Monte Carlo Methods - Chs. 3, 5 and 8

COMSEL

E. J. Maginn

1Department of Chemical and Biomolecular Engineering
University of Notre Dame
Notre Dame, IN 46556 USA

Monte Carlo Lecture
University of Notre Dame
Using Probabilistic Methods to Solve Problems

It has been known for a long time that it is possible to use probabilistic methods to estimate deterministic quantities.

- W. S. Gossett ("student") estimated the correlation coefficient in his “t” distribution with the help of sampling methods
- Joel Hildebrand (his graduate students) generated random samplings of spheres in a box to investigate liquid phase properties.
- Famous example: Lazzerini (Italian mathematician, 1901) and computing $\pi$
George Louis LeClerc Comte de Buffon

- French naturalist
- Discovered binomial theorem (age 20)
- Discovered theorem in mathematical probability (1777)

*If a needle of length \( \ell \) is thrown randomly onto a set of equally spaced parallel lines, \( d \) apart (\( d > \ell \)), the probability that the needle will cross a line is \( \frac{2\ell}{\pi d} \).*
Lazzarini’s “Experiment”

- Buffon’s theorem: a probabilistic way of estimating $\pi$
- Lazzerin(1901): Reportedly dropped a 2.5 cm needle 3408 times
- 1,808 crossings on 3 cm grid
- $\pi = (4/3) \times 3408/1808 = 3.14159292.$
- You could imagine how much easier this would be on a computer today...
Did Lazzarini Cheat?

\[ \pi = \left( \frac{4}{3} \right) \times \frac{3408}{1808} = 3.14159292. \]

This is a remarkable accurate result!

Estimate reduces to $\frac{355}{113}$

This fraction was the rational approximation of $\pi$ obtained by Chinese mathematician Tsu Ch’ung-chi around A.D. 500...

See *Mathematics Magazine*, 67, 83 (1994) for interesting article.
How lucky can you get?

- Take care when using stochastic methods
- For $d = 1$, $l = 0.7857$, two needles thrown, 1 crosses a line: $\pi \approx 3.1428$
- Even luckier than Lazzarini
Other ways of estimating $\pi$ stochastically

- Imagine a circular pan inscribed inside a square pan
- Place the pans in the rain
- Number of drops hitting the pans is $N_{\text{trials}}$
- Fraction landing inside the circular pan ($N_{\text{hits}}$)

Thus, $\pi = 4 \frac{N_{\text{hits}}}{N_{\text{trials}}}$
Random Numbers

▷ Key to these and other methods is the generation of “random” numbers (sequence of numbers that appear uncorrelated)
▷ Typically uniform on [0,1] and any range generated by suitable multiplication
▷ Many “random number generators” available
  ▷ intrinsic functions
  ▷ other functions, subroutines (we have given some examples)
▷ Perform statistical tests to ensure quality. Also want speed.
▷ Large literature on this - we will not discuss further. We just assume we can generate good random numbers!
Monte Carlo

- These are all examples of *Monte Carlo simulations*
- Conventional numerical discretization
  - Describe system mathematically with differential equations
  - Discretize and solve equations numerically
- Monte Carlo
  - Often simulate the process directly
  - No need to write down equations
  - Requirement: system under study must be characterized by probability density function (pdf)
  - Given a pdf, randomly sample from it
Easiest to think of MC used for simulating stochastic processes

This is too restrictive!

Can solve “deterministic” problems also with MC

Example: evaluating a definite integral

Key requirement: pose solution in terms of pdfs, then treat system stochastically

Definition: Monte Carlo methods encompass all methods that employ statistical simulation of some underlying system, whether or not the system represents a real physical process.
Evaluating Integrals with Monte Carlo

- So-called “sample mean method” is a way of determining integrals from stochastic “experiments”.
- Basic problem: evaluate \( I = \int_{x_1}^{x_2} f(x) \, dx \)
- \( f(x) \) is some arbitrary function
Chapter 3
Chapter 5
Chapter 8

Evaluating Integrals with Monte Carlo

$\int_{x_1}^{x_2} f(x) \, dx$ is re-written as

$$I = \int_{x_1}^{x_2} \left( \frac{f(x)}{\rho(x)} \right) \rho(x) \, dx \quad (1)$$

- $\rho(x)$ is an arbitrary probability density function.
- Then generate $N_{\text{trials}}$ random numbers $\xi$ from the distribution $\rho(x)$ in the range $(x_1, x_2)$
- Evaluate $f$ at each step.

(c) 2019 University of Notre Dame
Evaluating Integrals with Monte Carlo

We see that

$$I = \left\langle \frac{f(\xi)}{\rho(\xi)} \right\rangle_{\rho(x)}$$

(2)

Brackets indicate an average over the $N_{\text{trials}}$ evaluations, and that $\rho(x)$ used as probability distribution.
What should $\rho(x)$ be?

