

27 Nov 23 - Notes: Monte Carlo Simulations

Monte Carlo Simulations are a class of computational algorithms that rely on repeated random sampling to obtain numerical results. The method is named after the Monte Carlo Casino in Monaco, reflecting its foundational principle of randomness and chance. Initially developed as part of the Manhattan Project during World War II, Monte Carlo simulations have since evolved into a crucial tool in both physics and a myriad of other fields.

Overview of Monte Carlo Simulations

At its core, a Monte Carlo simulation involves using randomness to solve problems that might be deterministic in principle. It's particularly useful for systems with a large number of variables, where traditional analytical methods become impractical.

In physics, Monte Carlo methods are indispensable for studying complex systems where analytical solutions are unattainable. They are widely used in statistical physics, quantum mechanics, and astrophysics, among other areas. For instance, as we will show they can help model the behavior of gases, liquids, and solids at the atomic level, and simulate the evolution of stars and galaxies. In mathematics, these methods offer a way to approximate solutions to complex integrals and differential equations. While our primary focus here is on physics and astronomy, it's worth noting that Monte Carlo methods have been successfully applied in numerous other domains. For instance, in finance, they are used to model and predict stock market behaviors and in risk analysis. In engineering, they help in reliability analysis and optimization problems. Even in healthcare, they assist in radiation therapy planning and epidemiological studies.

How Monte Carlo Simulations Work

We will go into this later in detail

Monte Carlo simulations are grounded in the principle of random sampling and uses the power of statistical analysis to solve problems that, to be honest, they have no business working for. MC sims use randomness to explore a large set of possible solutions to a problem, where direct computation would be impractical or impossible. They then find the most likely outcomes and use them to approximate the true solution.

A key concept for this work is the use of probability distributions. In many physical problems, certain outcomes are more likely than others, and these probabilities can be used to guide the random sampling process. As we started to see with microstates and macrostates – at thermal equilibrium the most probable macrostate is easily determined.

Boltzmann Distribution

For example, in thermal physics, the distribution of particle energies can often be described by the Boltzmann distribution. This distribution tells us that the probability of a particle having a certain energy is proportional to the exponential of the negative of that energy divided by the temperature. This distribution is used in Monte Carlo simulations to determine the probability of a particle having a certain energy. The Boltzmann distribution is given by:

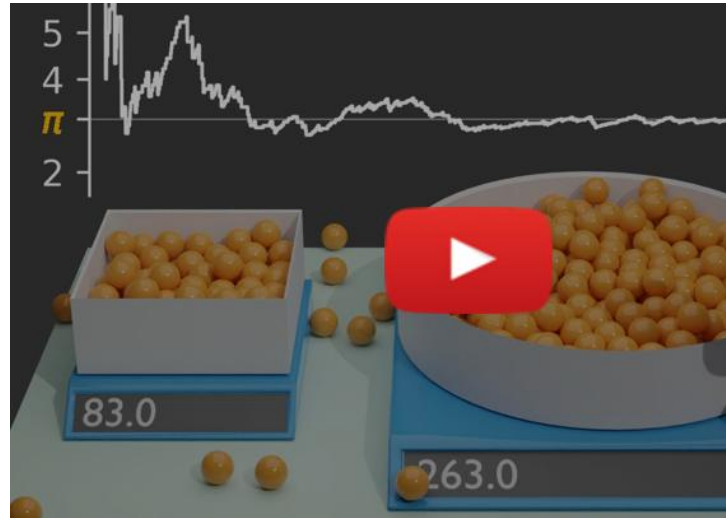
$$P(E) = \frac{e^{-E/kT}}{Z},$$

where E is the energy state, k is the Boltzmann constant, T is the temperature, and Z is the partition function.

Lecture Video

This introduction aims to provide a foundational understanding of Monte Carlo methods; we will delve deeper into the workings, applications, and mathematical aspects of these simulations.

> [Image not embedded: remote images are not included in PDF ex-



port. Check the original file for the image.]

- Non-Commercial Link: <https://inv.tux.pizza/watch?v=7ESK5SaP-bc>
- Commercial Link: <https://youtube.com/watch?v=7ESK5SaP-bc>

Basic Components of a Monte Carlo Simulation

The basic elements of any Monte Carlo Simulation are as follows:

1. **Random Number Generators (RNGs):** These are algorithms that generate sequences of numbers that approximate the properties of random

numbers. RNGs are the backbone of any Monte Carlo simulation. They are used to generate random samples from probability distributions, which are then used to approximate the solution to a problem.

