UPPSALA
UNIVERSITET

Department of Information Technology

Scientific Computing for Data Analysis

Davoud Mirzaei

Lecture 3: Convergence of Monte Carlo, Stochastic Processes

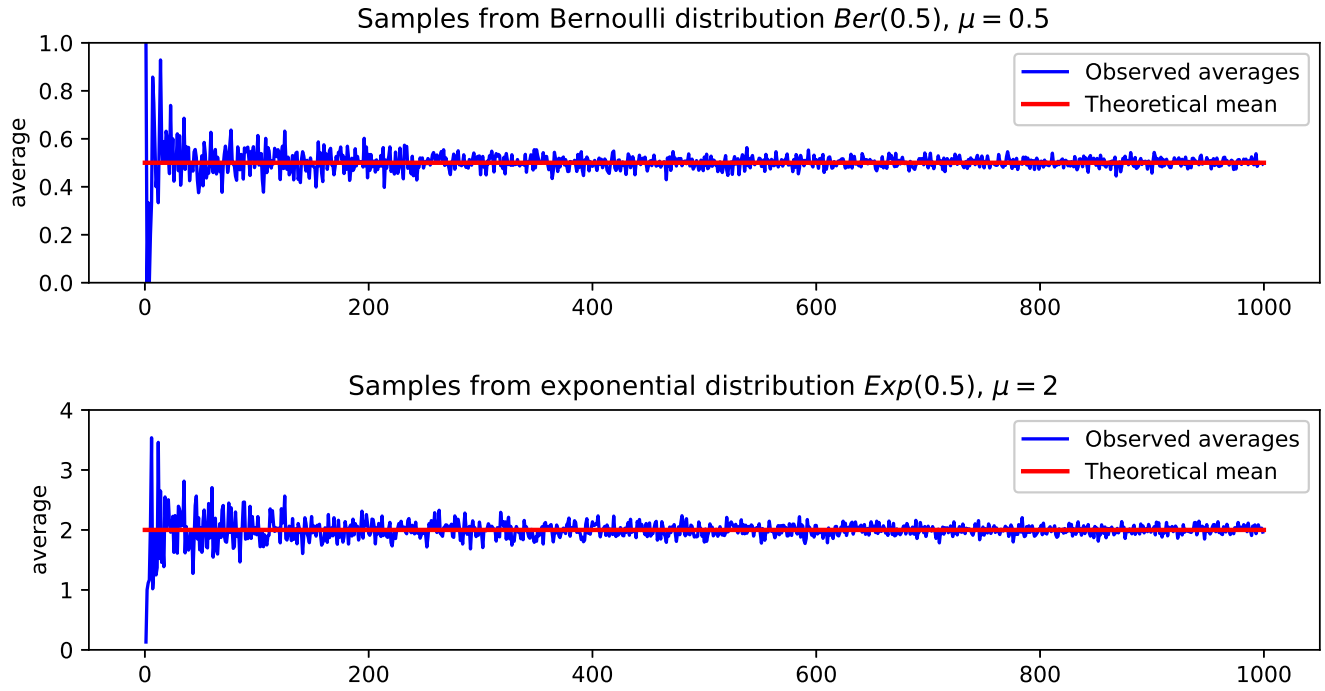
Agenda

- Law of large numbers and CLT
- Convergence of Monte Carlo method
- Stochastic Processes
- Markov processes (Random walk and Brownian motion)

Law of Large Numbers (Example)

Experiments with Bernoulli and Exponential distributions

$$\bar{x} = \frac{x_1 + \cdots + x_n}{n}, \quad n = 1, 2, \ldots, 1000$$



In both cases we observe that the sample mean converges to the theoretical mean of the variable. This is true for all other distributions.

The law of large numbers

If you repeat an experiment independently a large number of times, the average of the results will approach the expected value

Let $X_1, X_2, \ldots$ be independent and identically distributed (iid) random variables with expectation μ and variance σ^2. For each N, let

$$S_N = X_1 + X_2 + \cdots + X_N.$$

The law of large numbers states that

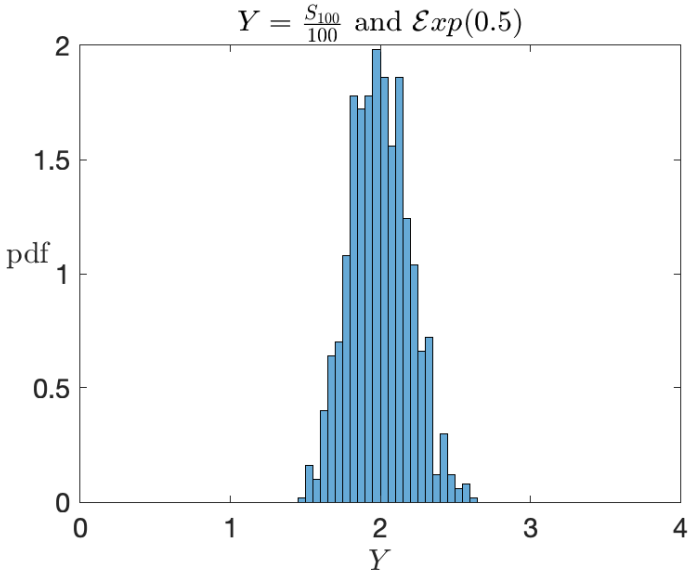
$$\frac{S_N}{N} \approx \mu, \quad \textit{for large values of } N.$$

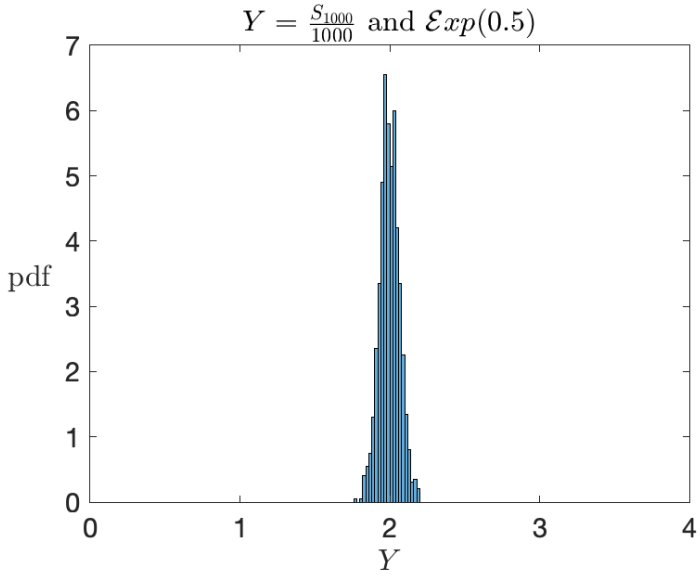
More precisely

$$\mathbb{P}\left(\lim_{N\to\infty} \frac{S_N}{N} = \mu\right) = 1$$

Central Limit Theorem (CLT)

Test: $X_1, \ldots, X_N$ from exponential $\mathcal{E}xp(0.5)$ and $S_N = X_1 + \cdots + X_N$.
Now, generate $M = 1000$ random numbers from new variable $Y = S_N/N$ and plot the histogram. Two cases: $N = 100$ and $N = 1000$:





- What is the distribution of S_N/N?
- As N is increased, the std is decreased while the mean remains unchanged

Central Limit Theorem

Central Limit Theorem: The mean of independent random variables from any distribution tends to a normal distribution. The rate of convergence is (the standard deviation decreases as) $\frac{1}{\sqrt{N}}$

The central limit theorem describes the limiting distribution of S_N/N. It states that the random sum S_N/N has a distribution that is approximately normal, when N is large.

In other words, for a large N we have

$$\frac{S_N}{N} \sim \mathcal{N}\left(\mu, \frac{\sigma^2}{N}\right)$$

Note: The CLT holds true regardless of how X_k are distributed.

