

Today's lecture: how to perform molecular dynamics at constant temperature, for systems with rigid bonds, and for systems with multiple time scales

Thermostats

As we have presented it so far, molecular dynamics is performed in the microcanonical ensemble of constant NVE variables (and technically also: total momentum P). We found that we could estimate the temperature using thermodynamic averages. It is often desirable, however, to specify the temperature a priori and perform a simulation in the **canonical** ensemble. There are several approaches to doing this. We must consider whether the methods

- preserve the correct **thermodynamics**, i.e., the correct microstate distribution in the canonical ensemble
- preserve realistic **dynamics** in that the equations of motion can be used to compute transport quantities accurately

With regards to the thermodynamics of the system, we want to reproduce the ensemble probabilities of the canonical ensemble:

$$\wp(\mathbf{p}^N, \mathbf{r}^N) \propto \frac{e^{-\beta H(\mathbf{p}^N, \mathbf{r}^N)}}{Q(T, V, N)}$$

where

$$Q(T, V, N) = \frac{1}{h^{3N} N!} \int e^{-\beta H(\mathbf{p}^N, \mathbf{r}^N)} d\mathbf{r}^N d\mathbf{p}^N$$

Velocity rescaling

We have already discussed this approach. Consider the following estimator of the temperature:

$$T = \frac{2\langle K \rangle}{k_B n_{\text{DOF}}}$$

The idea is that we rescale the velocities at each step (or after a preset number of steps) so that the kinetic energy gives the desired target temperature:

$$\mathbf{v}_{\text{new}}^N = \lambda \mathbf{v}^N$$

$$\lambda = \sqrt{\frac{T}{T_{\text{inst}}}} \quad T_{\text{inst}} = \frac{2K}{k_B n_{\text{DOF}}}$$

Does this approach generate the correct thermodynamic properties of the canonical ensemble? It turns out that it does not. Consider the limit in which we rescale the velocities at every time step. In this case, the kinetic energy will remain constant in time, with zero fluctuations. This is not correct, as the statistical mechanics of the canonical ensemble says that:

$$\sigma_K^2 = \frac{3}{2} N k_B^2 T^2$$

In other words, velocity rescaling does not capture the correct kinetic energy fluctuations.

Berendsen thermostat

The Berendsen thermostat [Berendsen et al., 1984] is similar to the velocity rescaling approach, but assigns a time scale for the updating of the velocities, rather than assume they are completely scaled to the target temperature at each time step. Underlying this approach is the assumption that the system is weakly coupled to a heat bath whose **coupling constant**, or time scale of heat transfer, is τ .

$$\lambda^2 = 1 + \frac{\delta t}{\tau} \left(\frac{T}{T_{\text{inst}}} - 1 \right)$$

Here, δt is the time step in the molecular dynamics simulation. Typically, $\tau = 0.1 - 0.4$ ps for $\delta t = 1$ fs. λ is the velocity rescaling factor as before.

The Berendsen thermostat, however, suffers from the same problems as velocity rescaling in that the energy fluctuations are not captured correctly. Moreover, it does not generate a correct canonical ensemble, even for the configurational degrees of freedom, and results in notable pathologies for some systems. Generally it should always be avoided.

Andersen thermostat

The Andersen thermostat [Andersen, 1980] introduces a stochastic element to the temperature by having **random collisions** of molecules with an imaginary heat bath at the desired temperature. In the **single-particle** approach, a random particle is chosen and its velocity is reassigned randomly from a Maxwell-Boltzmann distribution at the desired temperature:

$$\wp(v_{x,i}) = \left(\frac{m_i}{2\pi k_B T} \right)^{\frac{1}{2}} \exp\left(-\frac{m_i v_{x,i}^2}{2k_B T} \right)$$

This equation is applied to each component of the particle's velocity.

Alternatively, a **massive collision** can be used in which each velocity component of every particle is reassigned simultaneously, pulling separately from this distribution $3N$ times. After such events, it is typical to remove any center-of-mass motion in the system.