- Simplest to let $\rho(x)$ be a *uniform distribution*

$$\rho(x) = \frac{1}{(x_2 - x_1)}, \quad x_1 \leq x \leq x_2$$ (3)
Evaluating Integrals with Monte Carlo

With uniform distribution, integral is

\[ I \approx \frac{(x_2 - x_1)}{N_{\text{trials}}} \sum_{i=1}^{N_{\text{trials}}} f(\xi_i) \]  

Justification for eq 4: Mean Value theorem.
Algorithm for Evaluating Integrals with Monte Carlo

1. Generate random number, $\zeta$, uniformly on the range $(0, 1]$

2. Random value on $(x_1, x_2)$ from $\xi = x_1 + \zeta (x_2 - x_1)$.

3. Calculate $f(\xi)$

4. Accumulate the sum $\sum_i f(\xi_i)$ and estimate integral

$$I \approx \frac{(x_2 - x_1)}{N_{\text{trials}}} \sum_{i=1}^{N_{\text{trials}}} f(\xi_i)$$

Estimate gets better as $N_{\text{trials}}$ increases.

(c) 2019 University of Notre Dame
Monte Carlo Integration to Estimate $\pi$

- This approach can be used to estimate $\pi$ by considering the equation for a circle in the first quadrant

$$f(x) = (1 - x^2)^{-1/2}$$  \hspace{1cm} (6)

with $x$ between $x_1 = 0$ and $x_1 = 1$.

- If one uses the procedure outlined above, the estimate of $\pi$ after $10^7$ trials is 3.14169.

(c) 2019 University of Notre Dame
Monte Carlo Integration and Other Techniques

- For simple functions (i.e. one- or two-dimensional) Monte Carlo integration cannot compete with straightforward methods such as Simpson's rule.

- For example, using

\[
    f(x) = (1 - x^2)^{-1/2}
\]

Simpson's rule obtains \( \pi = 3.141593 \) after only \( 10^4 \) function evaluations. (Better with far fewer integrations).

- However, for the *multidimensional integrals* encountered in statistical mechanical applications, the sample mean method (with suitable choice of \( \rho(x) \)) is far superior to standard techniques. Why?
Consider the configurational integral for $N = 10$ atoms

$$Z = \int d\mathbf{r}^N \exp[-\beta \mathcal{V}(\mathbf{r}^N)]$$  (8)

We can think of $f(\mathbf{r}^N) \equiv f(\mathbf{r}_1, \cdots, \mathbf{r}_N) = \exp[-\beta \mathcal{V}(\mathbf{r}^N)]$

$Z$ is a $3N$–dimensional integral at fixed $NVT$
Monte Carlo and Multi-dimensional Integrals

For \( N = \mathcal{O}(10) \), \( Z \) is estimated using MC integration as

\[
Z \approx \frac{V^N}{N_{\text{trials}}} \sum_{i=1}^{N_{\text{trials}}} \exp[-\beta \mathcal{V}(r_1^{(i)}, \cdots, r_N^{(i)})]
\]  \hspace{1cm} (9)

where \((r_1^{(i)}, \cdots, r_N^{(i)})\) is a randomly selected point in the \(3N\)-dimensional configuration space. That is, it is \( N \) triplets of random numbers on \((0, L)\).
Monte Carlo and Multi-dimensional Integrals

- To evaluate $Z$ with a Simpson technique requires evaluating $f(r^N)$ at all nodes of a regular grid throughout the configuration space.
- If 10 points (nodes) per coordinate are used, this would entail $10^{3N}$ function evaluations!
- Clearly an unreasonable number even for this small–dimensional system.
- Monte Carlo integration, on the other hand, one can obtain a reasonable estimate for $N_{\text{trials}}$ much less than this.
- Of course, our estimate gets better as $N_{\text{trials}}$ gets larger, but we will converge fairly rapidly to the actual value.
Monte Carlo and Multi-dimensional Integrals

- For higher-dimensional systems, e.g. $N = O(100)$, even standard Monte Carlo integration becomes infeasible.
- Why?
- Consider MC integration of $Z$ for hard sphere fluid

$$Z \approx \frac{V^N}{N_{\text{trials}}} \sum_{i=1}^{N_{\text{trials}}} \exp[-\beta \mathcal{V}(\mathbf{r}_1^{(i)}, \ldots, \mathbf{r}_N^{(i)})]$$  \hspace{1cm} (10)

1. Pick 300 random numbers on $(0, L)$ to get $\mathbf{r}^N$
2. Calculate Boltzmann factor for this configuration.
3. **hard spheres**: Boltzmann factor will only be 1 if no two hard spheres overlap.
4. If only two atoms overlap, potential energy is infinite; Boltzmann factor for that trial will be zero.
5. At reasonable density, probability of generating a non-zero Boltzmann factor randomly goes to zero.
Monte Carlo and Multi-dimensional Integrals

- Similar problem with computing ensemble averages from MC integration.
- Example: estimate $\mathcal{A}$ from

$$< \mathcal{A} >_{NVT} = \frac{\int d\mathbf{r} \mathcal{A} \exp[-\beta V]}{\int d\mathbf{r} \exp[-\beta V]}$$

(11)

were $\approx \frac{\sum_{i=1}^{N_{\text{trials}}} A_i \exp[-\beta V_i]}{\sum_{i=1}^{N_{\text{trials}}} \exp[-\beta V_i]}