Convergence of Monte Carlo method

Let $X \sim f$, and $X_1, \ldots, X_N$ is a sample (iid) from X. Then

$$Y := \frac{1}{N} \sum_{k=1}^{N} g(X_k)$$

as a new random variable has its own mean and variance:
$\mu_Y = \mathbb{E}(Y) = ?,\ \sigma_Y^2 = \mathrm{Var}(Y) = ?$

$$\mathbb{E}[Y] = \frac{1}{N}\mathbb{E}\Big(\sum_{k=1}^{N} g(X_k)\Big) = \frac{1}{N}\sum_{k=1}^{N}\mathbb{E}[g(X_k)] = \frac{1}{N}(\mu + \cdots + \mu) = \mu$$

Using the *central limit theorem* for large values of N we have

$$Y \sim \mathcal{N}(\mu, \sigma^2/N), \quad \mu = \mathbb{E}[g(X)], \quad \sigma^2 = \mathrm{Var}[g(X)]$$

which means

$$\sigma_Y^2 = \frac{\sigma^2}{N}$$

or

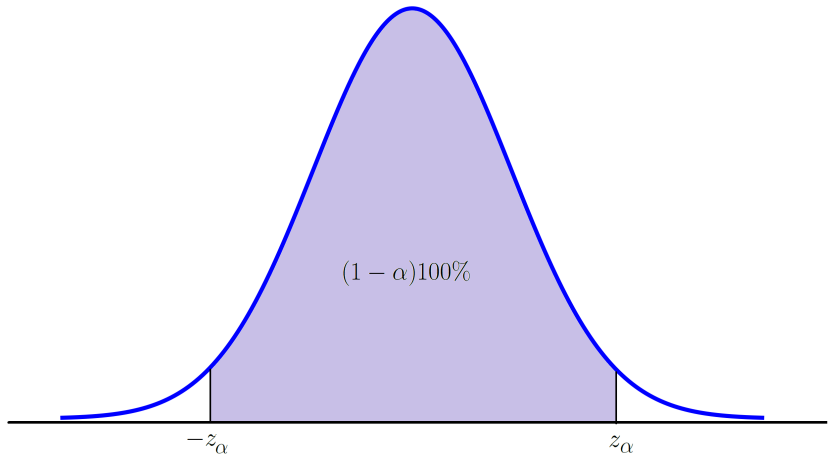
$$\sigma_Y = \frac{\sigma}{\sqrt{N}}$$

Convergence of Monte Carlo methods

$$Y \sim \mathcal{N}(\mu, \sigma^2/N)$$

means

$$\mathbb{P}\left(\mu - z_\alpha \frac{\sigma}{\sqrt{N}} \leqslant Y \leqslant \mu + z_\alpha \frac{\sigma}{\sqrt{N}}\right) = (1-\alpha)\%$$



If $\alpha = 0.05$ then $z_\alpha \doteq 1.96$, i.e.

$$\mathbb{P}\left(\mu - 1.96 \frac{\sigma}{\sqrt{N}} \leqslant Y \leqslant \mu + 1.96 \frac{\sigma}{\sqrt{N}}\right) = 95\%$$

For example:

- If $\alpha = 0.05$ then $z_\alpha \doteq 1.96$, which means that with 95% probability Y falls in *confidence interval*

$$\left[\mu - 1.96\frac{\sigma}{\sqrt{N}}\ ,\ \mu + 1.96\frac{\sigma}{\sqrt{N}}\right].$$

- If $\alpha = 0.01$ then $z_\alpha \doteq 2.576$, which means that with 99% probability Y falls in *confidence interval*

$$\left[\mu - 2.576\frac{\sigma}{\sqrt{N}}\ ,\ \mu + 2.576\frac{\sigma}{\sqrt{N}}\right].$$

- The probability will increase to 0.999% by $z_\alpha \doteq 3.29$.
- This confidence interval shows that the Monte Calro method *almost surly* converges with rate $N^{-1/2}$: with probability $(1-\alpha)\%$ the error e_N satisfies

$$|e_N| \leqslant z_\alpha \frac{\sigma}{\sqrt{N}} \Longrightarrow |e_N| = \mathcal{O}(N^{-1/2})$$

Convergence of Monte Carlo method

- The confidence interval shows that the accuracy of the estimator Y is determined by its std, i.e., $\sigma/\sqrt{N}$.
- Our aim was to estimate μ but for error estimation the value of σ is also required
- Usually, σ^2 (variance of $g(X)$) is unknown, but can be estimated with the *unbiased sample variance* s^2:

$$\overline{g} = \frac{1}{N}\sum_{j=1}^{N} g(x_j), \quad s^2 = \frac{1}{N-1}\sum_{j=1}^{N}(g(x_j) - \overline{g})^2,$$

  which tends to σ^2 by the law of large numbers
- For large values of N, the *approximate* confidence interval for Y is

$$\left[\mu - z_\alpha \frac{s}{\sqrt{N}} \ , \ \mu + z_\alpha \frac{s}{\sqrt{N}}\right].$$

Example

Use Pyhton to to estimate the integral below using MC integration. Estimate the errors for probability 95% with different values of $N = 10, 10^2, \cdots, 10^6$:

$$\int_{-\infty}^{\infty} (x^4 - x + 1) \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-x^2/2} dx$$

```
z = 1.96  # for a 95% probability
print("     N     mean     e_N" )
for k in range(1,7):
    N = 10**k
    X = np.random.randn(N,1)
    gX = X**4-X+1
    int_mc = np.mean(gX)
    e = z*np.std(gX)/np.sqrt(N)
    print("%9d" % N, "%.5f" % int_mc, "%.5f" % e)
```

A single execution gives:

```
        N  mc_sol   e_N
       10 4.39314 4.64702
      100 2.64882 0.72844
     1000 3.72380 0.48118
    10000 4.16994 0.21637
   100000 3.96749 0.05987
  1000000 3.99207 0.01926
 10000000 3.99711 0.00610
```

From old exams

6 **Analysis2_20230819**

Suppose that we have estimated the integral $\int_a^b f(x)\,dx$ using the Monte Carlo method with $N = 1000$ random numbers. Alongside, we have assessed the error through the length of the confidence interval (or standard deviation), which we found to be $\varepsilon = 0.2$. Now, if we seek to enhance the accuracy by increasing the number of random points to $N = 16000$, what would the resulting length of the confidence interval (or standard deviation) be?

Calculate and enter it here: [ ] .

---

5 **May2023: PartA: Ana-MC**

We use Monte-Carlo simulation to estimate $G = \int_0^1 g(x)\,dx$ with $g(x) = \frac{6}{1+x^2}$. We sample N samples $X_i, i = 1, \ldots, N$, from the uniform distribution $U(0,1)$ and estimate the G as $\hat{\mu}_N = \frac{1}{N}\sum_{i=1}^{N} g(X_i)$.
We also compute
$\frac{1}{99}\sum_{i=1}^{100}\left(g(X_i) - \hat{\mu}_{100}\right)^2 = 1.21$ and $\hat{\mu}_{100} = 4.75$
Use that information to determine the interval (with 2 decimals), within which the true integral value G lies with 95% probability using N=100. Which theorem is behind the approach you are using?

Monte Carlo: Conclusion

- Monte Carlo is a stochastic methods: perform N stochastic simulations and finally take average
- With a certain probability (for example 95%) the MC method converges with rate $\frac{1}{\sqrt{N}}$: To double the accuracy, you have to increase number of samplings with a factor 4
- Due to low accuracy and slow convergence, methods like mid-point, trapezoid and Simpson's are preferred
- . . . but, Monte Carlo accuracy is independent of dimension, whereas mid-point/trapezoid/Simpson are not
- Hence, MC integration is used when solving higher dimensional integrals

$$\int \cdots \int g(x_1, \ldots, x_n)\, dx_1 \cdots dx_n$$

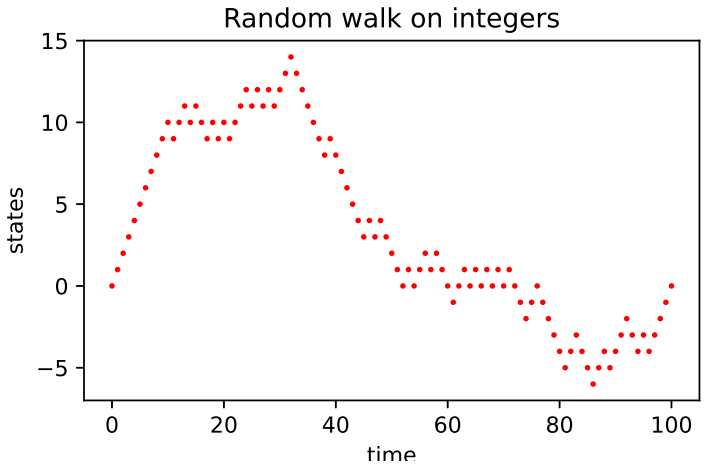
- MC methods are a group of "bad" numerical methods that still work when all "good" methods doesn't work at all

