Lecture 19
Monte Carlo Simulations

Understanding Molecular Simulations: Frenkel and Smit: Chapter 2


"Monte Carlo" Casino
GERMANY
LUXEMBOURG
LIECHTENSTEIN
FRANCE
SWITZERLAND
ITALY
SAN
MARINO
MONACO
ANDORRA
VATICAN
CITY

“*Do random events ever lead to concrete results? Seems unlikely – after all, they're random*.”

Burger EB, Starbird MP. *The Heart of Mathematics: an Invitation to Effective Thinking*. 2005.

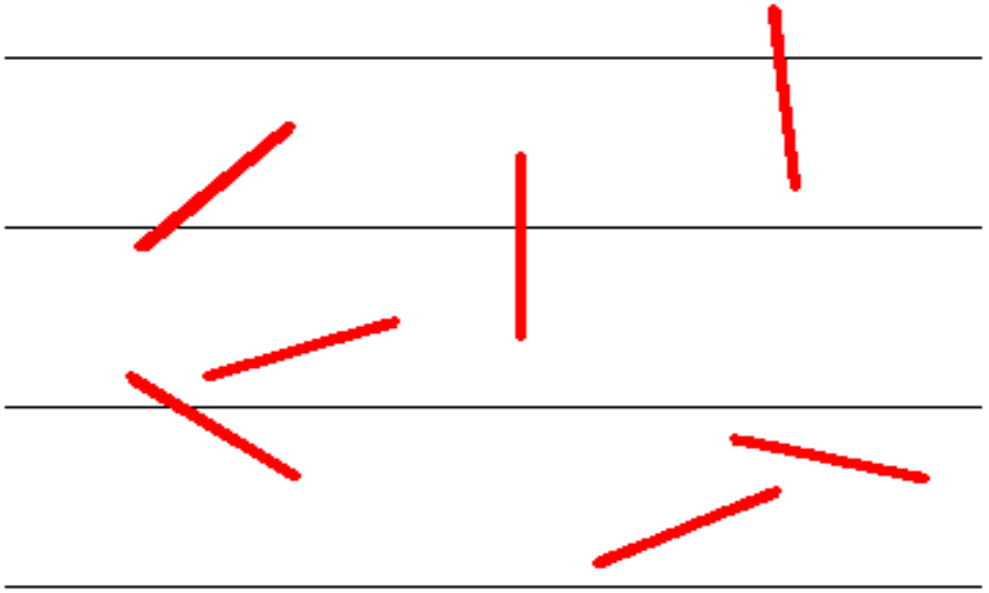
Outline

- A brief history
- Grid-based vs Stochastic integration methods
- Importance sampling
- Von Neumann rejection method
- Random walk and Markov chains
- Metropolis Algorithm
- MD vs MC

A brief history of Monte Carlo method

Buffon's needle experiment....determining the value of π..

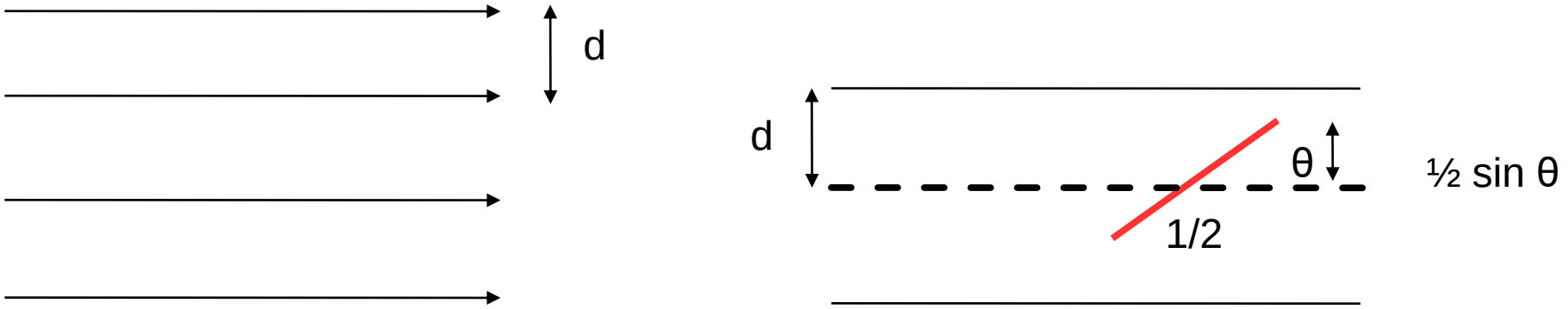
Georges Louis LeClerc, Comte de Buffon (1707-1788)



What is the probability that the needle crosses the line on the floor ?