In the Andersen scheme, one does not perform a collision with each molecular dynamics time step, but rather it is customary to adopt a **collision frequency** ν or collision time $\tau = 1/\nu$. The collision frequency should be chosen so as not to be too short to the time scales of molecular motions.

In the limit of an infinitely long trajectory averaged over many heat bath collisions, it can be shown that the Andersen thermostat **rigorously generates the correct canonical ensemble** probabilities. That is, the distribution of kinetic and potential energies, and the microstate probabilities of different configurations of the system, all rigorously approach their true form in the canonical ensemble.

However, the presence of random collisions causes the velocities of particles to decorrelate (“lose memory” of their initial values from some previous point in time) much faster than the NVE dynamics. As a result, true molecular **kinetics** are not preserved by the Andersen thermostat. For example, the computed diffusion constants for particles would give erroneous values.

Nose thermostat

Nosé devised a useful way of maintaining a system at constant temperature by a reformulation of the equations of motion [Nosé, 1984]. This can be rigorously formulated using the Lagrangian approach to classical mechanics; however, here we will only review the major results starting at the level of the Hamiltonian. For a detailed derivation, the reader is referred to Frenkel and Smit.

Recall for a Cartesian system that the Hamiltonian takes the form

$$\begin{aligned} H &= K(\mathbf{p}^N) + U(\mathbf{r}^N) \\ &= \frac{1}{2} \sum \frac{|\mathbf{p}_i|^2}{m_i} + U(\mathbf{r}^N) \end{aligned}$$

Nosé’s approach was to add two additional degrees of freedom to the system, with broad interpretations below:

- s - the “position” of an imaginary heat reservoir to which the system is coupled
- p_s - the conjugate “momentum” of the imaginary heat reservoir

In addition, the following parameter is introduced:

- Q - an effective “mass” associated with s , such that $p_s = Q\dot{s}$

The physical significance of these variables is not really evident at this point. The choice of their functional form is motivated by the way by which they transform the microcanonical partition function.

The system Hamiltonian is then constructed as follows:

$$H = \frac{1}{2} \sum \frac{|\mathbf{p}_i|^2}{m_i} + U(\mathbf{r}^N) + \frac{p_s^2}{2Q} + k_B T(3N + 1) \ln s$$

The added last two terms on the RHS give the kinetic and potential energies of the added degrees of freedom for the heat bath. Again, this form is chosen for reasons we will see later.

Here, the particle momenta are scaled and follow the relationship

$$\mathbf{p}_i = m_i \mathbf{v}_i \times s$$

It is through this scaling with the heat bath variable s that the momenta are coupled to constant temperature. The form of this relationship follows from the Lagrangian (not shown here).

The microcanonical partition function then follows:

$$\begin{aligned} \Omega &\sim \frac{1}{N!} \int \delta[H - E] s^{3N} d\mathbf{p}^N d\mathbf{r}^N dp_s ds \\ &= \frac{1}{N!} \int \delta \left[\frac{1}{2} \sum \frac{|\mathbf{p}_i|^2}{m_i} + U(\mathbf{r}^N) + \frac{p_s^2}{2Q} + k_B T(3N + 1) \ln s - E \right] s^{3N} d\mathbf{p}^N d\mathbf{r}^N dp_s ds \end{aligned}$$

The term s^{3N} comes from the scaling of the particle momenta and the fact that the true conjugate momenta of \mathbf{r}^N are \mathbf{p}^N scaled by s (this must be shown through the Lagrangian).

