Different ensembles in molecular dynamics simulations

I. Microcanonical (NVE) ensemble

Molecular dynamics (MD) is the method of simulating kinetic and thermodynamic properties of molecular systems using Newton equations of motions. Usually numerical integration of equation of motions in MD is accomplished using Verlet algorithm. If all the forces, which appear in the Newton equation of motions, are related to the potential energy of the system, then the total energy of the system $E=E_{kin}+E_{pot}$ is conserved. If the total number of atoms N and the volume V (of the unit cell) are also kept constant, then the MD simulations are said to be performed in the microcanonical (NVE) ensemble. Generally, if the simulation system is sufficiently large, the small part of it may be considered as a canonical system. For large NVE systems the fluctuations in temperature are small, and it may be considered approximately constant. There are situations, in which temperature must be kept constant. For example, studying temperature induced unfolding of proteins requires precise temperature control. Therefore, for these classes of problems MD must reproduce an isothermal ensemble, such as canonical NVT ensemble, in which the number of particles, volume, and temperature are fixed.

II. Rescaling of velocities

The most straightforward way of temperature control is based on velocity rescaling. The distribution of velocities is drawn from the Maxwell-Boltzmann (MB) distribution

$$P(v_{i,\alpha}) = \left(\frac{m}{2\pi k_B T} \right)^{\frac{1}{2}} e^{-\frac{mv_{i,\alpha}^2}{2k_B T}} , \tag{1}$$

where $v_{i,a}$ is the α (=x,y,z) component of the velocity of the atom i. From Eq. (1) we obtain that the average kinetic energy per degree of freedom (associated with the velocity component $v_{i,\alpha}$) is related to temperature T (the equipartion theorem) as

$$\left\langle \frac{mv_{i,\alpha}^2}{2} \right\rangle = \frac{1}{2} k_B T , \tag{2}$$

where <…> brackets indicate the ensemble average. Because the ensemble average corresponds to the average over velocities of all atoms, then for the finite size system one may define an instantaneous temperature $T(t)$

$$k_B T(t) = \frac{1}{N_f} \sum_{i,\alpha} mv_{i,\alpha}^2 , \tag{3}$$

where N_f is the number of degrees of freedom. *T(t)* does not coincide with *T* used to generate velocity distribution in Eq. (1), and it fluctuates between successive generations of random velocities. To keep the temperature *T(t)* = *T*, one can rescale the velocities as

$$v^{'}_{i,\alpha} = \sqrt{\frac{T}{T(t)}} v_{i,\alpha} . \qquad (4)$$

This temperature control method does not reproduce canonical ensemble. To show this let us compute the variance in temperature in canonical system

$$\sigma^2 = \frac{\left\langle T^2(t) \right\rangle - \left\langle T(t) \right\rangle^2}{\left\langle T(t) \right\rangle^2} \ , \qquad (5)$$

where (we set k_B=1)

$$\left\langle T(t) \right\rangle = \frac{1}{N_f} \sum_{i,\alpha} \left\langle m v^2_{i,\alpha} \right\rangle \qquad (6)$$

and

$$\left\langle T^2(t) \right\rangle = \frac{1}{N_f^2} \sum_{i,\alpha} \sum_{j,\beta} \left\langle m^2 v^2_{i,\alpha} v^2_{j,\beta} \right\rangle . \qquad (7)$$

The averaging in Eqs. (6,7) is done using Eq. (1) by considering multiple realizations of the simulation system of *N*(=N_f/3) atoms. Since Eq. (7) can be written as

$$\left\langle T^2(t) \right\rangle = \frac{1}{N_f^2} \left[\sum_{i,\alpha} \left\langle m^2 v^4_{i,\alpha} \right\rangle + \sum_{i \neq j, \alpha \neq \beta} \left\langle m v^2_{i,\alpha} \right\rangle \left\langle m v^2_{j,\beta} \right\rangle \right] = \frac{1}{N_f^2} \left[N_f \left\langle v^4_\alpha \right\rangle + N_f (N_f - 1) \left\langle v^2_\alpha \right\rangle^2 \right] (8)$$

we obtain

$$\sigma^2 = \frac{1}{N_f} \frac{\left\langle v^4_\alpha \right\rangle - \left\langle v^2_\alpha \right\rangle^2}{\left\langle v^2_\alpha \right\rangle^2} = \frac{2}{N_f} \qquad (9)$$