First instance of Monte Carlo simulation

Buffon's needle experiment

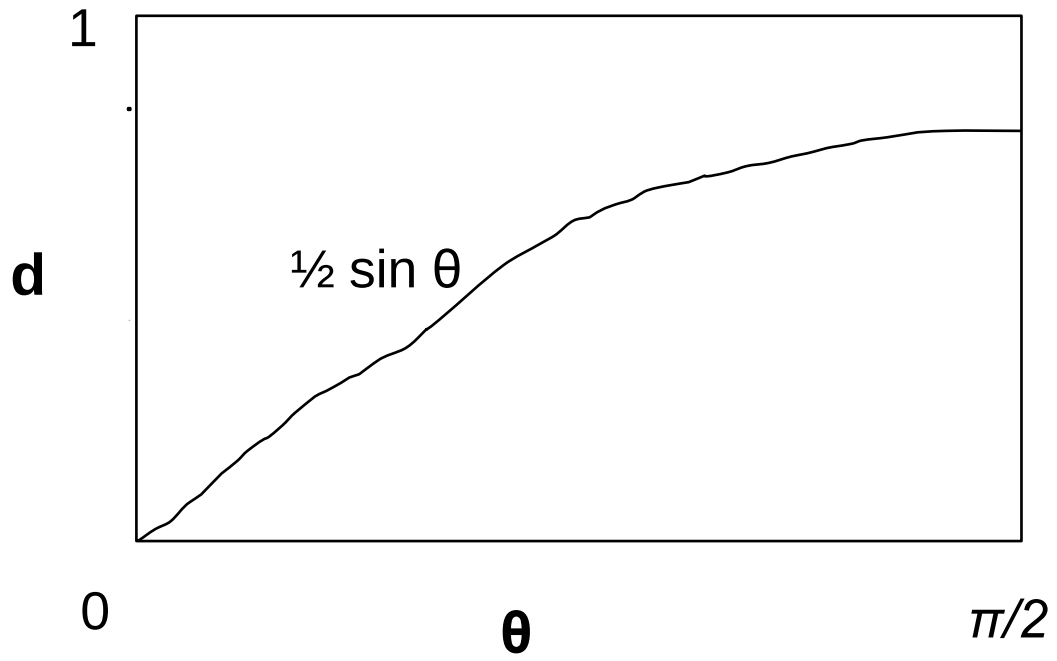


d = distance of the middle of the needle to closest line ($0 \le d \le 1$)

Θ = angle of needle relative to horizontal lines ($0 \le \theta \le \pi/2$)

The needle crosses line if $d \le \frac{1}{2} \sin \theta$

Buffon's needle experiment



Probability = $\frac{\text{Area under curve}}{\text{Area of rectangle}} = \int_0^{\pi/2} \frac{1}{2}\sin\theta d\theta = \frac{1}{2}$

Probability for one needle = $\frac{1/2}{\pi/2} = \frac{1}{\pi}$

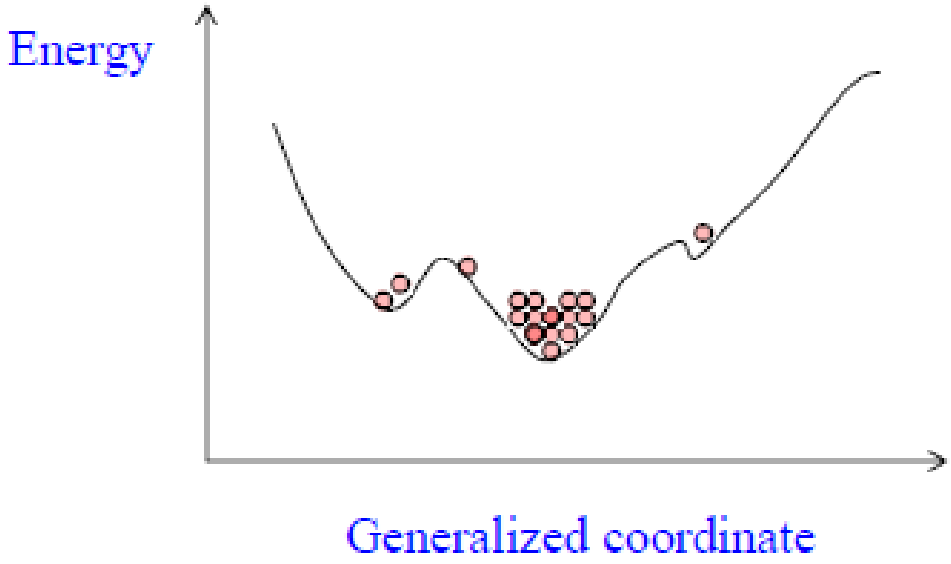
Repeat N times and count the number of intersections (successes) to be x
Probability = no. of successes/ total no. of tosses = x/N ~ 1/π

N/x ~ π

Ensemble averages

Computing thermal average of equilibrium systems

$$\langle A \rangle = \frac{\sum_i \exp(-\epsilon_i / k_B T) A_i}{\sum_i \exp(-\epsilon_i / k_B T)}$$



In the thermodynamic limit:

$$\langle A \rangle = \frac{\int \mathrm{d}\mathbf{p}^N \mathrm{d}\mathbf{r}^N \; A(\mathbf{p}^N, \mathbf{r}^N) \exp[-\beta \mathcal{H}(\mathbf{p}^N, \mathbf{r}^N)]}{\int \mathrm{d}\mathbf{p}^N \mathrm{d}\mathbf{r}^N \; \exp[-\beta \mathcal{H}(\mathbf{p}^N, \mathbf{r}^N)]}$$

High dimensional integral to be computed !

