Monte Carlo

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

Using Probabilitic Methods to Solve Problems

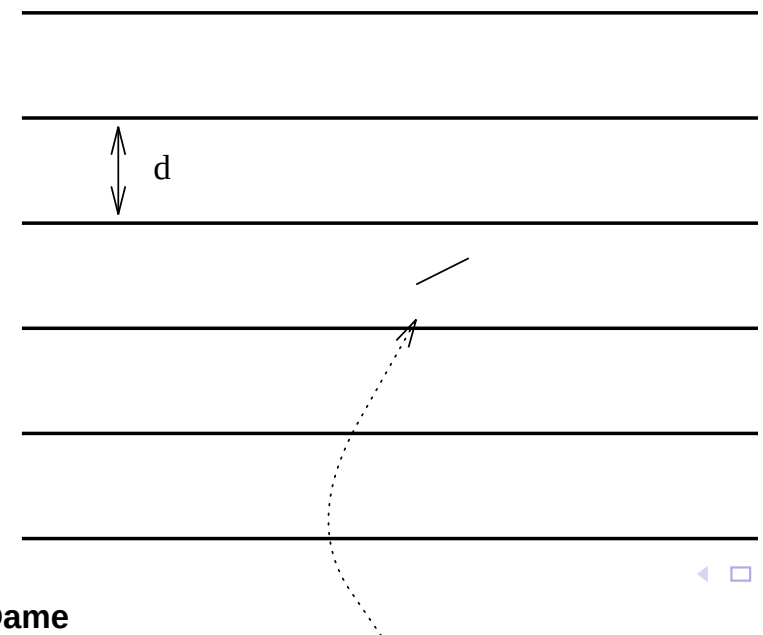
It has been known for a long time that it is possible to use probabilistic methods to determine deterministic quantities.

- 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 π

George Louis LeClerc Comte de Buffon

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

If a needle of length ℓ 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 probabilitic way of estimating π
- Lazzerin(1901): Reportedly dropped a 2.5 cm needle 3408 times
- 1,808 crossings on 3 cm grid
- $\pi = (4/3) * 3408/1808 = 3.14159292.$
- You could imagine how much easier this would be on a computer today...

Did Lazzarini Cheat?

- $\pi = (4/3) * 3408/1808 = 3.14159292.$
- This is a remarkable accurate result!
- Estimate reduces to 355/113
- This fraction was the rational approximation of π obtained by Chinese mathematician Tsu Ch'ung-chi around A.D. 500...
- See *Mathematics Magazine*, 67, 83 (1994) for interesting article

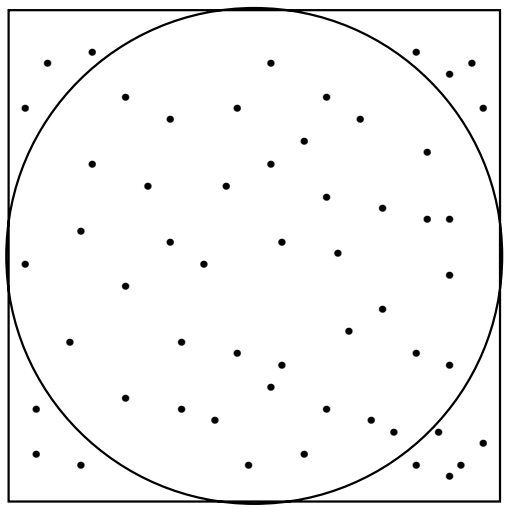
How lucky can you get?

- 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

Let's perform the simulations and see what we get! Run demo Buffon

Other ways of estimating π stochastically

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



Thus, $\pi = 4\frac{N_{hits}}{N_{trials}}$
Let's simulate it.

