3. Thermostats in Molecular dynamics

In these notes we briefly recap the velocity Verlet algorithm, and we discuss how this can be modified to sample the NVT ensemble (where the temperature T is fixed, together with the number of particle N and the volume of the simulation box V), instead of the NVE ensemble (where it is the total energy E which is constant, together with N and V). The NVT ensemble is relevant when the system of interest is in contact with a heat bath (this is the common situation in physics). Note that the NVE and NVT ensembles are also known as, respectively, the microcanonical and the canonical ensemble. The latter is familiar to us from the Monte-Carlo algorithm, which samples a system in thermodynamic equilibrium in this ensemble, where the weight of a microscopic state is given by its Boltzmann weight $\exp[-E/(k_B T)]$.

1.1. The velocity Verlet algorithm

We begin by briefly recapping the velocity Verlet algorithm, which allows us to integrate the Newton equations of motion, characteristic of the NVE ensemble, namely

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = -\sum_{j \neq i} \nabla_i U(|\mathbf{r}_i - \mathbf{r}_j|). \tag{1}$$

In Eq. 1 m_i is the mass of particle i, $\mathbf{r}_i = (x_i, y_i, z_i)$ is the position of particle i in 3-dimensional space, $\nabla_i = (\frac{\partial}{\partial x_i}, \frac{\partial}{\partial y_i}, \frac{\partial}{\partial z_i})$, while U denotes the potential energy.

The most used algorithm used in practice to integrate Eq. 1 is the velocity Verlet algorithm, whose steps are implemented as follows:

$$\begin{aligned}
\mathbf{r}_i(t+\delta t) &= \mathbf{r}_i(t) + \mathbf{v}_i(t)\delta t + \frac{\mathbf{f}_i(t)}{2m_i}\delta t^2 \\
\mathbf{v}_i(t+\delta t/2) &= \mathbf{v}_i(t) + \frac{\delta t}{2}\frac{\mathbf{f}_i(t)}{m_i} \\
\mathbf{f}_i(t+\delta t) &= \mathbf{f}_i(\mathbf{r}_i(t+\delta t)) \\
\mathbf{v}_i(t+\delta t) &= \mathbf{v}_i(t+\delta t/2) + \frac{\delta t}{2}\frac{\mathbf{f}_i(t+\delta t)}{m_i}
\end{aligned} \tag{2}$$

where $\mathbf{r}_i$, $\mathbf{v}_i$ and $\mathbf{f}_i$ denote the position of the $i-$th particle, its velocity, and the force it is subjected to. As there are no dissipative forces, the energy is conserved within this algorithm.

In practice, the following, equivalent, version of the velocity Verlet is probably the most used: The most used algorithm used in practice to integrate Eq. 1 is the velocity Verlet algorithm, whose steps are implemented as follows:

$$\begin{aligned}
\mathbf{v}_i(t+\delta t/2) &= \mathbf{v}_i(t) + \frac{\delta t}{2}\frac{\mathbf{f}_i(t)}{m_i} \\
\mathbf{r}_i(t+\delta t) &= \mathbf{r}_i(t) + \mathbf{v}_i(t+\delta t/2)\delta t
\end{aligned} \tag{3}$$

$$\begin{aligned}
\mathbf{f}_i(t+\delta t) &= \mathbf{f}_i(\mathbf{r}_i(t+\delta t)) \\
\mathbf{v}_i(t+\delta t) &= \mathbf{v}_i(t+\delta t/2) + \frac{\delta t}{2}\frac{\mathbf{f}_i(t+\delta t)}{m_i}.
\end{aligned}$$

As a simple exercise, you should convince yourself that the two versions in Eq. 2 and Eq. 3 are equivalent.

