

# Monte Carlo Basics

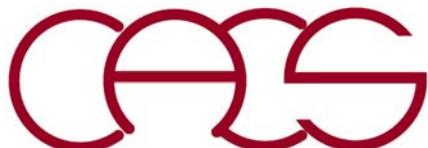
---

Aiichiro Nakano

*Collaboratory for Advanced Computing & Simulations  
Department of Computer Science  
Department of Physics & Astronomy  
Department of Chemical Engineering & Materials Science  
Department Quantitative & Computational Biology  
University of Southern California*

Email: [anakano@usc.edu](mailto:anakano@usc.edu)

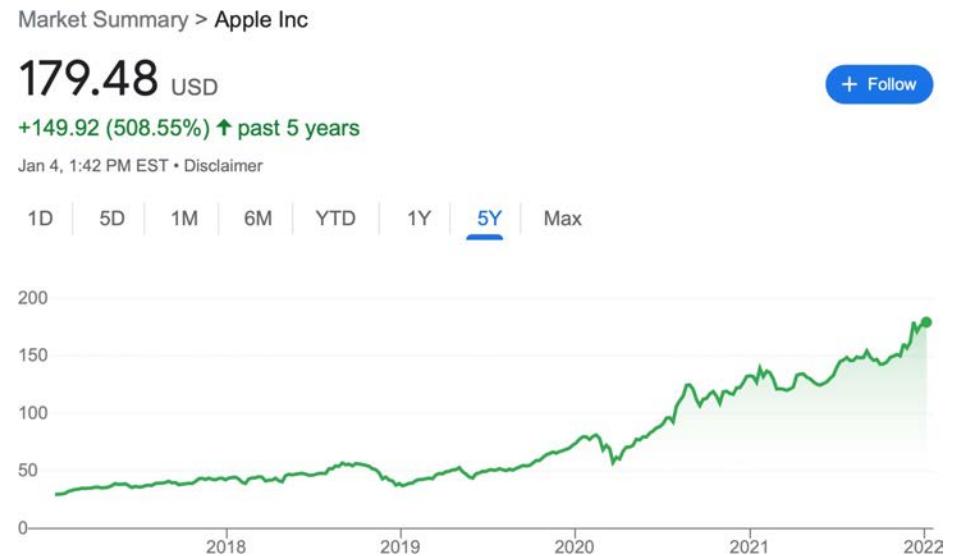
Goals: MC simulation in practice (visually understand & use)  
Minimal math = probability  
Markov-chain (Metropolis) MC



# Monte Carlo Method

- Monte Carlo (MC) method: A computational method that utilizes random numbers
- Major applications of the MC method:
  1. Multidimensional integrations (e.g., statistical mechanics in physics)
  2. Simulation of stochastic natural phenomena (e.g., stock price)

$$\langle A(\{\vec{r}_i\}) \rangle = \frac{\int d\vec{r}_1 \cdots \int d\vec{r}_N \exp(-\frac{V(\{\vec{r}_i\})}{k_B T}) A(\{\vec{r}_i\})}{\int d\vec{r}_1 \cdots \int d\vec{r}_N \exp(-\frac{V(\{\vec{r}_i\})}{k_B T})}$$



# Monte Carlo Integration

- Numerical integration

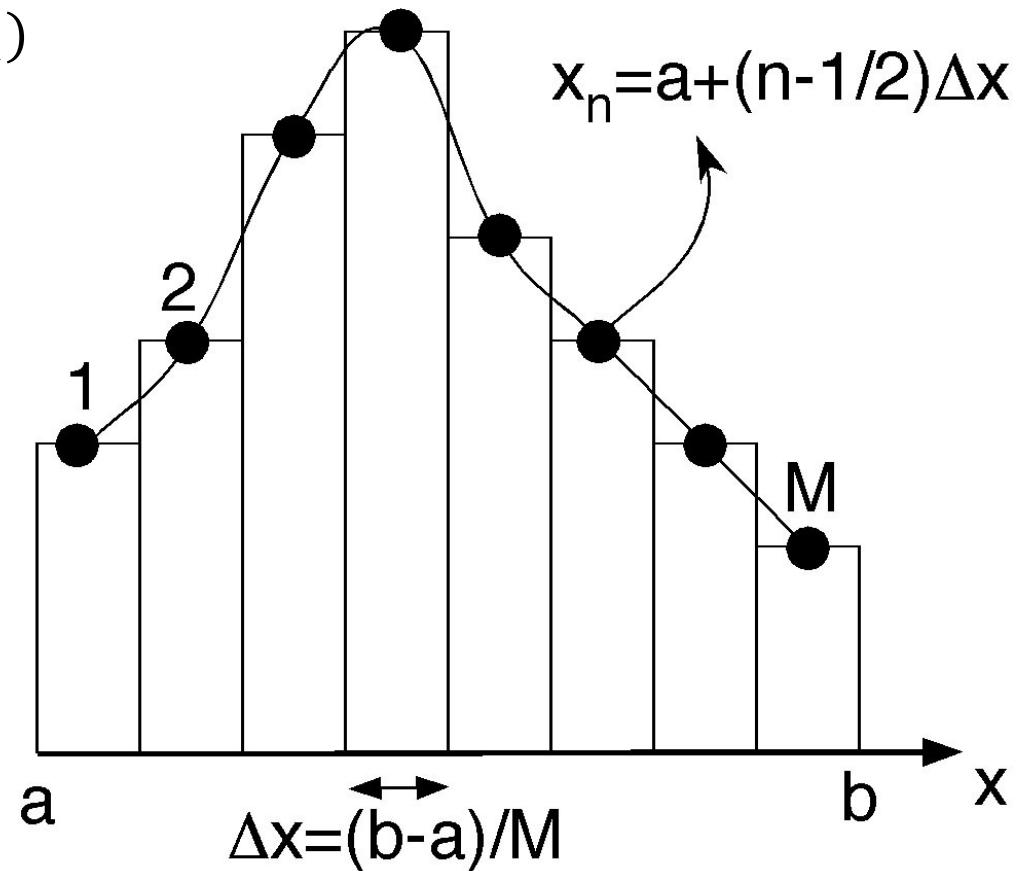
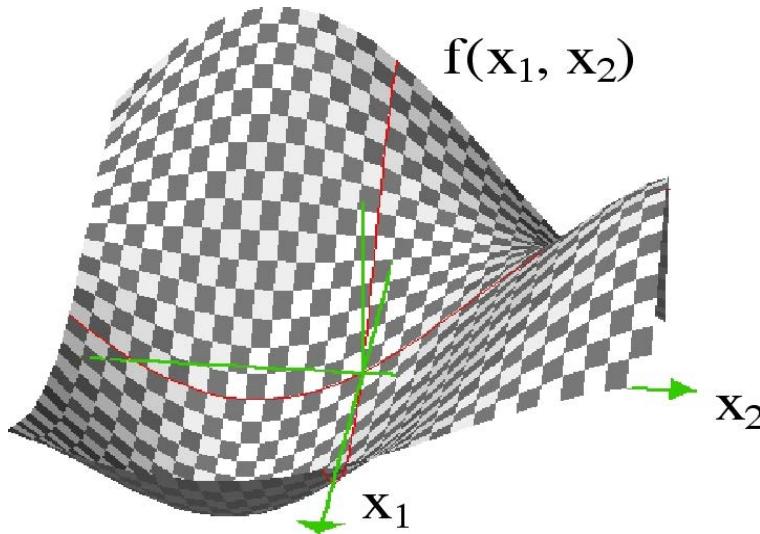
$$\int_a^b dx f(x) = \sum_{n=1}^M \Delta x f(x_n) = \sum_{n=1}^M \frac{b-a}{M} f(x_n)$$

- Monte Carlo integration ( $r_n \in [a,b]$ : random #): Error  $\sim 1/M^{1/2}$

$$\int_a^b dx f(x) \cong \sum_{n=1}^M \frac{b-a}{M} f(r_n)$$

- Multidimensional integration:

$$O(C^N) \int_0^1 dx_1 \cdots \int_0^1 dx_N f(x_1, \dots, x_N)$$



# Stochastic Model of Stock Prices

## Fluctuation in stock price

Market Summary > Apple Inc

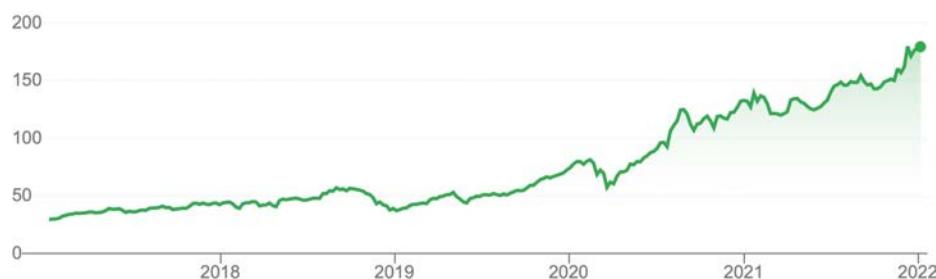
179.48 USD

+149.92 (508.55%) ↑ past 5 years

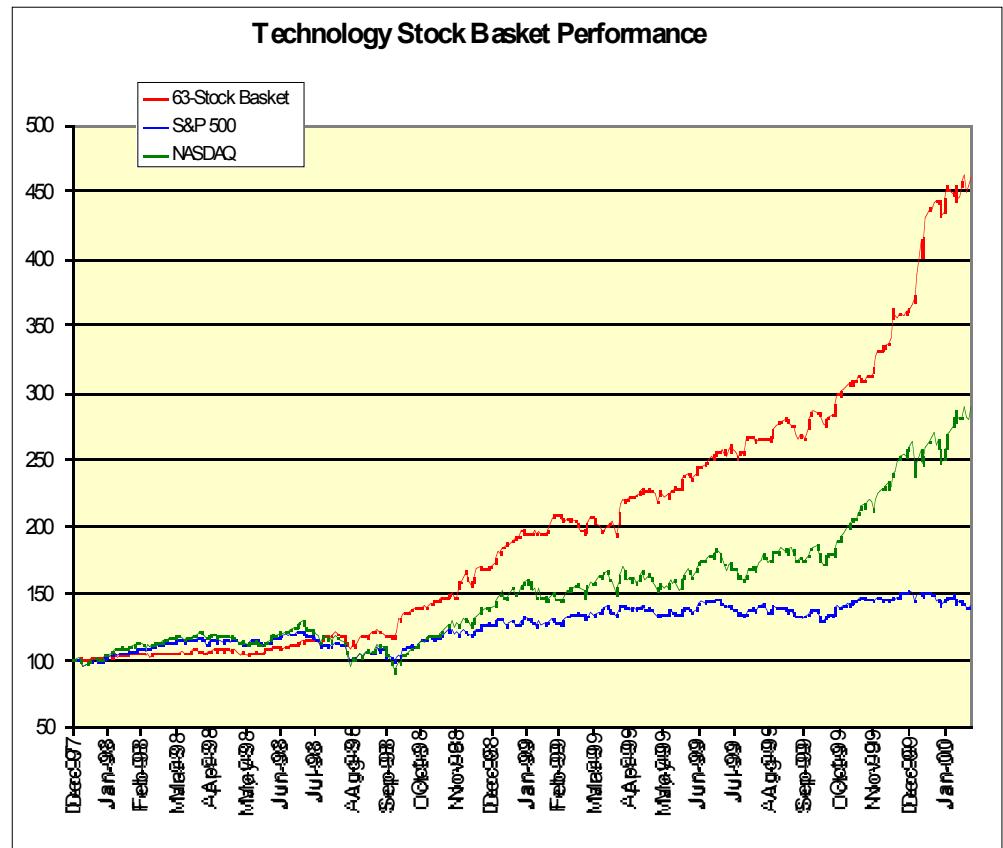
Jan 4, 1:42 PM EST • Disclaimer

+ Follow

1D | 5D | 1M | 6M | YTD | 1Y | 5Y | Max



## Computational stock portfolio trading



## Basis of Black-Scholes analysis of option prices



$$dS = \mu S dt + \sigma S \varepsilon \sqrt{dt}$$

(1997 Nobel Economy Prize to Myron Scholes)

Andrey Omelchenko (Quantlab)

