Monte Carlo Simulation of Liquids

E. J. Maginn, J. K. Shah

Department of Chemical and Biomolecular Engineering
University of Notre Dame
Notre Dame, IN 46556 USA

Monte Carlo Workshop, Brazil

UNIVERSITY OF NOTRE DAME

How to Perform MC Simulation of Liquids

Goal: Perform Metropolis MC on a simple liquid (i.e. Ar)

- Assume pairwise LJ interactions

$$\mathcal{V}_{ij}^{LJ} = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^{6} \right] \tag{1}$$

Task: Generate a sequence of configurations for N molecules in volume V that asymptotically samples the probability density of the canonical (NVT) ensemble.

How to Perform MC Simulation of Liquids

Nomenclature:

- Multi-dimensional space $\rightarrow$ configuration space, $(\mathbf{r}_1, \cdots, \mathbf{r}_M)$
- States $\rightarrow$ configurations (atom arrangements)
- $\rho_m \rightarrow \rho^{NVT}(\mathbf{r}_1^{(m)}, \cdots, \mathbf{r}_N^{(m)}) d\mathbf{r}^N$
  ($d\mathbf{r}^N$ is an elementary volume element in configuration space)
- $\rho_n / \rho_m \rightarrow \frac{\rho^{NVT}(\mathbf{r}_1^{(m)}, \cdots, \mathbf{r}_N^{(m)})}{\rho^{NVT}(\mathbf{r}_1^{(n)}, \cdots, \mathbf{r}_N^{(n)})} = \frac{\rho_n^{NVT}}{\rho_m^{NVT}}$

How to Perform MC Simulation of Liquids

Note that

$$\rho^{NVT}(\mathbf{r}_1^{(m)}, \cdots, \mathbf{r}_N^{(m)}) \equiv \rho_m^{NVT} = \frac{\exp[-\beta \mathcal{V}(\mathbf{r}_1, \cdots, \mathbf{r}_N)]}{Z(NVT)} \equiv \frac{\exp[-\beta \mathcal{V}_m]}{Z} \quad (2)$$

Thus,

$$\frac{\rho_n^{NVT}}{\rho_m^{NVT}} = \frac{\exp[-\beta \mathcal{V}_n]}{\exp[-\beta \mathcal{V}_m]} = \exp[-\beta(\mathcal{V}_n - \mathcal{V}_m)] = \exp[-\beta \Delta \mathcal{V}_{m \to n}] \quad (3)$$

UNIVERSITY OF NOTRE DAME

Metropolis Algorithm for Liquids

- Assume equal attempt probabilities $\alpha_{mn} = \alpha_{nm}$
- Start with configuration *m*, generate a new configuration *n*
- Compute the energy difference
- if $\mathcal{V}_n \leq \mathcal{V}_m$

$$min\left(1, \frac{\rho_n^{NVT}}{\rho_m^{NVT}}\right) = 1 \tag{4}$$

Accept the new configuration

- if $\mathcal{V}_n > \mathcal{V}_m$

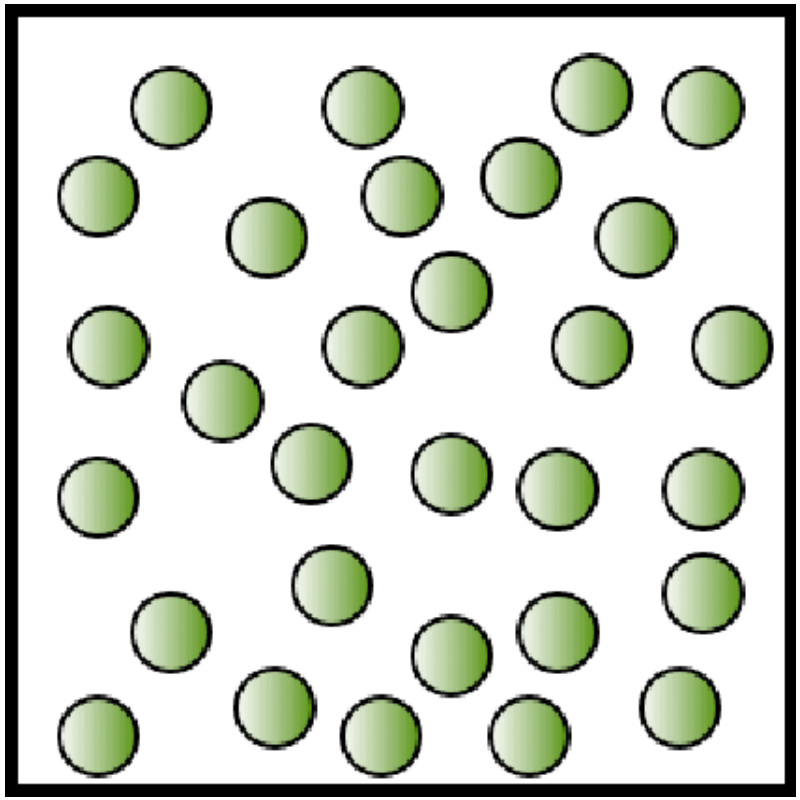
$$min\left(1, \frac{\rho_n^{NVT}}{\rho_m^{NVT}}\right) = \exp[-\beta\Delta\mathcal{V}_{m\rightarrow n}] \tag{5}$$