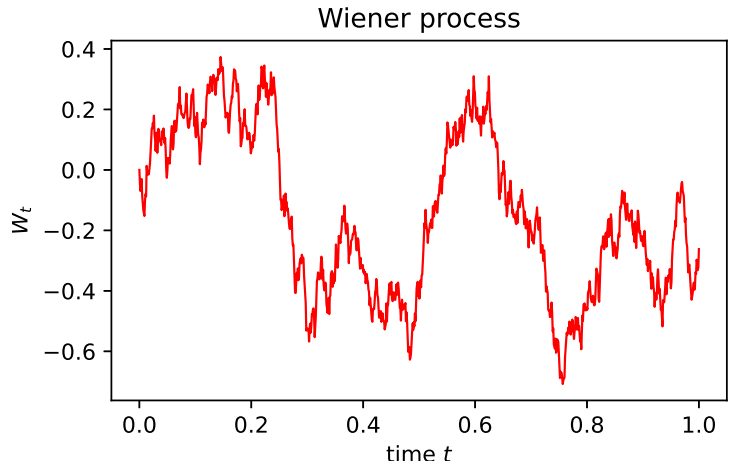
Stochastic Processes

Stochastic Process

Definition: A family of random variables $\{X_t : t \in T\}$ is called a **stochastic process** or random process.

- If $T = \{0, 1, 2, \ldots\}$ the sequence $X_0, X_1, \ldots$ is called a stochastic process with *discrete time parameter*
- If $T = [0, \infty)$ the family is called a stochastic process with *continuous time parameter*
- X_0 is called the *initial state* of the process; and the random variable X_t for a $t \in T$ is called the state of the process at time t
- Independent of T (discrete or continuous), states X_t may have either continuous or discrete pdf.





Example: Stochastic process

Example: A coin purse contains 5 quarters (each worth 25¢), 5 dimes (each 10¢) and 5 nickels (each 5¢).

- draw coins one by one and set on the table
- let $X_t =$ total value of coins set on the table after t draws
- $X_0 = 0$
- $\{X_t, t = 0, 1, 2, \ldots\}$ is a stochastic process (discrete time, discrete pdf)

Assume that in the first 6 draws all 5 nickels and 1 quarter are drawn, so $X_6 = 50¢$. What is X_7?, what is $\mathbb{P}(X_7 = 55)$?, what is $\mathbb{P}(X_7 = 60)$?

If we know $X_1, X_2, \ldots, X_6$ then we can answer questions like $\mathbb{P}(X_7 = x_7) =$?. The process in *memory-dependent*. X_{t+1} is conditioned to all previous X_js:

$$\mathbb{P}\left(X_{t+1} = x_{t+1} \middle| X_0 = x_0, X_1 = x_1, \ldots, X_t = x_t\right)$$

Example: Stochastic process (Markov process)

Example: In previous example, let us define

$$X_t = \text{ the count of various coin types on the table}$$

For example $X_0 = (0, 0, 0)$ and $X_6 = (1, 0, 5)$.

Now the process is memoryless: We can compute different probabilities for X_7 if we only know X_6 and not more. X_{t+1} depends solely on X_t:

$$\mathbb{P}\left(X_{t+1} = x_{t+1} \middle| X_0 = x_0, X_1 = x_1, \ldots, X_t = x_t\right) = \mathbb{P}\left(X_{t+1} = x_{t+1} \middle| X_t = x_t\right)$$

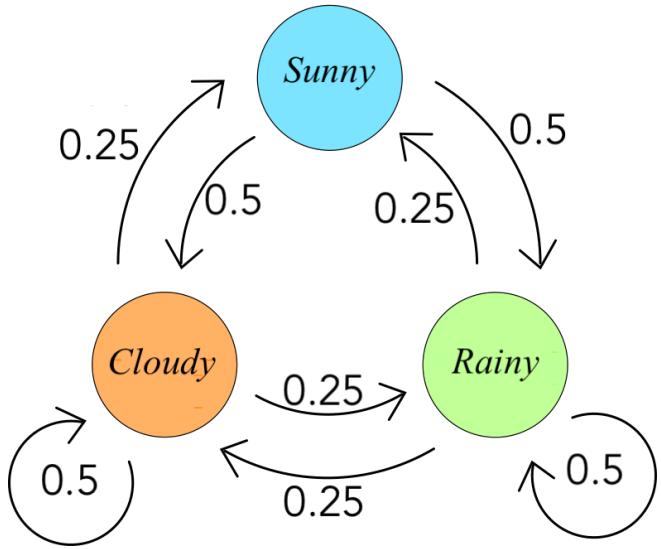
This type of stochastic process is called Markov Process.

A stochastic process is called Markov, roughly if one can make predictions for the future of the process based solely on it's present state.

Markov Process (example: discrete time)

Three state Markov process (daily morning weather in Uppsala): Three different states: (1) sunny, (2) cloudy, or (3) rainy. Observations of the weather forecasting office show that

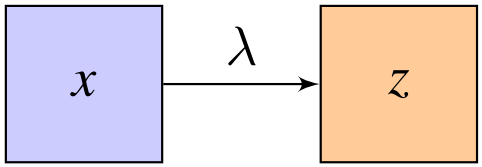
- a sunny day is never followed by another sunny day
- Rainy or cloudy weather is equally probable after a sunny day
- A rainy or cloudy day is followed by 50% probability by another day with the same weather
- If the weather is changing from cloudy or rainy weather, the following day will be sunny only in half of the cases



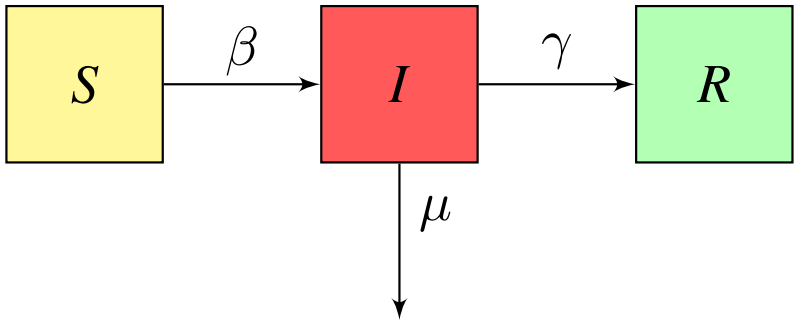
If today is cloudy then tomorrow is sunny with 25% probability:
$\mathbb{P}(X_1 = sunny | X_0 = cloudy) = 0.25$
The process is Markov. Why?

