Topics in Computational Chemistry, CHEM∗7500/CHEM 750 Fall 2021

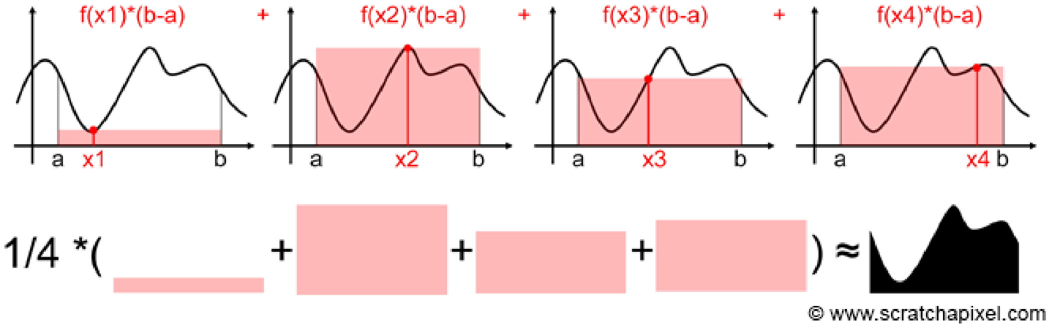
Week 5: Molecular Dynamics

Prof. Leanne D. Chen

October 13, 2021

Department of Chemistry

Week 3 Recap: Brute Force Monte Carlo



Source: Scratchapixel 2.0

Week 3 Recap: Importance Sampling

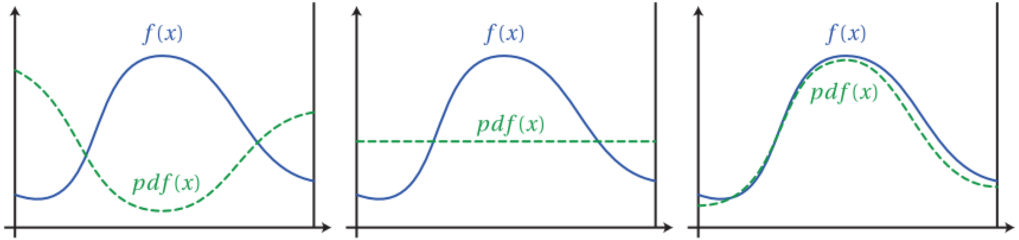


Figure A.2: Comparison of three probability density functions. The PDF on the right provides variance reduction over the uniform PDF in the center. However, using the PDF on the left would significantly increase variance over simple uniform sampling.

Source: Wojciech Jarosz Dissertation

Week 3 Recap: The Lennard-Jones Potential

- A commonly used intermolecular pair potentials is the **Lennard-Jones potential**, which is typically used to describe non-bonded interactions in a simulation.

$$U^{\mathrm{lj}}(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \tag{1}$$

- The simplest method to truncate potentials is to ignore all interactions beyond r_{c}:

$$U^{\mathrm{trunc}}(r) = \begin{cases} U^{\mathrm{lj}}(r), & r \leq r_{\mathrm{c}} \\ 0, & r > r_{\mathrm{c}} \end{cases} \tag{2}$$

UNIVERSITY of GUELPH

Week 3 Recap: Truncated and Shifted

- In molecular dynamics simulations, it is common to use another procedure.
- The potential is truncated and shifted, such that the potential vanishes at the cutoff radius:

$$U^{\text{tr-sh}}(r) = \begin{cases} U^{\text{lj}}(r) - U^{\text{lj}}(r_{\text{c}}), & r \leq r_{\text{c}} \\ 0, & r > r_{\text{c}} \end{cases} \tag{3}$$

- In this case, there are no discontinuities in the potential.

Molecular Dynamics

Overview

- Molecular dynamics (MD) is a technique for computing the equilibrium and transport properties of a classical many-body system.
- In this context, the word *classical* refers to non-quantum mechanical nuclear motion.
- In some ways, MD is similar to carrying out an experiment in real life.
- In an experiment, the chemist would need to prepare a sample of material, then connect it to some measuring instrument, and measure the property of interest during a certain time interval.

Overview

- If the measurements are subject to statistical noise, then the measurement becomes more accurate with more trials, or more time elapsed during the experiment.
- In an MD simulation, we take a similar approach.
- Analogous to sample preparation, a model system that contains N particles is selected, then Newton's equation of motion are solved for the system until its properties stabilize after a certain period of time, this is *equilibration*.
- After equilibration, then the "production runs" are conducted, and measurements are taken for some property of the system.

