Advanced Path Integral Methods in a Simulation of Water Across its Phase Diagram

Michele Ceriotti

June 2016

In this set of exercises we will perform simulations of room-temperature liquid water, using a simple empirical forcefield model based on TIP4P-like point charges [1]. We will start from quickly checking the convergence of standard path integral molecular dynamics, and then use this example to test a number of advanced PI features. For the sake of speed of execution, we will use a very small (32 molecules) simulation box. However, if you have more time to carry out the exercise, you can modify the input to use the initial configurations in `data/water-216.pdb`. You should be able to complete at least the first three exercises in a couple of hours. *Hints: Do not wait for the full simulations to be complete: as soon as you have collected enough statistics to understand what's going on and to compare with previous runs, you should move on to the next exercise.*

Exercise 1 A PIMD simulation of liquid water

1. Look at the i-PI input file in `ex-1/base.xml`. Observe the properties and trajectory files that are specified in the `<output></output>` section. Make sure it is clear what each part of the input does. For this example we will use the PILE-G thermostat[2] and constant-temperature sampling. We will initially preform a reference molecular dynamics run, with $P = 1$.

```
$ cd ex-1
$ i-pi base.xml &> log.ipi &
$ lmp_ubuntu < in.water &> log.lammps &
```

2. Modify the input so that you'll run with $P = 6$ beads, by changing the `nbeads` option of the `<initialize>` field. It is advisable to make these changes on a copy of the `ex-1/base.xml` input file. You may also want to adjust `prefix` so that it indicates the new number of beads (e.g. `prefix="base_p-6"`). Repeat with $P = 32$. *Hints: Simulations may take some time to complete so you might want to launch them and continue with the following exercises. In order to run multiple i-PI instances simultaneously you should also change the name of the UNIX-domain socket.*

3. Analyze the outcomes of the simulations and check for convergence. You can quickly compute averages for thermodynamic quantities from the command line using the `autocorr` code, e.g. for the potential energy. *Hints:*

`autocorr` will also compute error estimates that take into account the autocorrelation time for the observable. With short trajectories, this tends to be significantly under-estimated. Note also that we are discarding a short part of the trajectory for equilibration - in a production calculation you should carefully check equilibration and discard a part of the trajectory accordingly.

```
$ tail -n +50 base_p-1.out | awk '{print $6}' | \
·· ·· ·· autocorr -maxlag 100 | head
```

4. Have a look at the trajectories with the VMD software *Hints: You can launch VMD from the command line with e.g. `vmd base_p-1.pos_0.pdb`.* You can load all of the ring polymer beads as separate molecules to visualize the spread of the ring polymer

   ```
   $ vmd -m base_p-32.pos_*.pdb
   ```

   and you can also use some of the i-PI post-processing tools to visualize the connections within the ring polymer. *Hints: Note that you have to load the file from the TCL prompt in VMD to disable automatic bond detection.*

   ```
   $ i-pi-mergebeadspdb base_p-32 > base_p-32_merged.pdb
   $ vmd
   vmd > mol new base_p-32_merged.pdb type pdb autobonds 0
   ```

   You can see clearly the different spread of the ring polymers of H and O atoms.

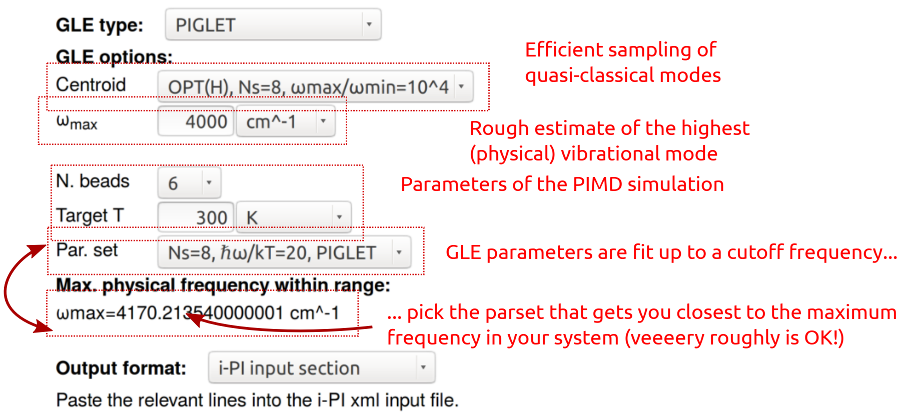
5. Compute radial distribution functions for different simulations. You can use any tool of your liking, but we suggest the `trajworks` post-processing tool that is part of the `toolbox` library. For instance, for one of the replicas in the 32-beads run,

   ```
   $ trajworks -ipdb -vbox -fstart 10 -gr -gr1 O -gr2 O \
   ·· ·· ·· ··-grmax 5 -grbins 250 -hwin triangle -hwinfac 2 \
   ·· ·· ·· ·· < base_p-32.pos_00.pdb > gOO.dat
   ```

Exercise 2 Colored-noise thermostatting

Move to the `ex-2/` folder. Here we will try to reduce the number of path integral replicas needed to converge a PIMD simulation by using a colored-noise thermostat, designed to selectively enhance fluctuations of high-frequency modes, so as to match the quantum predictions [3].