The outer integral in s can be performed analytically using the mathematical result

$$\int x^n \delta(a \ln x - b) dx = a^{-1} e^{b(n+1)/a}$$

The final result is:

$$\begin{aligned} \Omega &= \frac{1}{N!} \frac{e^{E/k_B T}}{3N + 1} \left[\int dp_s e^{\frac{-p_s^2/2Q}{k_B T}} \right] \left[\int e^{\frac{-K(\mathbf{p}^N) + U(\mathbf{r}^N)}{k_B T}} d\mathbf{r}^N d\mathbf{p}^N \right] \\ &= \text{const} \times Q(T, V, N) \end{aligned}$$

where $Q(T, V, N)$ is the canonical partition function for the original N particles.

The implication of this result is that a microcanonical simulation in the **extended system** (including heat bath degrees of freedom) returns a canonical ensemble for the original system. That is,

if we evolve the extended system according to the extended Hamiltonian above (more exactly, the extended underlying Lagrangian), we expect to obtain a canonical distribution of the particle positions and momenta. This evolution will correspond to solving the trajectory for the atomic coordinates and momenta *as well as* the heat bath degrees of freedom s and p_s .

Importantly, this thermostat rigorously generates canonical ensemble thermodynamics and can approximate the true dynamics of the system because the time evolution of the particles is **deterministic**, and doesn't involve stochastic changes.

Equations of motion can be developed for this extended Hamiltonian system. A value must be specified for the constant Q , which is important in determining the rate of exchange of system energy with the imaginary heat bath.

The Nosé-Hoover thermostat

The above formulation as written is not convenient to implement because the scaling of the particle momenta by s (which is variable throughout the simulation) implies that “real” time—the time we should use to evaluate time-averages and kinetic properties like diffusivity—is also variable in the simulation.

Hoover developed an alternate formulation of the Nosé approach that alleviates this problem [Hoover, 1986]. In this method,

$$H = \frac{1}{2} \sum \frac{|\mathbf{p}_i|^2}{m_i} + U(\mathbf{r}^N) + \frac{\xi^2 Q}{2} + 3Nk_B T \ln s$$

Here, ξ is a **friction coefficient** that, broadly speaking, replaces p_s . The time-evolution of the particle positions and momenta is governed by the following equations.

$$\begin{aligned} \frac{d\mathbf{r}_i}{dt} &= \mathbf{v}_i \\ \frac{d\mathbf{v}_i}{dt} &= -\frac{1}{m_i} \frac{\partial U(\mathbf{r}^N)}{\partial \mathbf{r}_i} - \xi \mathbf{v}_i \\ \frac{d\xi}{dt} &= \left(\sum m_i |\mathbf{v}_i|^2 - 3Nk_B T \right) / Q \\ \frac{d \ln s}{dt} &= \xi \end{aligned}$$

Notice that the velocity update of a particle resembles Newton's equations with an additional force that is proportional to the velocity. This is the friction term. Moreover, notice that the variable ξ no longer changes in time when the instantaneous kinetic energy equals $\frac{3}{2} Nk_B T$.

Numerical integration schemes, such as the velocity Verlet method, can be developed for these equations. Again, the heat bath mass Q is an important parameter to be chosen.

The Nosé-Hoover equations are broadly used in the simulation community. However, while they allow one to rigorously generate a canonical ensemble through molecular dynamics, they can at times exhibit pathological behavior for specific systems. Some methods have been developed to increase robustness of the approach; in **Nosé-Hoover chains**, multiple heat baths (multiple degrees of freedom s) are linked to enhance temperature equilibration.

Commentary

An overview of formulations of the equations of motions for various extended ensembles is in the book by Frenkel and Smit. These are all based on the Lagrangian formulation of classical mechanics, which is the most natural starting point for developing algorithms that correctly explore phase-space and have good numerical stability. More specifically, Martyna, Tuckerman, and coworkers have developed a rigorous formulation for the equations of motion for arbitrary ensembles. See for example [Martyna, Tuckerman, Tobias and Klein, Mol. Phys. 87, 1117 (1996)].

In general, if one is interested in microscopic dynamics and transport coefficients, it is recommended to use the **microcanonical ensemble, which is the only ensemble that correctly reproduces the true dynamics** of the simulation. A common approach is to equilibrate a system using a thermostat, but then to allow the production period to correspond to NVE dynamics.