- Estimating the numerator and denominator separately using the uniform sample mean method infeasible at high densities.
- Obvious solution: choose random coordinates from a non–uniform distribution ("importance sampling")
Importance Sampling

- Basic idea: choose random numbers from density distribution $\rho(x)$ which concentrates function evaluations in region that makes biggest contribution to integral

$$< f >_{\text{ens}} = \frac{\int d\mathbf{r}^N f(\mathbf{r}^N) \rho^{\text{ens}}(\mathbf{r}^N)}{\int d\mathbf{r}^N \rho^{\text{ens}}(\mathbf{r}^N)}$$

- Random configurations $\rightarrow$ few configurations contribute to integral

- Idea: Sample configuration space *nonuniformly* $\rightarrow$ more probable (i.e. low energy) states sampled more frequently
Importance Sampling

- What if we could sample states according to $\rho^{\text{ens}}$?
- Then calculating ensemble average $\left< f \right>$ amounts to taking a simple *arithmetic average* over the sampled microstates

$$\left< f \right>_{\text{ens}} = \frac{1}{N_{\text{trials}}} \sum_{i=1}^{N_{\text{trials}}} f(i)$$  \hspace{1cm} (13)

- Same as taking averages during MD!
- Metropolis and co–workers did this in late 1940s\(^1\)

Metropolis Monte Carlo

- Term “Monte Carlo” coined by Metropolis (1947)
- Described class of numerical techniques developed by von Neuman, Ulam, and Metropolis at Los Alamos near the end of World War II.
- Their interest: simulate the diffusion of neutrons in fissionable material
- Methods involved use of random numbers (casinos at Monte Carlo)
- Ideally suited for new MANIAC computer at Los Alamos
To understand the Metropolis methods, we need to understand the concept of a *Markov chain*

**Markov chain:** a sequence of trials (stochastic processes) that satisfies two conditions:

1. **The outcome of each trial belongs to a finite set of outcomes, called the *state space*. We describe the state space by \( \{\gamma_1, \gamma_2, \cdots, \gamma_m, \gamma_n, \cdots\} \).**
2. **The outcome of each trial depends only on the outcome of the *immediately preceding trial*. That is, the memory of the system only goes one step back.**
The transition probability links two states $\gamma_m$ and $\gamma_n$

Define $\pi_{mn}$ as the probability that a trial produces state $n$, given that the previous trial resulted in state $m$

$\pi_{mn}$ depends on values of $m$ and $n$, but is independent of where within the sequence the considered trial lies

Thus $\pi_{mn}$ is independent of “time” or number of trials

Given values of $\pi_{mn}$ for all possible $m$ and $n$, a transition probability matrix can be formed

This matrix contains all information on the “dynamics” governing the evolution of the Markov chain
The reliability of your computer follows a certain pattern.
- If it is up and running one day, there is a 60% chance that it will be up the next day.
- If it is down one day, there is a 70% chance it will be down the next day.

Our state space has only two components, “up” (U) and “down” (D).

The transition probability matrix is \( \pi = \begin{pmatrix} 0.6 & 0.4 \\ 0.3 & 0.7 \end{pmatrix} \)

The individual elements of the above matrix are \( \pi_{uu} = 0.6; \pi_{ud} = 0.4; \pi_{du} = 0.3; \pi_{dd} = 0.7 \)

Note that \( \sum_m \pi_{mn} = 1 \); the rows of the transition probability matrix sum to 1; the system must be in some state at the end of a trial. This makes \( \pi \) a stochastic matrix.
Assume that on day 1, the computer is equally likely to be up and down. That is \( \rho^{(1)} = (\rho_U \rho_D) = (0.5 \ 0.5) \)

What are the probabilities the computer will be up/down the next day?

\[
\rho^{(2)} = \rho^{(1)}\pi = (0.45, \ 0.55)
\]

Up: \( (0.5 \times 0.6) + (0.5 \times 0.3) = 0.45 \)

Down: \( (0.5 \times 0.7) + (0.5 \times 0.4) = 0.55 \)

Thus, there is a 45% chance your computer will be up on day 2, but a 55% chance it will be down
Assume that on day 1, the computer is equally likely to be up and down. That is $\rho^{(1)} = (\rho_U \rho_D) = (0.5 \ 0.5)$

What are the probabilities the computer will be up/down the next day?

$\rho^{(2)} = \rho^{(1)} \pi = (0.45, \ 0.55)$

Up: $(0.5 \times 0.6) + (0.5 \times 0.3) = 0.45$

Down: $(0.5 \times 0.7) + (0.5 \times 0.4) = 0.55$

Thus, there is a 45% chance your computer will be up on day 2, but a 55% chance it will be down.
Repeating the process, probabilities for day 3:
\[ \rho(3) = \rho(2) \pi = \rho(1) \pi \pi = (0.435, 0.565) \]

There is now only a 43.5% chance of it being up.