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

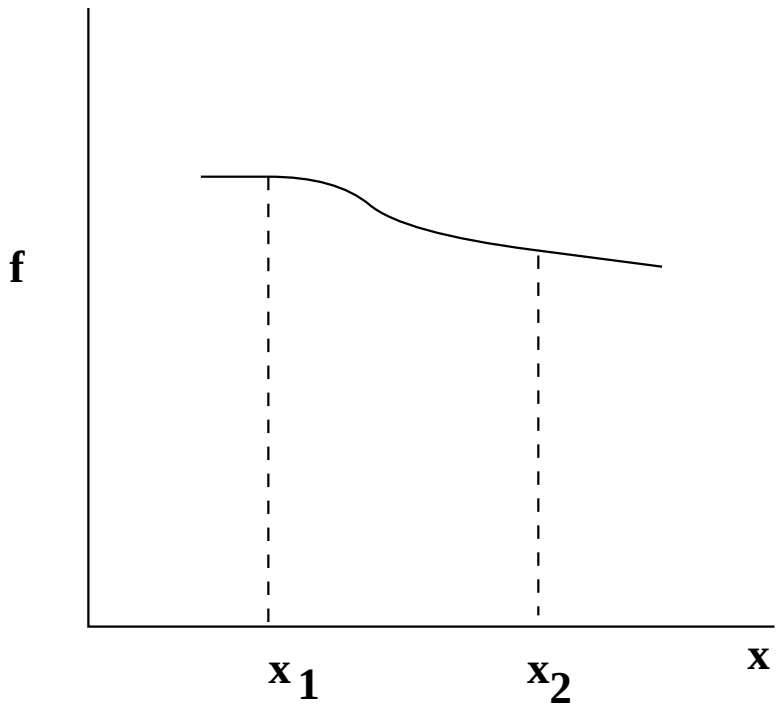
Monte Carlo

- Easiest to think of MC used for simulating stochastic processes
- *This is too restrictive!*
- Can solve “deterministic” problems also with MC
- Example: ealuating 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



Evaluating Integrals with Monte Carlo

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

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

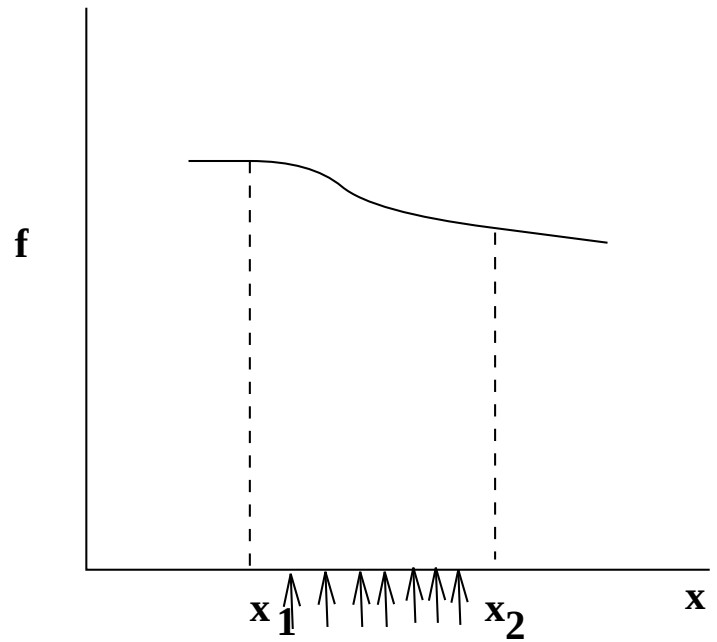
- $\rho(x)$ is an arbitrary probability density function.
- Then generate N_{trials} random numbers ξ from the distribution $\rho(x)$ in the range (x_1, x_2)
- Evaluate f at each step.