Markov Process (examples: continuous time)

- Radioactive decay: $X_t = amount\ of\ x\ at\ time\ t$



- SIR-model in epidemiology: $I_t = number\ of\ infected\ people\ at\ time\ t$



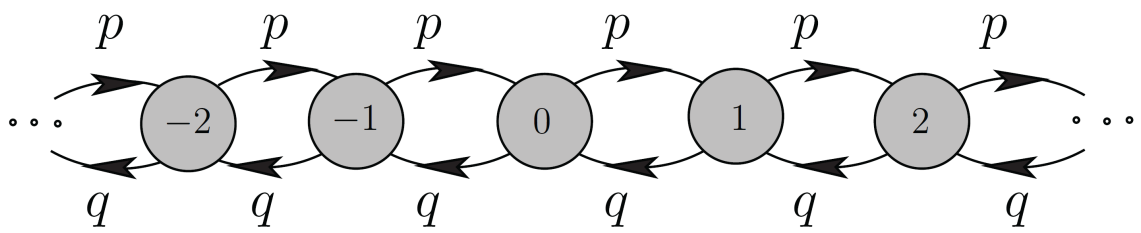
Three states:
S - the number of susceptible
I - the number of infectious
R - the number of recovered

Rates:
β - infection rate
γ - recovery rate
μ - death rate

Compared to the Uppsala daily weather, here the probabilities (propensity functions) are time dependent. (Later in the course we present Gillespies method to solve such models)

Markov Process (Simple Random Walk)

Let $p \in (0,1)$ be a real number and $q = 1 - p$. The Markov chain X with state space $\{\ldots, -2, -1, 0, 1, 2, \ldots\}$, initial state $X_0 = 0$, and following transition graph is called a **random walk** on integers:



The random walk X_t can be characterized as

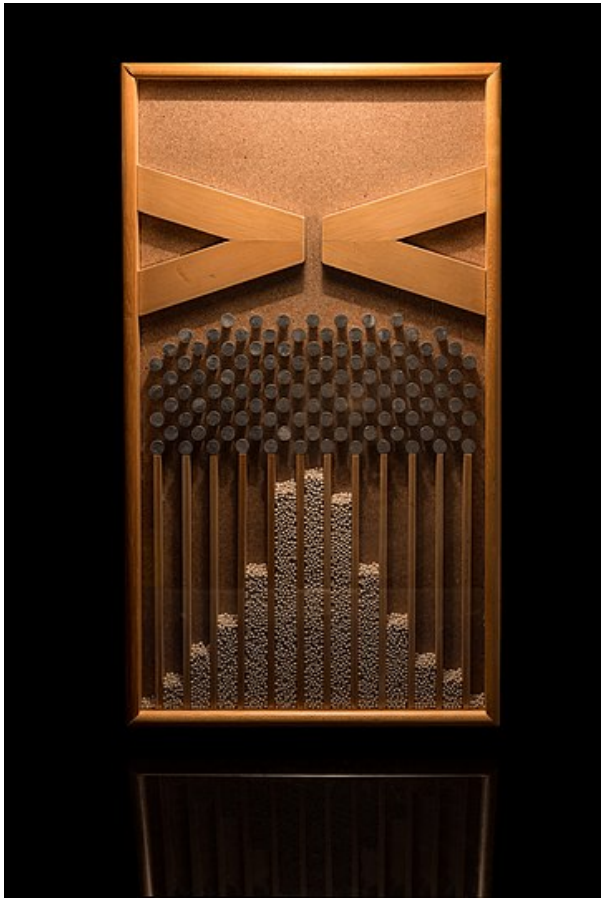
$$X_{t+1} = X_t + \tilde{B}_t$$

$\tilde{B}_t$ is Bernoulli distribution with state $\{-1, 1\}$ and probabilities q and p. Indeed $\tilde{B}_t = 2B_t - 1$ where $B_t \sim \mathcal{B}er(p)$ (standard Bernoulli). Equivalently:

$$X_t = \sum_{j=0}^{t-1} \tilde{B}_j$$

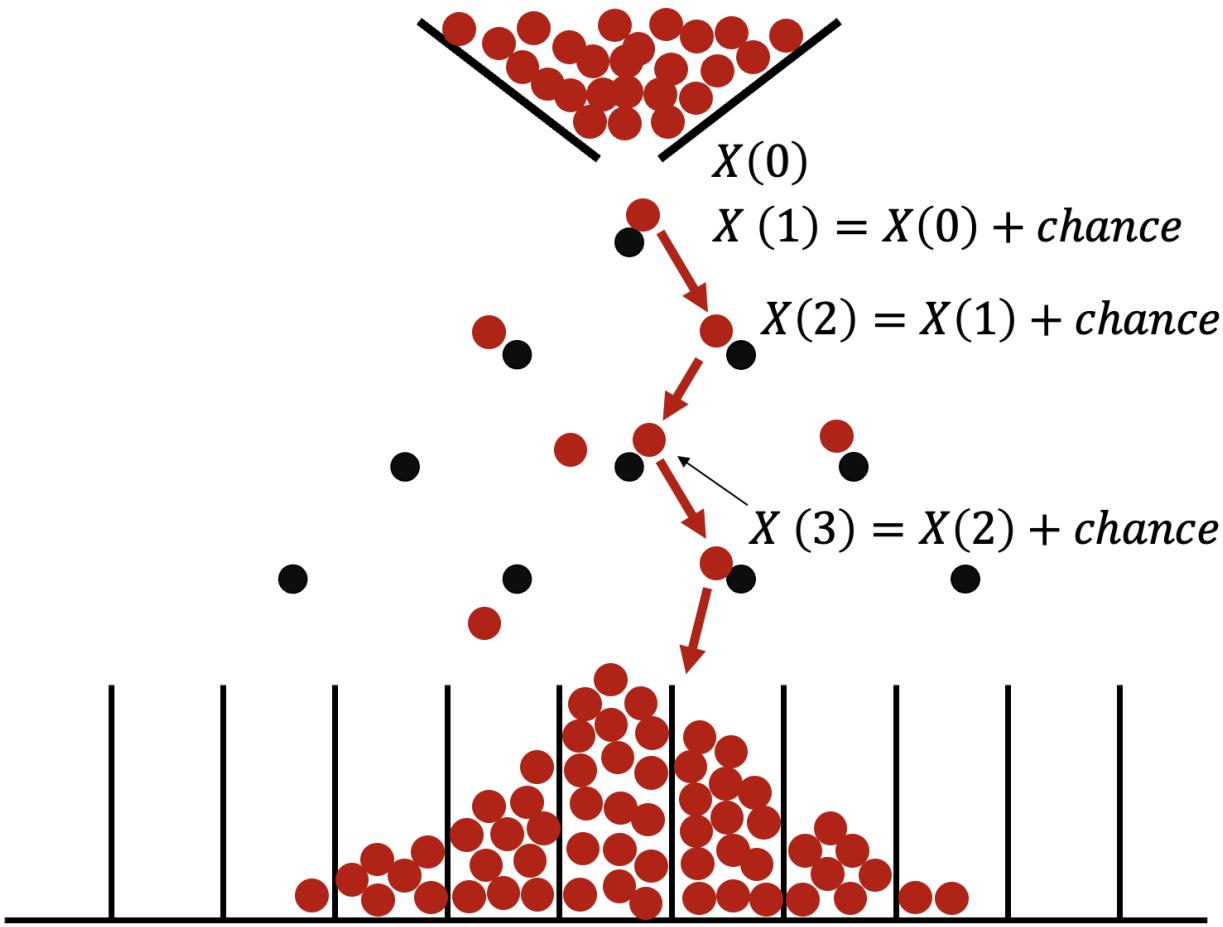
Sum of Bernoulli's is binomial and binomial for large t is approximately normal (CLT)