In writing Eq. (9) we took into account that

$$\left\langle v^2_\alpha \right\rangle = \int v^2_\alpha P(v_\alpha) dv_\alpha = \frac{k_B T}{m} \qquad (10)$$

and

$$\left\langle v^4_\alpha \right\rangle = \int v^4_\alpha P(v_\alpha) dv_\alpha = 3 \left(\frac{k_B T}{m} \right)^2 \qquad (11)$$

Thus, even in canonical system instantaneous temperature *T(t)* does fluctuate and fixing it *(=T)* seriously perturbs the canonical ensemble. In fact, velocity rescaling does not reproduce any known type of ensemble. It is usually used for equilibration purposes, when a new distribution of velocities (Eq. (1)) is generated every step and scaled using Eq. (4) to a prescribed temperature. Rescaling of velocities alone (without generating a velocity distribution using Eq. (1)) does not equilibrate the system well, because it tends to prolong existing temperature differences (between solvent and solute).

NAMD uses several keywords to control velocity rescaling (Box 1). The keyword rescaleFreq specifies the frequency (in steps) of rescaling velocities, and the keyword rescaleTemp sets the target temperature. The keyword temperature determines the temperature, at which the initial velocity distribution (Eq. (1)) is drawn.

III. Weak coupling with heat bath

Temperature may also be controlled using the method of weak coupling to a thermal bath proposed by Berendsen (*J. Chem. Phys.* **81**, 3684 (1984)). In this case coupling either removes or adds energy to the system to maintain constant temperature. The velocities are scaled at each step so that the rate of temperature change is given by

$$\frac{dT}{dt} = \frac{1}{\tau}\left(T_0 - T\right) \ , \qquad (12)$$

where T_0 is the target temperature. Eq. (12) implies that if $T<T_0$, the temperature will increase, but if $T>T_0$, the heat will be removed. Let the velocities be rescaled as $v_{i,a}' = \lambda v_{i,a}$. Then the change in kinetic energy upon rescaling is

$$\Delta E = \frac{1}{2}\left(\sum_{i,\alpha}(mv'^2_{i,\alpha} - mv^2_{i,\alpha})\right) = \left(\lambda^2 - 1\right)E_{kin} = \left(\lambda^2 - 1\right)\frac{3Nk_BT}{2} . \qquad (13)$$

Eq. (13) can be converted to express temperature change as

$$\Delta T = \left(\lambda^2 - 1\right)T . \qquad (14)$$

Approximating dT/dt with $\Delta T/\Delta t$ we get

$$\lambda^2 = 1 + \frac{\Delta t}{\tau}\left(\frac{T_0}{T} - 1\right), \qquad (15)$$

which gives the relation between scaling factor λ, the time step Δt and the coupling constant (the time interval between heat exchanges with the bath) τ. If $\Delta t << \tau$, then Eq. (15) implies that no rescaling takes place and we recover microcanonical ensemble. If $\Delta t = \tau$, then the standard velocity rescaling occurs (Section II). The usual compromise value for τ is 0.4 ps, which results in modest temperature fluctuations. The Berendsen thermostat does not strictly fix the temperature, but leads to exponential relaxation of instantaneous temperatures to a target one. Also from physical viewpoint the coupling to heat bath can be viewed as addition of a friction term $-\gamma mv$ to the Newton equations of motion, where $\gamma = \frac{1}{\tau}\left(\frac{T_0}{T} - 1\right)$.

There is no clear proof that Berendsen thermostat reproduces canonical distribution. Berendsen and coworkers have shown that weak coupling ($\tau >$ 0.01 ps) does lead to correct canonical averages, but still perturbs the fluctuations of various quantities.