Evaluating Integrals with Monte Carlo

- We see that

$$I = \left\langle \frac{f(\xi)}{\rho(\xi)} \right\rangle_{\rho(x)} \tag{2}$$

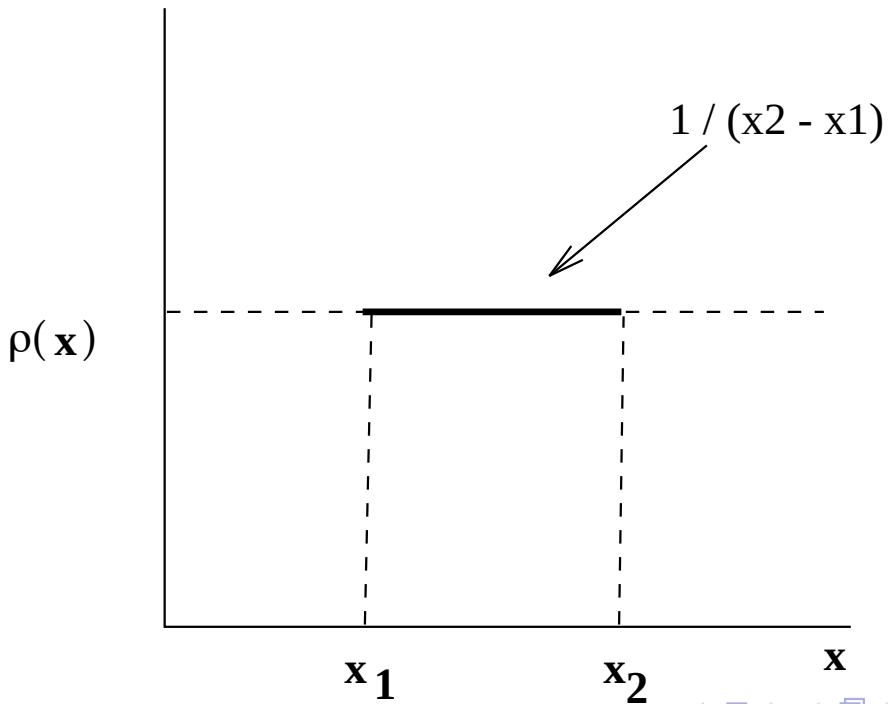
- Brackets indicate an average over the N_{trials} evaluations, and that $\rho(x)$ used as probability distribution.