Galton board



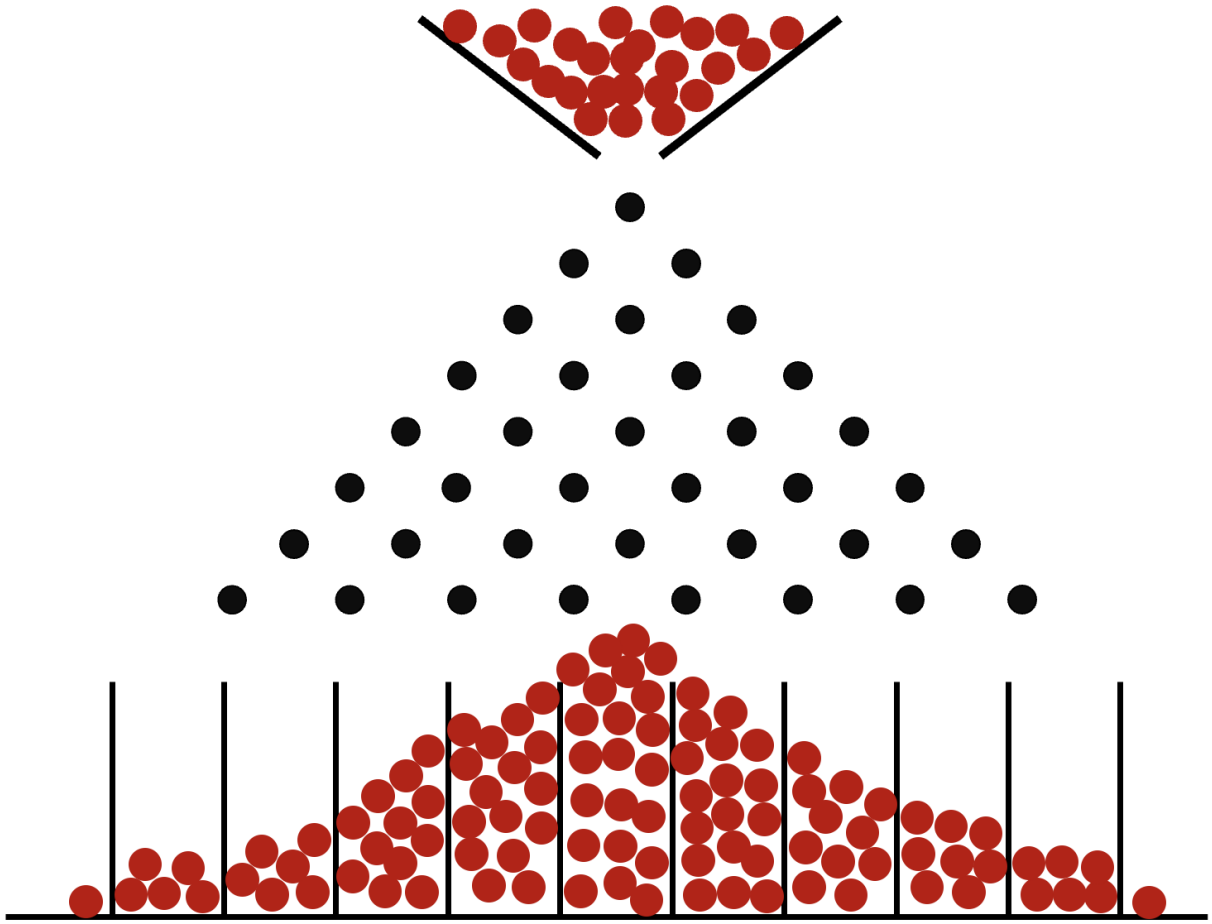
https://en.wikipedia.org/wiki/Galton_board

A random walk model in 1D



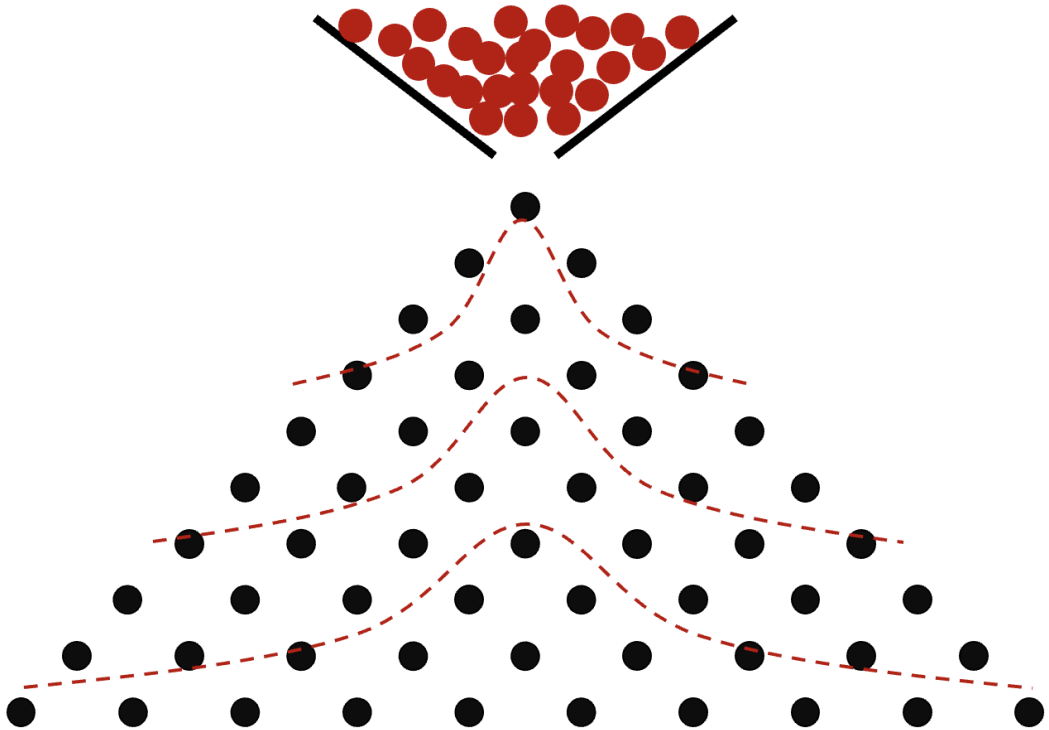
Each bullet (particle) follows a path. Distribution of $X(t)$ approaches the normal distribution

A random walk model in 1D



Increase the levels (more steps, larger t) leads to wider normal distribution

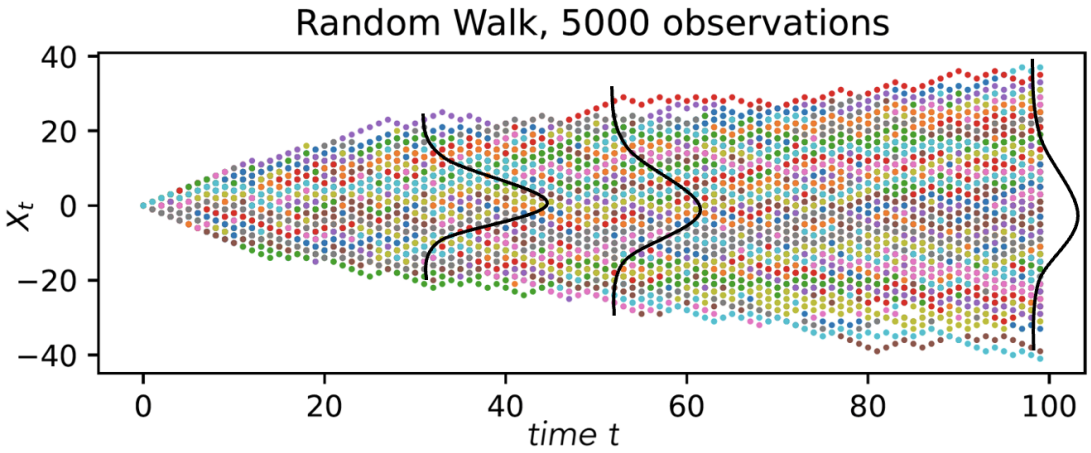
A random walk model in 1D

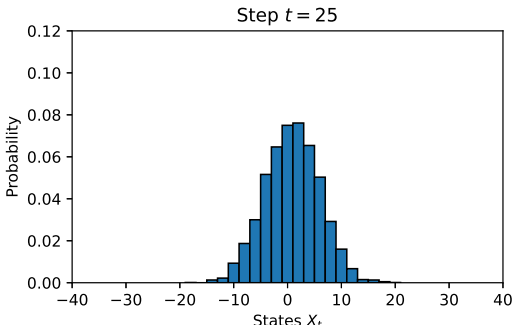


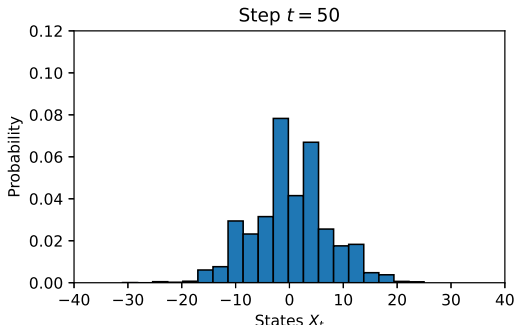
Wider distribution at each level - increased variance in each step/level. Equal increase in each step

