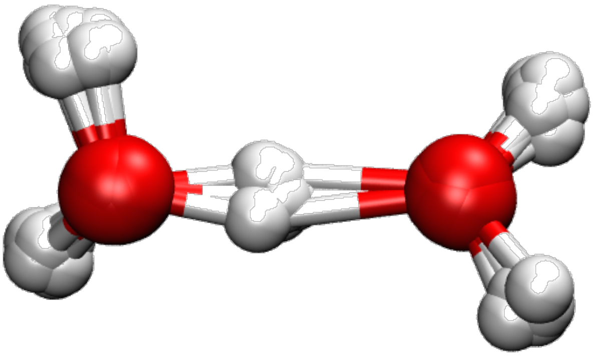
FHI-aims Developers' and Users' Meeting

Berlin, Germany, August 19 – August 22, 2014



Tutorial: Path integral molecular dynamics and the i-PI interface
Manuscript for Exercise Problems

Prepared by Mariana Rossi

Introduction

In this tutorial, the idea is to present the user with the infrastructure to calculate nuclear quantum effects through the i-PI interface with FHI-aims, by using path integral molecular dynamics (PIMD) and other methods derived from this formalism.

The Feynman path integral formalism [4] is the current method of choice to simulate nuclear quantum effects. Large systems can be treated with this formalism, and it can, in principle, be exact. Usually, exchange effects are ignored and the nuclei are treated as distinguishable particles. Exploring the fact that the quantum time propagator [$\exp(i\hat{H}t)$] is equal to the quantum density matrix [$\exp(-\beta\hat{H})$, where $\beta = 1/(k_BT)$] at imaginary time $i\beta\hbar$, one can rewrite quantum mechanics in the path integral formalism. The observables in quantum mechanics can be evaluated as quantum thermodynamical averages, provided that the partition function is known. The canonical partition function $Z =$Tr[$\exp(-\beta\hat{H})$] can be calculated by performing a Trotter factorization [5] of the trace, producing,

$$Z \approx \frac{1}{(2\pi\hbar)^{Nn}} \int d\mathbf{p}^{Nn} \int d\mathbf{q}^{Nn} \exp\left[\frac{-\mathcal{H}_n(\mathbf{q},\mathbf{p})}{nk_BT}\right], \tag{1}$$

where n is the so called Trotter number, corresponding to the number of identities introduced to factorize the trace. The non-commuting nature of the position and momentum operators gives rise to $\mathcal{H}_n$, which is nothing more than a classical ring-polymer Hamiltonian consisting of n “beads" connected by harmonic springs. It is given by

$$\mathcal{H}_n(\mathbf{q},\mathbf{p}) = \sum_I^N \sum_k^n \left[\frac{[\mathbf{p}_I^{(k)}]^2}{2M_I} + \frac{M_I\omega_n^2}{2}(\mathbf{q}_I^{(k)} - \mathbf{q}_I^{(k+1)})^2\right] + \sum_k^n V(\mathbf{q}^(k), \tag{2}$$

where V is the potential (e.g. the Born Oppenheimer potential from a DFT calculation), $\omega_n = nk_BT/\hbar$ and $\mathbf{q}_I^{(1)} = \mathbf{q}_I^{(n+1)}$. This formalism is exact (within the assumed approximations) in the limit of $n \to \infty$, but in practice one must use a finite amount of beads. Moreover, the kinetic energy term in Eq. 2 is not necessary for the evaluation of the canonical partition function, and is artificially introduced solely to allow sampling by molecular dynamics. The masses appearing in the denominator of this term are, thus, arbitrary - a property which can be explored in schemes of real time propagation which will be discussed below.

PIMD is, thus, 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.

PIMD increases the cost of a usual electronic structure 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 [3] 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.

Here we will treat the isolated Zundel cation ($H_5O_2^+$) since it is a small interesting system. Them whole machinery works for larger molecules and for periodic systems as well, but the computational resources available for this tutorial are not enough to have examples of those.

Exercise 1: Nuclear quantum effects in static properties

This exercise has two parts. In the first, a classical *ab initio* molecular dynamics simulation and a simulation containing approximate nuclar quantum effects will be perfomed – both already using the i-PI/FHI-aims interface. In the second part a path integral molecular dynamics simulation will be made, and the results for static properties (average potential and kinetic energy, and $g(r)$) will be compared for all different cases.

Exercise 1a: Approximate nuclear quantum effects with the colored noise thermostat

Here we will run a classical molecular dynamics simulation (for reference) and a simulation including nuclear quantum effects in an approximate fashion by using the colored noise GLE thermostats for the $H_5O_2^+$ molecule (the Zundel cation). We will use the LDA potential in FHI-aims simply because it is the faster DFT potential energy surface.