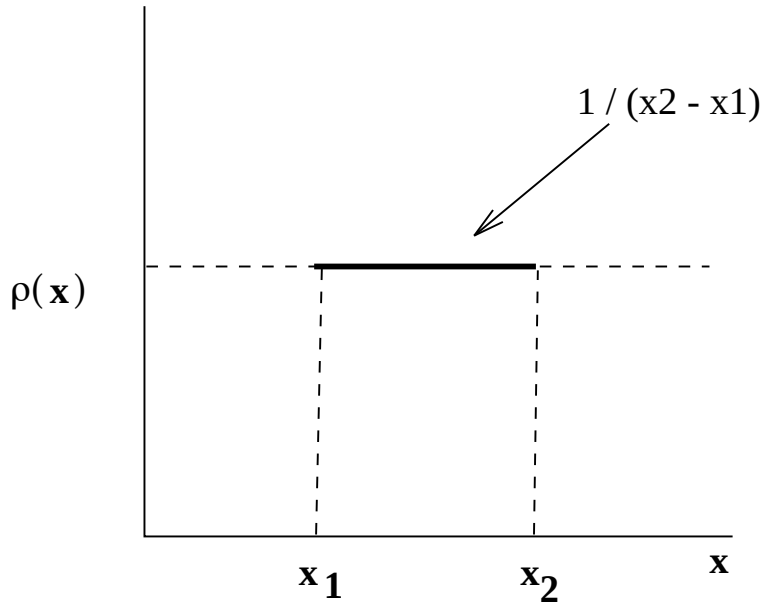
Evaluating Integrals with Monte Carlo

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

$$\rho(x) = \frac{1}{(x_2 - x_1)}, \; x_1 \leq x \leq x_2 \tag{3}$$



Evaluating Integrals with Monte Carlo



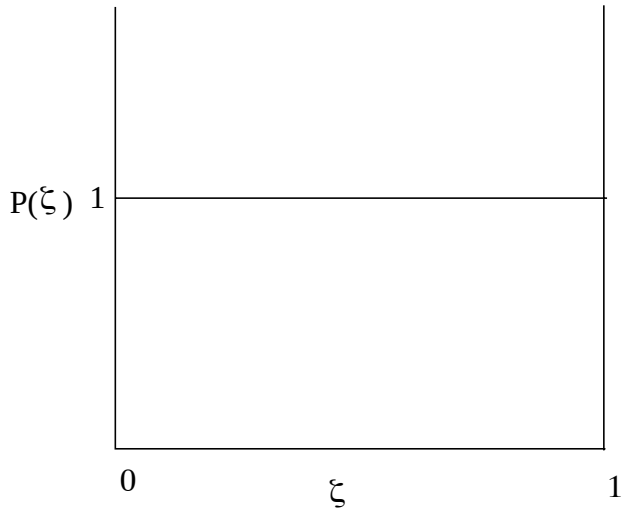
- With uniform distribution, integral is

$$I \approx \frac{(x_2 - x_1)}{N_{trials}} \sum_{i=1}^{N_{trials}} f(\xi_i) \tag{4}$$

- Justification for eq 4: Mean Value theorem.

Algorithm for Evaluating Integrals with Monte Carlo

1. Generate random number, ζ, 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

$$I \approx \frac{(x_2 - x_1)}{N_{trials}} \sum_{i=1}^{N_{trials}} f(\xi_i) \tag{5}$$

Monte Carlo Integration to Estimate π

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

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

  with x between $x_1 = 0$ and $x_1 = 1$.
- If one uses the procedure outlined above, the estimate of π after 10^7 trials is 3.14169.
- Exercise: try this on your own

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

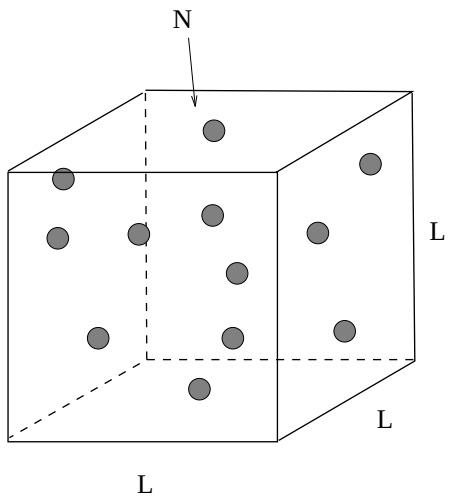
$$f(x) = (1 - x^2)^{-1/2} \tag{7}$$

  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?

Monte Carlo and Multi-dimensional Integrals

- Consider the configurational integral for $N = 10$ atoms

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



- 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

$$Z \approx \frac{V^N}{N_{trials}} \sum_{i=1}^{N_{trials}} \exp[-\beta \mathcal{V}(\mathbf{r}_1^{(i)}, \cdots, \mathbf{r}_N^{(i)})] \tag{9}$$

where $(\mathbf{r}_1^{(i)}, \cdots, \mathbf{r}_N^{(i)})$ 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 $f(\mathbf{r}^N)$ 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_{trials} much less than this.
- Of course, our estimate gets better as N_{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. $N = \mathcal{O}(100)$, even standard Monte Carlo integration becomes infeasible.
- Why?
- Consider MC integration of Z for hard sphere fluid

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

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

$$< \mathcal{A} >_{NVT} = \frac{\int d\mathbf{r} \mathcal{A} \exp[-\beta \mathcal{V}]}{\int d\mathbf{r} \exp[-\beta \mathcal{V}]} \tag{11}$$

  were $\approx \frac{\sum_{i=1}^{N_{trials}} \mathcal{A}_i \exp[-\beta \mathcal{V}_i]}{\sum_{i=1}^{N_{trials}} \exp[-\beta \mathcal{V}_i]}$

- 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*

$$< f >_{ens} = \frac{\int d\mathbf{r}^N f(\mathbf{r}^N)\rho^{ens}(\mathbf{r}^N)}{\int d\mathbf{r}^N \rho^{ens}(\mathbf{r}^N)} \tag{12}$$