Accept if $\zeta < \exp[-\beta(\mathcal{V}_n - \mathcal{V}_m)]$

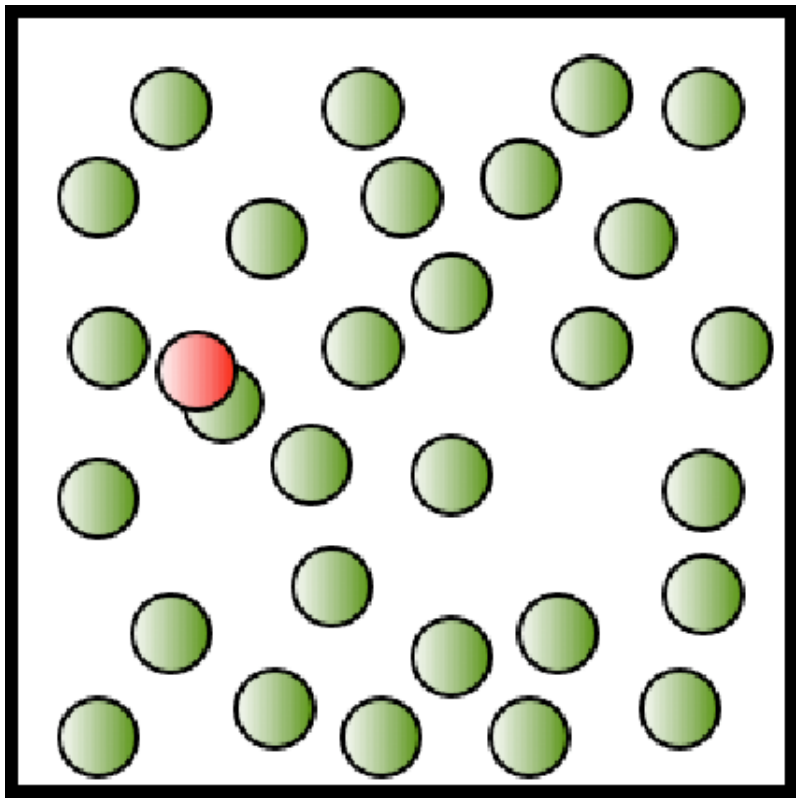
- Notice the Metropolis selection criterion *only involves potential energy*

Details, details...