These simulations could be run natively in FHI-aims, but since the idea is to learn how to use i-PI, we will use this machinery already here.

1. Ab initio MD with classical nuclei

   In the first part of this exercise we will simply run a canonical (constant temperature) *ab initio* molecular dynamics simulation at room temperature.

   - *Server:* An example of an input file for i-PI can be found inside the `exercise_1/` folder, called `input_class.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 `/UsrAndDev/ipi/doc/manual.pdf`.

     Besides setting the number of beads (1, since it is a classical simulation!), there are two other fields that should be completed in the file: The xml block `<address>` regarding the socket communication and the field `<thermostat>` regarding the thermostat.

     `<address>`: 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. `udberlin2014-ex1`. 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. More detailed instructions about the input file can be found in manual of i-PI, in Sec. 5.1.2.

     `<thermostat>`: For the classical simulation we will use a velocity rescaling stochastic thermostat (as proposed in Ref. [1]). For that add:

     ```
     <thermostat mode="svr">
        <tau units="femtosecond">25</tau>
     </thermostat>
     ```

   - *Client:* An example of the `control.in` file is in the `exercise_1` folder. The only extra flag needed 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:udberlin-2014-ex1`, 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 /UsrAndDev/ipi/i-pi input_class.xml > out_class &
```

- Launch one instance of FHI-aims by typing:

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

- You will notice that in the i-PI output the kinetic energies of different species (O and H) are outputted in columns 5 and 6. Plot them to see how they behave for this classical simulation.
- When at least 2ps of simulation have gone by, use the `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 class-h5o2 1
```

  where the first field is the prefix of the position files of all the beads and the second field is the number of beads. The script writes the radial distribution function $g(r)$, decomposed into OH, OO, and HH pairs. The first column is the distance (in ångstroms) and the second column a histogram count. Plot these quantities and try to understand them.
- The full run takes about 25 minutes.

2. 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):

$$\dot{p}(t) = -\gamma p(t) + \sqrt{2m\gamma T}\xi(t) \tag{3}$$

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

$$\begin{aligned} \dot{q} &= p/m \quad (4)\\ \begin{pmatrix} \dot{p} \\ \dot{\mathbf{s}} \end{pmatrix} &= \begin{pmatrix} -V'(q) \\ 0 \end{pmatrix} - \mathbf{A}_p \begin{pmatrix} p \\ \mathbf{s} \end{pmatrix} + \mathbf{B}_p \left(\boldsymbol{\xi}\right), \quad (5) \end{aligned}$$

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

$$\mathbf{A}_p\mathbf{C}_p + \mathbf{C}_p\mathbf{A}_p^T = \mathbf{B}_p\mathbf{B}_p^T, \tag{6}$$

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

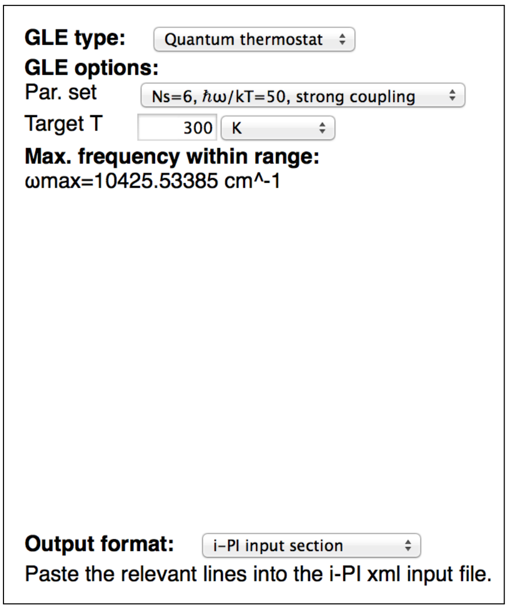
$$\begin{aligned} \dot{q} &= p/m \qquad (7) \\ \dot{p} &= -\frac{\partial V}{\partial q} - \int_{-\infty}^{t} Q(t-\tau)p(\tau) + \zeta(t), \qquad (8) \end{aligned}$$

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 feature that will be explored here.

- Make sure the previous calculation is over or kill it if you tou gathered already enough statistics for $g(r)$. In order to kill a calculation, type

  ```
  $ touch EXIT
  ```

  in the folder from where you launched the caculation.
- You will now use the `input_gle.xml` example to run i-PI. The input is essentially the same as the previous one, but the thermostat has to be changed.
- The parameters for this thermostat are quite complex matrices. Since we will use this thermostat to approximate nuclear quantum effects (we will not simulate a canonical ensemble), the matrices **A** and **C** (Eq. 5) 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 and substitute the thermostat block in the i-PI input file (starting from `<thermostat mode='gle'>`) .