1.2. The Nose-Hoover thermostat

The Nose-Hoover thermostat provides a way to simulate a system which is (asymptotically, i.e. at large times) in the NVT ensemble. The idea is to introduce a fictitious dynamical variable, whose physical meaning is that of a friction, ζ, which slows down or accelerates particles until the temperature (measured through the kinetic energy and the equipartition function, see below) is equal to the desired value. The equations of motions (in 3D) are:

$$m_i \frac{d^2\mathbf{r}_i}{dt^2} = \mathbf{f}_i - \zeta m_i \mathbf{v}_i \tag{4}$$

$$\frac{d\zeta(t)}{dt} = \frac{1}{Q}\left[\sum_{i=1}^{N} m_i \frac{\mathbf{v}_i^2}{2} - \frac{3N+1}{2}k_B T\right], \tag{5}$$

where Q in Eq. 5 determines the relaxation of the dynamics of the friction, $\zeta(t)$, while T denotes the target temperature. It can be seen that in steady state, where $\frac{d\zeta}{dt} = 0$, the kinetic energy is given by $\frac{3}{2}(N+1)k_B T$ as required by equipartition (there is a factor of $3N+1$ instead of $3N$ as there is one more degree of freedom, ζ). It is important to note that the temperature is therefore not fixed, rather it *tends to* the target value.

The equations of motion of the Nose-Hoover thermostat can be implemented by a small modification of the velocity Verlet algorithm – an option is given below.

The first four steps in the modified discretisation algorithm are:

$$\begin{aligned}
\mathbf{r}_i(t+\delta t) &= \mathbf{r}_i(t) + \mathbf{v}_i(t)\delta t + \left(\frac{\mathbf{f}_i(t)}{m_i} - \zeta(t)\mathbf{v}_i(t)\right)\frac{\delta t^2}{2} \qquad (6)\\
\mathbf{v}_i(t+\delta t/2) &= \mathbf{v}_i(t) + \frac{\delta t}{2}\left(\frac{\mathbf{f}_i(t)}{m_i} - \zeta(t)\mathbf{v}_i(t)\right) \\
\mathbf{f}_i(t+\delta t) &= \mathbf{f}_i(\mathbf{r}_i(t+\delta t)) \\
\zeta(t+\delta t/2) &= \zeta(t) + \frac{\delta t}{2Q}\left[\sum_{i=1}^{N} m_i \frac{\mathbf{v}_i(t)^2}{2} - \frac{3N+1}{2}k_B T\right].
\end{aligned}$$

To match the two-step character of the velocity Verlet algorithm, note that also $\zeta(t)$ is first updated at time $t+\delta t/2$. The final steps of the Nose-Hoover-Verlet algorithm are:

$$\zeta(t+\delta t) = \zeta(t+\delta t/2) + \frac{\delta t}{2Q}\left[\sum_{i=1}^{N} m_i \frac{\mathbf{v}_i(t+\delta t/2)^2}{2} - \frac{3N+1}{2}k_B T\right] \tag{7}$$

$$\mathbf{v}_i(t+\delta t) = \frac{\left[\mathbf{v}_i(t+\delta t/2) + \frac{\delta t}{2}\frac{\mathbf{f}_i(t+\delta t)}{m_i}\right]}{1+\frac{\delta t}{2}\zeta(t+\delta t)}$$

where the last equation is slightly more complicated than its counterpart in the NVE velocity Verlet algorithm, because the dissipative force over mass term, $\zeta\mathbf{v}$, is computed at time $t+\delta t$.

1.3. Brownian dynamics: a stochastic thermostat

Another option to simulate a system in the NVT ensemble is to use a stochastic thermostat, as opposed to the deterministic thermostat defined through the Nose-Hoover equations, Eqs. 4–5. This thermostat again requires the introduction of dissipative forces, through friction, which physically comes from fluctuating forces on a moving particle due to the chaotic motion of solvent molecules. For the system of equations to be physically meaningful, we also need to include a stochastic term to account for such fluctuating forces.

The presence of a stochastic force term renders this thermostat more complicated from a theoretical point of view. However, it is also one of the most commonly employed in practice, because of its very good performance. One of the reasons why this thermostat works so well is that, in this case, the friction is a real, physical parameter (incidentally, this means that it is now constant, and does not evolve as in the Nose-Hoover thermostat). The equations of motion of a system with a stochastic thermostat are known as Brownian dynamic equations, and for particle i (in 3D) these are:

$$\begin{aligned} m_i\frac{d^2\mathbf{r}_i}{dt^2} &= \mathbf{f}_i - \zeta m_i\mathbf{v}_i + \sqrt{2k_BT\zeta m}\mathbf{h}_i \qquad (8)\\ \langle\mathbf{h}_i\rangle &= \mathbf{0}\\ \langle\mathbf{h}_i(t)\cdot\mathbf{h}_j(t')\rangle &= 3\delta_{ij}\delta(t-t'), \end{aligned}$$