NAMD offers Berendsen thermostat as an extra option for temperature control. The keywords tCouple and tCoupleTemp switch the Berendsen coupling on and specify the bath temperature T_0, respectively. The keywords tCoupleFile and tCoupleCol indicate the name of the PDB file and the specific column with the reversed value of the coupling constant $1/\tau$, respectively. The column may correspond to coordinate, occupancy, or temperature factor fields. The advantage of Berendsen thermostat is simplicity and easy control over temperature coupling.

IV. Stochastic coupling

A simple method for temperature control was proposed by Andersen (*J. Chem. Phys.* **72**, 2384 (1980)). In this method each atom at each integration step is subject to small probability to experience collision with heat bath. Let us denote the frequency of collisions as ν. If collisions are uncorrelated, then the probability of next collision event is given by a Poisson distribution $P(t)=\nu exp(-\nu t)$. In the limit of small time steps Δt (i.e., << 1/ν) the probability of collision is $\nu\Delta t$. If an atom collides with the heat bath, its velocity is drawn from a MB distribution. Importantly, the velocities of other atoms are not affected.

It has been rigorously shown that by mixing deterministic MD with stochastic collisions Andersen thermostat creates a Markov process. Therefore, Andersen thermostat does correspond to canonical (NVT) ensemble. It is important to keep in mind that Andersen thermostat breaks the continuity of MD trajectories and does not preserve energy and momentum. In fact, Andersen thermostat leads to a series of microcanonical simulations, which are randomly interrupted, when the system is transferred to a new energy level. The values of ν should be chosen as a result of a compromise between two factors. Rare collisions do not create sufficient temperature fluctuations and the system remains close to microcanonical. Too frequent collisions effectively turn Andersen scheme into velocity reassignment.

The algorithm implementing Andersen thermostat is shown in Box 2 (p. 142 in Frenkel and Smit book), which shows that MD based on Andersen thermostat reproduces canonical ensemble. The comparison of MD and analytical results demonstrates excellent agreement (Fig. 6.2 in Frenkel and Smit book). Of special importance is Fig. 6.3, which shows that diffusion coefficient depends on the collision frequency. Effectively, collisions with the heat bath decorrelate atoms' velocities and reduce their diffusion coefficients. If collisions are rare (ν is small), the dependence of the average atom displacement $\Delta r^2(t)$ is close to a "true" one, but increasing the frequency progressively leads to slower diffusion behavior. Therefore, one cannot use Andersen thermostat to compute dynamic quantities, such as lifetime of water hydrogen bonds. NAMD does not implement Andersen thermostat as an option for temperature control.

V. Nose-Hoover method

Nose and Hoover proposed a method that generates a canonical ensemble (*J. Chem. Phys.* **81**, 511 (1984); *Mol. Phys.* **52**, 255 (1984); *Phys. Rev. A* **31**, 1695 (1985)). The sketch of the Nose-Hoover approach is as follows. Let us add to the energy function a fictitious degree of freedom (coordinate) *s*, which represents a heat bath. The potential energy associated with *s* is $(3N+1)k_BT \ln s$, where $3N+1$ is the total number of degrees of freedom. The kinetic energy related to *s* is $Q(ds/dt)^2/2$, where Q plays the role of fictitious mass. Consider an extended system, which includes a real system and a heat bath represented by *s*, and assume that the extended system is microcanonical. It can be shown that MD simulations of extended system produce a canonical ensemble in the real system due to heat exchange between fictitious degree of freedom and real system. The coupling between the two is controlled by Q. Small Q leads to high frequency oscillations of energy coupled with *s* and frequent collisions with the real system. In the limit of large Q microcanonical ensemble of real system is retained. The notable complication of Nose-Hoover thermostat is that time interval in the extended system $\Delta t'$ does not correspond to the real time Δt and must be scaled according to $\Delta t=s\Delta t'$.

This complication results in nonuniform integration step in the dynamics of real system. Nose-Hoover thermostat producing NVT ensemble is most often used in Charmm MD program, and is not implemented in NAMD.

