 066401 (2007)

[5] D. Frenkel and B. Smit, *Understanding Molecular Simulation: from algorithms to applications*, second edition, Academic Press 2002.

[6] S. Nosé, J. Chem. Phys. **81**, 511 (1984). W.G. Hoover, Phys Rev. A **31**, 1695 (1985)

[7] G. Bussi, D. Donadio, and M. Parrinello, J. Chem. Phys. **126**, 014101 (2007)

[8] M. Ceriotti, G. Bussi, M. Parrinello, JCTC **6**, 1170 (2010)
`https://epfl-cosmo.github.io/gle4md/`

[9] G. Martyna, M. Klein, M. Tuckerman, J. Chern. Phys. **97**, 2635 (1992)

[10] M. Ceriotti, J. More, D. E. Manolopoulos, CPC **185** 1019 (2013)

[11] M. Ceriotti, D. E. Manolopoulos, PRL **109**, 100604 (2012)

[12] M-P. Gaigeot, M. Martinez, and R. Vuilleumier, Mol. Phys. **105**, 2857 (2007)

[13] J. Borysow, M. Moraldi, and L. Frommhold, Molec. Phys. **56**, 913 (1985)

[14] R. Ramirez, T. Lopez-Ciudad, P. Kumar, and D. Marx, J. Chem. Phys. **121**, 3973 (2004)