Integration: $I = \int_a^b f(x)dx$

Grid-Based

- Define a set of grid points and associated weights

$$I \approx \sum_i w_i f(x_i)$$

- Error

$$\varepsilon \leq ch^k$$

- Computational efficiency will grow exponentially with dimensionality

$$T_c \propto (c/\varepsilon)^{d/k}$$

Stochastic

- Sample points randomly and uniformly in the interval [a:b]

$$I \approx \frac{b-a}{N} \sum_{i=1}^{N} f(x_i)$$

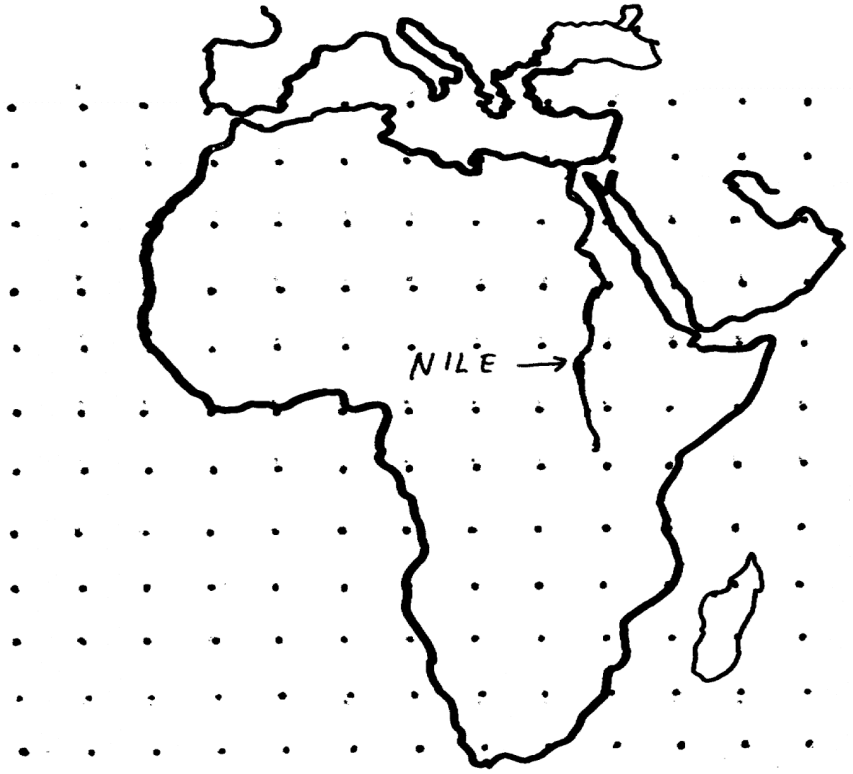
- Error: Central Limit Theorem

$$\varepsilon = \sqrt{\frac{\sigma^2}{N}} = \sqrt{\left(f^2 - f^2\right)/N}$$

$$T_c = t_0 \sigma^2 \varepsilon^{-2}$$

- Computational efficiency

Measure the depth of the river Nile



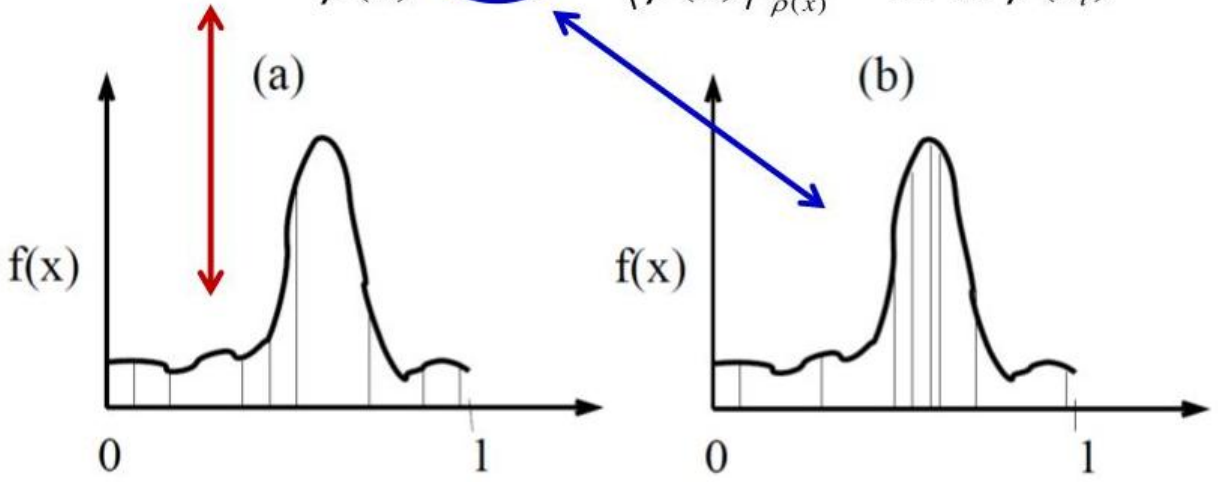
Using quadrature methods..
Grid-based methods