Keep going...
\[ \rho(5) = (0.4287, 0.5713) \]
\[ \rho(6) = (0.4286, 0.5714) \]
\[ \rho(20) = (0.4286, 0.5714) \]

Clearly, there is a limiting distribution which we reached, given by the formula
\[ \rho = \lim_{\tau \to \infty} \rho(1) \pi \tau \]

(c) 2019 University of Notre Dame
Markov Chain Example (after Allen and Tildesley)

- Repeating the process, probabilities for day 3:
  \[ \rho^{(3)} = \rho^{(2)} \pi = \rho^{(1)} \pi \pi = (0.435, 0.565) \]
- There is now only a 43.5% chance of it being up
- Keep going... \[ \rho^{(5)} = (0.4287, 0.5713) \]
  \[ \rho^{(6)} = (0.4286, 0.5714) \]
  \[ \rho^{(20)} = (0.4286, 0.5714) \]
- Clearly, there is a limiting distribution which we reached, given by the formula

\[
\rho = \lim_{\tau \to \infty} \rho^{(1)} \pi^\tau
\]  

(14)
Limiting Distributions

- We see from eqn 14 that the limiting distribution, $\rho$, must satisfy the eigenvalue equation

$$\rho \pi = \rho \quad \text{(15)}$$

$$\sum_m \rho_m \pi mn = \rho_n \quad \text{(16)}$$

- $\rho$ is an eigenvector of the stochastic matrix $\pi$ corresponding to an eigenvalue of 1.

- $\rho$ is completely determined by $\pi$, and independent of the initial condition ($\rho^{(1)}$). All memory of the initial state has been lost.

- Markov chains where one can go from any state to any other state are called ergodic or irreducible. Such chains always have a limiting probability distribution.
Transition Matrices and Ergodicity

- If the transition probability matrix is full, the chain is ergodic.
- If the transition probability matrix is block diagonal, the chain is non-ergodic.
- For statistical mechanical systems, the transition matrix is enormous.
- We assume it is stochastic and ergodic (though we can’t prove it).
- Interestingly, we do not know the elements of the matrix!
- We do know, however, the limiting distribution...(what is it?)
Metropolis Monte Carlo Algorithm

▶ Problem: the probability distribution of the multi–dimensional space is known, but the elements of the transition matrix are not

▶ Example: in NVT ensemble, the limiting distribution of a Markov chain is the vector with elements $\rho_m = \rho_{NVT}(\gamma_m)$ for each point $\gamma_m$ in phase space.

▶ Note: the multi–dimensional space may be discrete ($\rho_m$ is a probability in this case) or it may be continuous, ($\rho_m = \rho(\gamma_m) d\gamma$, where $\rho(\gamma_m)$ is a probability density and $d\gamma$ is an elementary volume in the multi–dimensional space centered at $\gamma_m$).

▶ Goal: An efficient numerical procedure for sampling the multi-dimensional space, according to the probability distribution $\{\rho_m\}$. 

(c) 2019 University of Notre Dame
Metropolis Monte Carlo Algorithm

What does “sampling” mean?

We wish to pick a finite set of points (states) $m_1, m_2, \ldots, m_t, \ldots, m_{N_{\text{trials}}}$ such that the probability of finding each state $m_t$ in the sequence is practically equal to $\rho m_t$

A given state may appear more than once in the sequence

If we can sample states this way, the average of any function $f$ defined on the state space is

$$< f > \approx \frac{1}{N_{\text{trials}}} \sum_{t=1}^{N_{\text{trials}}} f(m_i) \quad (17)$$

The solution to the problem of how to do this is known as Metropolis Monte Carlo
Metropolis Monte Carlo Algorithm

- What does “sampling” mean?
- We wish to pick a finite set of points (states) $m_1, m_2, \cdots, m_t, \cdots, m_{N_{\text{trials}}}$ such that the probability of finding each state $m_t$ in the sequence is practically equal to $\rho_{m_t}$
- A given state may appear more than once in the sequence
- If we can sample states this way, the average of any function $f$ defined on the state space is

$$< f > \approx \frac{1}{N_{\text{trials}}} \sum_{t=1}^{N_{\text{trials}}} f(m_i) \quad (17)$$

- The solution to the problem of how to do this is known as Metropolis Monte Carlo
Metropolis Monte Carlo Algorithm

- What does “sampling” mean?
- We wish to pick a finite set of points (states) $m_1, m_2, \cdots, m_t, \cdots, m_{N_{\text{trials}}}$ such that the probability of finding each state $m_t$ in the sequence is practically equal to $\rho m_t$
- A given state may appear more than once in the sequence
- If we can sample states this way, the average of any function $f$ defined on the state space is

$$\langle f \rangle \approx \frac{1}{N_{\text{trials}}} \sum_{t=1}^{N_{\text{trials}}} f(m_i)$$  \hspace{1cm} (17)

- The solution to the problem of how to do this is known as Metropolis Monte Carlo
What does “sampling” mean?