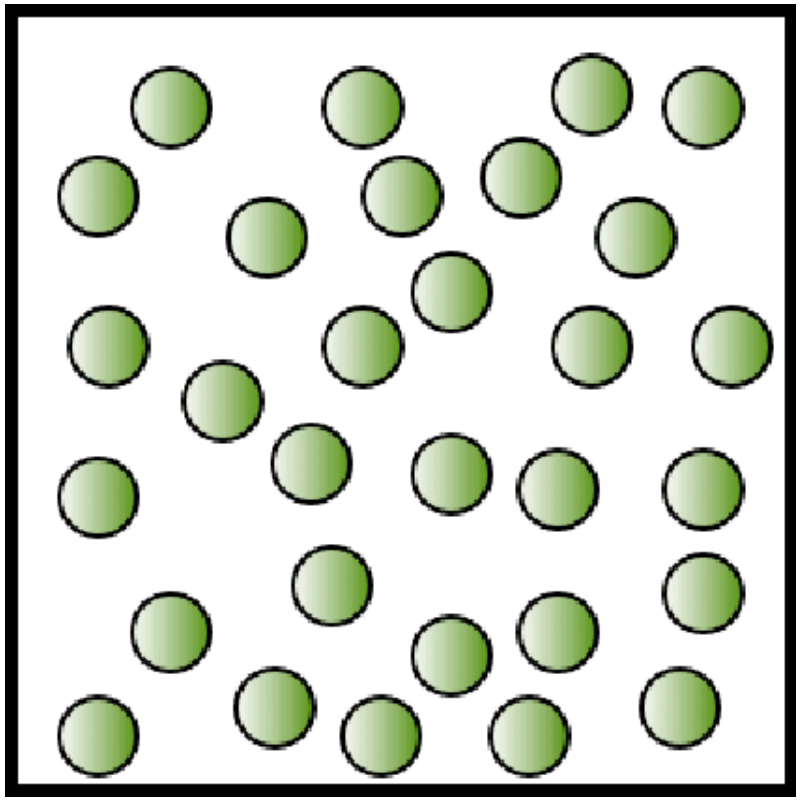
- Decide on form of α, the underlying stochastic matrix
- If moving between *m* and *n* is equally probable, $\alpha_{mn} = \alpha_{nm}$ and previous acceptance rules are OK
- Turns out small “local” moves are typically preferred over large “bold” moves. Why?



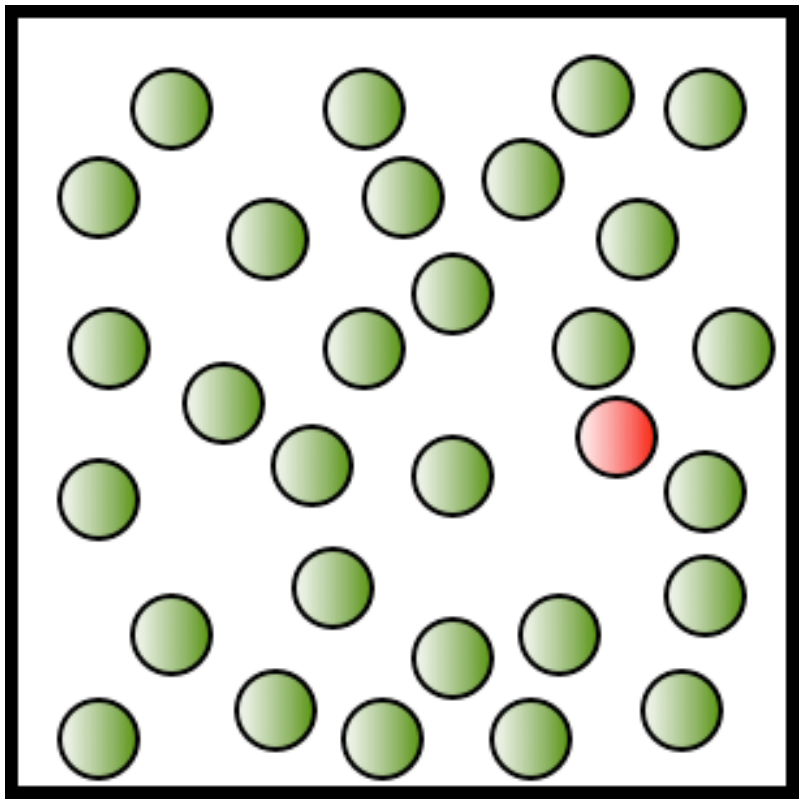
Initial configuration
Dense fluid



Move a molecule at random. High probability of overlap. *Move rejected*

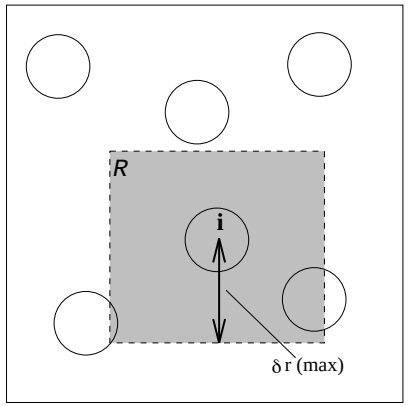


Initial configuration
Dense fluid



Local move reduces overlap probability *Adjust* α *to achieve desired acceptance rates*