Barostats

Often it is desirable to maintain a simulated system at both constant temperature and pressure. In a thermodynamic sense, systems at constant pressure are ones that can exchange volume with their surroundings (e.g., by way of a piston). Their volume therefore fluctuates. Likewise, simulated systems at constant pressure involve **volume fluctuations**.

A number of different **barostat** techniques exist for maintaining a target pressure by way of adjusting the simulation volume. Many of these are counterparts to thermostats we described earlier. We won't discuss the algorithms in detail, but the following lists a number of common barostat techniques:

- **volume rescaling** – the instantaneous pressure is made to equal the target pressure by rescaling the system volume at periodic intervals
- **Berendsen barostat** – the pressure is weakly coupled to a “pressure bath” and the volume periodically rescaled

- **extended ensemble barostat** – the system is coupled to a fictitious “pressure bath” using an extended Lagrangian and the introduction of new degrees of freedom, similar to the Nose-Hoover thermostat. This approach was pioneered by Andersen and is often termed the **Andersen barostat**.

Langevin dynamics

Many simulated systems employ implicit solvation, in one of two ways:

- the effects of the solvent free energy are incorporated into the intramolecular potential energy function in the form of **effective pairwise interactions**
- the solvent free energy is included using an **implicit solvation method**, like the Poisson-Boltzmann / Surface Area (PBSA) approach or the Generalized Born / Surface Area approach (GBSA).

When explicit solvent atoms are removed, the solute molecules that remain no longer experience a solvent viscosity since this relates, in part, to the time scales of the solvent degrees of freedom. In other words, the implicit solvent free energy assumes that the solvent degrees of freedom equilibrate instantaneously to any change in the solute conformation.

Often the lack of a solvent viscosity is desirable, as it increases the exploration of the solute phase-space and thermodynamic properties can be evaluated more quickly.

On the other hand, if realistic dynamics are desired, one must build viscous effects back into the equations of motion. This can be accomplished using **Langevin dynamics**. Here, Newton’s equations of motion take the form:

$$m_i \frac{d\mathbf{v}_i}{dt} = -\frac{\partial U(\mathbf{r}^N)}{\partial \mathbf{r}_i} + \mathbf{f}_{\text{Langevin}}(\mathbf{v}_i)$$

Here, the force that a particle experiences is due both to a gradient in the potential energy surface and a frictional force due to the solvent:

$$\mathbf{f}_{\text{Langevin}}(\mathbf{v}_i) = -\gamma_i m_i \mathbf{v}_i + \mathbf{R}$$

Here, γ_i is a **friction coefficient**. For spherical particles of radius a in a bulk medium of viscosity η , an approximate relation is

$$\gamma_i = \frac{6\pi a \eta}{m}$$

The component \mathbf{R} is a random force due to stochastic collisions with the solvent, and it must “balance” the viscous force in order to recover the proper canonical ensemble. It is Gaussian with zero mean and a variance that depends on the particle mass, γ_i , and the temperature:

$$R_x = \sqrt{2\gamma k_B T M} R_{Gauss}$$

The above prescription for the equations of motion can be implemented in numerical, finite-difference methods similar to the integrators described previously. The Langevin equations rigorously converge to the canonical ensemble. One important difference associated with the Langevin approach is that the random force \mathbf{R} introduces a stochastic component to the numerical trajectories.

Rigid bond constraints

Motions along bonds in a simulated system are typically the fastest degrees of freedom and thus set an upper limit on the time step that can be used for integrating the equations of motion. One way to move to larger time steps—which ultimately enables longer simulations—is to rigidly constrain bonds to fixed lengths. This is termed **constraint dynamics**.

Here we discuss the proper approach. Rigid bonds and constraint dynamics are common in particular in many models of liquid water, and in simulating organic macromolecules where bonds with hydrogen atoms are often constrained.