We wish to pick a finite set of points (states) $m_1, m_2, \cdots, m_t, \cdots, m_{N_{\text{trials}}}$ such that the probability of finding each state $m_t$ in the sequence is practically equal to $\rho_m_t$

A given state may appear more than once in the sequence

If we can sample states this way, the average of any function $f$ defined on the state space is

$$\langle f \rangle \approx \frac{1}{N_{\text{trials}}} \sum_{t=1}^{N_{\text{trials}}} f(m_i)$$  \hspace{1cm} (17)

The solution to the problem of how to do this is known as Metropolis Monte Carlo
Metropolis Monte Carlo Algorithm

- Idea: Generate sequence of points, each from the previous one, according to certain stochastic rules.
- Create a Markov chain of states
  \[ m_1 \to m_2 \to \cdots \to m_t \to \cdots \to m_{N_{\text{trials}}} \]
- Enforce a condition known as “detailed balance”. Many variations, but for our purpose we define it as

\[ \pi_{mn} \alpha_{mn} \rho_m = \pi_{nm} \alpha_{nm} \rho_n \quad (18) \]

- \( \alpha_{mn} \) is the probability of attempting a move from state \( m \) to state \( n \).
Metropolis Monte Carlo Algorithm

- Satisfy detailed balance by selecting the transition matrix of the Markov chain (for moving from $m$ to $n$) as

\[
\pi_{mn} = \frac{\alpha_{nm}}{\alpha_{mn}}, \quad \text{if } \rho_n \geq \rho_m, \ n \neq m
\] (19)

\[
\pi_{mn} = \frac{\alpha_{nm} \rho_n}{\alpha_{mn} \rho_m}, \quad \text{if } \rho_n < \rho_m, \ n \neq m
\] (20)

- It is also possible to stay in the same state

\[
\pi_{mm} = 1 - \sum_{m \neq n} \pi_{mn}
\] (21)
Let’s apply detailed balance to the uptime example
\[ \rho_U = 0.4287 \]
\[ \rho_d = 0.5713 \]

If attempt probabilities are equal \((\alpha_{mn} = \alpha_{nm})\)

\[
\frac{\pi_{ud}}{\pi_{du}} = \frac{0.5713}{0.4287} = 0.4/0.3 \quad (22)
\]
Attempt Matrix and Metropolis Acceptance Rule

- Often moves are attempted only between close-lying states.
- That is, most of the elements of $\alpha$ are zero.
- Typically, $\alpha_{mn}$ is uniform in a small region of the space around state $m$ – meaning that an attempted move to any of the nearby states $n$ is done with equal probability.
- Also, $\alpha$ is often symmetric (i.e. $\alpha_{mn} = \alpha_{nm}$) and stochastic (i.e. $\sum_n \alpha_{mn} = 1$).
- Under these conditions, the probability of accepting a move from $m$ to $n$, the so-called “Metropolis selection criterion” is

$$\pi_{mn} = \min(1, \frac{\rho_n}{\rho_m})$$

(23)

- “min” is Fortran minimum function; accepts the minimum of two arguments.
Metropolis Acceptance Rule

With this selection criterion

1. If new state, $n$, is more probable that the state $m$ from which we started, then accept the new state $n$ as the next state in the Markov chain.
2. If the new state, $n$, is less probable than the old state $m$, accept the new state $n$ with probability

$$\frac{\rho_n}{\rho_m} < 1$$  \(24\)

3. When a new state is accepted, count it in properties for averaging.
4. If the new state $n$ is not accepted, state $m$ is retained as the next state in the chain and its characteristics are entered again in the averaging procedure.
\[ \pi \rho = \rho \] (25)

- The above property guarantees that, regardless of the starting point of the Markov chain, a chain long enough will asymptotically sample the probability distribution of interest, defined by \( \rho \).
- The acceptance rule means \( \rho \) only needs to be known up to a multiplicative constant
  - Only probability ratios appear in the Metropolis scheme
  - This makes the method very useful in statistical mechanical applications.
Example: Metropolis in the Canonical Ensemble

What is $\rho_{NVT}$

- Only need to know it to a multiplicative constant
\[
\rho_{NVT} \propto \exp(-\beta V)
\]

\[
\pi_{mn} = \min \left(1, \frac{\rho_n}{\rho_m} \right) = \min \left(1, \frac{\exp(-\beta V_n)}{\exp(-\beta V_m)} \right)
\]

\[
\pi_{mn} = \min (1, \exp[-\beta (V_n - V_m)])
\]
Example: Metropolis in the Canonical Ensemble

- What is $\rho_{NVT}$?
- Only need to know it to a multiplicative constant

$$\rho_{NVT} \propto \exp\left(-\beta V\right)$$

$$\pi_{mn} = \min\left(1, \frac{\rho_n}{\rho_m}\right) = \min\left(1, \frac{\exp\left(-\beta V_n\right)}{\exp\left(-\beta V_m\right)}\right) \quad (26)$$

$$\pi_{mn} = \min\left(1, \exp\left[-\beta (V_n - V_m)\right]\right) \quad (27)$$
1. Starting from initial state $m$, an elementary move is attempted to neighboring state $n$

$$x_n = x_m + [\text{rand()} - 0.5] \times \text{delx}$$

2. The probabilities of states $m$ and $n$ are compared.

3. Metropolis selection criterion:
   - If state $n$ is more probable, the move is immediately accepted.
   - If state $n$ is less probable, a random number $\zeta$ on $(0, 1)$ is generated. If $\frac{\rho_n}{\rho_m} \geq \zeta$ accept the move and the new state is $n$. If $\frac{\rho_n}{\rho_m} < \zeta$, reject the move.
   - The “new” state ($n$ for accepted moves, $m$ for rejected moves) is taken as the current state, and used in computing averages.