Overview

- To measure an observable quantity in an MD simulation, we must first of all be able to express this observable as a function of the positions and momenta of the particles in the system.
- For example, a convenient definition of the temperature in a classical, many-body system makes use of the equipartition of energy over all degrees of freedom that enter quadratically in the Hamiltonian of the system.
- In particular, for the average kinetic energy per degree of freedom, we have

$$\left\langle \frac{1}{2} m v_\alpha^2 \right\rangle = \frac{1}{2} k_\mathrm{B} T \tag{4}$$

Overview

- In a simulation, we use the above equation as an operational definition of the temperature.
- In practice, we would measure the total kinetic energy of the system and divide this by the number of degrees of freedom N_f.
- Here, $N_\mathrm{f} = 3N - 6$ for a system of N total particles with total fixed momentum.
- As the total kinetic energy of a system fluctuates, so does the instantaneous temperature:

$$T(t) = \sum_{i=1}^{N} \frac{m_i v_i^2(t)}{k_\mathrm{B} N_\mathrm{f}} \tag{5}$$

The Maxwell-Boltzmann Distribution

Now, we must determine the constants of proportionality K_x, K_y, and K_z. Similar to our previous derivation for the pressure, we begin with the x direction only. We note that a particle must have a velocity in the x direction, between the range $-\infty < v_x < \infty$, which means that integration over the full range of possible values of v_x must give a total probability of 1:

$$\int_{-\infty}^{\infty} f(v_x) \mathrm{d}v_x = 1 \tag{6}$$

Substituting in the expression for $f(v_x)$,

$$1 = K_x \int_{-\infty}^{\infty} e^{-mv_x^2/2k_\mathrm{B}T} \mathrm{d}v_x \tag{7}$$

So we need to evaluate the above integral.

The Maxwell-Boltzmann Distribution

The definite integral of an arbitrary Gaussian function is

$$\int_{-\infty}^{\infty} e^{-a(x+b)^2} \mathrm{d}x = \left(\frac{\pi}{a}\right)^{1/2} \tag{8}$$

So

$$K_x \int_{-\infty}^{\infty} e^{-mv_x^2/2k_\mathrm{B}T} \mathrm{d}v_x = K_x \left(\frac{2\pi k_\mathrm{B}T}{m}\right)^{1/2} = 1 \tag{9}$$

And therefore K_x is equal to the inverse of the expression in parentheses:

$$f(v_x) = \left(\frac{m}{2\pi k_\mathrm{B}T}\right)^{1/2} e^{-mv_x^2/2k_\mathrm{B}T} \tag{10}$$

UNIVERSITY of GUELPH

The Maxwell-Boltzmann Distribution

Now we go back to finding an expression for $f(v)\mathrm{d}v$, which is the product of $f(v_x)f(v_y)f(v_z)\mathrm{d}v_x\mathrm{d}v_y\mathrm{d}v_z$. The probability that a molecule has a velocity in the range v_x to $v_x + \mathrm{d}v_x$, v_y to $v_y + \mathrm{d}v_y$, and v_z to $v_z + \mathrm{d}v_z$ is

$$\begin{aligned} f(v_x)f(v_y)f(v_z)\mathrm{d}v_x\mathrm{d}v_y\mathrm{d}v_z &= \left(\frac{m}{2\pi k_\mathrm{B}T}\right)^{3/2} \overbrace{e^{-mv_x^2/2k_\mathrm{B}T}e^{-mv_y^2/2k_\mathrm{B}T}e^{-mv_z^2/2k_\mathrm{B}T}}^{e^{-m\left(v_x^2+v_y^2+v_z^2\right)/2k_\mathrm{B}T}}\mathrm{d}v_x\mathrm{d}v_y\mathrm{d}v_z \\ &= \left(\frac{m}{2\pi k_\mathrm{B}T}\right)^{3/2} e^{-mv^2/2k_\mathrm{B}T}\mathrm{d}v_x\mathrm{d}v_y\mathrm{d}v_z \qquad (11) \end{aligned}$$

where

$$v^2 = v_x^2 + v_y^2 + v_z^2 \qquad (??)$$

from before.

The Maxwell-Boltzmann Distribution

Finally, we substitute all molecular constants (k, m) with molar constants (R, M), where $R = N_\mathrm{A}k_\mathrm{B}$ and $m/k_\mathrm{B} = mN_\mathrm{A}/R = M/R$, to obatin

$$f(v) = 4\pi\left(\frac{M}{2\pi RT}\right)^{3/2} v^2 e^{-Mv^2/2RT} \tag{12}$$

Equation 12 is called the **Maxwell-Boltzmann distribution of speeds**. Note that, in common with other distribution functions (such as the normal distribution), $f(v)$ only has physical meaning after it is multiplied by the range of speeds of interest.