Measure the depth of the river Nile

Better approach:
Stochastic Method =
Importance sampling

Importance Sampling: Reducing the Variance

$$A = \int_a^b f(x)\,dx = \int_a^b \frac{f(x)}{\rho(x)}\,\rho(x)\,dx \equiv \left\langle \frac{f(x)}{\rho(x)} \right\rangle_{\rho(x)} \cong \frac{1}{M}\sum_{i=1}^{M} \frac{f(x_i)}{\rho(x_i)}$$

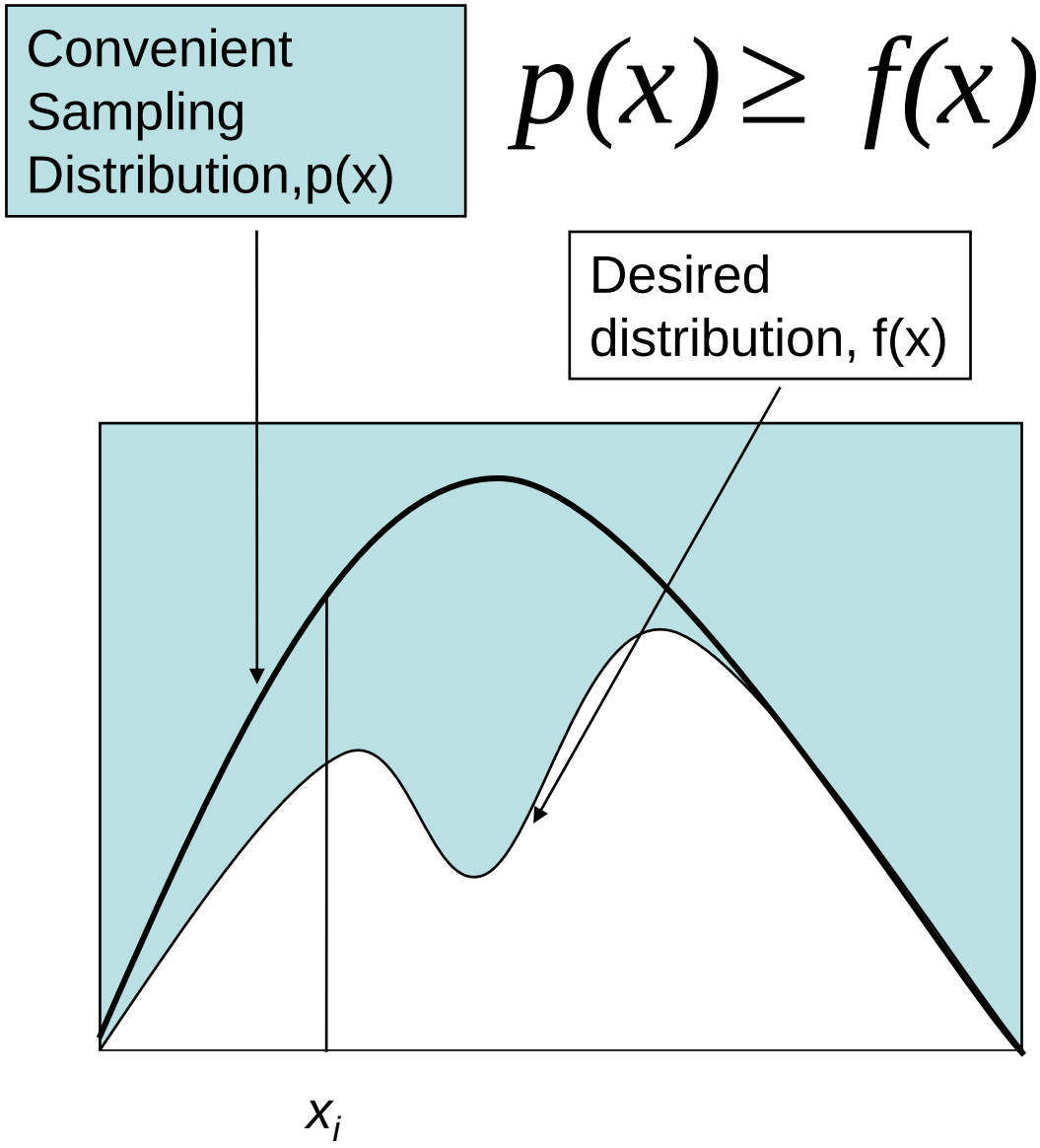


- Sampling distribution p(x) must be finite and non-negative.
- Sampling points from uniform distribution may not be the best way.
- Weight of the integral (p(x)) is large where f(x) is large.

$$\sigma^2 = \left\langle \frac{f^2}{p^2} \right\rangle - \left\langle \frac{f}{p} \right\rangle^2$$