4. Iterate many times and compute averages. Stop when convergence is reached.
Metropolis Flowsheet

Metropolis Monte Carlo

Initial state \( m \)

Attempt elementary move:
move to neighboring state \( n \) with probability \( \rho_{mn} \). \( n \) is typically selected randomly from states near \( m \).

Compare a priori probabilities,
\[ \rho_n \rho_m \]

\( \rho_n \geq \rho_m \): state \( n \) is taken as the next state in the Markov chain. \( n \) becomes the current state.

\( \rho_n < \rho_m \): generate random number, compare with ratio \( \rho_n / \rho_m \)

state \( n \) taken as the next state in chain with probability \( \rho_n / \rho_m \)

state \( m \) retained as next state in chain with probability \( 1 - \rho_n / \rho_m \)

move accepted

move rejected

Has desired number of states been sampled?

no

yes

finished!

(c) 2019 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

\[ V_{ij}^{LJ} = 4\epsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^{6} \right] \] (28)

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 → configuration space, \((r_1, \cdots, r_M)\)
- States → configurations (atom arrangements)
- \(\rho_m \rightarrow \rho^{NVT}(r^{(m)}_1, \cdots, r^{(m)}_N)\,d\mathbf{r}^N\)
  \((d\mathbf{r}^N\) is an elementary volume element in configuration space\)
- \(\rho_n/\rho_m \rightarrow \frac{\rho^{NVT}(r^{(m)}_1, \cdots, r^{(m)}_N)}{\rho^{NVT}(r^{(n)}_1, \cdots, r^{(n)}_N)} = \frac{\rho^{NVT}_n}{\rho^{NVT}_m}\)
How to Perform MC Simulation of Liquids

Note that

\[ \rho_{NVT}^{(m)}(r_1^{(m)}, \ldots, r_N^{(m)}) \equiv \rho_m^{NVT} = \frac{\exp[-\beta \mathcal{V}(r_1, \ldots, r_N)]}{Z(NVT)} \equiv \frac{\exp[-\beta \mathcal{V}_m]}{Z} \]

(29)

Thus,

\[ \frac{\rho_{n}^{NVT}}{\rho_{m}^{NVT}} = \frac{\exp[-\beta \mathcal{V}_n]}{\exp[-\beta \mathcal{V}_m]} = \exp[-\beta (\mathcal{V}_n - \mathcal{V}_m)] = \exp[-\beta \Delta \mathcal{V}_{m \rightarrow n}] \]

(30)
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{31}
  
  \text{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{32}
  
  \text{Accept if } \zeta < \exp[-\beta (\mathcal{V}_n - \mathcal{V}_m)]
  
- Notice the Metropolis selection criterion only involves potential energy

(c) 2019 University of Notre Dame
Details, details...

- Decide on form of $\alpha$, 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 $\alpha$ 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 \( \mathbf{r}^{(m)}_i \)
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 \( \delta r_{\text{max}} \)
3. Denote local region by \( \mathcal{R} \). Note that if we use a cube (as done below), the sides are \( 2\delta r_{\text{max}} \) 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

\[
\alpha_{mn} = 1/N_\mathcal{R}; \quad \mathbf{r}^{(n)}_i \in \mathcal{R} \tag{33}
\]

\[
\alpha_{mn} = 0; \quad \mathbf{r}^{(n)}_i \notin \mathcal{R} \tag{34}
\]

(c) 2019 University of Notre Dame
To implement: an atom is chosen at random and given a uniform, random displacement along each Cartesian axes.

An adjustable parameter, $\delta r_{\text{max}}$, 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.

$\delta r_{\text{max}}$ is typically adjusted during equilibration so that about 50% of the attempted moves are successful

$$rx_{\text{new}} = rx(i) + (2.0 \times r\text{ranf}(\text{iseed}) - 1.0) \times dr_{\text{max}}$$
$$ry_{\text{new}} = ry(i) + (2.0 \times r\text{ranf}(\text{iseed}) - 1.0) \times dr_{\text{max}}$$
$$rz_{\text{new}} = rz(i) + (2.0 \times r\text{ranf}(\text{iseed}) - 1.0) \times dr_{\text{max}}$$
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, $\delta V_{nm} \leq 0$ and the move is always accepted. $\pi_{mn} = \alpha_{nm}/\alpha_{mn} = 1$

For “uphill” moves, a random number $\zeta$ is generated uniformly on (0,1)

If $\zeta < \exp[-\beta V_{nm}]$, ($\zeta_1$ in the figure), the move is accepted

Otherwise, ($\zeta_2$), the move is rejected

Over the course of the simulation, the net result is that energy changes such as $\delta V_{nm}$ are accepted with probability $\exp[-\beta \delta V_{nm}]$
What if $\alpha_{mn} \neq \alpha_{nm}$?

Recall the acceptance rule that satisfies detailed balance:

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

(35)

Simply retain $\alpha$ in the acceptance rule.

Key: In this case you must know what $\alpha_{mn}$ and $\alpha_{nm}$ are!