The Maxwell-Boltzmann Distribution

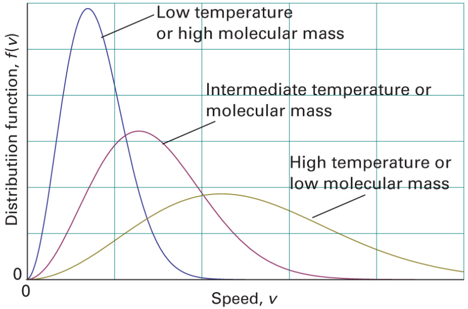


Figure 1B.4 The distribution of molecular speeds with temperature and molar mass. Note that the most probable speed (corresponding to the peak of the distribution) increases with temperature and with decreasing molar mass, and simultaneously the distribution becomes broader.

UNIVERSITY of GUELPH

Mean Values of the Maxwell-Boltzmann Distribution

The average value of v^n is calculated as

$$\langle v^n \rangle = \int_0^\infty v^n f(v) \mathrm{d}v \tag{13}$$

In particular, integration with $n = 2$ results in the mean square speed, $\langle v^2 \rangle$, of the molecules at a temperature T:

$$\langle v^2 \rangle = \frac{3RT}{M} \tag{14}$$

Then, the root-mean-square speed of the molecules of a gas is

$$v_{\mathrm{rms}} = \langle v^2 \rangle^{1/2} = \left(\frac{3RT}{M} \right)^{1/2} \tag{15}$$

UNIVERSITY of GUELPH

Back to MD: the Algorithm

- Let us consider a simple program that is designed to carry out MD simulations; in terms of the algorithm, the steps are as follows:
  1. First, we read in the parameters that specify the conditions of the run; e.g., initial temperature, number of particles, density, and time step.
  2. Then, we initialize the system by selecting initial positions and velocities.
  3. Next, we compute the forces on all particles.
  4. After we have obtained the forces, we are then able to integrate Newton's equations of motion; the force computations and integrations are repeated until the system has evolved for a sufficiently long time.
  5. Finally, the average values of quantities we wish to measure are computed, and the program is completed.

UNIVERSITY of GUELPH

The Algorithm

- To begin the simulation, we first place the particles on a square or cubic lattice (depending on the total number of dimensions in the system).
- Then, an initial velocity in each dimension is given to each particle, which is taken from a Maxwell-Boltzmann distribution.
- The force calculation is typically the most time-consuming step for an MD simulation.

The Force Calculation

- We need to calculate the force acting on *every particle*; for a system that only has pairwise additive interactions, we would need to consider the contribution to the force on particle i due to all its neighbours.
- If we consider only the interaction between a particle and the nearest image of another particle, this implies that, for a system of N particles, we must evaluate $N \times (N-1)/2$ pair distances.
- Without using efficient algorithms, then, the evaluation of forces in the most naive way scales as N^2.

The Force Calculation

- We first compute the distance in the *x*, *y* (and *z* if working in three dimensions) directions between each pair of particles *i* and *j*.
- We enforce periodic boundary conditions and ensure that we only count interactions that are within one half of the box length.
- This means that we limit the evaluation of intermolecular interactions between *i* and *j* only to the interaction between *i* and the nearest periodic image of *j*.
- If a given pair of particles is close enough to interact, we must compute the force between these particles, and the contribution to the potential energy.

The Force Calculation

- Suppose that we would like to compute the x-component of the force:

$$\begin{aligned} f_x(r) &= -\frac{\partial u(r)}{\partial x} \\ &= -\left(\frac{x}{r}\right)\left(\frac{\partial u(r)}{\partial r}\right) \end{aligned} \tag{16}$$

For a Lennard-Jones system (in reduced units), the force is

$$f_x(r) = \frac{48x}{r^2}\left(\frac{1}{r^{12}} - 0.5\frac{1}{r^6}\right) \tag{17}$$

UNIVERSITY of GUELPH

Integrating the Equations of Motion

- After we have computed the forces between all pairs of particles, we can then integrate Newton's equations of motion.
- There are many different algorithms that accomplish this goal.
- The one we will focus on is the Verlet algorithm, which is one of the simplest and best-performing algorithms.
- To derive the Verlet algorithm, we start with a Taylor expansion of the coordinate of a particle around time t

$$r\left(t+\Delta t\right) = r\left(t\right) + v\left(t\right)\Delta t + \frac{f(t)}{2m}\Delta t^2 + \frac{\Delta t^3}{3!}\dddot{r} + \mathcal{O}\left(\Delta t^4\right) + \ldots \qquad (18)$$