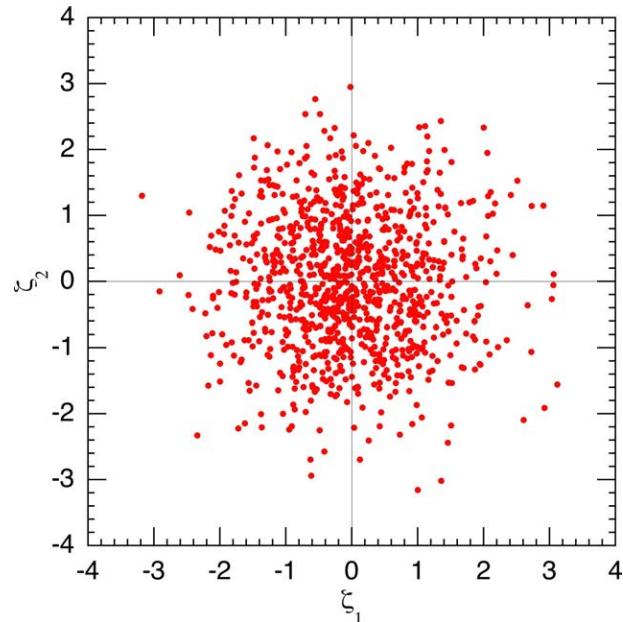
# Goals: Monte Carlo Basics

## 1. Sample-mean MC

$$\int dx f(x) p(x) \cong \bar{f} \pm \sqrt{\frac{\bar{f}^2 - (\bar{f})^2}{M-1}} \quad \left( \bar{f} = \frac{1}{M} \sum_{n=1}^M f(r_n) \right)$$

## 2. Visualize probability density by a set of points: nonuniform random-number generation by coordinate transformation

$$P'(\zeta_1, \dots, \zeta_N) = \frac{P(r_1, \dots, r_N)}{|\partial(\zeta_1, \dots, \zeta_N)/\partial(r_1, \dots, r_N)|}$$



## 3. Metropolis (Markov-chain) MC algorithm for importance sampling Random attempt → conditional acceptance with probability $\exp(-\Delta V/k_B T)$

# Monte Carlo Integration: Hit-&-Miss

Visualize probability density function

Program hit.c: Calculate  $\pi$

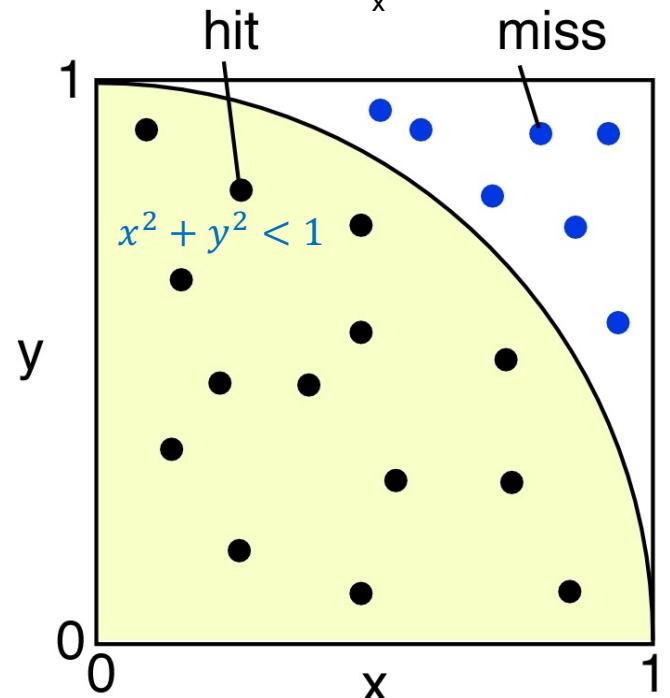
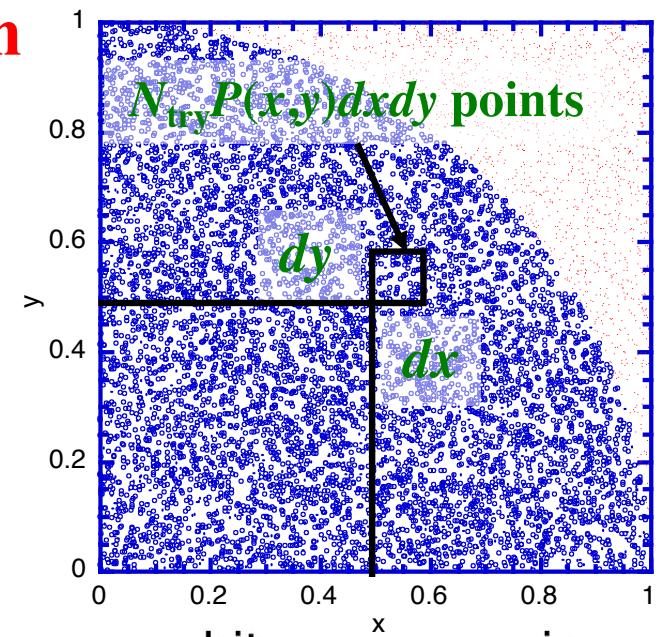
<https://aiichironakano.github.io/phys516/src/MCbasics/hit.c>

```
#include <stdio.h>
#include <stdlib.h>
main() {
    double x, y, pi;
    int hit = 0, try, ntry;
    printf("Input the number of MC trials\n");
    scanf("%d", &ntry);
    srand((unsigned)time((long *)0));
    for (try=0; try<ntry; try++) {
        x = rand()/(double)RAND_MAX;
        y = rand()/(double)RAND_MAX;
        if (x*x+y*y < 1.0) ++hit;
    }
    pi = 4.0*hit/ntry;
    printf("MC estimate for PI = %f\n", pi);
} https://aiichironakano.github.io/phys516/01-1MCbasics.pdf, p.3
```

discovery.usc.edu> more /usr/include/stdlib.h

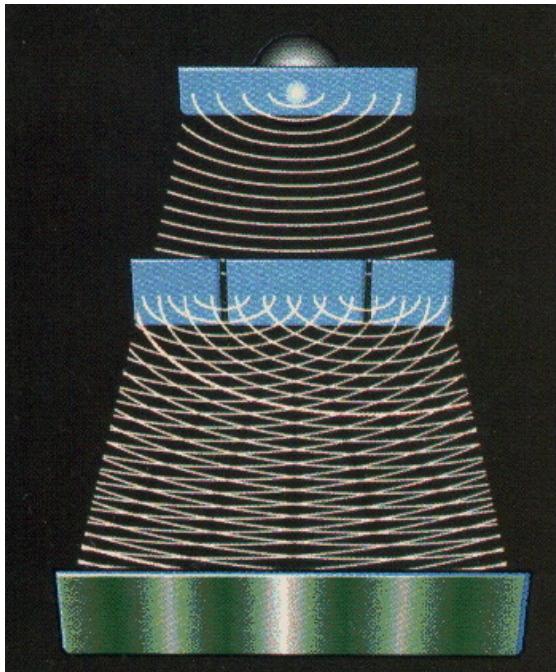
...

```
#define RAND_MAX          2147483647 = 231-1
rand( ) ∈ [1,RAND_MAX-1]
```



# Single-Electron Double-Slit Experiment

Visualize probability density as point crowd



<http://rdg.ext.hitachi.co.jp/rd/moviee/doubleslite-n.mpeg>

**Akira Tonomura (Hitachi, Ltd.)**

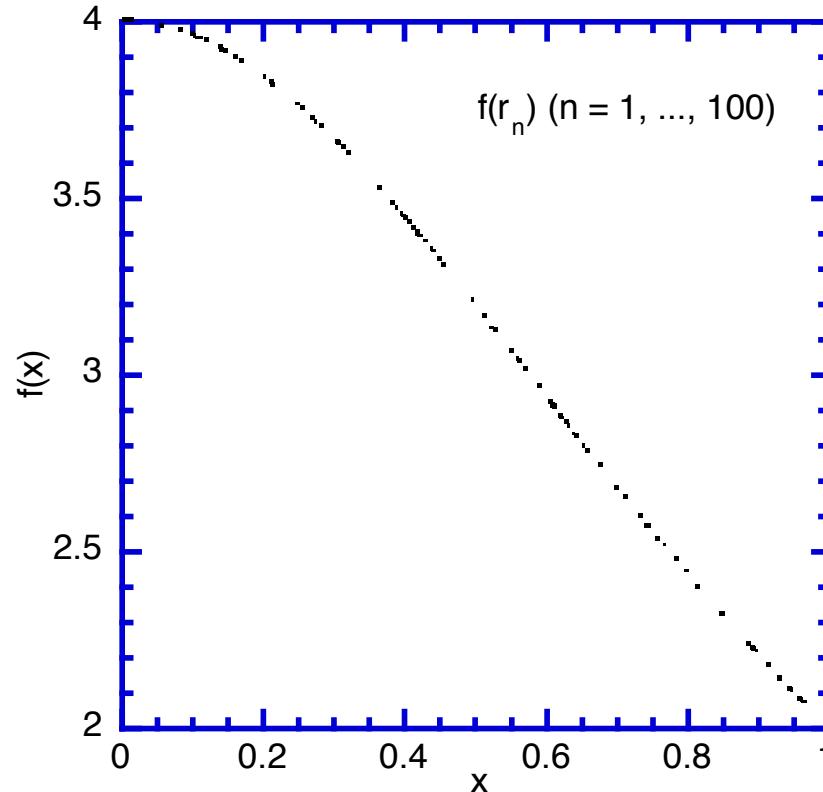
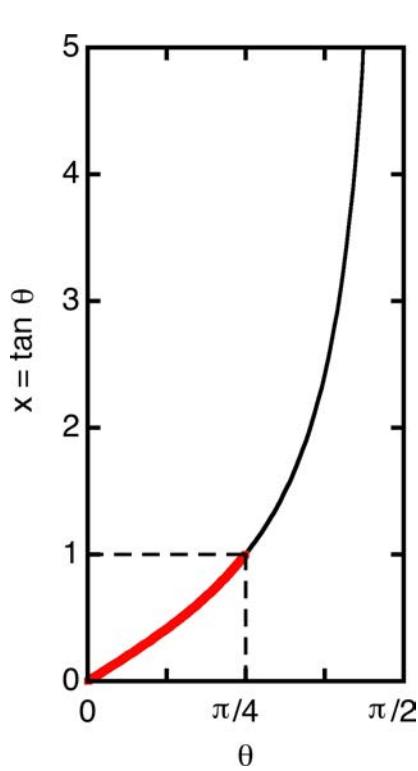
# Monte Carlo Integration: Sample Mean

**Bottom line: MC simulation as integration**

$$\int_a^b dx f(x) \cong \sum_{n=1}^M \frac{b-a}{M} f(r_n) = (b-a) \left[ \frac{1}{M} \sum_{n=1}^M f(r_n) \right] = (b-a) \langle f(r_n) \rangle$$

**average**

$$\left\langle \frac{4}{1+x^2} \right\rangle = \int_0^1 dx \frac{4}{1+x^2} = \int_0^{\pi/4} \frac{d\theta}{\cos^2 \theta} \frac{4}{1+\tan^2 \theta} = \int_0^{\pi/4} 4 d\theta = \pi$$



# Sample Mean MC Program

## Program mean.c: Calculate $\pi$

<https://aiichironakano.github.io/phys516/src/MCbasics/mean.c>

```
#include <stdio.h>
#include <stdlib.h>
main() {
    double x, pi, sum = 0.0;
    int try, ntry;
    printf("Input the number of MC trials\n");
    scanf("%d", &ntry);
    srand((unsigned)time((long *)0));
    for (try=0; try<ntry; try++) {
        x = rand()/(double)RAND_MAX;
        sum += 4.0/(1.0 + x*x); sum = sum + 4.0/(1.0 + x*x);
    }
    pi = sum/ntry;
    printf("MC estimate for PI = %f\n", pi);
}
```

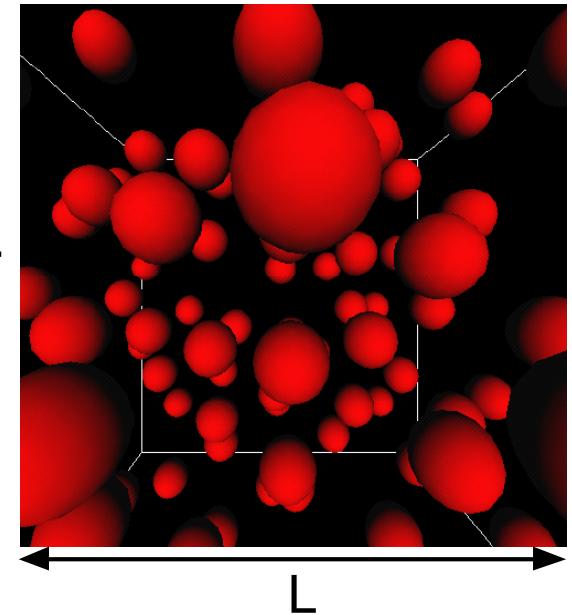
<https://aiichironakano.github.io/phys516/01-1MCbasics.pdf>, p.4

# Multidimensional Integration

## Statistical mechanics

$$\langle A(\{\vec{r}_i\}) \rangle = \frac{\int d\vec{r}_1 \cdots \int d\vec{r}_N \exp(-\frac{V(\{\vec{r}_i\})}{k_B T}) A(\{\vec{r}_i\})}{\int d\vec{r}_1 \cdots \int d\vec{r}_N \exp(-\frac{V(\{\vec{r}_i\})}{k_B T})}$$

$k_B = 1.38 \times 10^{-16}$  erg/K: Boltzmann constant



Need Monte Carlo!