2. **Simulation Algorithms:** An example is the **Metropolis-Hastings algorithm**, widely used in statistical physics. This algorithm decides whether to accept or reject a new state based on a probability criterion, allowing for a detailed exploration of the state space of a system.

In the Metropolis-Hastings algorithm, the probability of moving from a state i to a state j is given by:

$$P(i \rightarrow j) = \min \left(1, \frac{e^{-E_j/kT}}{e^{-E_i/kT}} \right),$$

where E_i and E_j are the energies of states i and j respectively.

Applications in Physics and Astronomy

Monte Carlo simulations are widely used in physics and astronomy, where they help in modeling complex systems and phenomena. Here are some examples:

Statistical Physics

- **Phase Transitions and Critical Phenomena:** Monte Carlo simulations are instrumental in studying phase transitions, like the transition from a ferromagnetic to a paramagnetic state, by allowing for the exploration of large lattice systems.
- **Ising Model of Ferromagnetism:** This model uses a lattice where each site has a spin that interacts with its neighbors. Monte Carlo methods help in simulating and understanding the magnetic properties of materials.

Quantum Mechanics

- **Quantum Monte Carlo Methods:** These are used to study systems of many interacting quantum particles, providing insights into the ground state and excited state properties.
- **Applications in Atomic and Molecular Physics:** Monte Carlo methods help in calculating the properties of atoms and molecules, which are otherwise difficult due to the complexity of quantum interactions.

Astronomy and Astrophysics

- **Stellar Evolution Simulations:** By simulating the life cycles of stars, Monte Carlo methods help in understanding phenomena like supernovae, neutron stars, and black hole formation.

- **Galactic Dynamics and Dark Matter Modeling:** They are used to simulate galaxy formation and the distribution of dark matter in the universe.

Cross-disciplinary Examples

- **Climate Modeling:** Monte Carlo simulations are used to predict climate change by accounting for the numerous variables and uncertainties in climate systems.
- **Biological Systems:** In biology, they assist in modeling complex systems like protein folding and the spread of diseases.

A Common Example: Estimating Pi

The two most common examples of Monte Carlo are:

- **Estimating Pi:** A simple application of Monte Carlo is the estimation of π by randomly placing points in a square and counting how many fall inside a quarter circle.
- **Roulette Simulations:** A basic gambling game like roulette can be simulated to understand probabilities and expected outcomes in games of chance.

To illustrate the basic principles of Monte Carlo simulations, let's look at estimating the value of π . The code below drops dots on a plane and determines if the dots are inside or outside a circle. The ratio of dots inside the circle to the total number of dots is then used to approximate the value of π . The sliders on the widget demonstrate how the accuracy of the approximation improves with the number of dots.

```
import matplotlib.pyplot as plt
import numpy as np
import ipywidgets as widgets
from IPython.display import display

def monte_carlo_pi_plot(num_samples):
    inside_circle = []
    outside_circle = []

    for _ in range(num_samples):
        x, y = np.random.uniform(-1, 1), np.random.uniform(-1, 1)
        if x**2 + y**2 <= 1:
            inside_circle.append((x, y))
        else:
            outside_circle.append((x, y))

    estimated_pi = 4 * len(inside_circle) / num_samples
```

```

# Plotting
fig, ax = plt.subplots()
circle = plt.Circle((0, 0), 1, color='black', fill=False)
ax.add_artist(circle)
ax.set_xlim(-1, 1)
ax.set_ylim(-1, 1)
ax.set_aspect('equal', adjustable='box')
ax.scatter(*zip(*inside_circle), color='blue', s=1)
ax.scatter(*zip(*outside_circle), color='red', s=1)
ax.set_title(f"Monte Carlo Estimation of Pi: {estimated_pi:.4f} (Samples: {num_samples})")
plt.show()

# Create a slider to control the number of samples
slider = widgets.IntSlider(
    value=100,
    min=100,
    max=10000,
    step=100,
    description='Samples:',
    continuous_update=False
)

# Create an interactive widget
widgets.interactive(monte_carlo_pi_plot, num_samples=slider)

interactive(children=(IntSlider(value=100, continuous_update=False, description='Samples:',

```

Additional Resources

Lecture Video

If you want to see more about how Monte Carlo simulations can work, in particular, with respect to data science, check out this video.

> [Image not embedded: remote images are not included in PDF ex-



port. Check the original file for the image.]

- Non-Commercial Link: <https://inv.tux.pizza/watch?v=EaR3C4e600k>
- Commercial Link: <https://youtube.com/watch?v=EaR3C4e600k>