UNIVERSITY of GUELPH

Integrating the Equations of Motion

- Similarly,

$$r\left(t-\Delta t\right)=r\left(t\right)-v\left(t\right)\Delta t+\frac{f(t)}{2m}\Delta t^{2}-\frac{\Delta t^{3}}{3!}\dddot{r}+\mathcal{O}\left(\Delta t^{4}\right)+\ldots \qquad (19)$$

If we sum Equations 18 and 19, we obtain

$$r\left(t+\Delta t\right)+r\left(t-\Delta t\right)=2r\left(t\right)+\frac{f(t)}{m}\Delta t^{2}+\mathcal{O}\left(\Delta t^{4}\right)+\ldots \qquad (20)$$

Which we can approximate with

$$r\left(t+\Delta t\right)\approx 2r\left(t\right)-r\left(t-\Delta t\right)+\frac{f(t)}{m}\Delta t^{2} \qquad (21)$$

Integrators

- The estimate of the new position contains an error that is of order Δt^4, where Δt is the time step in the molecular dynamics scheme.
- Once the new positions are computed, the positions at time $t - \Delta t$ are discarded.
- The current positions become the old positions, and the new positions become the current positions.
- For the velocity, we have

$$v(t) = \frac{r(t + \Delta t) - r(t - \Delta t)}{2\Delta t} + \mathcal{O}\left(\Delta t^2\right) \tag{22}$$

which is based on the mean-value theorem

$$f'(c) = \frac{f(b) - f(a)}{b - a}. \tag{23}$$

Integrators

- We can also look at some alternatives to the Verlet algorithm.
- The most naive algorithm is based simply on a truncated Taylor expansion of the particle coordinates:

$$r\,(t + \Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 + \dots \tag{24}$$

- If we truncate this expansion beyond the term in Δt^2, we obtain the *Euler algorithm*.
- Although it looks similar to the Verlet algorithm, it is not time reversible, and may suffer from an energy drift.

Integrators

- Another algorithm, which is equivalent to the Verlet scheme, is the Leap Frog algorithm.
- This algorithm evaluates the velocities at half-integer steps, and uses these velocities to compute the new positions.
- To derive the Leap Frog algorithm from the Verlet algorithm, we start by defining the velocities at half-integer time steps as follows:

$$v\left(t - \frac{\Delta t}{2}\right) \equiv \frac{r(t) - r(t - \Delta t)}{\Delta t} \tag{25}$$

and

$$v\left(t + \frac{\Delta t}{2}\right) \equiv \frac{r(t + \Delta t) - r(t)}{\Delta t} \tag{26}$$

Integrators

- From Equation 26 we obtain an expression for the new positions, based on the old positions and velocities:

$$r\left(t+\Delta t\right) = r(t) + \Delta t v\left(t + \frac{\Delta t}{2}\right) \tag{27}$$

- From the Verlet algorithm, we get the following expression for the update of the velocities

$$v\left(t+\frac{\Delta t}{2}\right) = v\left(t - \frac{\Delta t}{2}\right) + \frac{f(t)}{m}\Delta t \tag{28}$$

Time-Reversal Symmetry

- Time-reversal symmetry means that: at a given time, if we reverse all the momenta (without changing the positions at that time), then the system will trace its trajectory exactly back into the past.
- E.g. we have a system of particles, with $\mathbf{r}_i(0)$, $\mathbf{p}_i(0)$ as the initial conditions, that then is evolved for time t, so that the conditions at time t are $\mathbf{r}_i(t)$, $\mathbf{p}_i(t)$.
- Then, if the momenta are reversed at time t, i.e., $\mathbf{p}_i(t) \to -\mathbf{p}_i(t)$ (the positions are kept the same), the system will return to its initial conditions after evolving for another time period t.

Python

Bubble Sort

1. Loop through the list up to the second last element.
2. Compare each pair of values, the current value and the i+1th value, and swap them if they are out of order.
3. After you get to the end of the list, reduce the number of elements by 1 in the next loop (since the largest value is now at the end of the list and thus sorted).
4. Run through steps 1 and 2 until the whole list is sorted (when your starting point converges with your end point).

Insertion Sort

1. Create a new list, starting with the first element in the old or unsorted list.
2. Take the next element, and place it in the correct position in the sorted list (which currently only contains one element).
3. Continue iterating through the old list, and insert each element in the correct position in the new, sorted list.

Thank you!

Please feel free to unmute yourself or raise your hand to ask a question.