Elements on the Widom code

Gabriel Stoltz (stoltz@cermics.enpc.fr)

February 7, 2010

1 Overall structure

The source files are in the directoty `src/`, the input files are in the directory `input/`, the output files are in the directory `output/`, and the present documentation is in `doc/`. The codes should be compiled and run within the `src/` directory as they stand. The input file read by the executable should be self-explanatory, otherwise the user should have a look at the corresponding file `input_METHOD.cpp` which has been compiled depending on the method (see the `makefile` command file for a list of files required for the compilation).

Besides header files (extension `.hpp`), the source files include `gaussian.cpp` which is used to produce gaussian random numbers; `vector.cpp` and `matrix.cpp` which are simple vector and matrix classes defined in order to require the least amount of `C++` libraries; `main.cpp` which is the (concise) main files, in which a call to the function `load` is performed. This is the master function in the codes, which is defined in a file whose name depends on the method at hand; finally, forces and energy computations are performed with `hamiltonian.cpp`.

2 Model and input files

The parameters describing the microscopic interactions and the thermodynamic state of the system are the same for all input files:

- `Friction coefficient` γ used in the Langevin dynamics;
- `Inverse temperature` β;
- `Total number of particles` N (should be the square of an integer);
- `Elementary cell size` a, so that the density is $\rho = a^{-2}$;
- `LJ equilibrium distance` σ;
- `LJ energy epsilon` ε;
- `Cut off radius` $r_{\rm cut}$.

In all input files, the time step Δt used in the Langevin dynamics is given by the field `Time step`.

3 Methods

3.1 Canonical sampling

A simple canonical sampling can be performed by producing `sampling++` with the command `make sampling` There are two output files: `energy`, which gives the energy as a function of time (format: time, and then current energy, potential energy, insertion energy); while `xmakemol.xyz` is a movie of the system readable with the software `XMakeMol` (see `www.nongnu.org/xmakemol/`).

The additional parameters are

- `Number of iterations` is the total number of iterations;
- `Xmakemol output frequency` is the number of steps before a new configurations is written down;
- `Other outputs frequency` is the frequency at which the energies are saved.

3.2 Free-energy perturbation

The standard FEP method is obtained with the command `make fep`, which produces `FEP++`. This executable reads the input data in the file `input_file_FEP`:

- `Number of iterations` is the total number of integration steps
- `Number of thermalization steps` is the number of Langevin steps before the actual computation
- `Output frequency` gives the number of steps after which the output files are updated.

Output files are `energy` (same format as for the simple canonical sampling) and `DeltaF`, which is the current estimate of the free energy difference.

Umbrella sampling with respect to the distribution μ_θ can be performed. First, the code should be compiled as `make us` (which generates `US++`), and needs the additional line in `input_file_FEP` giving the parameter θ, namely `Umbrella Samp. param.` Same output files.

3.3 Thermodynamic integration

The compilation option is `make ti`, which produces `TI++`. The output files are: `energy` (same format as for the simple canonical sampling); `current_mean_force` which gives the running average of the mean force; `mean_force` and `PMF` which give respectively the estimated mean forces and free energy profiles (obtained by numerical integration of the mean force) at the end of the simulation.

The parameters are

- `Number of iterations` is the number of sampling iterations per value of the alchemical parameter λ;
- `Number of thermalization steps` is the number of preliminary thermalization steps before current averages of the mean force are computed;

- `Output frequency`

- `Number of intermediate stages` is the number of intermediate values of the alchemical parameter at which the mean force is estimated.

The λ schedule (*i.e.* the points chosen for the quadrature) should be changed directly in the file `TI.cpp`, see the function `schedule`.

3.4 Nonequilibrium dynamics

There are two options for simple nonequilibrium dynamics.

- The plain nonequilibrium dynamics with one-sided estimate of the free energy difference is compiled with `make jarz` (producing `jarz++`).

- Estimates obtained witht the bridge sampling method can be obtained with `make Bjarz` ('B' for 'BAR'), thereby producing `Bjarz++`. Both forward and backward switchings are realized.

The output files are, in all cases, `energy` (same format as for the simple canonical sampling), `work` which gives instantaneous work distributions at given times, and `DeltaF` which are the final estimates of the free energy difference (three results for the bridge sampling version, corresponding to forward, backward and bridge estimations). For the simple nonequilibrium dynamics, the final free energy profile is written in `PMF`.

The additional parameters are

- `Number of realizations` is the number of independent realizations of the estimation process;

- `Number of replicas` is the number of work values used for the average;

- `Number of thermalization steps` is the number of iterations used to generate the initial conditions starting from the crystal geometry;

- `Work distribution output freq.` is the number of iterations steps after which the instantaneous work distributions are saved;

- `Other output frequency` is the number of iterations steps after which the current energies are recorded;

- `Switching time` is the switching time for the schedule;

- `Selection` is set to 1 when selection is required.

The λ schedule (*i.e.* the points chosen for the quadrature) should be changed directly in the files `jarz.cpp` and `Bjarz.cpp`, see the function `schedule`.

3.5 Path sampling

Path sampling strategies can be resorted to with `make path` and `path++`. The output files are: `work` which gives the final work distributions, and `DeltaF` which is the current running estimate of the free energy difference.

The additional parameters are

- `Number of MC iterations` is the number of Monte-Carlo moves;
- `Number of thermalization steps` is an initial thermalization to have the first configuration on the first path;
- `Output frequency` is the number of iterations steps after which the current estimate of the free energy is recorded;
- `Switching time` is the switching time for the schedule; Switching time : 4.
- `Path weighting coefficient` is the coefficientα defining the path weight.

The λ schedule (*i.e.* the points chosen for the quadrature) should be changed directly in the files `jarz.cpp` and `Bjarz.cpp`, see the function `schedule`.