Move Algorithm



To move from state *m* to state *n*

1. Choose atom at random; assume atom *i* at position $\mathbf{r}_i^{(m)}$
2. Define a "local" or "neighboring" environment by a square (cube or sphere in three dimensions) centered on *i*. Edge length (or radius) of the local region is δr_{max}
3. Denote local region by $\mathcal{R}$. Note that if we use a cube (as done below), the sides are $2\delta r_{max}$ long.
4. There is a large but finite set of new configurations, $N_{\mathcal{R}}$ within the cube $\mathcal{R}$. If each is of equal probability

$$\alpha_{mn} = 1/N_{\mathcal{R}};\ \mathbf{r}_i^{(n)} \in \mathcal{R} \tag{6}$$

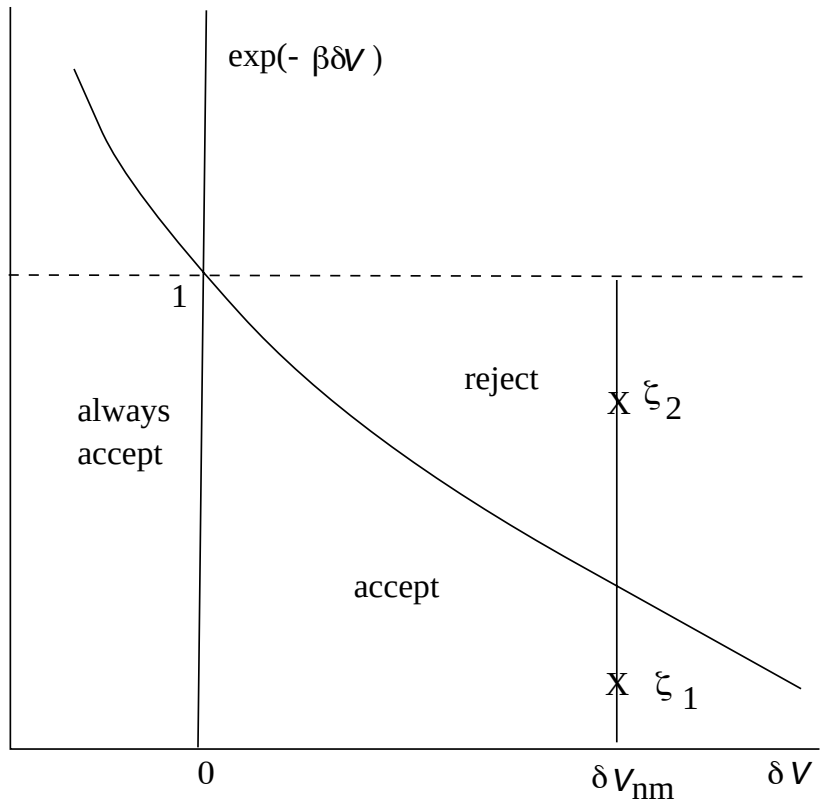
$$\alpha_{mn} = 0;\ \mathbf{r}_i^{(n)} \notin \mathcal{R} \tag{7}$$

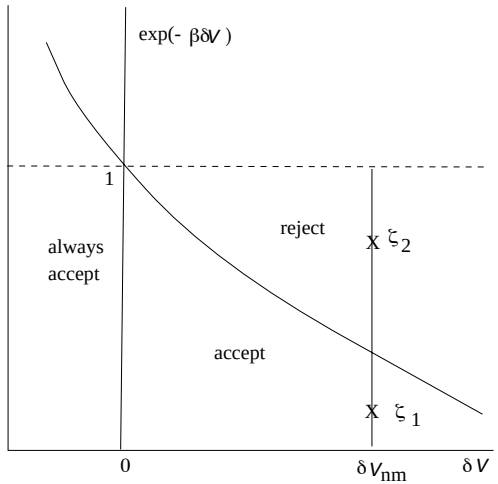
- To implement: an atom is chosen at random and given a uniform, random displacement along each Cartesian axes.
- An adjustable parameter, δr_{max}, controls the "boldness" of the attempted move: small displacements give high acceptance rates but slow evolution; large displacements yield large configurational changes, but get rejected more often.
- δr_{max} is typically adjusted during equilibration so that about 50% of the attempted moves are successful

```
rxnew = rx(i) + (2.0 * rranf(iseed) - 1.0) * drmax
rynew = ry(i) + (2.0 * rranf(iseed) - 1.0) * drmax
rznew = rz(i) + (2.0 * rranf(iseed) - 1.0) * drmax
```