GLE type: Quantum thermostat

GLE options:

Par. set Ns=6, ħω/kT=50, strong coupling

Target T 300 K

Max. frequency within range:

ωmax=10425.53385 cm^-1

Output format: i-PI input section

Paste the relevant lines into the i-PI xml input file.

- Launch the server by typing:

  ```
  $ python /UsrAndDev/ipi/i-pi input_gle.xml > out_gle &
  ```

- **Change** the address of the `control.in` file of FHI-aims to match the one of i-PI and launch one instance of FHI-aims by typing:

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

- While waiting for the simulations to complete, you can 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.

3. Again plot the kinetic energies of the different species from the i-PI output ('.out' file). What differences do you see?
4. Calculate the $g(r)$ also in this case by running the command below and compare it to the classical one

   ```
   $ python get_rdf_h5o2_PI.py gle-h5o2 1
   ```

   For which $g(r)$ of you see the larger differences between the classical and "quantum thermostat" distribution? Could you predict the differences? Can you understand them?

Exercise 1b: Nuclear quantum effects with path integral molecular dynamics

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.

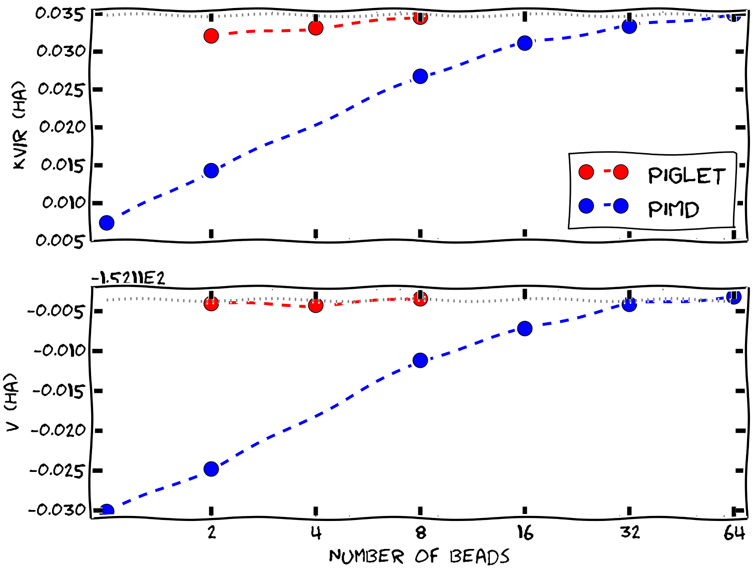


Figure 1: Convergence of the average virial estimator for kinetic energy $\langle K_{cv} \rangle$ and average potential energy $\langle V \rangle$ for the $H_5O_2^+$ molecule at 300K with path integral molecular dynamics.

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

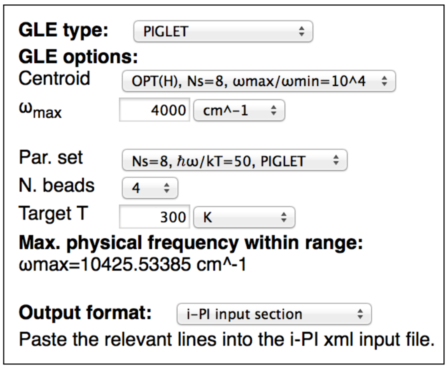
$$\langle V \rangle = \frac{1}{n} \sum_{i=1}^{n} \langle V(\mathbf{q}_i^1, ..., \mathbf{q}_i^N) \rangle \tag{9}$$

$$\langle K_{cv} \rangle = \frac{3Nk_BT}{2} + \frac{1}{2n} \sum_{j=1}^{N} \sum_{i=1}^{n} \langle (\mathbf{q}_i^j - \bar{\mathbf{q}}^j) \cdot \nabla V(\mathbf{q}_i^1, ..., \mathbf{q}_i^N) \rangle \tag{10}$$

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. [6]. 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:* Now you will use the `input_piglet.xml` example in your exercise folder. Perform three modifications: (1) Set the number of beads to **4**; (2) Write a new address for the UNIX socket; (3) modify the thermostat block as follows

- 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.