# Probability: Foundation of MC

---

Minimal probability theory enough to understand MC simulation

- **Random variable:** A variable that takes different values at different measurements.
- **Probability distribution function**

$$p(x) = (\# \text{ of samples with result } x) / (\text{Total } \# \text{ of samples})$$

- **Normalization**

$$\sum_k p(x_k) = 1$$

- **Expectation (weighted sum)**

$$E[x] = \langle x \rangle = \sum_k x_k p(x_k)$$

- **Variance (spread)**

$$Var[x] = \langle (x - \langle x \rangle)^2 \rangle = \langle x^2 - 2x\langle x \rangle + \langle x \rangle^2 \rangle = \langle x^2 \rangle - \langle x \rangle^2$$

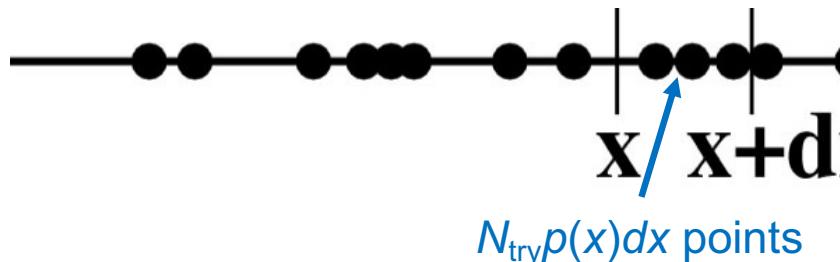
- **Standard deviation**

$$Std[x] = \sqrt{Var[x]}$$

# Continuous Random Variables

- Probability “density” function (note the unit)

$p(x)dx = (\# \text{ of samples in the range } [x, x+dx]) / (\text{Total } \# \text{ of samples})$

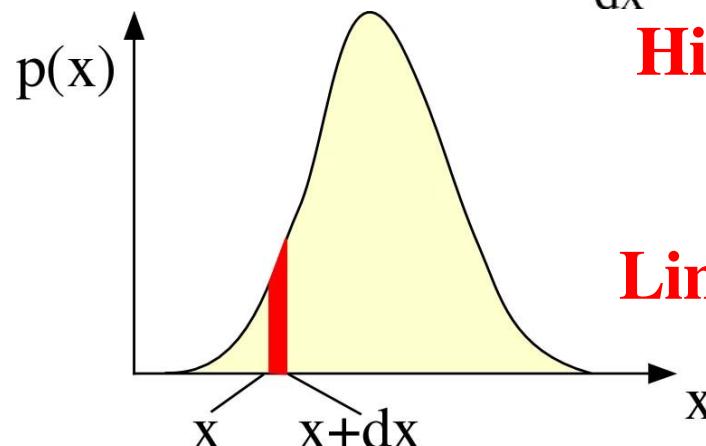
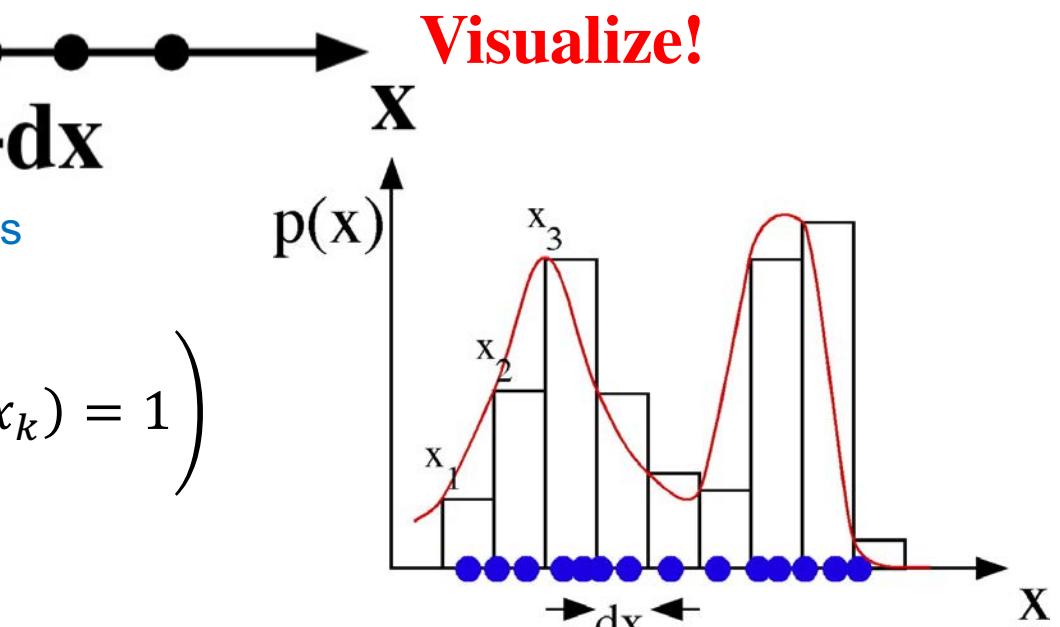


- Normalization

$$\int_{-\infty}^{\infty} dx p(x) = 1 \quad \left( \lim_{\Delta x \rightarrow 0} \sum_k \Delta x p(x_k) = 1 \right)$$

- Expectation

$$E[x] = \langle x \rangle = \int_{-\infty}^{\infty} dx x p(x)$$



# MC Estimation

- Distinguish infinite vs. sample ( $\sim$  poll) means

$$\langle f(x) \rangle = \int dx f(x) p(x) \quad \text{vs.} \quad \overline{f(x)} = \frac{1}{M} \sum_{n=1}^M f(r_n)$$

takes random values random variable

- **Unbiased estimator:** A statistical quantity, using  $M$  samples, which converges to the underlying population value for large  $M$
- The mean of  $M$  sampled random variables,  $\overline{f(x)}$ , converges to its expected value  $\langle f(x) \rangle$ :  $\overline{f(x)}$  is an unbiased estimator of  $\langle f(x) \rangle$
- The error of an MC estimation decreases as  $1/M^{1/2}$

## Central Theorem of MC Sampling

Bottom line

$$\int dx f(x) p(x) \cong \bar{f} \pm \sqrt{\frac{\overline{f^2} - (\bar{f})^2}{M-1}}$$

Exact

“Finite” sampling

# Proof of the Central Theorem

- Unbiased estimator of the mean

$$E\{\bar{f(x)}\} = \frac{1}{M} \sum_{n=1}^M E\{f(r_n)\} = \frac{1}{M} \sum_{n=1}^M \langle f(x) \rangle = \frac{1}{M} M \langle f(x) \rangle = \langle f(x) \rangle$$

- Error estimate—variance of the sample mean

$$\text{Var}\{\bar{f(x)}\} = \text{Var}\left\{\frac{1}{M} \sum_{n=1}^M f(r_n)\right\} = \frac{1}{M^2} \sum_{m=1}^M \sum_{n=1}^M \langle [f(r_m) - \langle f(x) \rangle][f(r_n) - \langle f(x) \rangle] \rangle$$

For uncorrelated random variables,  $\langle AB \rangle = \langle A \rangle \langle B \rangle$ , and thus

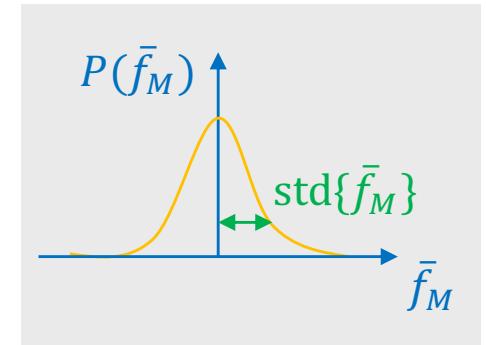
$$\langle [f(r_m) - \langle f(x) \rangle][f(r_n) - \langle f(x) \rangle] \rangle = \begin{cases} \text{Var}\{f(x)\} & (m = n) \\ \langle f(r_m) - \langle f(x) \rangle \rangle \langle f(r_n) - \langle f(x) \rangle \rangle = 0 & (\text{else}) \end{cases}$$

$$\therefore \text{Var}\{\bar{f(x)}\} = \frac{1}{M^2} \sum_{m=1}^M \text{Var}\{f(x)\} = \frac{\text{Var}\{f(x)\}}{M}$$

or  $\text{Std}\{\bar{f(x)}\} = \frac{\text{Std}\{f(x)\}}{\sqrt{M}}$

Spread of underlying  
mother population

Error bar of poll



Error decreases as  $1/M^{1/2}$ , but what is the prefactor  $\sim \text{Var}\{f(x)\}$ ?

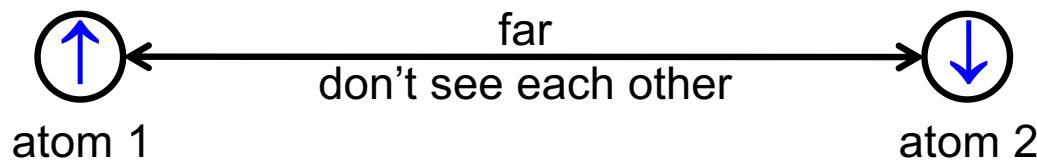
# An Example of Correlated Random Variables

- A spin variable,  $s$ , takes the value of +1 (spin up,  $\uparrow$ ) with probability 0.5, & -1 (spin down,  $\downarrow$ ) with probability 0.5

$$\langle s \rangle = 1 \times \frac{1}{2} + (-1) \times \frac{1}{2} = 0$$

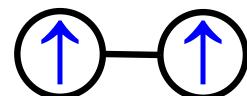
(Uncorrelated spins)

$$\langle s_1 s_2 \rangle = \langle s_1 \rangle \langle s_2 \rangle = 0$$



(Correlated spins) Ferromagnetic spins: they always align with each other  
(spin configurations  $\uparrow\uparrow$  with probability 0.5 and  $\downarrow\downarrow$  with probability 0.5)

$$\langle s_1 s_2 \rangle = 1^2 \times \frac{1}{2} + (-1)^2 \times \frac{1}{2} = 1 \neq \langle s_1 \rangle \langle s_2 \rangle$$



# Estimate of the Variance of Sample Mean

- Unbiased estimator of the variance

$$s_M = \frac{1}{M} \sum_{n=1}^M f^2(r_n) - \left[ \frac{1}{M} \sum_{n=1}^M f(r_n) \right]^2$$

Experimentally measurable  
by finite sampling

$$\begin{aligned}\langle s_M \rangle &= \frac{1}{M} \sum_{n=1}^M \langle f^2(x) \rangle - \frac{1}{M^2} \sum_{m=1}^M \sum_{n=1}^M \langle f(r_m) f(r_n) \rangle \\ &= \langle f^2(x) \rangle - \frac{1}{M^2} \sum_{m=1}^M \langle f^2(x) \rangle - \frac{1}{M^2} \sum_{m=1}^M \sum_{n=1, n \neq m}^M \langle f(r_m) f(r_n) \rangle.\end{aligned}$$

For uncorrelated random variables  $\langle f(r_m) f(r_n) \rangle = \langle f(x) \rangle \langle f(x) \rangle$