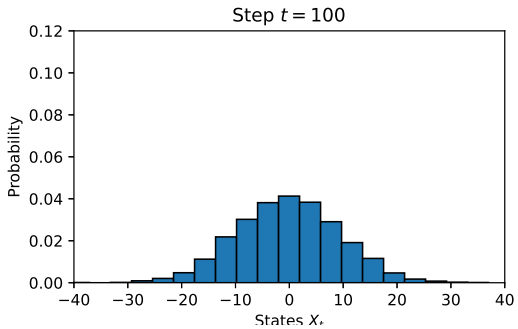
Markov Process (Simple Random Walk)

5000 random paths for $p = 0.5$









The distribution of X_t for large t approaches normal pdf with mean 0, the std gets larger as t increases

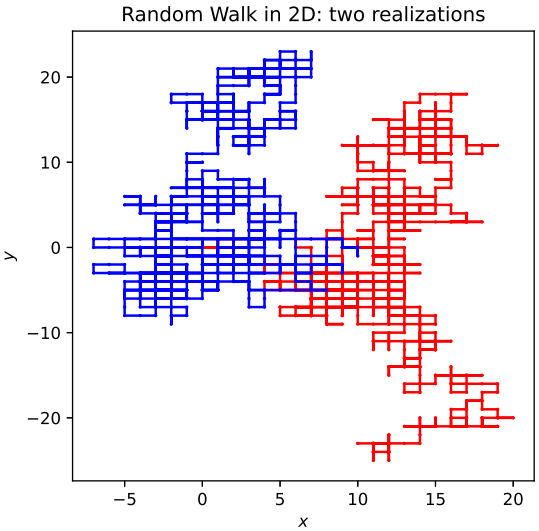
Random walk in higher dimensions

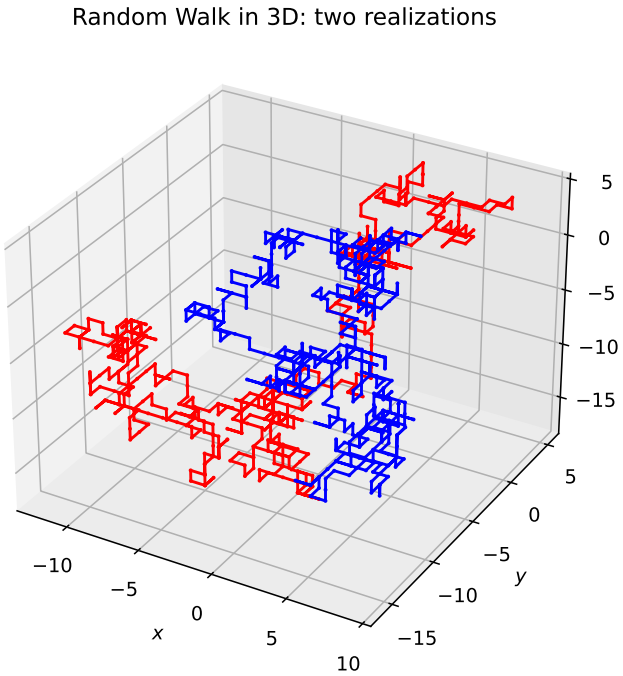
In general we have

$$X_{t+1} = X_t + B$$

where $B \sim \mathcal{DD}([x_1, \ldots, x_{2d}], [p_1, \ldots, p_{2d}])$ (discrete distribution)

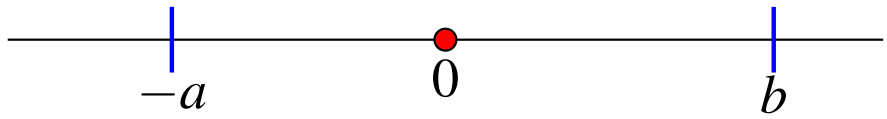
- An example in 2D: $p_1 = p_2 = p_3 = p_4 = 1/4$ and $x_1 = [-1, 0]$, $x_2 = [1, 0]$, $x_3 = [0, -1]$, and $x_4 = [0, 1]$





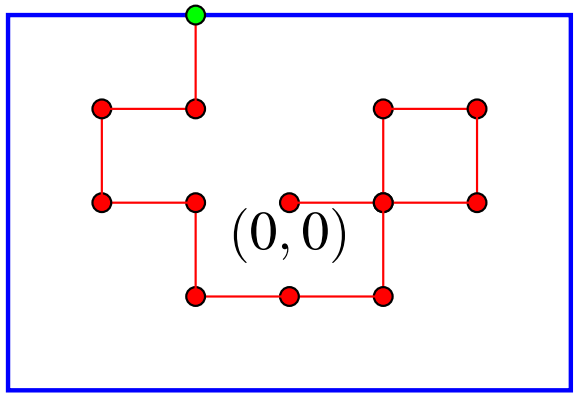
Hitting the boundaries

- If a and b are positive integers, the Monte Carlo algorithm can be used to estimate the expected number of steps until a simple 1D random walk starting at 0 first hits b or $-a$ (boundaries)



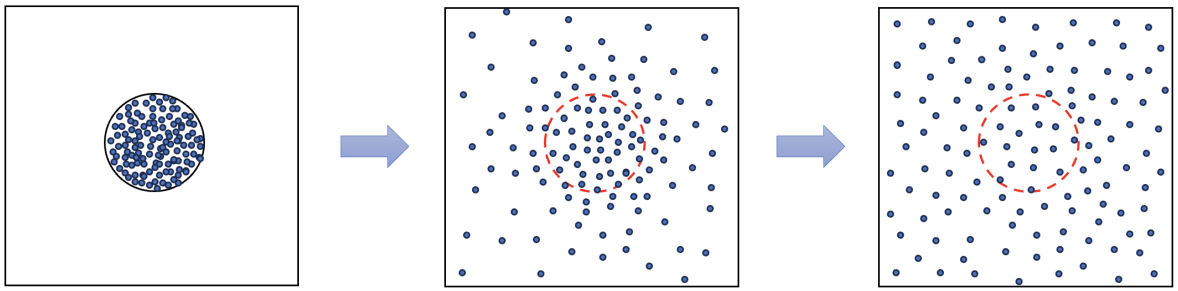
  Also compute the probability that this walk will hit b before $-a$. Hint: the exact solutions are ab and $a/(a+b)$, respectively.

- Extend the algorithm to 2D and 3D by replacing points a and b by edges of a box in 2D or 3D. Think about possible changes in the algorithm to force it working in 2D and 3D.