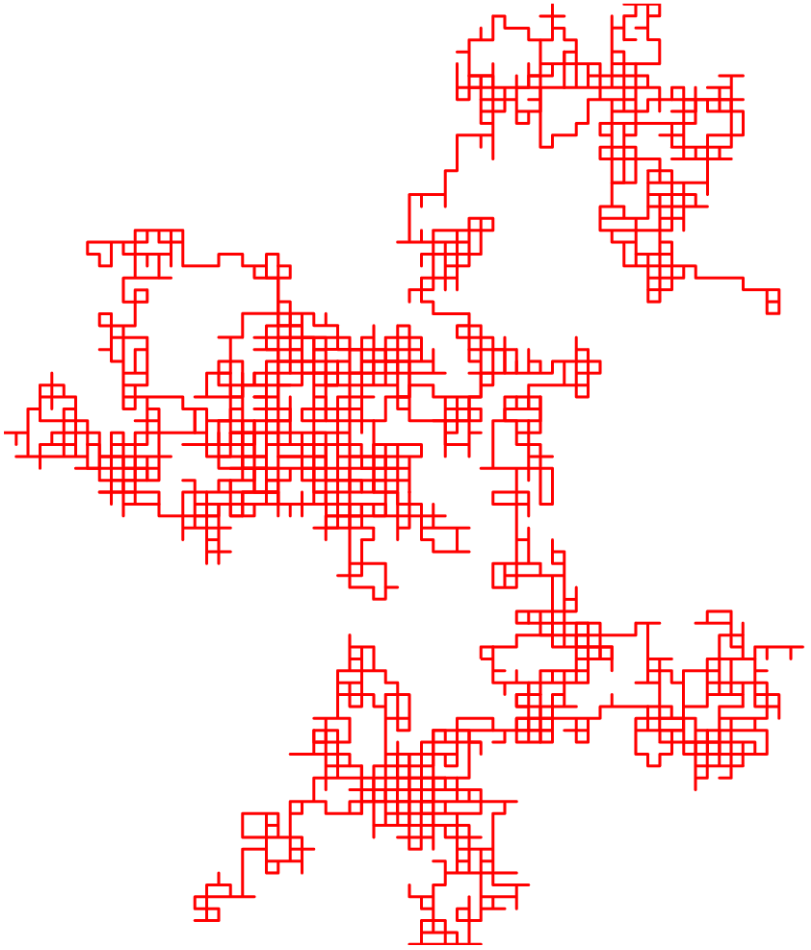
Non-uniform Distributions: von Neumann Rejection Method

1. Sample x_i from the distribution $p(x_i)$ and compute $f(x_i)$.
2. Sample a random number ξ uniformly distributed between 0 and 1.
3. If $f(x_i)/p(x_i) > \xi$, accept x_i; otherwise reject x_i.



Random Walk

- A drunk person moving in all directions randomly.
- He forgets the past steps, new steps are independent of previous steps.
- Forms a MARKOV chain.



Random Walk

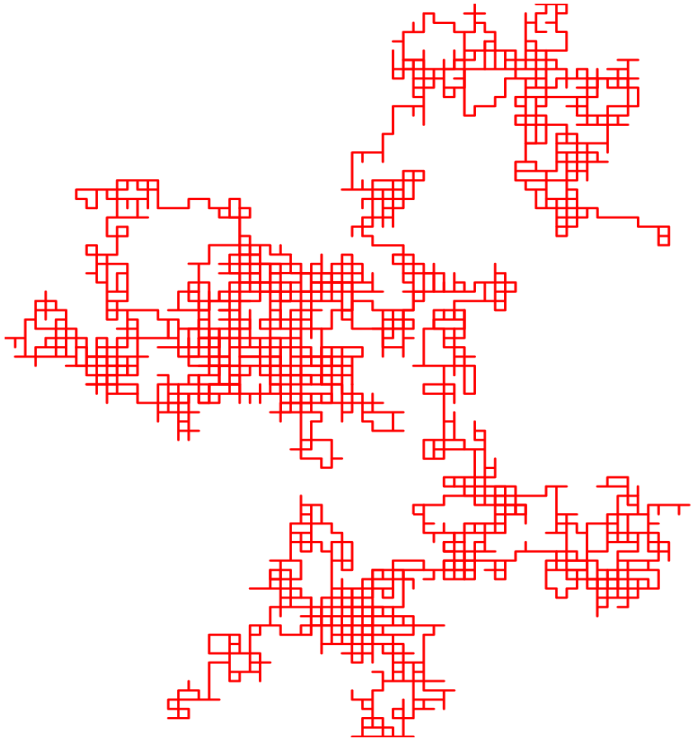
- A set of probabilistic rules which allow for the motion of the state point of the system through configuration space.
- A walker is a moving state point.
- Features of random walk:

1. The **available configuration space** and **attributes** of walker must be defined.
2. An ensemble of walkers represents the probability distribution of the system at some point in time, P(x,t).
3. **Source distribution** at time t=0 must be defined.

$$P(x,0) \geq 0, \int P(x,0)dx = 1$$

4. The system's kinetics will be determined by the **transition matrix, Txx'**

$$T_{xx'} \geq 0 \text{ and } \int T_{xx'} dx' \geq 1$$



Density of arrivals at some point x, is denoted by χ(x).