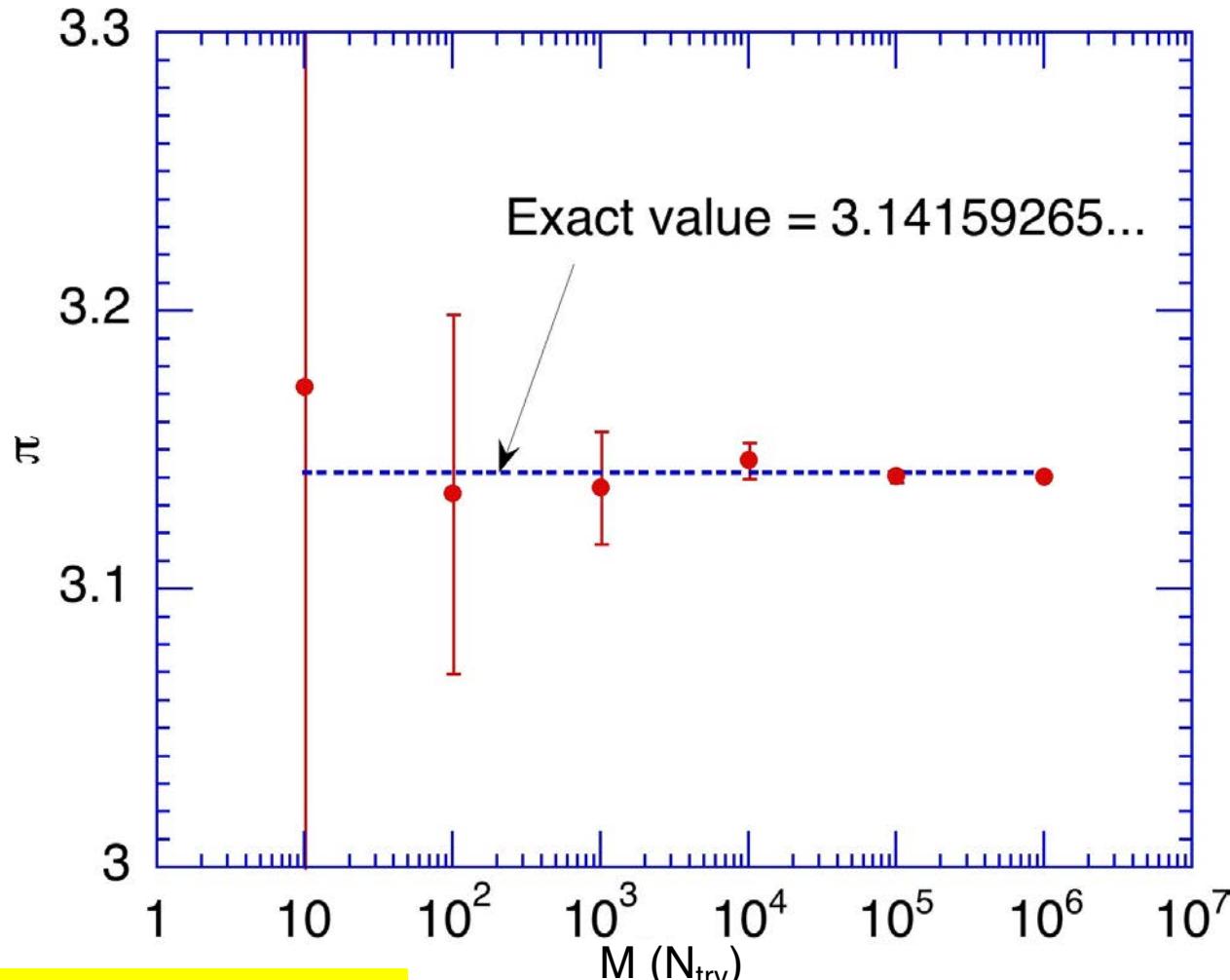
$$\begin{aligned}\langle s_M \rangle &= \left(1 - \frac{1}{M}\right) \langle f^2(x) \rangle - \frac{M(M-1)}{M^2} \langle f(x) \rangle \langle f(x) \rangle \\ &= \frac{M-1}{M} \text{Var}\{f(x)\}\end{aligned}$$

$$\therefore \frac{\langle s_M \rangle}{M-1} \xrightarrow{\text{estimates}} \frac{\text{Var}\{f(x)\}}{M} \xrightarrow{\text{estimates}} \text{Var}\{\bar{f}(x)\}$$

$$\therefore \int dx f(x) p(x) \cong \bar{f} \pm \text{Std}\{\bar{f}\} \cong \bar{f} \pm \frac{\text{Std}\{f(x)\}}{\sqrt{M}} \cong \bar{f} \pm \sqrt{\frac{\bar{f}^2 - (\bar{f})^2}{M-1}}$$

# Running mean.c

MC estimation of  $\pi$ : unbiased estimators for the mean & its standard deviation

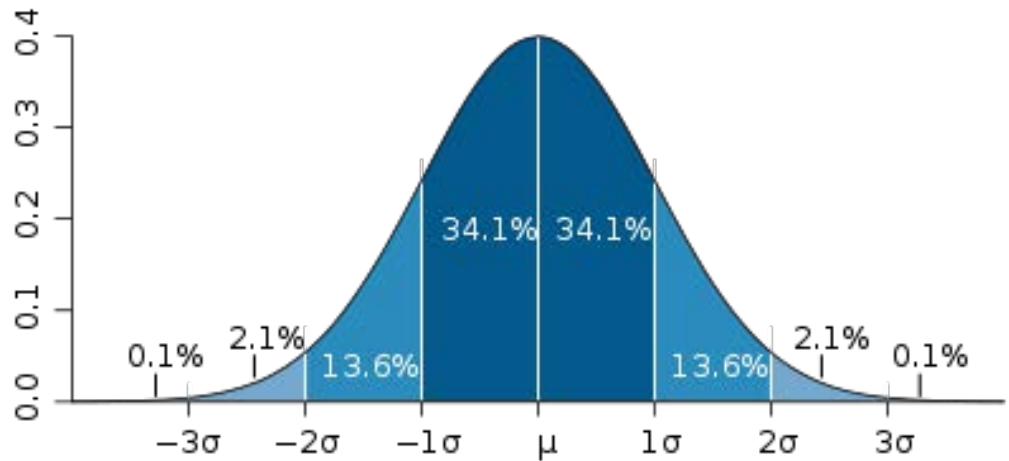
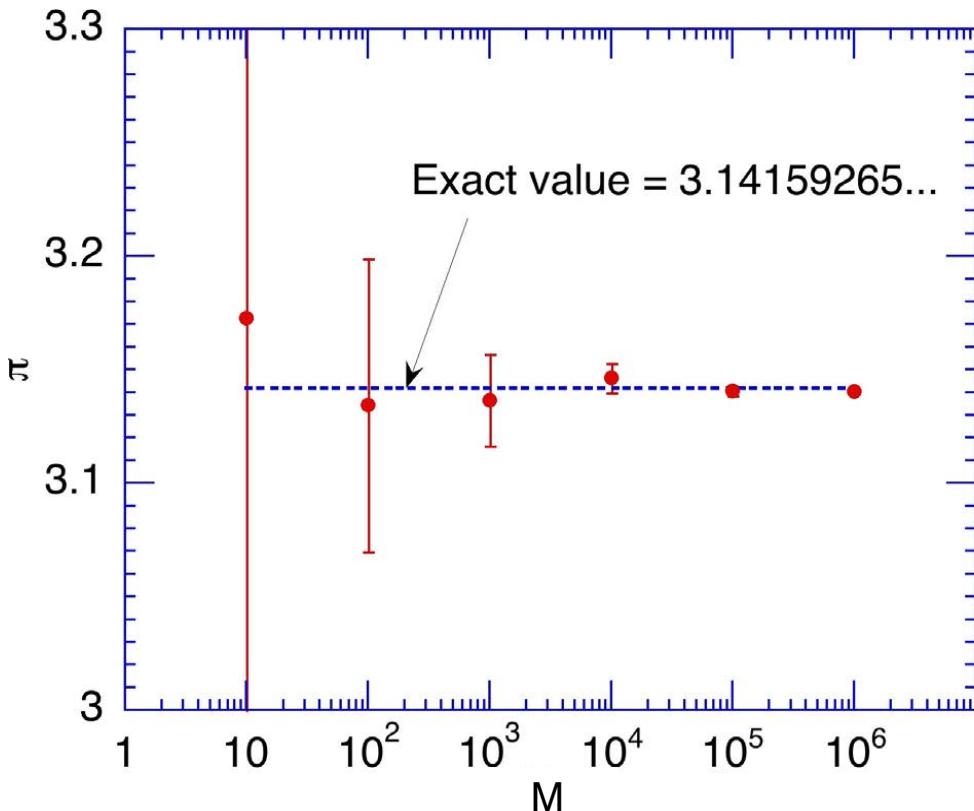


$$\int_0^1 dx f(x) \cong \bar{f} \pm \sqrt{\frac{\bar{f}^2 - (\bar{f})^2}{M-1}}$$

$$\bar{f} = \frac{1}{M} \sum_{n=1}^M f(r_n) \quad \bar{f}^2 = \frac{1}{M} \sum_{n=1}^M f^2(r_n) \quad f(r) = \frac{4}{1+r^2}$$

# Error Bar & the Exact Value

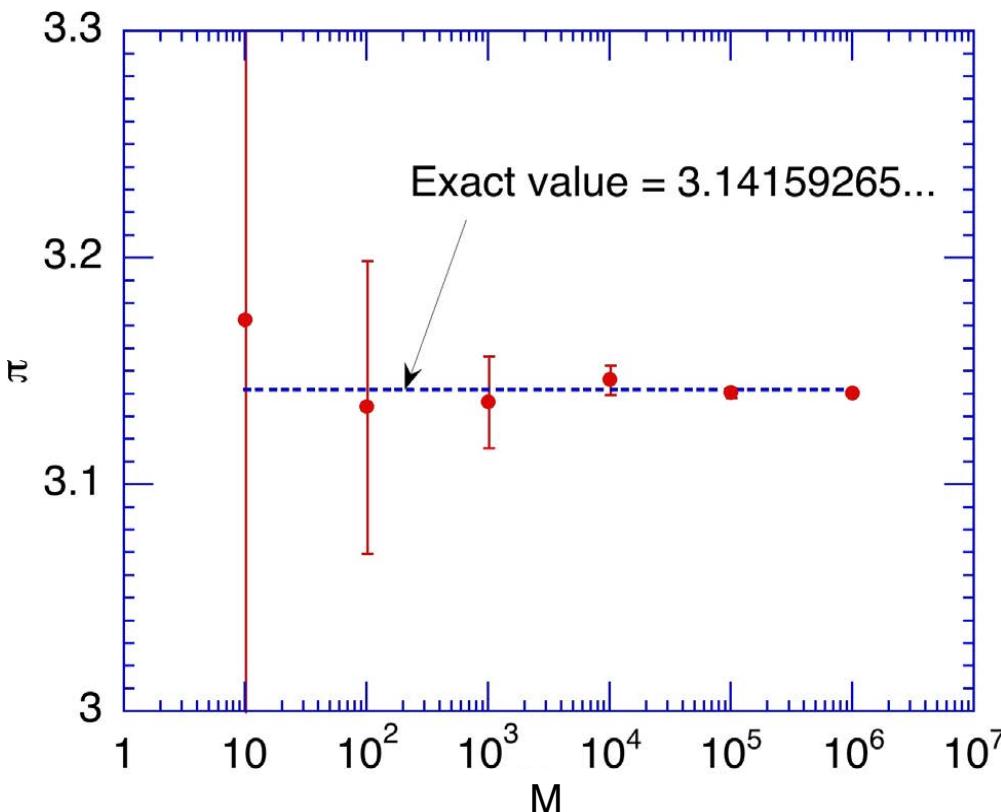
**Q:** Should the exact value always fall within the error bar?



**A:** No, only ~68% of the time in the case of the Gaussian distribution

# Standard Deviation of mean.c

MC estimation of  $\pi$ : unbiased estimators for the mean & its standard deviation



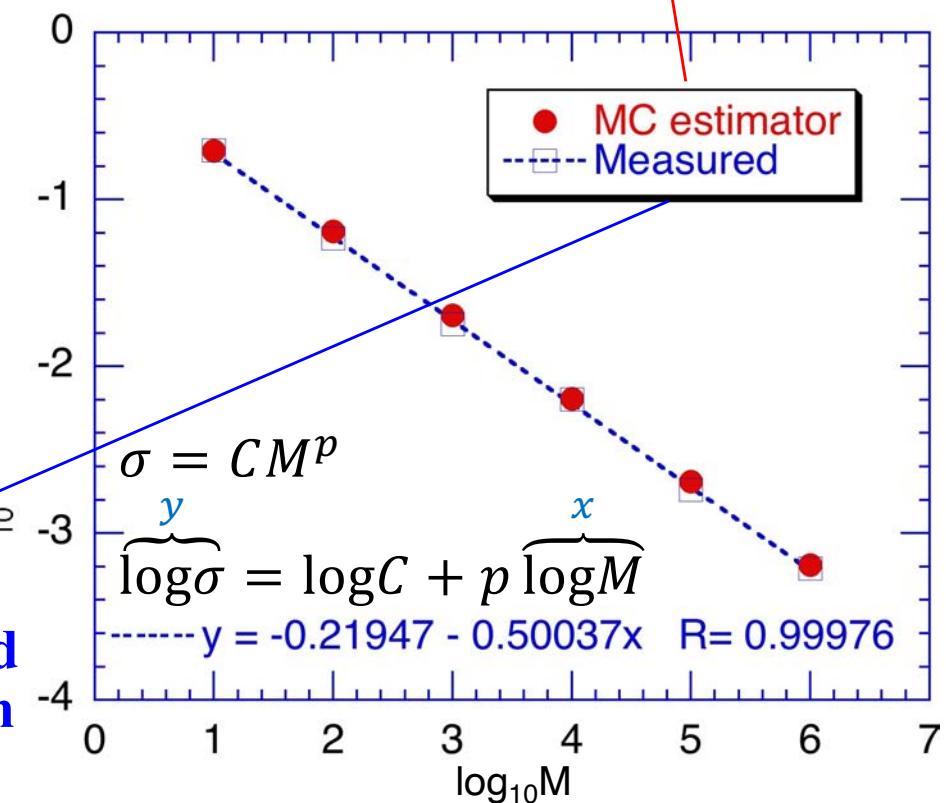
$$\sigma_M = \sqrt{\frac{1}{N_{\text{seed}}} \sum_{i=1}^{N_{\text{seed}}} \pi_i^2 - \left[ \frac{1}{N_{\text{seed}}} \sum_{i=1}^{N_{\text{seed}}} \pi_i \right]^2}$$