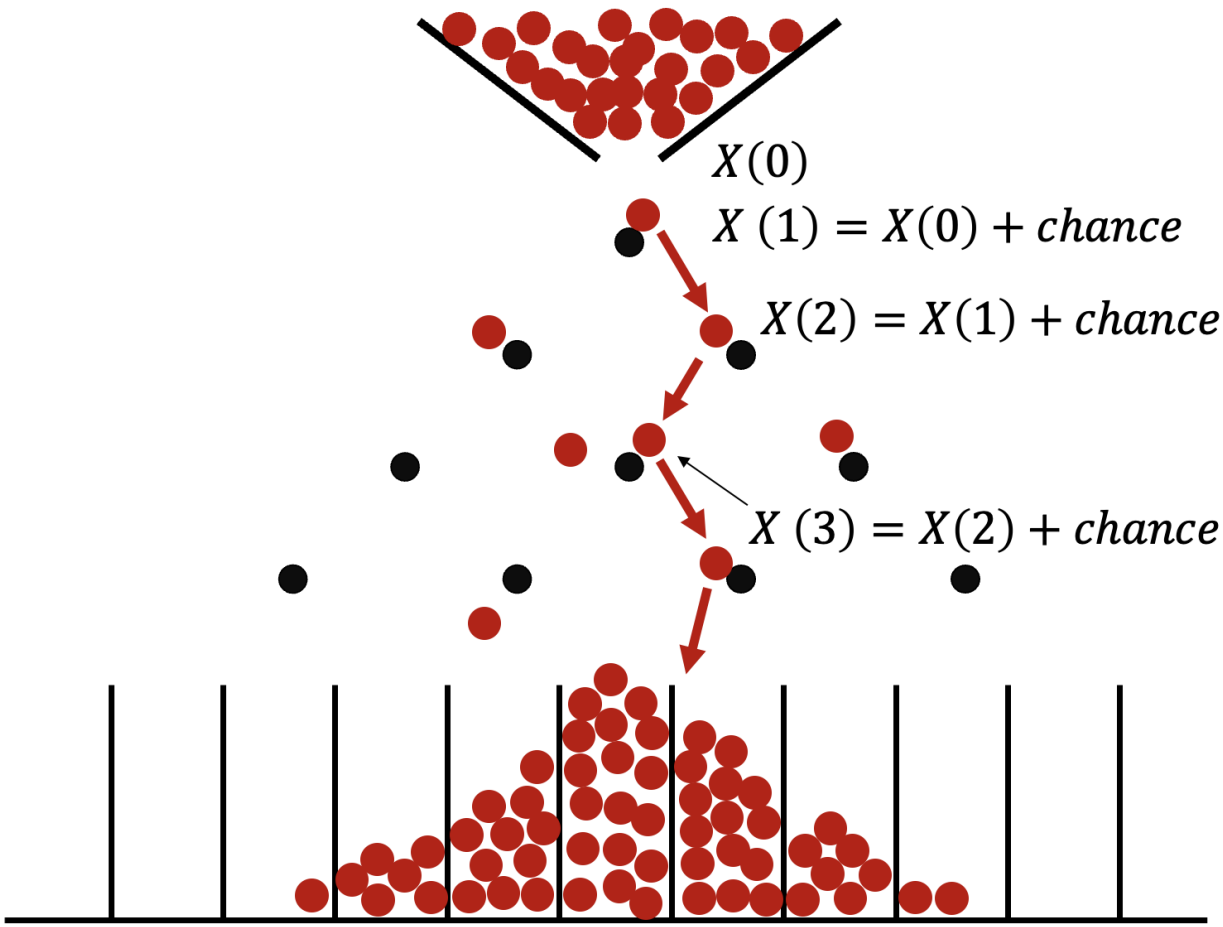
Brownian motion

- Is a Markov process with continuous time variable
- A continuous version of random walk (a random walk with tiny step lengths)
- Introduced by Robert Brown 1827, motion of pollen particles in water
- Also called Wiener process (special case of a large class of processes called diffusion processes)
- Describes the irregular motion of single moving particles, e.g. large protein moving in water, where large number of water molecules “pushes” the protein in different directions



- Linked to diffusion (“to spread out”),
- Application to prediction in Stockmarkets

Brownian motion: An approximate discrete model (random walk)



Each bullet (particle) follows a path. Distribution of $X(t)$ approaches the normal distribution

Brownian motion

To generalized the model to continuous time parameter, consider a small time step h and define for real time t:

$$X(t+h) = X(t) + chance$$

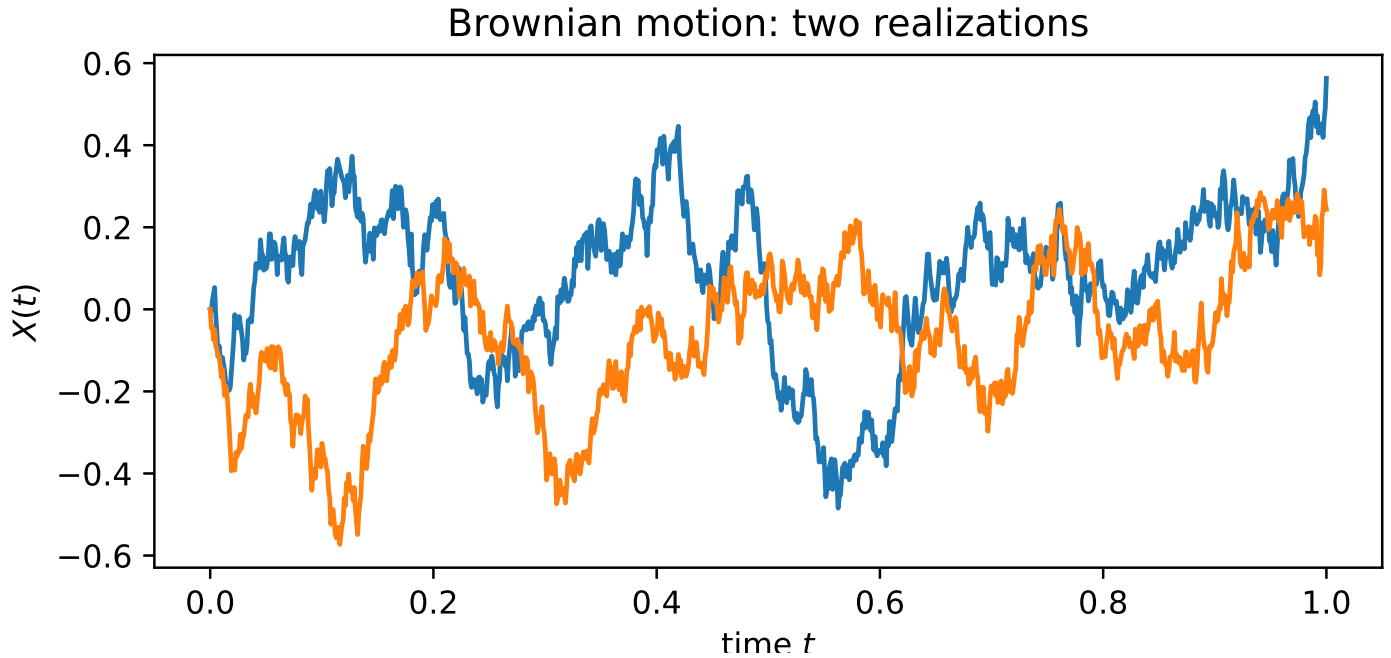
starting with initial state $X(0)$

What is happening in time $t+h$ only depends on step t, i.e. "memoryless". So Brownian motion is a Markov process

- How do we describe $chance$
- From the model we observe
  - Normal distribution
  - "width", variance increases the longer step we take - increases with h
  - Same expected value all the time
- Idea: $chance \sim \mathcal{N}(0,h) = \sqrt{h}\,\mathcal{N}(0,1)$

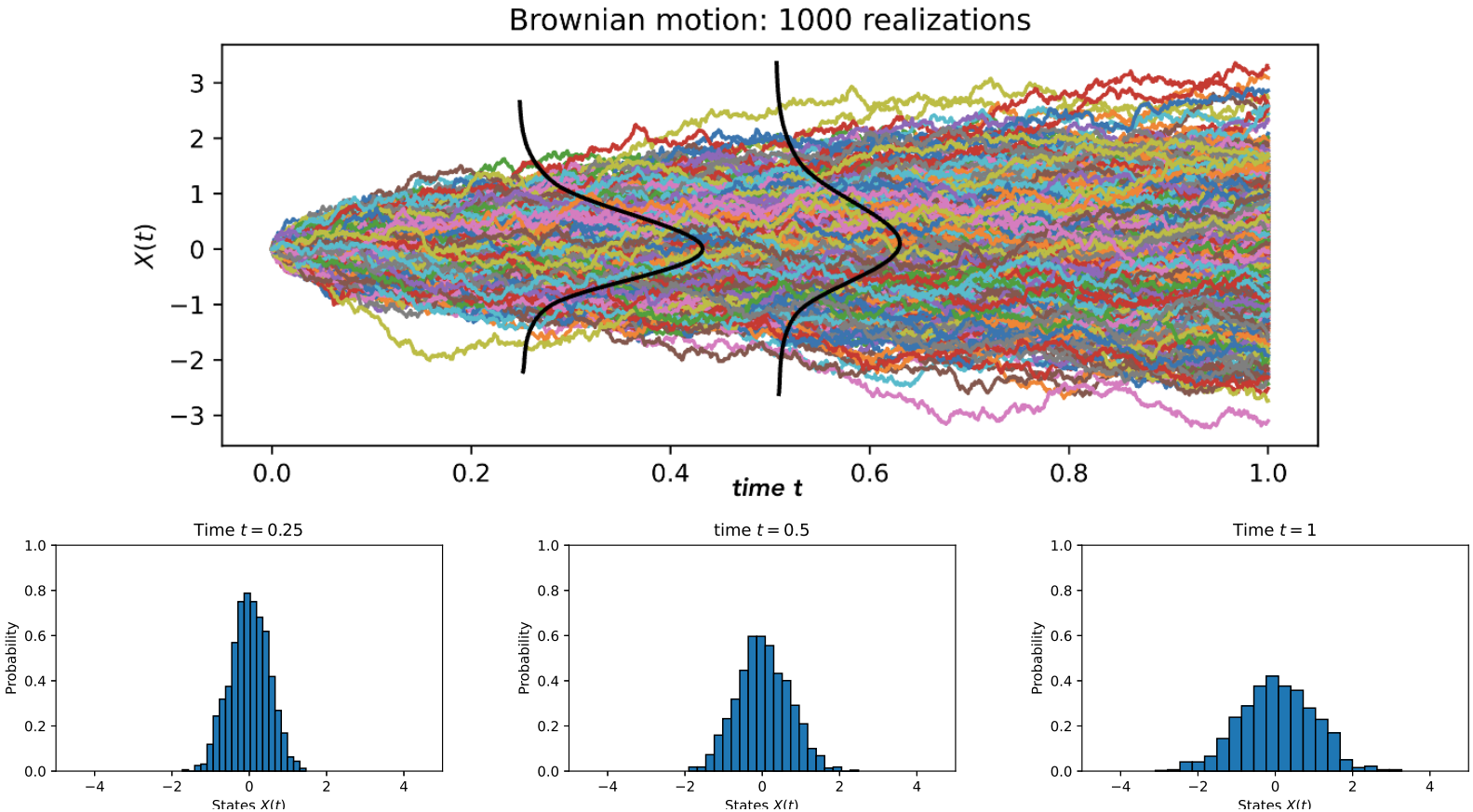
Realizations via Python

```
def BrownMotion(initial_pos, end_time, time_step):
    X = initial_pos
    n, m = initial_pos.shape
    k = 0
    while k*time_step <= end_time:
        chance = np.sqrt(time_step)*np.random.normal(0,1,n)
        new_pos = X[:,-1] + chance
        X = np.column_stack((X,new_pos))
        k += 1
    return X
```