- Random configurations $\longrightarrow$ few configurations contribute to integral
- Idea: Sample configuration space *nonuniformly* $\longrightarrow$ more probable (i.e. low energy) states sampled more frequently

Importance Sampling

- What if we cold sample states according to ρ^{ens}?
- Then calculating ensemble average $< f >$ amounts to taking a simple *arithmetic average* over the sampled microstates

$$< f >_{ens} = \frac{1}{N_{trials}} \sum_{i=1}^{N_{trials}} f(i) \tag{13}$$

- Same as taking averages during MD!
- Metropolis and co–workers did this in late 1940s[1]

[1] Metropolis, N; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; and Teller, E., *J. Chem. Phys.*, 21, 1087-1092, (1953)

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

Markov Chains

- To understand the Metropilis 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.

Markov Chains

- The *transition probability* links two states γ_m and γ_n
- Define π_{mn} as the *probability that a trial produces state n, given that the previous trial resulted in state m*
- π_{mn} depends on values of *m* and *n*, but is *independent of where within the sequence the considered trial lies*
- Thus π_{mn} is independent of "time" or number of trials
- Given values of π_{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

Markov Chain Example (after Allen and Tildesley)

- 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 π a *stochastic matrix*.

Markov Chain Example (after Allen and Tildesley)

- Assume that on day 1, the computer is equally likely to be up and down. That is

$$\boldsymbol{\rho}^{(1)} = (\rho_U \ \rho_D) = (0.5 \ 0.5)$$

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

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

- Up: $(0.5 * 0.6) + (0.5 * 0.3) = 0.45$
- Down: $(0.5 * 0.7) + (0.5 * 0.4) = 0.55$
- Thus, there is a 45% chance your computer will be up on day 2, but a 55% chance it will be down

Markov Chain Example (after Allen and Tildesley)

- Assume that on day 1, the computer is equally likely to be up and down. That is

$$\boldsymbol{\rho}^{(1)} = (\rho_U \ \rho_D) = (0.5 \ 0.5)$$

- What are the probabilities the computer will be up/down the next day?
  $\boldsymbol{\rho}^{(2)} = \boldsymbol{\rho}^{(1)}\pi = (0.45,\ 0.55)$
- Up: $(0.5 * 0.6) + (0.5 * 0.3) = 0.45$
- Down: $(0.5 * 0.7) + (0.5 * 0.4) = 0.55$
- Thus, there is a 45% chance your computer will be up on day 2, but a 55% chance it will be down

Markov Chain Example (after Allen and Tildesley)

- Repeating the process, probabilities for day 3: $\boldsymbol{\rho}^{(3)} = \boldsymbol{\rho}^{(2)}\pi = \boldsymbol{\rho}^{(1)}\pi\pi = (0.435,\ 0.565)$
- There is now only a 43.5% chance of it being up
- Write a small program or Excel worksheet to compute the probabilities for days 5 and 6 and 20
- Result $\rho^{(5)} = (0.4287,\ 0.5713)$ $\rho^{(6)} = (0.4286,\ 0.5714)$ $\rho^{(20)} = (0.4286,\ 0.5714)$
- Clearly, there is a limiting distribution which we reached, given by the formula

$$\rho = \lim_{\tau \to \infty} \rho^{(1)}\pi^{\tau} \tag{14}$$

Markov Chain Example (after Allen and Tildesley)

- Repeating the process, probabilities for day 3: $\boldsymbol{\rho}^{(3)} = \boldsymbol{\rho}^{(2)}\pi = \boldsymbol{\rho}^{(1)}\pi\pi = (0.435,\ 0.565)$
- There is now only a 43.5% chance of it being up
- Write a small program or Excel worksheet to compute the probabilities for days 5 and 6 and 20
- Result $\boldsymbol{\rho}^{(5)} = (0.4287,\ 0.5713)$ $\boldsymbol{\rho}^{(6)} = (0.4286,\ 0.5714)$ $\boldsymbol{\rho}^{(20)} = (0.4286,\ 0.5714)$
- Clearly, there is a limiting distribution which we reached, given by the formula

$$\boldsymbol{\rho} = \lim_{\tau \to \infty} \boldsymbol{\rho}^{(1)}\pi^{\tau} \tag{14}$$