- It is the generated quantity of interest in random walk simulation.
- It is the expected number of times the configuration *x* is sampled when one sums over all possible random walks generated with a given transition matrix.

Markov Chains

- A sequence of configurations $\{x_i\}$, generated in a random walk is termed a Markov chain if the transition matrix element, $T_{x,\, x+1}$, which moves the system from state x_i to x_{i+1}, depends only on the current state x_i of the system.
- Successive configurations in a Markov chain will be statistically correlated.
- Transition matrix elements must be positive.
- *Stochastic matrix* satisfies the condition $T_{xx'} \geq 0 \text{ and } \int T_{xx'} dx' = 1$

The transition matrix must be a stochastic matrix:

$$T_{\mathbf{x}\mathbf{x}'} \geq 0 \qquad \text{and} \qquad \int T_{\mathbf{x}\mathbf{x}'} d\mathbf{x}' = 1$$

The master equation

$$\frac{dP(\mathbf{x},t)}{dt} = \sum_{\mathbf{x}'} T_{\mathbf{x}'\mathbf{x}} P(\mathbf{x}',t) - \sum_{\mathbf{x}'} T_{\mathbf{x}\mathbf{x}'} P(\mathbf{x},t)$$

At equilibrium or in the stationary state,

$$\sum_{\mathbf{x}'} T_{\mathbf{x}'\mathbf{x}} P(\mathbf{x}',t) = \sum_{\mathbf{x}'} T_{\mathbf{x}\mathbf{x}'} P(\mathbf{x},t)$$

Using the conservation condition, we can then write

$$\sum_{\mathbf{x}'} T_{\mathbf{x}'\mathbf{x}} P(\mathbf{x}',t) = P(\mathbf{x},t)$$

Detailed balance:

$$T_{\mathbf{x}'\mathbf{x}} P(\mathbf{x}',t) = T_{\mathbf{x}\mathbf{x}'} P(\mathbf{x},t)$$

Metropolis Algorithm

The transition matrix elements are:

$$T_{\mathbf{x}\mathbf{x}'} = F_{\mathbf{x}\mathbf{x}'} A_{\mathbf{x}\mathbf{x}'}$$

- Proposal step: Proposal matrix with elements $F_{\mathbf{x}\mathbf{x}'}$ such that $\int F_{\mathbf{x}\mathbf{x}'} d\mathbf{x}' = 1$.
- Acceptance step: acceptance matrix with elements $A_{\mathbf{x}\mathbf{x}'}$.

The detailed balance condition:

$$F_{\mathbf{x}\mathbf{x}'} A_{\mathbf{x}\mathbf{x}'} P(\mathbf{x}) = F_{\mathbf{x}'\mathbf{x}} A_{\mathbf{x}'\mathbf{x}} P(\mathbf{x}')$$

Acceptance matrix must satisfy:

$$\frac{A_{\mathbf{x}\mathbf{x}'}}{A_{\mathbf{x}'\mathbf{x}}} = \frac{F_{\mathbf{x}'\mathbf{x}} P(\mathbf{x}')}{F_{\mathbf{x}\mathbf{x}'} P(\mathbf{x})} = q_{\mathbf{x}\mathbf{x}'}.$$

The original form of $A_{\mathbf{x}\mathbf{x}'}$ proposed by Metropolis et al was:

$$A_{\mathbf{x}\mathbf{x}'} = \min(1, q_{\mathbf{x}\mathbf{x}'})$$

Clearly if $q_{\mathbf{x}\mathbf{x}'} > 1$, then $A_{\mathbf{x}\mathbf{x}'} = 1$. Moreover, in this case, the acceptance probability for the reverse move will be:

$$A_{\mathbf{x}'\mathbf{x}} = \min(1, q_{\mathbf{x}'\mathbf{x}}) = \min(1, 1/q_{\mathbf{x}\mathbf{x}'}) = 1/q_{\mathbf{x}\mathbf{x}'}.$$

The usual choice for the transition matrix is:

$$\begin{aligned} F_{\mathbf{x}\mathbf{x}'} &= (1/\delta) \qquad \text{if} \qquad |\mathbf{x} - \mathbf{x}'| \leq \delta/2 \\ &= 0 \qquad \text{otherwise} \end{aligned} \tag{1}$$

Acceptance criterion:

$$\frac{A_{\mathbf{x}\mathbf{x}'}}{A_{\mathbf{x}'\mathbf{x}}} = \frac{(1/\delta)P(\mathbf{x}')}{(1/\delta)P(\mathbf{x})} = \frac{P(\mathbf{x}')}{P(\mathbf{x})}$$

Metropolis Algorithm

- Start with a configuration $\mathbf{r}^{(n)}$, denoted by $\mathbf{r}_{old}$, of the random walk and the corresponding potential energy, $V_{old} = V(\mathbf{r}^n)$.