GLE type: PIGLET

GLE options:

Centroid OPT(H), Ns=8, ωmax/ωmin=10^4

ω_{max} 4000 cm^−1

Par. set Ns=8, ℏω/kT=50, PIGLET

N. beads 4

Target T 300 K

Max. physical frequency within range:

ωmax=10425.53385 cm^-1

Output format: i−PI input section

Paste the relevant lines into the i-PI xml input file.

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

- *Client:* The `control.in` for FHI-aims is the same as before, but you must again change the UNIX socket address so that it communicates with the server.

- Launch the server by typing:

```
$ python /UsrAndDev/ipi/i-pi input_piglet.xml > out_piglet &
```

- Launch four instances of FHI-aims by typing:

```
$ for i in {1..4};
  do (/UsrAndDev/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.

- The simulation here may take a long time, but you can look at the rdfs and other quantities while the simulation is running. Meanwhile take your time to read the introduction of the i-PI manual, which explains in detail the architecture of the code and its capabilities.
- Compare the $g(r)$ obtained here with the ones from the previous part of the exercise. For which species do you see most differences between the classical and the quantum case? Can you explain these results?
- Also look and compare quantities like the potential energy and the kinetic energy – how do they change between the three simulations that were run?

Exercise 2 (optional): Harmonic vibrations, "classical" anharmonic vibrations, and "quantum" 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 [10]:

$$I(\omega) = F(\omega) \int_{-\infty}^{\infty} dt\, \mathrm{e}^{i\omega t} \langle \mathrm{Tr}[\exp(-\beta \hat{H}) \hat{M}(0) \hat{M}(t)] \rangle_{t_0} \tag{11}$$

The angular brackets denote a statistical time average for the auto-correlation of the dipole moment of the molecule. What our simulations actually approximate is Kubo-transformed autocorrelation functions, defined by:

$$\tilde{c}_{MM} = \frac{1}{\beta Z} \int_0^{\beta} d\lambda \mathrm{Tr} \left\{ \exp[-(\beta - \lambda)\hat{H}] \hat{M}(0) \exp[-\lambda \hat{H}] \hat{M}(t) \right\}, \tag{12}$$

The relation between the Fourier transform of the canonical autocorrelation function and the Kubo-transformer one gives rise to the factor $F(\omega)$, which is proportional to ω^2 (see lecture for more details). Eq. 11 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.

Usually, one approximates the autocorrelation function by the one coming from a classical (AI)MD simulation, such that only classical anharmonicities are included. Within the path integral formalism, approximating these autocorrelation functions calls for the development of new approximate theories, since the time dependent information is not directly accessible in PIMD. Here we will use thermostated ring polymer molecular dynamics method (TRPMD) (see Ref. [7]). As will be seen in the exercise, although this method can capture quantum anharmonicities to a good extent, it cannot capture explicitly coherent

A harmonic (0K) spectrum of the molecule calculated with LDA is to be found in the folder `exercise_2/harmonic_vibs` for comparison.

The instructions follow:

1. **Classical anharmonic vibrations**

   - We will simulate at $T = 300$K (you may try other temperatures if you wish). The control.in and geometry.in files for FHI-aims will be the same as the one in the previous exercise, so copy it to the exercise_2/classical directory. There is an example for the i-PI input file to use there as well.

   - In the exercise_1/1a folder there should be a file called class-h5o2.restart. Copy it to this folder, since we will use this file to start a microcanonical (NVE) simulation from an already thermalized geometry. This is made possible in i-PI by simply specifying

     ```
     <initialize nbeads="1">
       <file mode="chk"> class-h5o2.restart </file>
       <velocities mode="thermal" units="kelvin"> 300 </velocities>
     </initialize>
     ```

     in the <system> block.

   - Launch the server by typing:

     ```
     $ python /UsrAndDev/ipi/i-pi input_nve.xml > out_nve &
     ```

   - Launch one instance of FHI-aims by typing:

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

   - The total dipole of the molecule will be gathered by i-PI in a file called class-h5o2-nve.dipole_0. For the analysis of this simulation, we will use the script auto-correlate-PI.py in order to see the evolution of the auto correlation function and of the spectrum with the time of the run (it is in the scripts folder). 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).

     In order to run auto-correlate-PI.py, you need to prepare an input file called 'control.autocorr.in'. Type:

     ```
     $ python auto-correlate-PI.py -h
     ```

     for an explanation of the flags that should be in control.autocorr.in and other help messages from the code. An example of this file is provided in the folder.

     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-AVG.py control.autocorr.in
     ```

     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).

- You can always watch the time-evolution of the molecule by looking at the positions output with the `VMD` program.
- There is not enough time to perform long enough simulations here – a lot more than we can simulate here is needed to achieve convergence. The resulting spectra of this one will be noisy. A spectra obtained from a longer run is provided in the solutions folder.
- 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?

2. **Anharmonic vibrations from thermostated ring polymer molecular dynamics**

   Here we will run a thermostated ring polymer molecular dynamics (TRPMD) simulation for $H_5O_2^+$. Even though these methods are more suitable for condensed phase situations (where the spectra become continuous), there is useful information that can be gathered from this small test-system. In particular $H_5O_2^+$ has a fermi resonance related to the shared proton (well studied in the literature), which cannot be described by TRPMD (or any standard PIMD-derived method). Quantum anharmonicities due to zero point energy and tunneling are well described, though.

   For this simulation, it is not possible to use the PIGLET thermostat to reduce the necessary number of replicas, since it would disturb the real time correlations. Therefore, the simulation here is too large to run in the computers and time we have available. The next steps are intended just to explain how to set up such a simulation. If you wish to leave it running over night, the results for one simulation should be ready in around 12 hours, but more simulations are needed. Results for comparison are provided in the `solution` folder. If more processors were available and we could run, say, 16 or 32 replicas where each one would use 4 CPUs, it would take the same amount of time as the classical simulation in the previous step, since the replicas are trivially paralellizable.

   - Again, the `control.in` and `geometry.in` files for FHI-aims will be the same as the one in the previous exercise, so copy it to the `exercise_2` directory.
   - In the `exercise_1/1b` folder there should be a file called `piglet-h5o2.restart`. Copy it to this folder, since we will use this file to start a TRPMD simulation from an already thermalized (PIMD) configuration. Here we will use 16 beads, which is still not fully converged at 300K, but enough for the purposes of the current exercise. Write in the `input_trpmd.xml` file:

     ```
     <initialize nbeads="16">
       <file mode="chk"> piglet-h5o2.restart </file>
       <velocities mode="thermal" units="kelvin"> 300 </velocities>
     </initialize>
     ```

     in the `<system>` block.

- A TRPMD simulation works by switching off any thermostats related to the centroid mode of the ring polymer and thermostating all internal modes with a optimally-damped Langevin thermostat at the corresponding free ring polymer frequency. It is very similar to the PILE thermostat proposed in Ref. [8], with no thermostat in the centroid. Therefore, in order to perform this simulation, your <ensemble> block in the example input_trpmd.xml file should look as follows:

  ```
  <ensemble mode="nvt">
    <thermostat mode="pile_g">
        <tau units="femtosecond">3000</tau>
        <pile_lambda> 0.5 </pile_lambda>
    </thermostat>
    <timestep units="femtosecond"> 0.5</timestep>
    <temperature units="kelvin">300</temperature>
  </ensemble>
  ```

  The very large τ parameter works to effectively turn off the centroid thermostat. Note also that we are using a smaller time step here, in order to correctly perform the integration of the equations of motion of the internal modes of the ring polymer.

- With the above modifications in place, launch again the server by typing

  ```
  $ python /UsrAndDev/ipi/i-pi input_trpmd.xml > out_trpmd &
  ```

- Make sure the UNIX socket address in the control.in file of FHI-aims matcher the one in the i-PI input file and run eight instances of FHI-aims:

  ```
  $ for i in {1..8};
    do (/UsrAndDev/bin/aims.ipi.x > aims${i}.out &) ; done
  ```

- In order to visualize the dynamics of all beads during the simulation you can load all bead positions in VMD

  ```
  $ vmd -m trpmd-h5o2.pos_0[0-8].xyz trpmd-h5o2.pos_1[1-5].xyz
  ```

- Now the total dipole of each bead will be gathered by i-PI in files called class-h5o2-nve.dipole_XX. In order to run auto-correlate-PI.py, you can copy and modify the previous control.autocorr.in file to analyze 16 replicas and run as you did in the classical case.

- For the sake of immediate comparison, take the converged spectrum from the solution folder and compare them. Which trends do you observe? (The outputs are in arbitrary units, therefore you should scale one of the two spectra in order to compare them.)

References

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

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

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

[4] R. P. Feynman and A. R. Hibbs, *Quantum Mechanics and Path Integrals*, Emended Edition, Dover (2010)

[5] H. F. Trotter, Proc. Amer. Math. Soc. **10**, 545 (1959)

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

[7] M. Rossi, M. Ceriotti, D. E. Manolopoulos, JCP **140**, 234116 (2014)

[8] M. Ceriotti, M. Parrinello, T. E. Markland, D. E. Manolopoulos, JCP **133**, 124104 (2010)

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

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