Monte Carlo
What types of problems can we solve with the help of random numbers?

We can compute (potentially) complicated averages:

1. Where does “the average” web surfer end up? (PageRank)
2. How much is my stock portfolio/option going to be worth?
3. What are my odds to win a certain competition?
Random number generators

- Computers are deterministic - operations are reproducible
- How do we get random numbers out of a determinist machine?

Demo “Playing around with random number generators”

- Pseudo-random numbers
  - Numbers and sequences appear random, but they are in fact reproducible
  - Good for algorithm development and debugging

- How truly random are the pseudo-random numbers?
Example: Linear congruential generator

\[ x_o = \text{seed} \]
\[ x_{n+1} = (a \ x_n + c) \ (mod \ M) \]

- If we keep generating numbers using this algorithm, will we eventually get the same number again? Can we define a period?

**Demo “Random numbers”**
Good random number generator

- Random pattern
- Long period
- Efficiency
- Repeatability
- Portability
Random variables

We can think of a random variable $X$ as a function that maps the outcome of unpredictable (random) processes to numerical quantities.

Examples:
- How much rain are we getting tomorrow?
- Will my buttered bread land face-down?

We don’t have an exact number to represent these random processes, but we can get something that represents the average case.

To do that, we need to know how likely each individual value of $X$ is.
Discrete random variables

Each random value $X$ takes values $x_i$ with probability $p_i$

for $i = 1, \ldots, m$ and $\sum_{i=1}^{m} p_i = 1$

Example:

Random variable $X$ = # top of die after each roll

Possible values $x_i$:

$x_1 = 1 \quad \rightarrow \quad p_1 = \frac{1}{6}$

$x_2 = 2 \quad \rightarrow \quad p_2 = \frac{1}{6}$

$\vdots$

$x_6 = 6 \quad \rightarrow \quad p_6 = \frac{1}{6}$
Coin toss example

Random variable $X$: result of a toss can be heads or tails

$x_1 = X = 1$: toss is heads $\rightarrow p_1 = 0.5$

$x_2 = X = 0$: toss is tail $\rightarrow p_2 = 0.5$

**Expected value:** $E(X) = \sum_{i=1}^{m} p_i x_i$

**Roll:** $E(X) = \frac{1}{6}(1) + \frac{1}{6}(2) + \frac{1}{6}(3) + \ldots + \frac{1}{6}(6) = \frac{7}{2}$

**Toss:** $E(X) = \frac{1}{2}(1) + \frac{1}{2}(0) = 0.5$
# Coin toss example

**Numerical experiment:**

<table>
<thead>
<tr>
<th># tosses</th>
<th># heads</th>
<th># heads/N(*)</th>
<th>0.45</th>
<th>0.52</th>
<th>0.54</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>45</td>
<td>0.45</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>52</td>
<td>0.52</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>54</td>
<td>0.54</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(M)</td>
<td>61</td>
<td>0.51</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*\(M\): number of numerical experiments*

- Increase \(M\) ⇒ better estimate for expected value
- Increase \(N\) ⇒ decrease variance

\(\text{(*)expected value}\)
Texas Hold'em Game

**Question**: for each starting pair of cards, what is the probability of winning?

1. **Numerical experimental** → play \( N \) games

**Game**: set of 7 cards

\[ \text{tied, win, or loss} \]
**Texas Holdem Game**

**Question**: for each starting pair of cards, what is the probability of winning?

Starting hand (deterministic variable $S$):

Dealer hand (random variable $D$):

Opponent hand (random variable $O$):

For $i = 1, N$ (games)

generate $D, O$

who win ($S, D, O$) → use poker rules to decide who wins.
Texas Holdem Game

\[ X = \text{Win}(S, O, D) \]

\( X = [1,0,0] \): starting hand wins
\( X = [0,1,0] \): starting hand loses (opponent wins)
\( X = [0,0,1] \): tie

Numerical experiment of \( N=50 \) games

game 1 \( \rightarrow \) \( X = \text{WhoWin}(S, O, D) = [0,1,0] \)
2 \( \rightarrow \)
\( N \rightarrow \)

\[ X = [0,1,0] \]

(\( \oplus \) \[ \#\text{time win}, \#\text{time lose}, \#\text{time tie} \])
Texas Holdem Game

Starting hand: pair of aces

Plotting the number of wins for 100 numerical experiments

\[ N = 50 \text{ games} \]  \[ N = 10,000 \text{ games} \]

- 72% at 36 wins
- 84% at 40 wins
- 96% at 44 wins

- 84% at 8,420 wins
- 85% at 8,450 wins
- 86% at 8,520 wins

\( M = \frac{\text{Number of Wins}}{\text{Number of Occurrences}} \)

Increase \( N \rightarrow \text{reduce variance} \)
Monte Carlo methods

- You just implemented an example of a Monte Carlo method!

- Algorithm that compute APPROXIMATIONS of desired quantities based on randomized sampling

→ Often used to approximate areas/volumes of complicated surfaces.
Example: Approximate the number $\pi$

1 numerical experiment:
- sample $N$ points inside domain
- count # points that are inside circle $\rightarrow N_0$

$A_0 \propto N_0 = N$
$A_0 \propto N_0$

$$r^2 = 4 \frac{N_0}{N}$$

$$A_0 = 4 \frac{N_0}{N}$$

$$(\pi r^2)_{\text{approx}} = 4 \frac{N_0}{N}$$
What can we learn about this simple numerical experiment?

- What is the cost of this numerical experiment? What happens to the cost when we increase the number of sampling points ($n$)?

- Does the method converge? What is the error?

- CONS: Slow convergence rate when using Monte Carlo Methods
- PROS: Efficiency does not degrade with increase in the dimension of the problem (try to modify the demo to approximate the area of an sphere)