1. Look at the input file. The only significant difference relative to Exercise 1 is the change in the `<thermostat>` section. The matrix elements that are included in this section determine the behavior of the colored-noise dynamics, and have been fitted to give the correct quantum fluctuations in the harmonic limit. Parameters that are suitable for a broad range of conditions can be obtained from the on-line repository `http://epfl-cosmo.github.io/gle4md/index.html?page=matrix` – familiarize yourself with the interface.



GLE type: PIGLET

GLE options:

Centroid OPT(H), Ns=8, ωmax/ωmin=10^4

ω_{max} 4000 cm^-1

N. beads 6

Target T 300 K

Par. set Ns=8, ħω/kT=20, PIGLET

Max. physical frequency within range:

ωmax=4170.213540000001 cm^-1

Output format: i-PI input section

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

Efficient sampling of quasi-classical modes

Rough estimate of the highest (physical) vibrational mode

Parameters of the PIMD simulation

GLE parameters are fit up to a cutoff frequency...

... pick the parset that gets you closest to the maximum frequency in your system (veeeery roughly is OK!)

2. Run your simulation – the commands are pretty much the same as what you used in Exercise 1. Compute the mean quantum and kinetic energy, and compare them with the results you had from conventional PIMD.
3. Look also at the temperature of the ring polymer system. Remember: this temperature is only an indication of the (classical) sampling of the ring-polymer Hamiltonian and has no relation with the kinetic energy of the system. You will see that the mean temperature differs from the one specified in `<ensemble>`. This is an indication of the out-of-equilibrium nature of PIGLET, and is perfectly normal – nothing to worry about!
4. Look also at the radial distribution functions. Consider that sampling time is also an issue: from these short simulations one cannot really make conclusive statements on the convergence of different techniques.

Exercise 3 Variable-cell sampling

Let's now try to use the constant-pressure sampling features of i-PI to perform (colored-noise) path integral dynamics of water at high temperature and GPa pressure. Note that the most interesting physics in these conditions relates to high rates of self-ionization, that is observed in classical ab initio simulations [4], and is much enhanced by quantum effects [5]. Unfortunately, in this tutorial we can only use a non-dissociable forcefield and so we will not be able to explore this aspect.

1. A working example is provided in the input file `ex-3/npt.xml`, but don't run it just yet. What thermostat is being used? How many beads? Although one may think that at such a high temperature quantum effects will be mild, it is a good idea to use a PIGLET thermostat just to be sure. Generate input parameters from `http://epfl-cosmo.github.io/gle4md/index.html?page=matrix`, adapting them to the thermodynamic conditions considered here, substitute the `<thermostat>` field and launch your simulation. *Hints: Note that the barostat (a kind of a Langevin piston [6], adapted to PIMD [5]) contains a separate* `<thermostat>`*. You should not touch this, or at least*

make sure you keep a classical thermostat, since the degrees of freedom of the cell cannot be thermostatted with a GLE (why? think of how the GLE infers the fluctuations from the time scale of oscillations, and how the time scale of the cell dynamics is an arbitrary parameter).

2. Analyze the run by observing the convergence of the cell volume to the equilibrium value, and check that the pressure does fluctuate around the target value. You can also visualize the variable-cell trajectory in VMD. *Hints: You can visualize the simulation box with the TCL command*

   ```
   vmd > pbc box
   ```

Exercise 4 Doing PIMD Like a Pro

As a final example (in folder `ex-4/`) we will combine all the bits and pieces, performing a simulation of ice at 100K, combining two different techniques to speed up the calculation. We will use a fourth-order Suzuki-Chin factorization of the Boltzmann operator [7, 8], combined with a finite-difference integrator [**?**]. Ring-polymer contraction [9] can also be used, that speeds up simulations by evaluating long-range interactions on a reduced number of beads. We will not cover here the theory behind these techniques, so if you are not familiar with it, you are encouraged to read up the relevant literature before embarking in this exercise. *Hints: These start to be serious simulations, and it will not be possible to converge them at any decent level with a few minutes on a laptop. You can let them run overnight and/or run them on a more powerful computer if you want to familiarize yourself with the convergence behavior of these accelerated PI methods.*