If $\alpha_{mn} \neq \alpha_{nm}$ and you think they are equal, the standard Metropolis algorithm will give you the wrong answer and you will not know it!
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

(c) 2019 University of Notre Dame
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 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.*
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.
An important advantage of the Monte Carlo method is that it allows one to use any arbitrary probability distribution to generate system configurations. So far, we have used the canonical distribution (NVT), but we can also use the isothermal-isobaric (NpT) or grand canonical (GC) distributions.

The NpT Monte Carlo method is useful as it allows you to control “experimental” variables P and T.
To switch ensembles, we should simply use a different probability distribution in the detailed balance equation.

\[ \pi_{mn} \alpha_{mn} \rho_m^{\text{ens}} = \pi_{nm} \alpha_{nm} \rho_n^{\text{ens}} \]  

(36)

What is \( \rho^{NPT} \)?
Recall the canonical ensemble partition function

\[ Q(N, V, T) = \frac{1}{\Lambda^{3N}N!} \int_0^L dr^N \exp \left[ -\beta U(r^N) \right] \] \quad (37)

where \( \Lambda = \frac{\hbar}{\sqrt{2\pi mkT}} \) is the deBroglie wavelength.
Consider “expanding” the canonical ensemble to allow for volume fluctuations in the range $0 \leq V \leq \infty$

$$\Delta (N, p, T) = \int_0^\infty e^{-\beta PV} Q (N, V, T) dV$$  \hspace{1cm} (38)

The probability that the system has a volume $V$ is

$$\rho^{NPT} (V) = \frac{Q (N, V, T) e^{-\beta PV}}{\Delta (N, p, T)}$$  \hspace{1cm} (39)

The probability of observing a microstate is then

$$\rho^{NPT} (V, r^N) = \frac{1}{\Lambda(T)^{3N} N!} \frac{e^{-\beta PV} e^{-\beta U(r^N)}}{\Delta (N, p, T)} dr^N dV$$  \hspace{1cm} (40)
NpT Monte Carlo

We need moves that correctly sample the NpT probability distribution given by equation 40. To achieve this, Cassandra uses typical thermal equilibration moves (translation, rotation, configurational-biased regrowths) and volume moves.

Before we define what volume moves are, it is convenient to define scaled atomic coordinates $s_i$ as

$$s_i = \frac{1}{L}r_i$$

where $L = V^{1/3}$, $r_i$ are the atomic cartesian coordinates for $i = 1, 2, \ldots, N$. 
In Cassandra, new volumes are sampled as follows:

1. Pick a random volume $\Delta V$ with uniform probability from the interval $[-\delta V_{max}, \delta V_{max}]$. The trial volume is $V + \Delta V$.
2. Scale the box lengths uniformly.
3. Scale the center of mass of each molecule uniformly.

Note that the probability of selecting $\Delta V$ is the same as selecting $-\Delta V$ which makes scaling the volume symmetric. In Cassandra, the center of mass of each molecule are scaled consistently with the proposed volume change. To write this as a ratio of attempt probabilities, we can use scaled coordinates as

$$\frac{\alpha_{nm}}{\alpha_{mn}} = \frac{dr_n^N}{dr_m^N} = \frac{V_n^N ds_n^N}{V_m^N ds_m^N} = \frac{V_n^N}{V_m^N}$$

(42)
Substituting the attempt probability ratio and the NpT probability distribution in the detailed balance condition, we can get the acceptance rule for volume moves

\[
\pi_{mn} = \text{min} \left[ 1, \left( \frac{V_n}{V_m} \right)^N e^{-\beta \Delta U - \beta P \Delta V} \right] \tag{43}
\]
The Gibbs Ensemble Monte Carlo (GEMC) method is a widely used method for computing vapor-liquid equilibria. A distinguishing feature is that two boxes are used to represent each bulk phase. There is no explicit interface.
This method was designed in response to the challenge of simulating VLE using systems with explicit interfaces.

<table>
<thead>
<tr>
<th>Number of particles</th>
<th>% of particles near interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,000</td>
<td>49%</td>
</tr>
<tr>
<td>64,000</td>
<td>14%</td>
</tr>
<tr>
<td>1,000,000</td>
<td>6%</td>
</tr>
</tbody>
</table>

Measuring VLE bulk properties can require large system sizes if we use an interface! We would need to arbitrarily define where each phase is.
The GEMC method was designed to satisfy the thermodynamic conditions for phase equilibria:

\[ T_{vapor} = T_{liquid} \quad (44) \]
\[ P_{vapor} = P_{liquid} \quad (45) \]
\[ \mu_{i,\text{vapor}} = \mu_{i,\text{liquid}} \quad (46) \]
Gibbs Ensemble Monte Carlo

There are two versions of the GEMC Method. The first one, GEMC-NVT, is used to simulate the VLE of pure substances. It requires the temperature as a simulation input, consistent with the Gibbs phase rule (one intensive variable to fully specify a state of a single species with two phases).

The second one, GEMC-NPT, is used to simulate the VLE of mixtures. It requires temperature and pressure as simulation inputs (two intensive variables to fully specify a state of a mixture with two species and two phases).