Unbiased estimates & measured values of the standard deviation  
See lecture on least square fit

$$\int_0^1 dx f(x) \cong \bar{f} \pm \sqrt{\frac{\bar{f}^2 - (\bar{f})^2}{M-1}}$$

$$\bar{f} = \frac{1}{M} \sum_{n=1}^M f(r_n) \quad \bar{f}^2 = \frac{1}{M} \sum_{n=1}^M f^2(r_n)$$

$$f(x) = 4 / (1 + x^2)$$



# Random Number Generation

- **Uniform random number generator:** A routine to generate a sequence of random numbers within a specified range (typically [0,1])
- **Linear congruential generator:** Generates a sequence  $X_i$  of positive integers based on recursion

$$X_{i+1} = aX_i \bmod M$$

in the range  $[1, M-1]$  & generates a random number in the range  $[0,1]$  as

$$r_i = X_i/M$$

(Example)  $M = 7$

Good:  $a = 3$

$$X_{i+1} = 3X_i \bmod 7$$

$i$	$X_i$
1	1
2	3
3	2
4	6
5	4
6	5
7	1

Period  $M-1 = 6$

Bad:  $a = 2$

$$X_{i+1} = 2X_i \bmod 7$$

$i$	$X_i$
1	1
2	2
3	4
4	1

Period  $3 < M-1$

# Linear Congruential Generator

- Most common generator: cycle  $\sim 2$  billion

$$(M, a) = (2^{31} - 1 = 2147483647, 7^5 = 16807)$$

- Overflow avoidance algorithm (using 32-bit integer arithmetic)

Let  $M = aq + r$ , i.e.,  $q = [M/a]$ ,  $r = M \bmod a$  then

$$az \bmod M = \begin{cases} a(z \bmod q) - r[z/q] & \text{if } \geq 0 \\ a(z \bmod q) - r[z/q] + M & \text{otherwise} \end{cases}$$

```
#define IA 16807
#define IM 2147483647
#define AM (1.0/IM)
#define IQ 127773
#define IR 2836
float ran0(int *idum) {
    long k;
    float ans;

    k = (*idum)/IQ;
    *idum = IA*(*idum - k*IQ) - IR*k;
    if (*idum < 0) *idum += IM;
    ans = AM*(*idum);
    return ans;
}
```

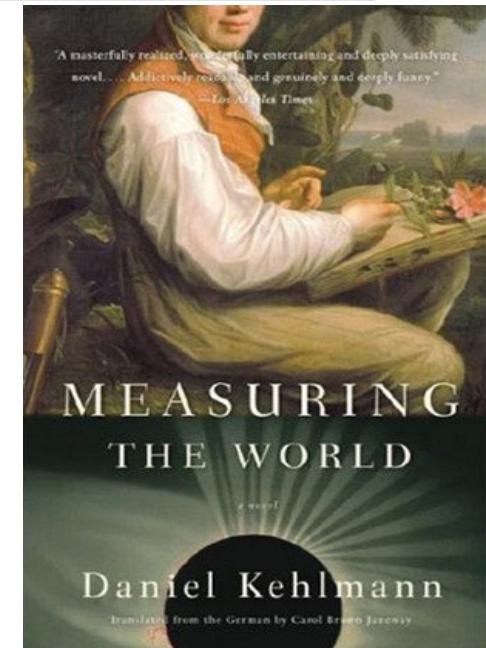
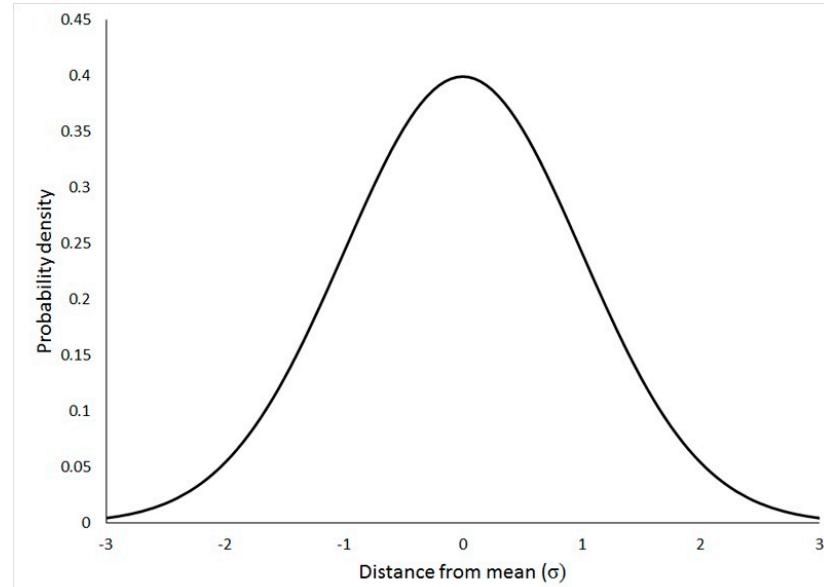
# Gaussian (Normal) Distribution

$$\rho(\xi) = \sqrt{\frac{1}{2\pi}} \exp\left(-\frac{\xi^2}{2}\right)$$

$$\int_{-\infty}^{\infty} d\xi \rho(\xi) = 1$$

$$Var\{\xi\} = \int_{-\infty}^{\infty} d\xi \xi^2 \rho(\xi) = 1$$

See Appendix A & problem in [MC Basics lecture](#)



Let's use coordinate transformation to generate nonuniform random numbers

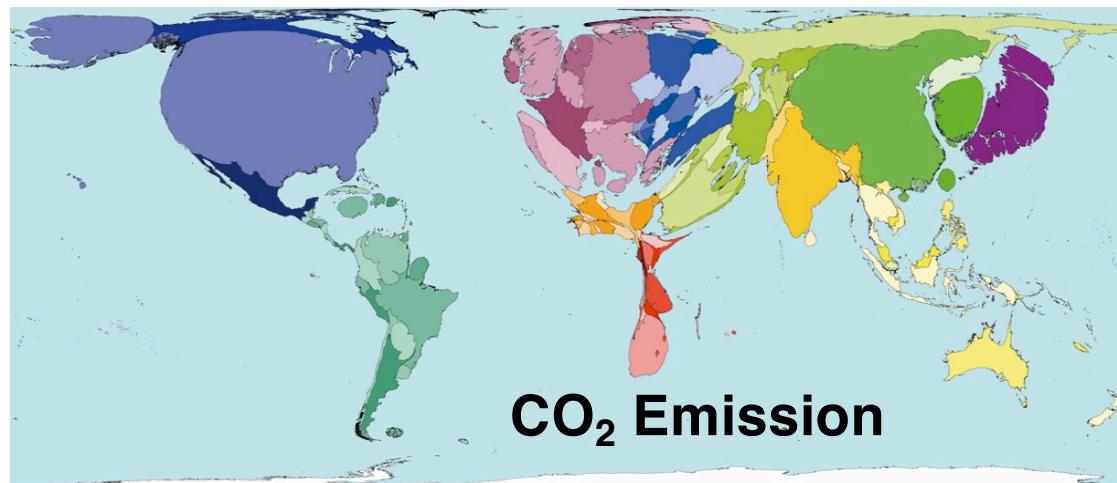
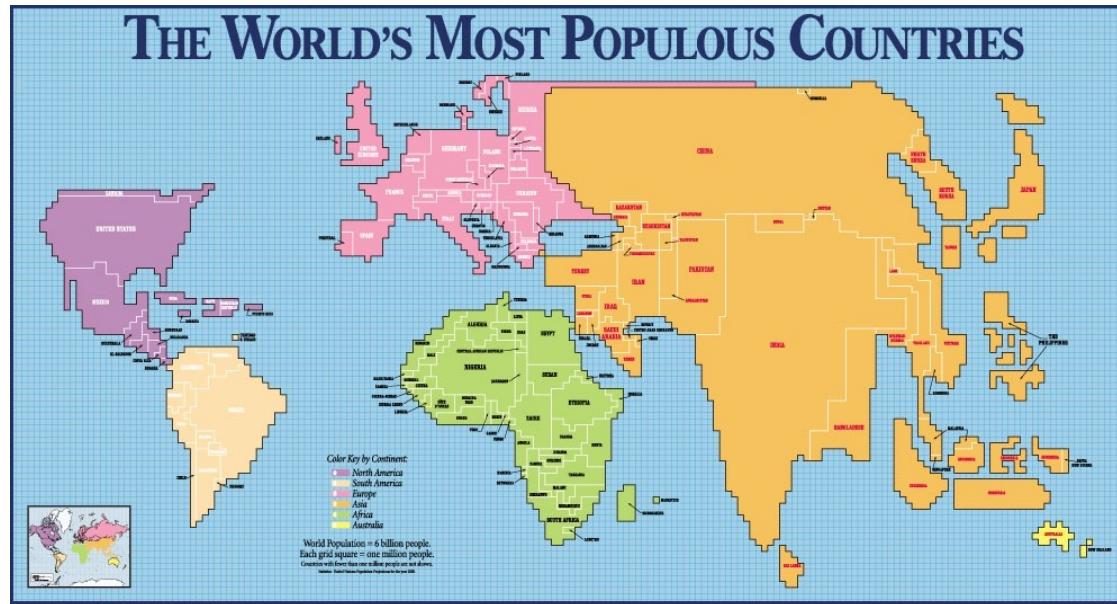
# Greenland Phenomenon

Spherical coordinate system (latitude,  $\theta$  & longitude,  $\phi$ ) is curved



# Transformation Method

Uniformly generate population in these distorted maps  
to generate nonuniform distributions