where in the last equation δ_{ij} denotes a Kronecker delta, while $\delta(t-t')$ denotes a Dirac delta. Note that in Eq. 8 the stochastic term $\mathbf{h}_i$ has a coefficient, $\sqrt{2k_BT\zeta m_i}$, which contains the friction, ζ. This relation is a consequence of the so-called "fluctuation-dissipation theorem", which relates fluctuation-related quantities (such as diffusion coefficient, or noise strength) to dissipative quantities (such as friction). Another consequence of the fluctuation-dissipation theorem is that the diffusion coefficient of a particle in a solvent is given by $D = k_BT/(m_i\zeta)$ – a relation which only holds when the dissipative and stochastic forces have exactly the form in Eq. 8. (The proof of this requires a bit more knowledge of statistical physics and in particular of the Langevin equation: we do not give it here.) Finally, note that the noise term $\mathbf{h}_i$ is assumed to be Gaussian, so that it is uniquely determined by its first two moments (given in Eq. 8).

Finally, we give explicitly here a modified (velocity) Verlet algorithm which uses Brownian dynamics as a thermostat. The simplest way to do this is to

define a total force as follows:

$$\begin{aligned} \mathbf{f}_i^{\text{tot}}(t) &= \mathbf{f}_i(t) - \zeta m \mathbf{v}_i(t) + \sqrt{\frac{2k_B T \zeta m}{\delta t}} \tilde{\mathbf{h}}_i(t) \\ \langle \tilde{\mathbf{h}}_i(t) \rangle &= \mathbf{0} \\ \langle \tilde{h}_{i,\alpha} \tilde{h}_{i,\beta}(t) \rangle &= \delta_{\alpha\beta}. \end{aligned} \tag{9}$$

In the equations above, α and β denote Cartesian components (x, y or z in 3D), and, importantly, $\tilde{\mathbf{h}}_i$ is now a standard vector, each component of which are Gaussian random variable with mean zero and variance 1 (as opposed to a Dirac delta, which is difficult to discretise). Note that the stochastic term has been scaled by $\frac{1}{\sqrt{\delta t}}$, where δt is the time step – this is dimensionally sensible (as $\delta(t - t')$ has the dimension of an inverse time) – the deep reason behind this choice however requires, once more, a bit more work, and we leave it out. Once the continuum noise $\mathbf{h}_i$ has been replaced by $\frac{\tilde{\mathbf{h}}_i}{\sqrt{dt}}$, it is a simple matter to generate this noise through standard random number generator (as all we need is a random number generated according to a Gaussian probability distribution).

Once the force is defined in such a way, we can include it in an implementation of the velocity Verlet algorithm. Here we give the discretisation used in the code LAMMPS (an acronym which stands for Large-scale Atomic/Molecular Massively Parallel Simulator), which is a very well known molecular dynamics code. The relevant discretised formulas are,

$$\begin{aligned} \mathbf{v}_i(t + \delta t/2) &= \mathbf{v}_i(t) - \frac{\delta t}{2} \left(\frac{\nabla_i U(t)}{m_i} + \zeta \mathbf{v}_i(t) \right) + \sqrt{\frac{\delta t k_B T \zeta}{m_i}} \tilde{\mathbf{h}}_i \\ \mathbf{r}_i(t + \delta t) &= \mathbf{r}_i(t) + \mathbf{v}_i(t + \delta t/2) \delta t \\ \mathbf{v}_i(t + \delta t) &= \mathbf{v}_i(t + \delta t/2) - \frac{\delta t}{2} \left(\frac{\nabla_i U(t + \delta t)}{m_i} + \zeta \mathbf{v}_i(t + \delta t/2) \right) + \sqrt{\frac{\delta t k_B T \zeta}{m_i}} \tilde{\mathbf{h}}_i, \end{aligned} \tag{10}$$

where you should note that due to the Verlet scheme the time step in each of the two velocity updates is now $\delta t/2$, rather than δt. You should also recall that $\tilde{\mathbf{h}}_i$ is a random vector, each components of which have zero mean and unit variance: a different random vector should be generated, for each particle, for each velocity update.