Let’s start with GEMC-NVT!
Gibbs Ensemble Monte Carlo

Consider two boxes A and B in thermal equilibrium with a heat bath at temperature $T$. Each box has a number of particles $N_A$ and $N_B$, respectively. Similarly, the boxes have volume of $V_A$ and $V_B$. The number of moles and the volume of each box are kept constant. The partition function of the combined system is:

$$Q_{Gibbs}(T, V_A, V_B, N_A, N_B) = Q_A(T, V_A, N_A)Q_B(T, V_B, N_B) \quad (47)$$

$$Q_A(T, V_A, N_A) = \left[ \frac{1}{N_A! \Lambda(T)^{3N_A}} \int e^{-\beta U(r^{N_A})} \, dr^{N_A} \right] \quad (48)$$

$$Q_B(T, V_B, N_B) = \left[ \frac{1}{N_B! \Lambda(T)^{3N_B}} \int e^{-\beta U(r^{N_B})} \, dr^{N_B} \right] \quad (49)$$
Gibbs Ensemble Monte Carlo

Now, let’s allow exchange of volume between these two boxes keeping the total system volume constant

\[ V = V_A + V_B = \text{constant}. \]

\[
Q_{\text{Gibbs}}(T, V, N) = \int_0^V Q_A(T, V_A, N_A) Q_B(T, V_B, N_B) \, dV_A \tag{50}
\]

If we allow exchange of particles between these two boxes keeping the total particle number constant

\[ N = N_A + N_B = \text{constant}, \]

the partition function becomes

\[
Q_{\text{Gibbs}}(T, V, N) = \sum_{N_A=0}^N \int_0^V Q_A(T, V_A, N_A) Q_B(T, V_B, N_B) \, dV_A \tag{51}
\]
By using scaled atomic coordinates, the partition function can be explicitly written as

\[ Q_{\text{Gibbs}}(T, V, N) = \frac{1}{\Lambda(T)^{3N}} \sum_{N_A=0}^{N} \frac{1}{N_A! N_B!} \]

\[ = \frac{1}{\Lambda(T)^{3N}} \int_0^V dV_A \left[ (V_A)^{N_A} (V_B)^{N_B} \right] \left[ \int e^{-\beta U(s^{N_A})} ds^{N_A} \right] \left[ \int e^{-\beta U(s^{N_B})} ds^{N_B} \right] \]
We can extract the ensemble probability distribution by looking at the partition function we just wrote.

\[ \varphi_{\text{Gibbs}}(T, V, N) \propto \frac{V_A^{N_A} V_B^{N_B}}{N_A! N_B!} e^{-\beta [U(s^{N_A}) + U(s^{N_B})]} \] (52)
The acceptance rule for thermal equilibration moves is

$$\frac{\text{acc}(o \rightarrow n)}{\text{acc}(n \rightarrow o)} = \frac{\alpha_{nm}}{\alpha_{mn}} e^{-\beta \Delta U_A}$$  \hspace{1cm} (53)

For volume exchange, we obtain

$$\frac{\text{acc}(o \rightarrow n)}{\text{acc}(n \rightarrow o)} = \frac{(V_A + \Delta V)^{N_A} (V_B - \Delta V)^{N_B}}{V_A^{N_A} V_B^{N_B}} e^{-\beta \Delta U_A - \beta \Delta U_B}$$  \hspace{1cm} (54)
For particle transfers, we need to build the attempt probability. For simplicity, let us assume we have propane (1 fragment). We will transfer a molecule from box A to box B. The forward attempt probability $\alpha_{mn}$ is

1. Pick box A, 1/2
2. Pick a molecule in box A, $1/N_A$
3. Pick a new conformation of propane from the library, $\frac{e^{-\beta U(q_{N+1}^{int})} dq_n}{Z_{int}}$
4. Pick an orientation of this fragment, $\frac{dq_{rot}}{Z_{rot}}$
5. Generate $N_{\text{trials}}$ trial positions in box B for center-of-mass insertions, $dr_{\text{COM}}/N_{\text{trials}}$
6. Pick one of the trial positions with probability $\frac{e^{-\beta \Delta U^\text{ins}_k}}{\sum_k e^{-\beta \Delta U^\text{ins}_k}}$

The reverse attempt probability can be built analogously.

(c) 2019 University of Notre Dame
The acceptance rule for particle transfers is

\[
\frac{\text{acc}(o \rightarrow n)}{\text{acc}(n \rightarrow o)} = \frac{\alpha_{nm}}{\alpha_{mn}} e^{-\beta \Delta U_A - \beta \Delta U_B}
\]  

(55)
Features of the GEMC-NPT method

- Used to simulate multicomponent systems
- Volume of each box is allowed to fluctuate independently from the other box (i.e. total volume is not constant)
- Pressure must be specified as a simulation parameter in addition to the temperature.

The derivation of the partition function is very similar to GEMC-NVT. Here we present the probability distribution of microstates

$$p_m \propto (V_A)^{N_A}(V_B)^{N_B}e^{-\beta U(s^{N_A})-\beta U(s^{N_B})-\beta P(V_A+V_B)}$$

(56)