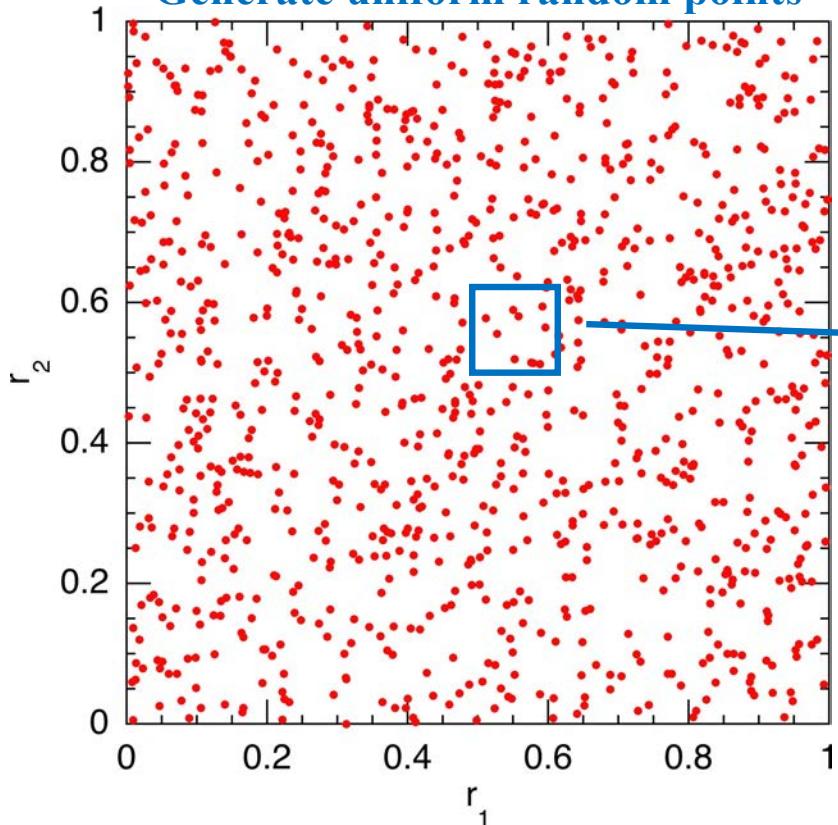


# Box-Muller Transformation

## Uniform

$$r_1, r_2 \in [0,1]$$

Generate uniform random points

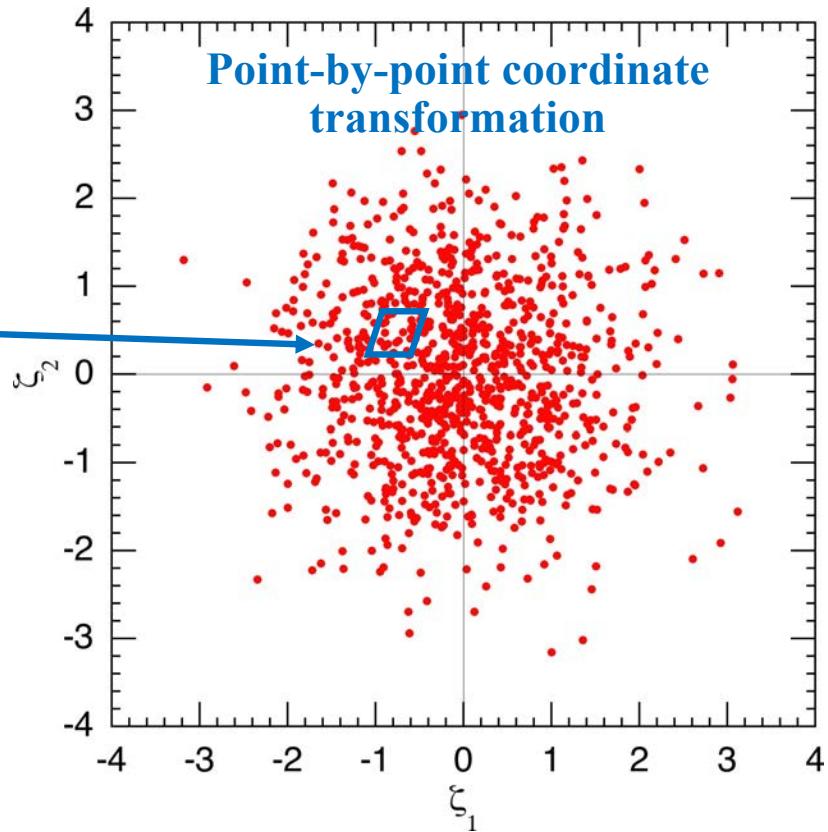


$$P(r_1, r_2) = 1$$

## Non-uniform

$$\begin{cases} \xi_1 = \sqrt{-2 \ln r_1} \cos(2\pi r_2) \\ \xi_2 = \sqrt{-2 \ln r_1} \sin(2\pi r_2) \end{cases}$$

Point-by-point coordinate transformation



$$P(\xi_{1/2}) = \sqrt{\frac{1}{2\pi}} \exp\left(-\frac{\xi_{1/2}^2}{2}\right)$$

# Box-Muller Algorithm

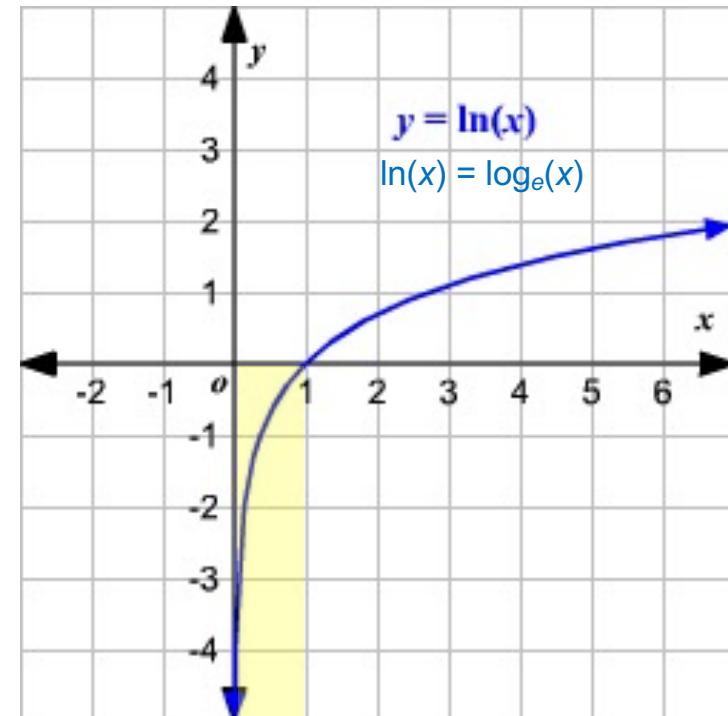
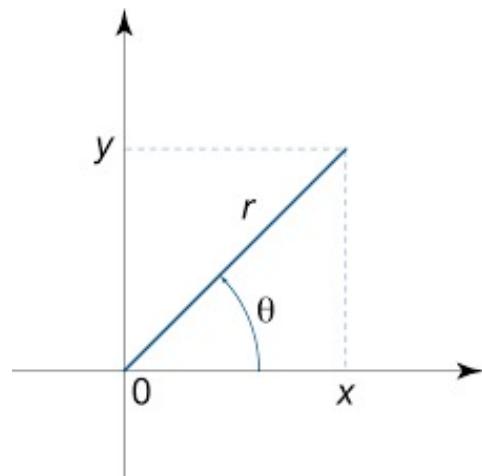
Uniform random number  $\in [0,1]$

```
r1 = rand()/(double)RAND_MAX; // cast  
r2 = rand()/(double)RAND_MAX;  
z = sqrt(-2*log(r1))*cos(2*M_PI*r2); // sin() works as well
```

Natural logarithm,  $\log_e()$        $\pi$  in C math library

cf. Polar coordinate

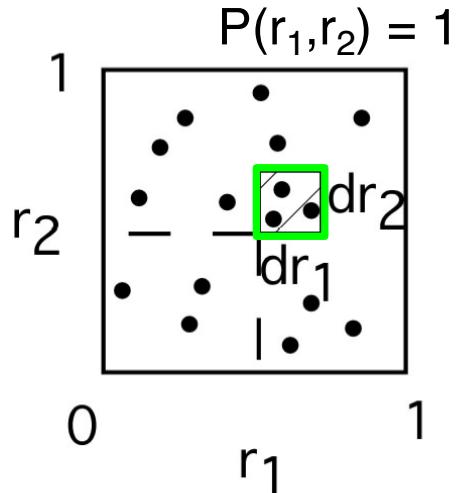
$$\begin{cases} r = \sqrt{-2\ln r_1} \in [0, \infty] \\ \theta = 2\pi r_2 \in [0, 2\pi] \end{cases}$$



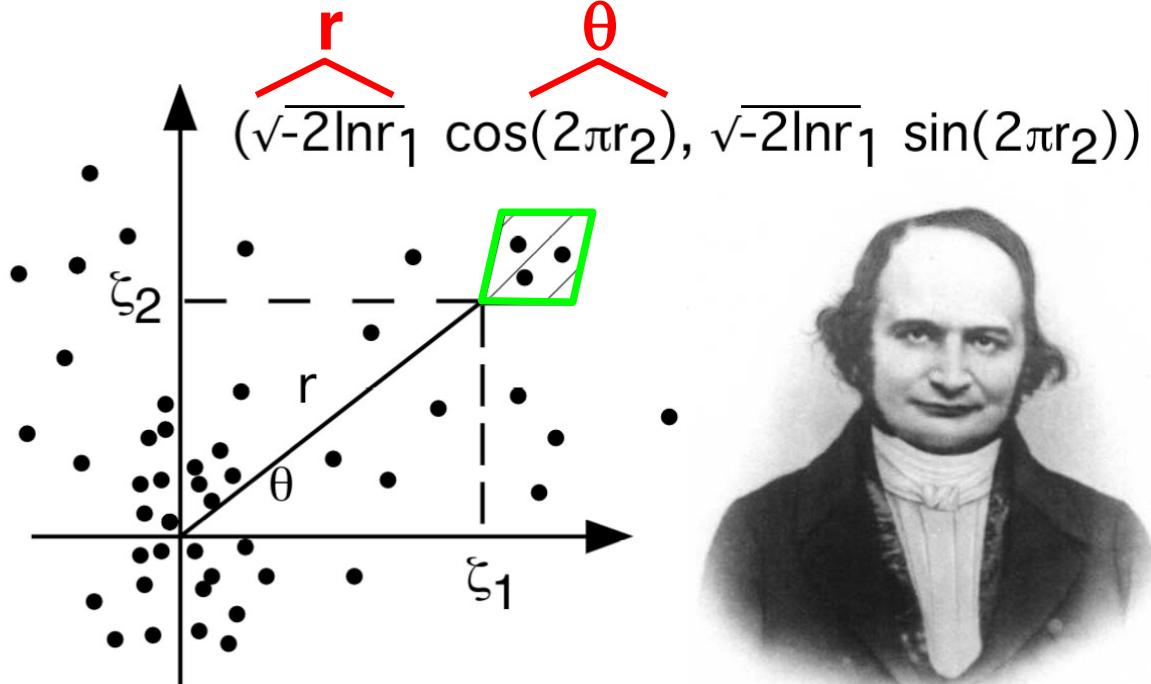
# Nonuniform Random Numbers

## Coordinate transformation method

Uniform random number



map



$\text{total sample size}$ $\frac{1}{N} \int P(r_1, r_2) dr_1 dr_2$	$\text{density}$ $P'(\zeta_1, \zeta_2)$	$\text{area}$ $S(r_1, r_2, dr_1, dr_2)$
--	--	--

**Number of samples:**  $N \overbrace{P(r_1, r_2)}^1 dr_1 dr_2 = N \times P'(\zeta_1, \zeta_2) S(r_1, r_2, dr_1, dr_2)$

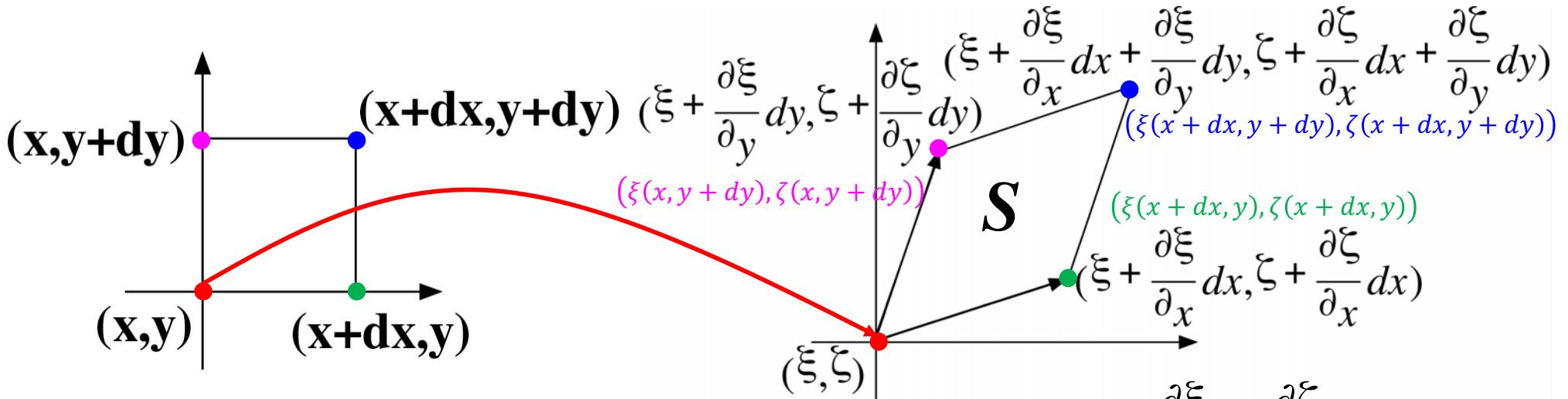
**Areal transformation:**  $S = \left| \frac{\partial(\zeta_1, \zeta_2)}{\partial(r_1, r_2)} \right| dr_1 dr_2$  **Jacobian = areal scaling**

<https://aiichironakano.github.io/phys516/JacobianInversion.pdf>

$$\therefore P'(\zeta_1, \zeta_2) = \left[ \frac{\partial(\zeta_1, \zeta_2)}{\partial(r_1, r_2)} \right]^{-1} \overbrace{P(r_1, r_2)}^1 = \left| \frac{\partial(r_1, r_2)}{\partial(\zeta_1, \zeta_2)} \right|$$