Statistical mechanics

It may seem surprising, but a system of rigid bonds is *not* equivalent to the same system with harmonic bonds in the limit that the force constants approach infinity. One clue to this phenomenon is that a harmonic degree of freedom will always contribute to the potential energy of the system, regardless of the force constant, while a rigid bond will not. In the limit that the harmonic degree of freedom is decoupled from other degrees of freedom (typically the case of very large force constants), one expects a potential energy contribution of $\frac{1}{2}k_B T$.

Problem formulation

The correct way to treat rigid bonds is to add additional Lagrange multipliers to the Lagrangian. We will not cover this method in detail, but the following sketches some of the basic physical ideas. A detailed discussion is in Frenkel and Smit.

Consider that we wish to constrain a bond between atoms 1 and 2 in our simulation to a length of d . Physically what would hold two atoms at a fixed bond length would be a force along the

bond vector that opposed any interatomic forces acting on the atoms. In Cartesian coordinates, we add this yet unknown bond force to the potential energy function. We use the following form

$$U'(\mathbf{r}^N) = U(\mathbf{r}^N) + \lambda\sigma(\mathbf{r}_1, \mathbf{r}_2)$$

Here, $\sigma(\mathbf{r}_1, \mathbf{r}_2)$ is a constraint equation that we desire to be equal to zero in order to satisfy the bond length between atoms 1 and 2:

$$\sigma(\mathbf{r}_1, \mathbf{r}_2) = r_{12}^2 - d^2$$

The variable λ is a Lagrange multiplier that will change in time so as to ensure $\sigma(\mathbf{r}_1, \mathbf{r}_2) = 0$. Notice that as long as the constraint is zero, we maintain the same potential energy function as the original.

The force acting on atom 1 is

$$\begin{aligned} \mathbf{f}'_1 &= -\frac{\partial U'(\mathbf{r}^N)}{d\mathbf{r}_1} \\ &= -\frac{\partial U(\mathbf{r}^N)}{d\mathbf{r}_1} - \lambda \frac{d}{d\mathbf{r}_1} (r_{12}^2 - d^2) \\ &= \mathbf{f}_1 + 2\lambda(\mathbf{r}_2 - \mathbf{r}_1) \end{aligned}$$

For atom 2,

$$\mathbf{f}'_2 = \mathbf{f}_2 + 2\lambda(\mathbf{r}_1 - \mathbf{r}_2)$$

Newton's equations of motions for particle 1 gives

$$\frac{d^2\mathbf{r}_1}{dt^2} = \frac{\mathbf{f}'_1}{m_1}$$

If we integrate the position of particle 1 a short step in time δt using the Verlet algorithm,

$$\begin{aligned} \mathbf{r}_1(\delta t) &= 2\mathbf{r}_1(0) - \mathbf{r}_1(-\delta t) + \frac{\mathbf{f}'_1}{m_1} \delta t^2 \\ &= 2\mathbf{r}_1(0) - \mathbf{r}_1(-\delta t) + \frac{\mathbf{f}_1}{m_1} \delta t^2 + \frac{2\lambda}{m_1} [\mathbf{r}_1(0) - \mathbf{r}_2(0)] \delta t^2 \\ &= \hat{\mathbf{r}}_1(\delta t) + \frac{2\lambda}{m_1} [\mathbf{r}_1(0) - \mathbf{r}_2(0)] \delta t^2 \end{aligned}$$