An important method of temperature control is Langevin dynamics. In NPT ensemble, pressure and temperature are kept constant. The NPT ensemble is used for comparison of MD simulations with experiments. Temperature in NPT ensemble is controlled using Langevin method. The NPT ensemble is implemented in NAMD.

Box 1. Example of the NAMD configuration file for Aβ11-42 dimer

```
# input system......
structure  ab_m2_dimer_solv.psf
coordinates      ./output/ab_quench04149.coor
bincoordinates   ./output/abi_quench04149.coor
binvelocities     ./output/abi_quench04149.vel

#..force field.......................
paratypecharmm on
parameters  par_all22_na.inp
parameters  par_all22_prot.inp
exclude  scaled1-4
1-4scaling       1.0
dielectric       1.0
switching  on
switchdist  8.0
cutoff  12.0
pairlistdist 13.5
margin  0.0
stepspercycle 20
rigidBonds      all
rigiΔTolerance  0.00001
rigidIterations 100

# Ewald EL.........................
PME             on
PMETolerance    0.000001
PMEGridSizeX    32                        # grids for fast evaluation in Fourier sum
PMEGridSizeY    32                        # note that 32=2^5
PMEGridSizeZ    32                        #

#integrator ............
timestep 1.0
fullElectFrequency  4

#output...................
outputenergies 1000
outputtiming 1000
binaryoutput no
outputname output/ab_quench04150
restartname     output/abi_quench04150
restartfreq     10000
binaryrestart   yes
DCDfile         output/ab_quench04150.dcd
dcdfreq  1000

#MD protocol.............
seed                32204150
numsteps        40000
```

```
temperature     300                        # temperature of initial MB velocity distribution

#..........Velocity rescaling…..
rescaleFreq     1                          # frequency of velocities’ rescaling
rescaleTemp     300                        # target temperature for velocity rescaling

#……………….OR Berendsen thermostat……………
tCouple         on                         # turning Berendsen thermostat on
tCoupleTemp     300                        # target temperature
tcoupleFile     ab_Berendsen.pdb           # name of the pdb file with coupling constants
tCoupleCol      O                          # column in pdb file with coupling constants
……………………………………………..

#reassignFreq 1
#reassignTemp   300
#reassignIncr 0.001
#reassignHold 300

# periodic boundary conditions......
cellBasisVector1   57.8   0.0  0.0
cellBasisVector2    0.0  57.8  0.0
cellBasisVector3    0.0   0.0 57.8
cellOrigin          0.0   0.0  0.0
wrapWater        on
```

Box 2. Verlet integrator with Andersen thermostat

```
program MDAndersen

call init(temp)                         # initialization of velocities and coordinates
call force(f,en)                        # computation of forces
t=0
do while(t< tmax)
  call integrate(1,f,en,temp)           # applying Verlet algorithm for new positions
  call force(f,en)                      # getting forces for new position
  call integrate(2,f,en,temp)           # getting new velocities and attempting
                                        # collision
  t = t + dt
enddo
stop
end program MDAndersen

subroutine integrate(switch,f,en,temp)
 if(switch == 1)then                    # first pass through the Verlet algorithm
   do i=1,npart
      x(i) = x(i) + dt*v(i)+dt*dt*f(i)/2   # new positions
      v(i) = v(i) + dt*f(i)/2              # first half of new velocities
   enddo
 elseif(switch == 2) then                  # second pass through Verlet algorithm
  tempa = 0                                # current temperature initialization
  do i=1,npart
    v(i) = v(i) + dt*f(i)/2                # new velocities
    tempa = tempa + v(i)*v(i)              # getting current temperature
  enddo
  tempa = tempa / npart
  do i=1,npart                             # applying Andersen thermostat
     if(ranf() < nu*dt) v(i) = gauss()     # on particle I with the probability n*ΔT
  enddo                                    # velocities are drawn from Gaussian
                                           # distribution
  endif
  return
end subroutine integrate
```