1. Start by running the Suzuki-Chin simulation, that is

   ```
   $ i-pi highorder.xml &> log.ipi &
   $ lmp_ubuntu < in.ice &> log.lammps &
   ```

   While the simulation runs, look at the input file. You'll notice that we are outputting four more properties: `kinetic_opsc, kinetic_tdsc, potential_opsc , potential_tdsc`. These are the 'thermodynamic' and 'operator' versions of the Suzuki-Chin estimators for kinetic and potential energy. Note that – contrary to the Trotter case – the thermodynamic estimators are well-behaved when the number of beads is increased. You should see that after a few steps both the Td and the Op versions converge to very similar values, and that they are both significantly different from the conventional estimators (that do not correspond to any physical observable when sampling a fourth-order Hamiltonian, and are output only for didactic purposes). *Hints: Remember, when you compute averages, that you'll have to discard the (rather long) equilibration part, and that you should use the columns that correspond to the S-C estimators.*

2. Compute radial distribution functions. When inspecting trajectories, remember that in SC path integrals you can evaluate structural observables by analyzing the even beads from the trajectory, and ignoring the odd ones.

You can try to compute RDFs for even and odd beads separately, and verify that the two subsets sample different ensembles.

3. Finally, try to run the input file that also uses RPC

```
$ i-pi ultimate.xml &> log.ipi &
$ lmp_ubuntu < in.ice_long &> log.lammps &
$ lmp_ubuntu < in.ice_short &> log.lammps &
```

Try to follow the complex simulation set up. If you are familiar with the structure of LAMMPS input files, you'll notice that `in.ice_long` computes only Lennard-Jones and Coulomb terms, while bonded terms contain dummy values. Conversely, `in.ice_short` only computes the bonded (stretch and bend) interactions. These two LAMMPS instances connect to two separate `<ffsocket>` objects, that provide evaluators for the short and long-range components of the force. Now, within the description of the `<system>`, you will find the setup of the physical interactions, as they will be used by the integrator:

```
 <forces>
·· <force forcefield=’lammps.long’ nbeads=’8’></force>
·· <force forcefield=’lammps.short’ ></force>
 </forces>
```

This specifies that long-range forces should be computed on a contracted ring polymer with only eight beads. Conversely, the short-range component should be computed on the default (total) number of beads, as specified in the `<initialize>` section. Now compare the results with what you obtained with the `highorder.xml` input. You should observe only small differences, although considerably fewer long-range force evaluations have been used. *Hints: For a cheap potential as the one we consider here, communication overhead is dominant, and so the speed-up will not be very noticeable. Nevertheless, this strategy can be very advantageous when the long-range part of the interaction is time-consuming, such as in an ab initio simulation.*

References

[1] Scott Habershon, Thomas E Markland, and David E Manolopoulos. Competing quantum effects in the dynamics of a flexible water model. *J. Chem. Phys.*, 131:24501, 2009.

[2] Michele Ceriotti, David E Manolopoulos, and Michele Parrinello. Accelerating the convergence of path integral dynamics with a generalized Langevin equation. *J. Chem. Phys.*, 134:84104, 2011.

[3] Michele Ceriotti and David E Manolopoulos. Efficient First-Principles Calculation of the Quantum Kinetic Energy and Momentum Distribution of Nuclei. *Phys. Rev. Lett.*, 109:100604, 2012.

[4] Eric Schwegler, Giulia Galli, François Gygi, and Randolph Hood. Dissociation of Water under Pressure. *Phys. Rev. Lett.*, 87:265501, 2001.

[5] Michele Ceriotti, Joshua More, and David E. Manolopoulos. i-PI: A Python interface for ab initio path integral molecular dynamics simulations. *Comp. Phys. Comm.*, 185:1019–1026, 2014.

[6] Giovanni Bussi, Tatyana Zykova-Timan, and Michele Parrinello. Isothermal-isobaric molecular dynamics using stochastic velocity rescaling. 130:74101, 2009.

[7] M Suzuki. Hybrid exponential product formulas for unbounded operators with possible applications to Monte Carlo simulations. *Phys. Lett. A*, 201:425–428, 1995.

[8] Siu A. Chin. Symplectic integrators from composite operator factorizations. *Phys. Lett. A*, 226:344–348, 1997.

[9] Thomas E Markland and David E Manolopoulos. An efficient ring polymer contraction scheme for imaginary time path integral simulations. *J. Chem. Phys.*, 129:024105, 2008.

[10] M. Tuckerman, B. J. Berne, and G. J. Martyna. Reversible multiple time scale molecular dynamics. *J. Chem. Phys.*, 97:1990, 1992.