The above code works for higher dimensions as well

Realizations via Python



The distribution of $X(t)$ at any time t is normal with mean $\mu = 0$, the std gets larger as t increases. Indeed $Var(X(t)) = t$ (linear increase in time, diffusion)

Definition of Brownian motion

A more precise definition: A stochastic process $X(t),\ t \geqslant 0$ with following properties is called a Brownian motion:

- $X(0) = 0$
- $X(t)$ is continuous
- $X(t)$ has independent increments
- Increments are normally distributed:

$$X(t) - X(s) \sim \mathcal{N}(0, \sigma^2 = t - s), \quad 0 \leqslant s \leqslant t$$

Higher dimension Brownian motions

- Extension to higher dimensions is straightforward
- $X(t)$ is a d-vector and the $chance$ is normally distributed with mean $\mathbf{0}$ and covariance matrix Σ (multi-dimensional normal dist. $\mathcal{N}(\mathbf{0}, \Sigma)$)
- In a symmetric case Σ is diagonal matrix with same on-diagonals h:

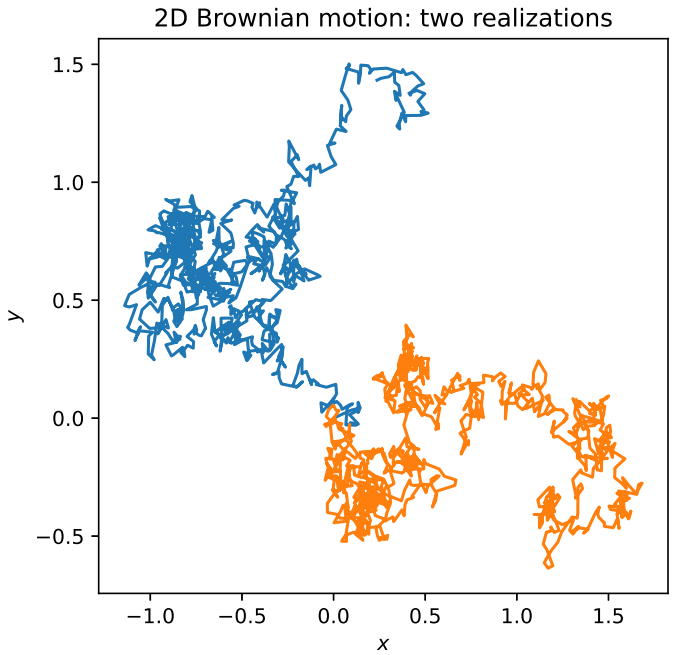
$$X(t+h) = X(t) + \sqrt{h}\,Z, \quad Z \sim \mathcal{N}(\mathbf{0}, I)$$

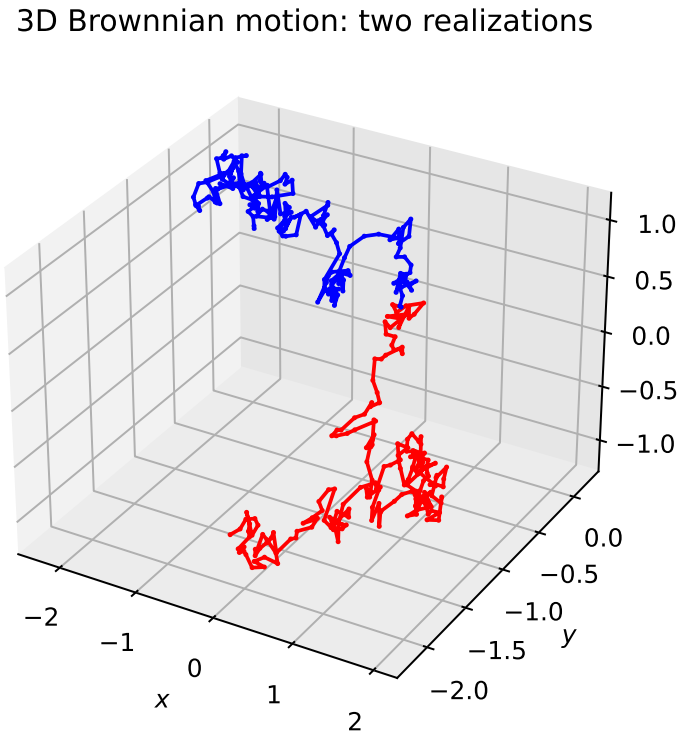
The above code works for all dimensions. We call it for 2 realizations in 2D. The 3D case is similar (see lab exercise 1):

```
# 2D Brownian motion
import matplotlib.pyplot as plt

initial_pos = np.zeros([2,1])
end_time, time_step = 1, 0.001
X = BrownMotion(initial_pos, end_time, time_step)
Y = BrownMotion(initial_pos, end_time, time_step)
plt.plot(X[0,:],X[1,:])
plt.plot(Y[0,:],Y[1,:])
```

Higher dimension Brownian motions: realization using Python

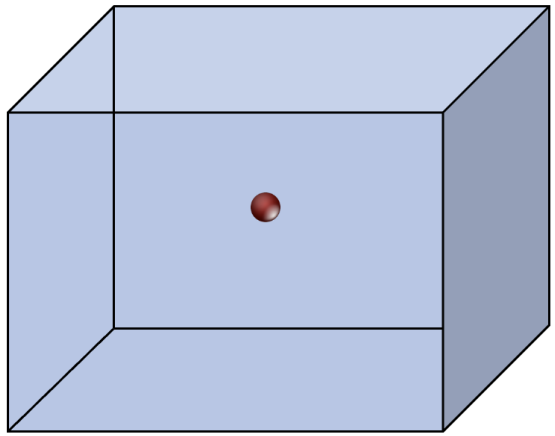




Many applications, e.g. financial math (stocks, options), in physics and chemistry (e.g. particles), biology

Hitting the boundary

- Suppose that we model the motion of a single large protein in surrounding of water molecules and that the system in confined to a box of some fixed sizes.



- We want to know the expected time until the particle hits one of the sides of the box if it starts from the center
- Monte Carlo is used to generate many 3D Brownian motion paths (realizations), say 10^4. For each realization measure and store the time it took until the particle hit one of the sides. Finally compute the average time (mean).