- Generate a configuration randomly in the neighbourhood of this $\mathbf{r}_{old}$ using

$$\mathbf{r}_{new} = \mathbf{r}_{old} + \Delta \tag{48}$$

  The random displacement vector Δ may be chosen in a number of ways and will be discussed below. Compute the potential energy $V_{new} = V(\mathbf{r}_{new})$.

- Compute the ratio of the Boltzmann

$$w = \exp(-\beta V_{new})/\exp(-\beta V_{old}) = \exp(\beta(V_{old} - V_{new})) \tag{49}$$

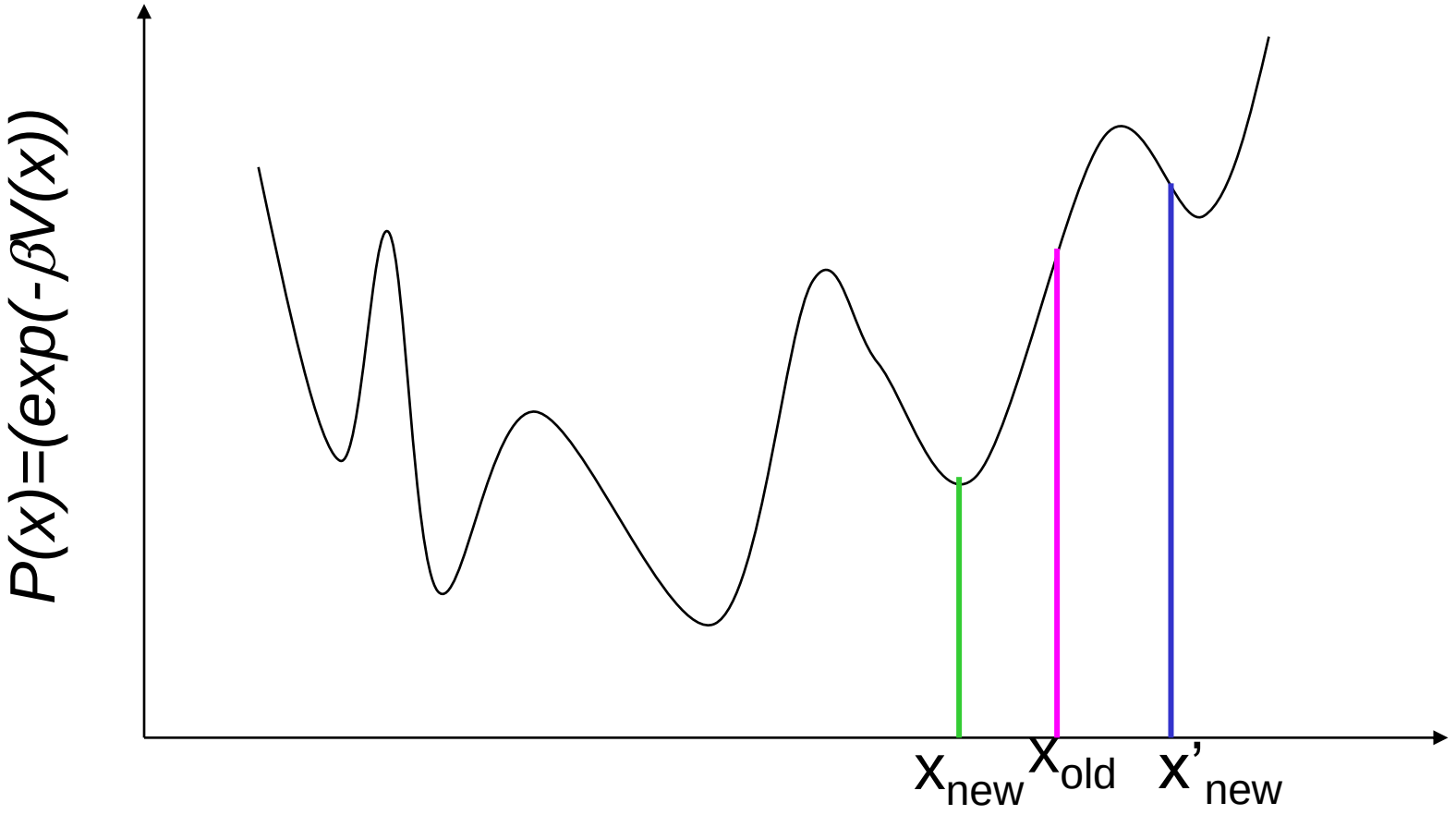
  (a) If $w > 1$, then accept the new configuration. This implies that the $(n+1)$-th configuration in the random walk is $\mathbf{r}^{(n+1)} = \mathbf{r}_{new}$.

  (b) If $w \leq 1$, then generate a random number ξ uniformly.
  (i) If $\xi < w$, then accept $\mathbf{r}_{new}$. The $(n+1)$-th configuration in the random walk will then be $\mathbf{r}_{new}$ i.e. $\mathbf{r}^{(n+1)} = \mathbf{r}_{new}$.
  (ii) If $\xi > w$, then reject $\mathbf{r}_{new}$. The $(n+1)$-th configuration in the random walk will then be a repeat of $\mathbf{r}_{old}$ i.e. $\mathbf{r}^{(n+1)} = \mathbf{r}_{old}$.