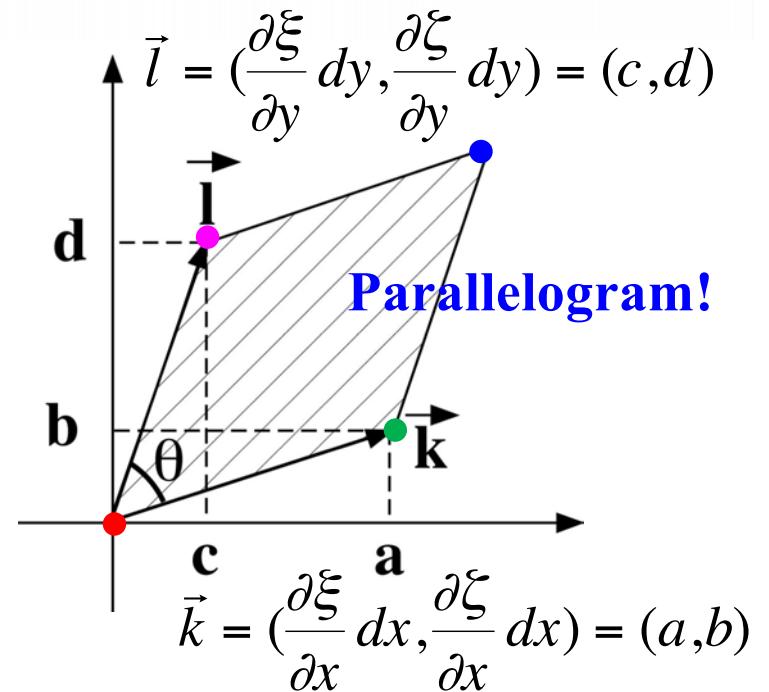
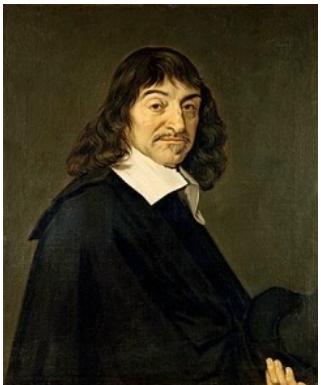
density  $\propto$  (areal magnification) $^{-1}$

# Areal Transformation



$$S = \begin{vmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \zeta}{\partial x} & \frac{\partial \zeta}{\partial y} \end{vmatrix} dxdy = \left( \frac{\partial \xi}{\partial x} \frac{\partial \zeta}{\partial y} - \frac{\partial \xi}{\partial y} \frac{\partial \zeta}{\partial x} \right) dxdy$$

See Appendix B in [MC Basics lecture](#)



$$S = |\vec{k}| |\vec{l}| \sin \theta = \sqrt{a^2 + b^2} \sqrt{c^2 + d^2} \sin \theta = |ad - bc|$$

# Multidimensional Integration

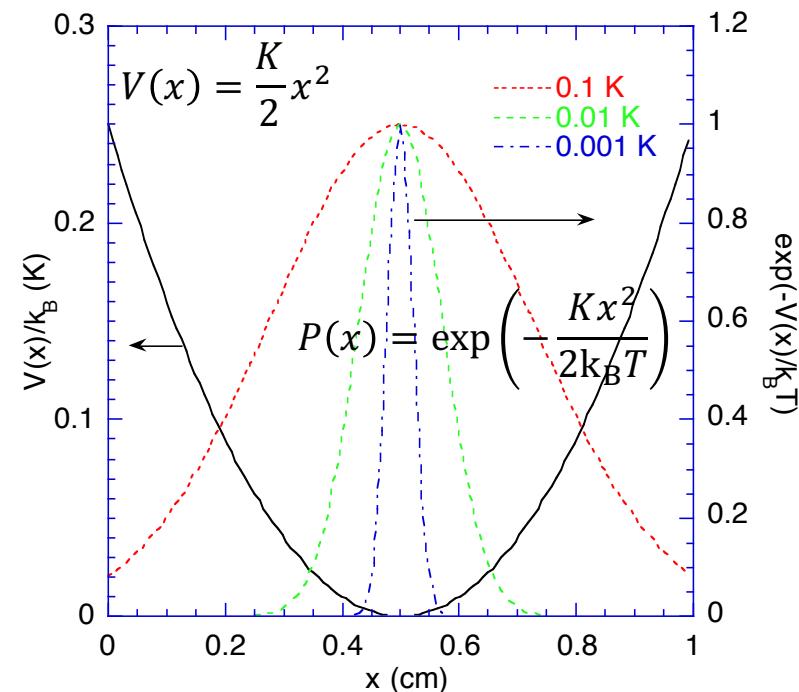
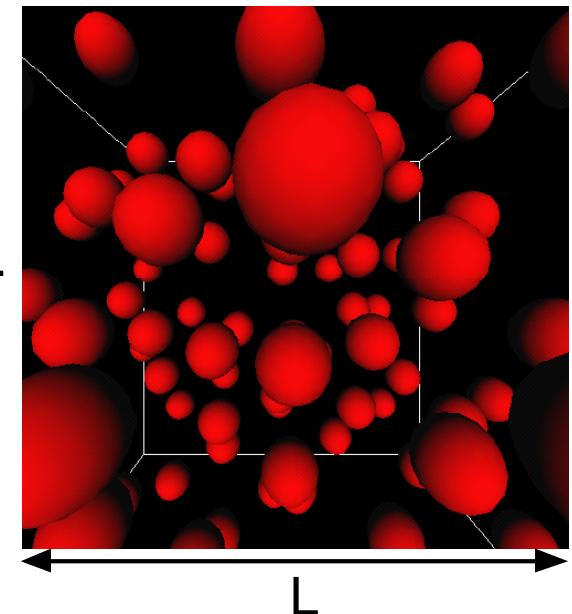
## Statistical mechanics

$$\langle A(\{\vec{r}_i\}) \rangle = \frac{\int d\vec{r}_1 \cdots \int d\vec{r}_N \exp(-\frac{V(\{\vec{r}_i\})}{k_B T}) A(\{\vec{r}_i\})}{\int d\vec{r}_1 \cdots \int d\vec{r}_N \exp(-\frac{V(\{\vec{r}_i\})}{k_B T})}$$

$k_B = 1.38 \times 10^{-16}$  erg/K: Boltzmann constant



$$S = k_B \log W$$



**Need importance sampling!**  
(Example: harmonic oscillators)

$0.2^{100} = 1.3 \times 10^{-70}$  for 100 oscillators

# Importance Sampling

(Problem) How to generate  $3N$ -dimensional sample points according to the Boltzmann probability distribution?

$$\langle A \rangle = \frac{\int d\vec{r}^N A(\vec{r}^N) \exp(-V(\vec{r}^N)/k_B T)}{\int d\vec{r}^N \exp(-V(\vec{r}^N)/k_B T)} \approx \frac{\sum_{n=1}^M A(\vec{r}^N) \exp(-V(\vec{r}^N)/k_B T)}{\sum_{n=1}^M \exp(-V(\vec{r}^N)/k_B T)}$$

- **Importance sampling:** Chooses a sequence of random numbers from a probability density function,  $p(x)$
- **Markov chain:** A sequence of trials that satisfies
  - (a) The outcome of each trial belongs to a finite set of outcomes,  $\{\Gamma_1, \Gamma_2, \dots, \Gamma_N\}$ , called the state space
  - (b) The outcome of each trial depends only on the outcome of the trial that immediately precedes it
- **Transition probability matrix:** Its element  $\pi_{mn}$  is the conditional probability that the next state is  $\Gamma_m$  given that the current state is  $\Gamma_n$ , i.e., the probability of state transition,  $n \rightarrow m$

$$p_m^{(\tau)} = \sum_{n=1}^N \pi_{mn} p_n^{(\tau-1)}$$

# Markov Chain

- **Normalization:** Given that the system is in the state,  $\Gamma_n$ , then the next state must be one of the  $N$  possible states

$$\sum_{m=1}^N \pi_{mn} = 1$$

- **Matrix-vector notation**

$$\Pi = \begin{bmatrix} \pi_{11} & \pi_{12} & \cdots & \pi_{1N} \\ \pi_{21} & \pi_{22} & & \\ \vdots & & \ddots & \\ \pi_{N1} & & & \pi_{NN} \end{bmatrix} \quad \rho^{(\tau)} = \begin{pmatrix} p_1^{(\tau)} \\ p_2^{(\tau)} \\ \vdots \\ p_N^{(\tau)} \end{pmatrix}$$

$$\rho^{(\tau)} = \Pi \rho^{(\tau-1)} = \Pi^2 \rho^{(\tau-2)} = \dots = \Pi^\tau \rho^{(0)}$$

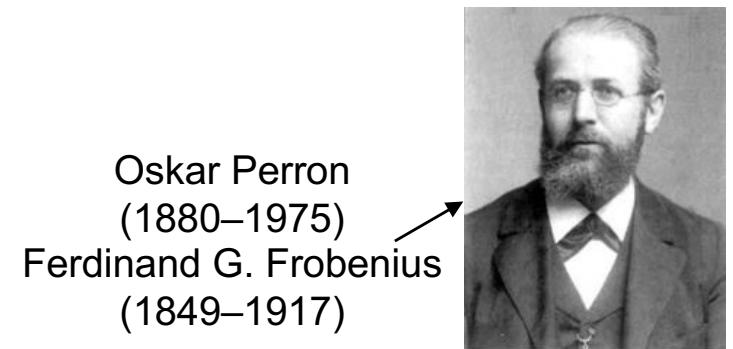


Andrey Markov  
(1856-1922)

**(Perron-Frobenius theorem) Eigenvalues:**  $\Pi \rho_\nu = \varepsilon_\nu \rho_\nu \quad (\varepsilon_1 = 1, |\varepsilon_2|, \dots, |\varepsilon_N| < 1)$

**1. Fixed point:**  $\Pi \rho_1 = \rho_1$

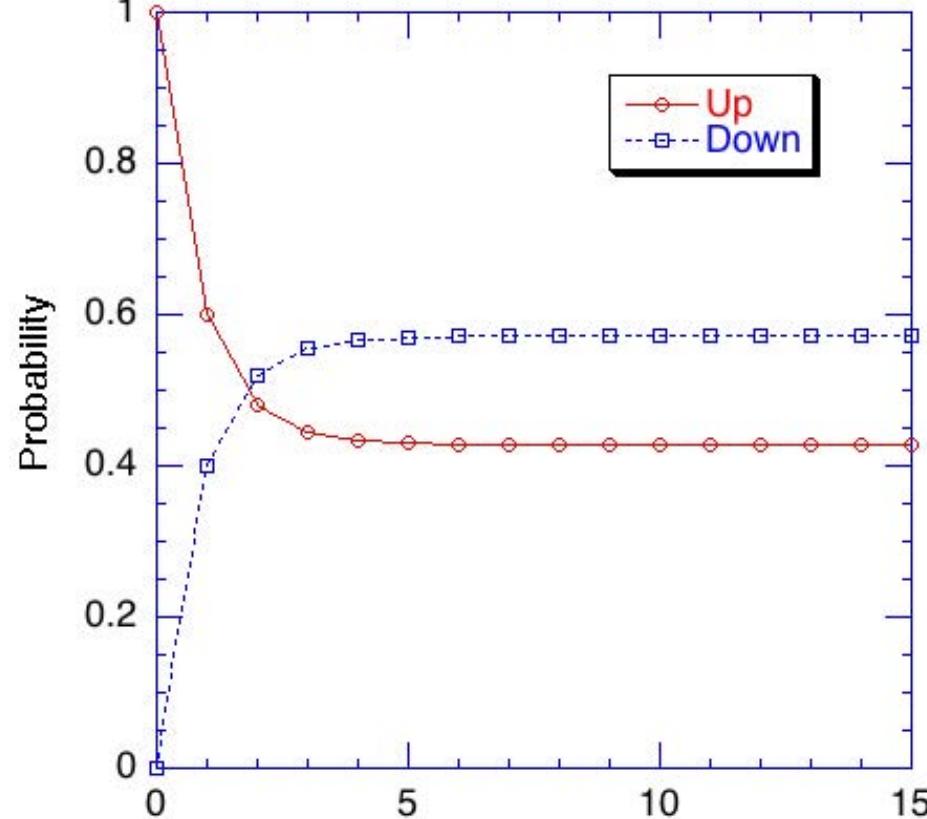
**2. Filter:**  $\Pi^\tau \left( \sum_{\nu=1}^N c_\nu \rho_\nu \right) = \sum_{\nu=1}^N c_\nu \varepsilon_\nu^\tau \rho_\nu \xrightarrow{\tau \rightarrow \infty} c_1 \rho_1$



Oskar Perron  
(1880–1975)  
Ferdinand G. Frobenius  
(1849–1917)

# Example: Two-Level System

$$\Pi = \begin{pmatrix} \uparrow & \downarrow \\ \downarrow & \uparrow \end{pmatrix} = \begin{pmatrix} a & 1-b \\ 1-a & b \end{pmatrix} = \begin{pmatrix} 0.6 & 0.3 \\ 0.4 & 0.7 \end{pmatrix} \quad (a = 0.6, b = 0.7)$$



$$\begin{pmatrix} p_{\uparrow}^{(\tau)} \\ p_{\downarrow}^{(\tau)} \end{pmatrix} = \begin{pmatrix} 0.6 & 0.3 \\ 0.4 & 0.7 \end{pmatrix}^{\tau} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \xrightarrow[\tau \rightarrow \infty]{} \begin{pmatrix} 0.4286 \\ 0.5714 \end{pmatrix}$$