UNIVERSITY OF NOTRE DAME

- After displacement is made, energy of the new state is compared to the energy of the old state.
- The Metropolis selection rule is used to decide whether or not this new state is accepted





- If move from *m* to *n* is downhill, $\delta\mathcal{V}_{nm} \leq 0$ and the move is always accepted. $\pi_{mn} = \alpha_{nm}/\alpha_{mn} = 1$
- For "uphill" moves, a random number ζ is generated uniformly on (0,1)
- If $\zeta < \exp[-\beta\mathcal{V}_{nm}]$, ($\zeta_1$ in the figure), the move is accepted
- Otherwise, (ζ_2), the move is rejected
- Over the course of the simulation, the net result is that energy changes such as $\delta\mathcal{V}_{nm}$ are accepted with probability $\exp[-\beta\delta\mathcal{V}_{nm}]$

Using Cassandra

Use Cassandra to reproduce the properties of the LJ equation of state

- Study a series of cases to explore the effect of various simulation parameters
- Potential cutoff
- System size
- Uniqueness of trajectories
- Computing averages

Stochastic Matrix

- What if $\alpha_{mn} \neq \alpha_{nm}$?
- Recall acceptance rule that satisfies detailed balance

$$\pi_{mn} = min(1, \frac{\alpha_{nm}\rho_n}{\alpha_{mn}\rho_m}) \tag{8}$$

- Simply retain α in the acceptance rule
- Key: In this case *you must know what* α_{mn} *and* α_{nm} *are!*
- If $\alpha_{mn} \neq \alpha_{nm}$ and you think they are equal, standard Metropolis algorithm will give you the wrong answer *and you will not know it!*

UNIVERSITY OF NOTRE DAME

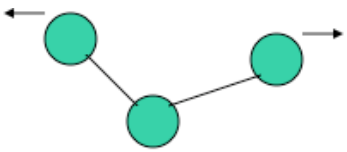
Demonstration of Violation of Detailed Balance

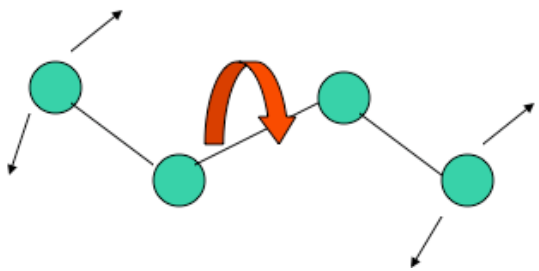
Run demo to see what happens when $\alpha_{mn} \neq \alpha_{nm}$ but you assume it is.

Molecular Liquids

- For molecular systems, the elementary moves must change all the configurational degrees of freedom
  - rigid translation
  - rigid rotation
  - rotation about bonds
  - bond distortion
- More on this later

bond distortions





bond rotations

Initial Configuration

- Final result should be independent of initial configuration
- But how do you start a simulation?
  - Want to start in high probability (low energy) state, to minimize time spent in equilibration
  - Traditional approach: start from an fcc lattice and “melt” to obtain a liquid. This ensures none of the molecules are initially overlapping.
  - Alternative: Randomly shoot molecules into a box and the do energy minimization to relax overlaps.
  - Last configuration of a MC run can be used as starting point if conditions are similar.

Equilibration

- Initial configuration is “far” from equilibrium
- Do not count in averages
- Simulations performed in two stages
  1. “Equilibration” phase: Markov chain asymptotically approaches limiting distribution
  2. “Production” phase: collect averages of “equilibrium” state
- At the end of the equilibration period, *all memory of the starting configuration should be lost*.

Equilibration

- To check whether the system has in fact reached equilibrium:
  - Monitor the potential energy and pressure. Run equilibration until there is no systematic drift in either quantity, only fluctuations about a mean.
  - If you started from a lattice, make sure all indications of initial order have vanished. (Translational and orientational order parameters show no order in fluid).
  - For fluid simulations, the mean–square displacement should grow linearly with time, indicating”diffusive” behavior.
- Rule of thumb is: low–molecular weight systems require 500N – 1000N steps to equilibrate.

Equilibration and the Ising Lattice

Go back and re-run Ising lattice simulation but run an equilibration and then restart the production run from the ending configuration. Do you see much difference? Why or why not?