Limiting Distributions

- We see from eqn 14 that the limiting distribution, ρ, must satisfy the eigenvalue equation

$$\rho\pi = \rho \tag{15}$$

$$\sum_m \rho_m \pi_{mn} = \rho_n \tag{16}$$

- ρ is an eigenvector of the stochastic matrix π corresponding to an eigenvalue of 1.
- ρ is *completely determined* by π, 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 γ_m in phase space.
- Note: the multi–dimensional space may be discrete (ρ_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 γ_m).
- Goal: An efficient numerical procedure for sampling the multi-dimensional space, according to the probability distribution $\{\rho_m\}$.

Metropolis Monte Carlo Algorithm

- What does “sampling” mean?
- We wish to pick a finite set of points (states) $m_1, m_2, \cdots, m_t, \cdots, m_{N_{trials}}$ such that the probability of finding each state m_t in the sequence is practically equal to ρ_{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

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

- 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) $m_1, m_2, \cdots, m_t, \cdots, m_{N_{trials}}$ such that the probability of finding each state m_t in the sequence is practically equal to ρm_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

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

- 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) $m_1, m_2, \cdots, m_t, \cdots, m_{N_{trials}}$ such that the probability of finding each state m_t in the sequence is practically equal to ρm_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

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

- 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) $m_1, m_2, \cdots, m_t, \cdots, m_{N_{trials}}$ such that the probability of finding each state m_t in the sequence is practically equal to ρm_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

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

- 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
  $m_1 \longrightarrow m_2 \longrightarrow \cdots \longrightarrow m_t \longrightarrow \cdots \longrightarrow m_{N_{trials}}$
- Enforce a condition known as "detailed balance". Many variations, but for our purpose we define it as

$$\pi_{mn}\alpha_{mn}\rho_m = \pi_{nm}\alpha_{nm}\rho_n \tag{18}$$

- α_{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

$$\pi_{mn} = \frac{\alpha_{nm}}{\alpha_{mn}}, \ \textit{if}\ \rho_n \geq \rho_m, n \neq m \tag{19}$$

$$\pi_{mn} = \frac{\alpha_{nm}}{\alpha_{mn}} \frac{\rho_n}{\rho_m}, \ \textit{if}\ \rho_n < \rho_m, n \neq m \tag{20}$$

- It is also possible to stay in the same state

$$\pi_{mm} = 1 - \sum_{m \neq n} \pi_{mn} \tag{21}$$

Attempt Matrix and Metropolis Acceptance Rule

- Often moves are attempted only between close–lying states
- That is, most of the elements of α are zero
- Typically, α_{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, α 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

$$\pi_{mn} = min(1, \frac{\rho_n}{\rho_m}) \tag{22}$$

- "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

$$\frac{\rho_n}{\rho_m} < 1 \tag{23}$$

  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.

Additional Details

$$\pi \boldsymbol{\rho} = \boldsymbol{\rho} \tag{24}$$

- 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 ρ.
- The acceptance rule means $\boldsymbol{\rho}$ 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.

Other Algorithms

- There are other algorithms that satisfy the transition matrix criteria listed above. So–called Barker sampling yields

$$\pi_{mn} = \alpha_{mn}\rho_n/(\rho_n + \rho_m),\ m \neq n \tag{25}$$

$$\pi_{mn} = 1 - \sum_{n \neq m} \pi_{mn} \tag{26}$$

- Eqns 25 - 26 also satisfy microscopic reversibility, but in general are not as efficient as the Metropolis algorithm.