Here, $\hat{\mathbf{r}}_1(\delta t)$ gives the position of the particle if we had not accounted for the constraint forces. In other words, $\hat{\mathbf{r}}_1(\delta t)$ would be the new position if we had simply integrated using the original potential energy function without rigid bonds. This new position will not satisfy the bond length

constraint and thus a correction is needed. Instead, we need to find the value of λ that will give the correct bond. This is achieved by solving:

$$|\mathbf{r}_1(\delta t) - \mathbf{r}_2(\delta t)|^2 = d^2$$

Plugging in the above formula and a similar one for $\mathbf{r}_2(\delta t)$,

$$\left| \hat{\mathbf{r}}_1(\delta t) + \frac{2\lambda}{m_1} [\mathbf{r}_1(0) - \mathbf{r}_2(0)] \delta t^2 - \hat{\mathbf{r}}_2(\delta t) - \frac{2\lambda}{m_2} [\mathbf{r}_2(0) - \mathbf{r}_1(0)] \delta t^2 \right|^2 = d^2$$

$$\left| \hat{\mathbf{r}}_1(\delta t) - \hat{\mathbf{r}}_2(\delta t) + \frac{4\lambda}{m_1} [\mathbf{r}_1(0) - \mathbf{r}_2(0)] \delta t^2 \right|^2 = d^2$$

This equation is quadratic in λ and can be solved to give a value of the constraint multiplier. This enables one to predict the particle positions after a time step δt using the predicted particle positions prior to the application of the rigid constraints.

For multiple constraints, one generalizes this approach using multiple Lagrange multipliers λ_k for each constrained bond k . For systems involving more than one constraint on each atom (such as a triatomic molecule), the solution to all of the λ_k requires solving a system of k coupled non-linear equations that can quickly become unwieldy.

Bond angles and torsional angles can also be held at fixed values by using multiple distance constraints. For example, a water molecule can be constrained to maintain the same H-O-H bond angle by applying three distance constraints between all pairs of atoms.

Incidentally, the force along each constrained bond due to the Lagrange multipliers makes a contribution to the virial of the system. In other words, the constrained bonds contribute to the total pressure of the system.

SHAKE and RATTLE

The **SHAKE algorithm** [Ryckaert, Ciccotti, & Berendsen, 1977] is a numerical procedure designed to solve for the bond Lagrange multipliers at each time step in a molecular dynamics simulation. It uses an iterative approach:

1. Consider all rigid bonds in succession.
2. For each bond k in turn, compute the value of λ_k for fixed constant values of all other λ .
3. Go back to step 1 until all of the λ no longer change by some tolerance. This procedure must be iterated because an atom may be involved in one or more constraints and therefore the value of λ for one constraint is coupled to that of another.

The SHAKE algorithm is designed for use with the Verlet integrators (as presented above). Alternatively, the **RATTLE algorithm** [Andersen, 1983] is formulated for use with the velocity Verlet integrator. Other algorithms are also available for other integrators.

Efficient Fortran implementations of these algorithms, as well as many others, can be found at the online Computational Chemistry List at <http://www.ccl.net/chemistry/index.shtml>.

Multiple time steps

As we mentioned in the previous section on rigid bonds, bond vibrations typically set a maximum time step and hence maximum length of our simulation run. A second and perhaps more general approach to reaching longer time scales, beyond constraint dynamics, is to integrate separately the slow and fast degrees of freedom in our system. The **r-RESPA (reversible reference system propagation algorithm)** method was pioneered by [Tuckerman et al, 1992] and is a **multiple time step** approach:

- One identifies terms in the force field that give rise to fast (e.g., bond vibrations) and slow (e.g., van der Waals interactions) time scales.
- Separate molecular dynamics time steps are used for the slow (δt_{slow}) and fast (δt_{fast}) degrees of freedom, such that $\delta t_{\text{slow}} > \delta t_{\text{fast}}$.
- For every slow time step taken, multiple intermediate fast time steps ($n = \delta t_{\text{slow}}/\delta t_{\text{fast}}$) are performed. A time savings is achieved because only the fast force field components need to be evaluated for the intermediate steps.

In rRESPA, any number of different time steps can be used. One can have separate Δt values for bond vibrations, angle bending, torsional interactions, and so on and so forth. Moreover, the rRESPA algorithm was formulated to be time-reversible and area-preserving (via the Liouville formalism). The resulting equations closely resemble the velocity Verlet-type integrator.