# Eigenvalue Problem

- **Eigensystem**  $\Pi w = \lambda w$   $w = \begin{pmatrix} u \\ v \end{pmatrix}$   
$$\begin{pmatrix} a & 1-b \\ 1-a & b \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \lambda \begin{pmatrix} u \\ v \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} \lambda - a & b-1 \\ a-1 & \lambda - b \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

- For a nontrivial (other than  $u = v = 0$ ) solution,  $(\Pi - \lambda I)^{-1}$  should not exist

$$|\Pi - \lambda I| = (\lambda - a)(\lambda - b) - (a - 1)(b - 1) = (\lambda - a - b + 1)(\lambda - 1) = 0$$

- **Eigenvalues:**  $\lambda_+ = 1$  and  $-1 < \lambda_- = a+b-1 < 1$  **Perron-Frobenius!**  
for a positive matrix:  $a, b > 0$

- **Eigenvectors:**

$$w_+ = \begin{pmatrix} u_+ \\ v_+ \end{pmatrix} = \frac{1}{2-a-b} \begin{pmatrix} 1-b \\ 1-a \end{pmatrix} \quad \text{and} \quad w_- = \begin{pmatrix} u_- \\ v_- \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

# Telescopic Technique

$$\begin{cases} \Pi \begin{pmatrix} u_+ \\ v_+ \end{pmatrix} = \lambda_+ \begin{pmatrix} u_+ \\ v_+ \end{pmatrix} \\ \Pi \begin{pmatrix} u_- \\ v_- \end{pmatrix} = \lambda_- \begin{pmatrix} u_- \\ v_- \end{pmatrix} \end{cases} \quad \text{or} \quad \begin{pmatrix} \Pi_{11} & \Pi_{12} \\ \Pi_{21} & \Pi_{22} \end{pmatrix} \begin{pmatrix} U \\ v_+ \end{pmatrix} = \begin{pmatrix} u_+ & u_- \\ v_+ & v_- \end{pmatrix} \begin{pmatrix} \Lambda \\ 0 \end{pmatrix}$$

$$\Pi U = U \Lambda \quad \text{or} \quad \Pi = U \Lambda U^{-1} \quad \text{Eigen decomposition}$$

$$\begin{pmatrix} p_1^{(n)} \\ p_2^{(n)} \end{pmatrix} = \Pi^n \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = U \Lambda U^{-1} U \Lambda U^{-1} \dots U \Lambda U^{-1} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = U \boxed{\Lambda^n} U^{-1} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}$$

Filtering

$$\begin{pmatrix} \lambda_+^n & 0 \\ 0 & \lambda_-^n \end{pmatrix} \xrightarrow{n \rightarrow \infty} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} p_1^{(n)} \\ p_2^{(n)} \end{pmatrix} = \Pi^n \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \xrightarrow{n \rightarrow \infty} \begin{pmatrix} u_+ \\ v_+ \end{pmatrix} = \begin{pmatrix} \frac{1-b}{2-a-b} \\ \frac{1-a}{2-a-b} \end{pmatrix} = \begin{pmatrix} 0.3/0.7 = 0.4286 \\ 0.4/0.7 = 0.5714 \end{pmatrix}$$

Perron-Frobenius theorem in action!

# Detailed Balance Condition

- Perron-Frobenius theorem → repeated application of  $\Pi$  converges to its eigenvector,  $\rho$ , with eigenvalue 1

$$\rho^{(\tau)} = \Pi^\tau \rho^{(\tau=0)} \xrightarrow[\tau \rightarrow \infty]{} \rho$$

where  $\Pi\rho = \rho$  or  $\sum_{n=1}^N \Pi_{mn} \rho_n = \rho_m$

(Inverse problem = importance sampling) Find a transition probability matrix  $\Pi$  whose 1st eigenvector equals a given  $\rho$ , e.g.,

$$\rho_n = \frac{\exp(-V_n / k_B T)}{\sum_n \exp(-V_n / k_B T)}$$

(Solution = detailed balance condition) Let  $\Pi$  satisfy, for any pair of states,

$$\pi_{mn} \rho_n = \pi_{nm} \rho_m$$

then  $\sum_{n=1}^N \pi_{mn} \rho_n = \rho_m$ .

$$\therefore \sum_{n=1}^N \pi_{mn} \rho_n = \boxed{\sum_{n=1}^N (\pi_{nm})} \rho_m$$



# Metropolis Algorithm

---

(Problem) Find a transition probability matrix  $\Pi$  that satisfies the detailed balance condition for a given  $\rho$ :

$$\pi_{mn}\rho_n = \pi_{nm}\rho_m$$

(Solution) Let  $\alpha_{mn} = \alpha_{nm}$  be a symmetric attempt matrix, then

$$\pi_{mn} = \begin{cases} \alpha_{mn} & \rho_m \geq \rho_n \quad m \neq n \\ (\rho_m / \rho_n)\alpha_{mn} & \rho_m < \rho_n \quad m \neq n \\ 1 - \sum_{m' \neq n} \pi_{m'n} & m = n \end{cases}$$

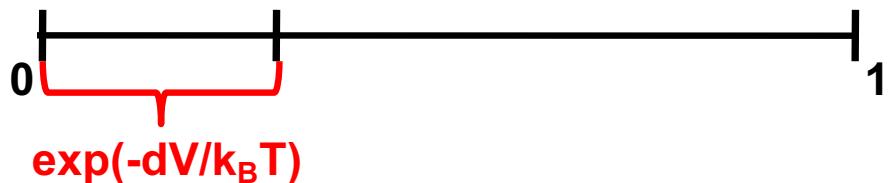
(Example) Statistical mechanics: no need for the absolute weight!

$$\frac{P(\{\vec{r}_i^{\text{next}}\})}{P(\{\vec{r}_i^{\text{current}}\})} = \frac{\exp(-V(\{\vec{r}_i^{\text{next}}\})/k_B T)}{\exp(-V(\{\vec{r}_i^{\text{current}}\})/k_B T)} = \exp(-\delta V(\{\vec{r}_i^{\text{next}}\}, \{\vec{r}_i^{\text{current}}\})/k_B T)$$

# Metropolis Algorithm

Let  $X$  be a state variable (e.g. positions of atoms  $\in \mathbb{R}^{3N}$ ) and calculate the statistical average of function  $A(X)$  with probability density  $P(X) \propto e^{-V(X)/k_B T}$

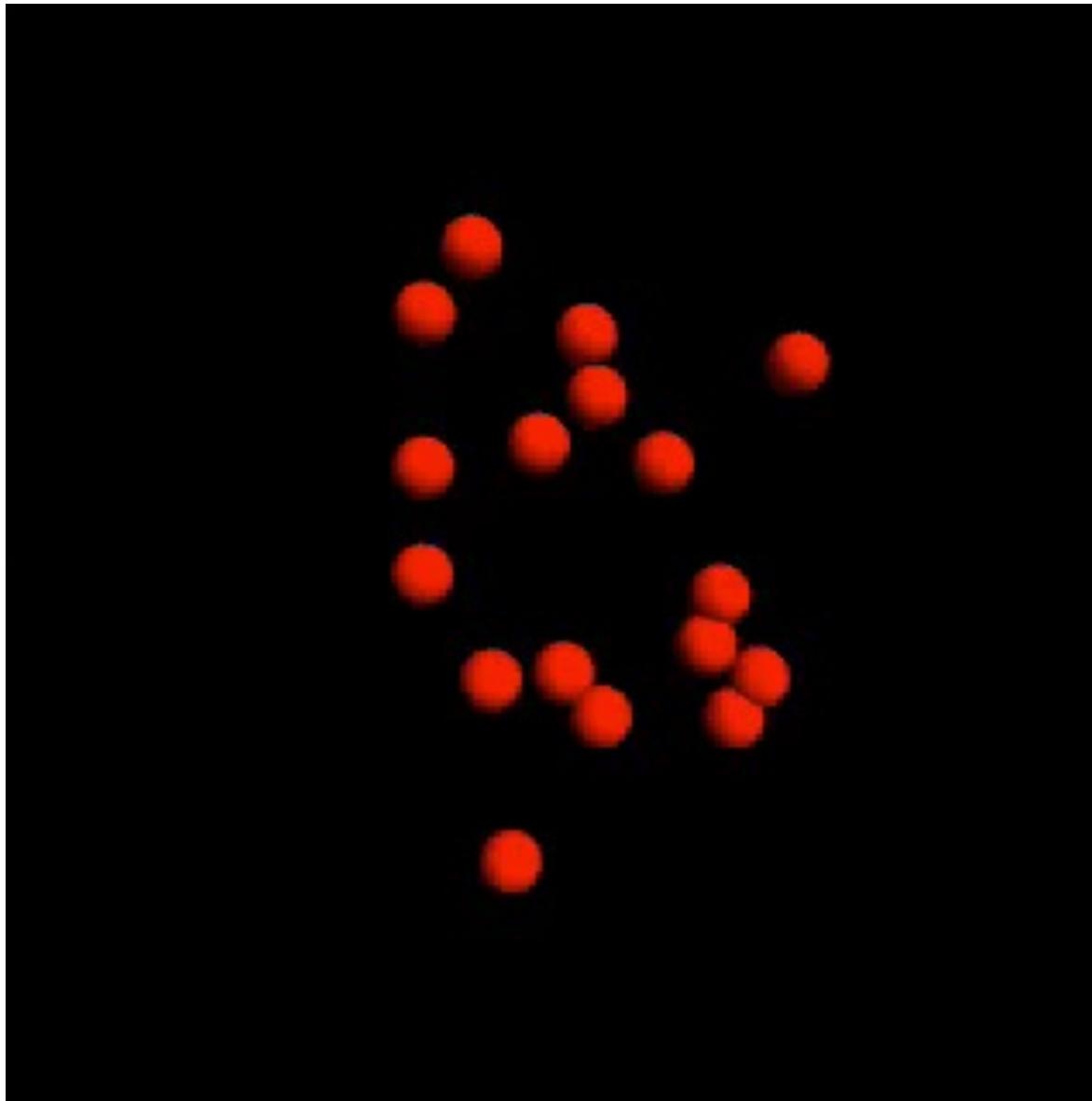
```
Sum_A = 0
for n = 1 to Ntrial
    X' ← X + dX /* Random state-transition attempt */
    Compute dV = V(X') - V(X)
    if dV < 0 then /* Greedy */
        Accept X ← X'
    else if rand() ≤ exp(-dV/k_B T) then /* Occasional uphill move */
        Accept X ← X'
    /* else the state remains X */
    endif
    Sum_A = Sum_A + A(X) /* Count X more than once, if it remains */
endfor
Avg_A = Sum_A/Ntrial
```



Note: rand() here is a uniform random number in the range  $[0, 1]$

# Monte Carlo Simulation

---



$$V = \sum_{i < j} \frac{e^2}{|\vec{r}_i - \vec{r}_j|}$$

**Random trial → acceptance by a cost criterion**

# Summary: Monte Carlo Basics

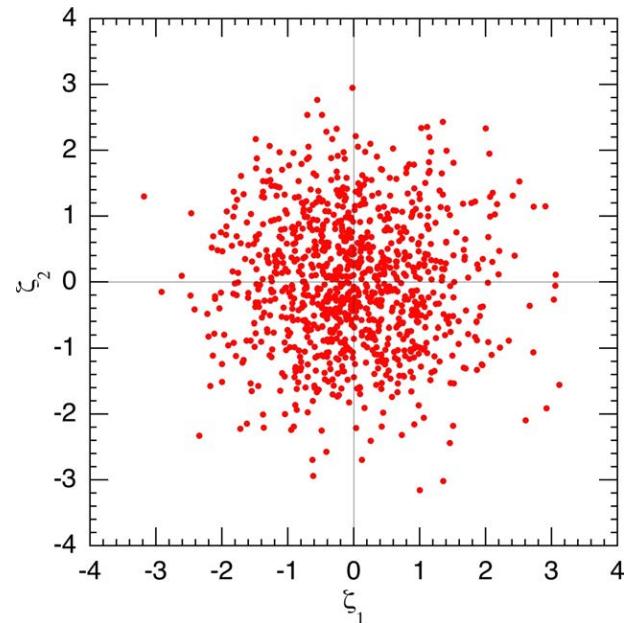
## 1. Sample-mean MC

$$\int dx f(x) p(x) \cong \bar{f} \pm \sqrt{\frac{\bar{f}^2 - (\bar{f})^2}{M - 1}}$$

## 2. Visualize probability density by a set of points: nonuniform random-number generation by coordinate transformation

$$P'(\zeta_1, \dots, \zeta_N) = \frac{P(r_1, \dots, r_N)}{|\partial(\zeta_1, \dots, \zeta_N)/\partial(r_1, \dots, r_N)|}$$

Jacobian =  
Areal magnification factor



## 3. Metropolis (Markov-chain) MC algorithm for importance sampling

Random attempt → conditional acceptance with probability  $\exp(-\Delta V/k_B T)$