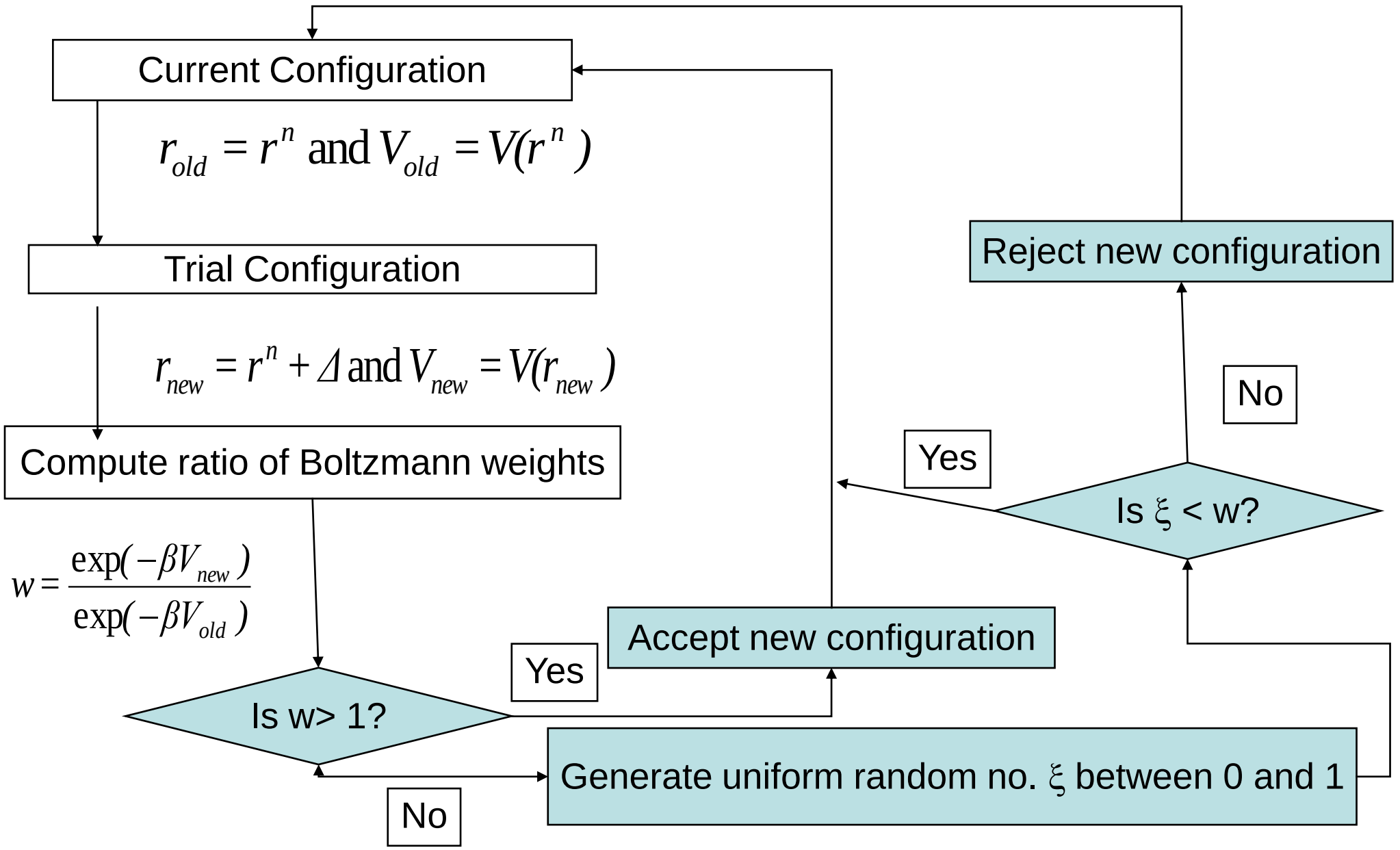
- Monitor the ratio of the number of acceptances to the number of trial moves. This ratio should be maintained at approximately 50% by changing Δ.

Classical Monte Carlo

Generating a set of configurations, $\{x_i, i=1,N\}$ distributed according to their Boltzmann weights, $P(x_i)=(exp(-\beta V(x_i))$



Generating the Boltzmann Distribution



Molecular Dynamics vs Monte Carlo

Monte Carlo:
-Microstates generated by stochastic sampling involving random numbers
- Equilibrium properties only (no time dependence, no dynamics)
- Stochastic sampling: solves mathematical equations
-randomly selects values to fit a probability distribution to create scenarios of a problem.

Molecular dynamics:
- Microstates generated by the positions and momenta of the particles
- Solves Newton's equations of motion
- Microstates generated by integration over time
- Time evolution, dynamic behaviour of the system
- Gives both equilibrium and dynamic properties.