Example: Metropolis in the Canonical Ensemble

- What is ρ_{NVT}
- Only need to know it to a multiplicative constant

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

$$\pi_{mn} = min\left(1, \frac{\rho_n}{\rho_m}\right) = min\left(1, \frac{\exp\left(-\beta \mathcal{V}_n\right)}{\exp\left(-\beta \mathcal{V}_m\right)}\right) \tag{27}$$

$$\pi_{mn} = min\left(1, \exp\left[-\beta(\mathcal{V}_n - \mathcal{V}_m)\right]\right) \tag{28}$$

Example: Metropolis in the Canonical Ensemble

- What is ρ_{NVT}
- Only need to know it to a multiplicative constant
  $\rho_{NVT} \propto \exp(-\beta\mathcal{V})$

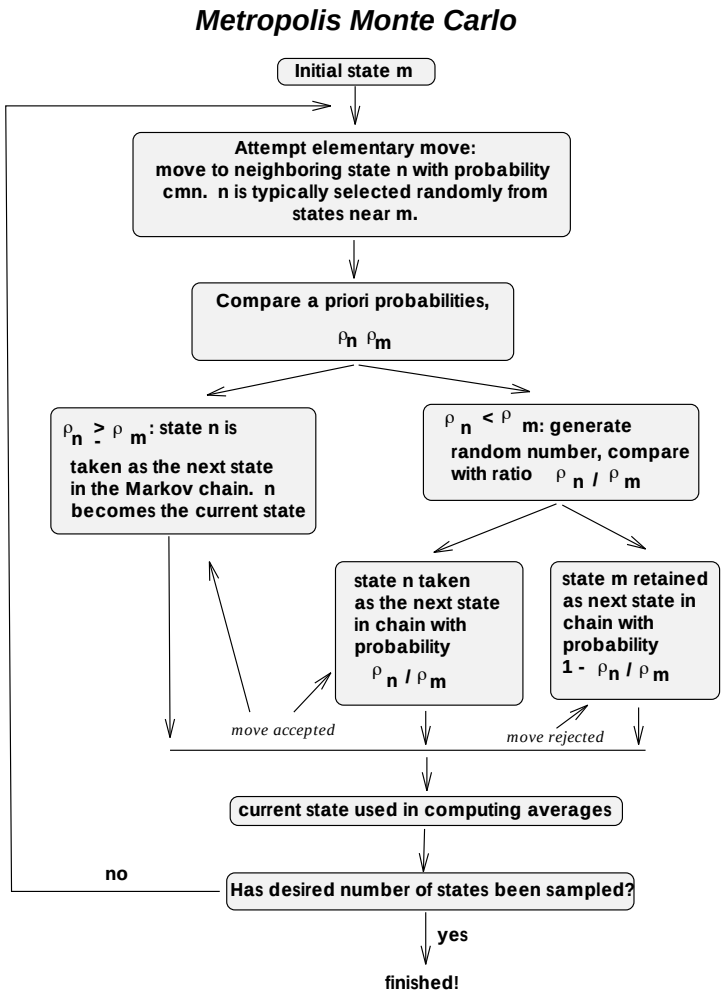
$$\pi_{mn} = min\left(1, \frac{\rho_n}{\rho_m}\right) = min\left(1, \frac{\exp(-\beta\mathcal{V}_n)}{\exp(-\beta\mathcal{V}_m)}\right) \tag{27}$$

$$\pi_{mn} = min\left(1, \exp\left[-\beta(\mathcal{V}_n - \mathcal{V}_m)\right]\right) \tag{28}$$

Metropolis Flowsheet

1. Starting from initial state m, an elementary move is attempted to neighboring state n
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 ζ on $(0, 1)$ is generated. If $\rho_n/\rho_m \geq \zeta$ accept the move and the new state is n. If $\rho_n/\rho_m < \zeta$, 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 simulation of 3-state model

- Revist the three-state model and compute the average energy and probability of each state at 200 K, 300 K and 1000 K.
- Compare with analytical result.
- Method 1: don't count "old" state of rejected moves in average
- Method 2: do count "old" state of rejected moves in average
- Explore accuracy of result vs. number of iterations.