

# Monte Carlo Method and its applications

Xiaoshu Wang

# What is Monte Carlo Method?

- ❖ Monte Carlo methods, or Monte Carlo experiments, are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. The underlying concept is to use randomness to solve problems that might be deterministic in principle. (Wikipedia)
- ❖ Or more frankly speaking, Monte Carlo method can simulate the stochastic properties of a system by constructing a probabilistic model similar to the performance of the system and conducting lots and lots of randomized trials. It is a simulation process.

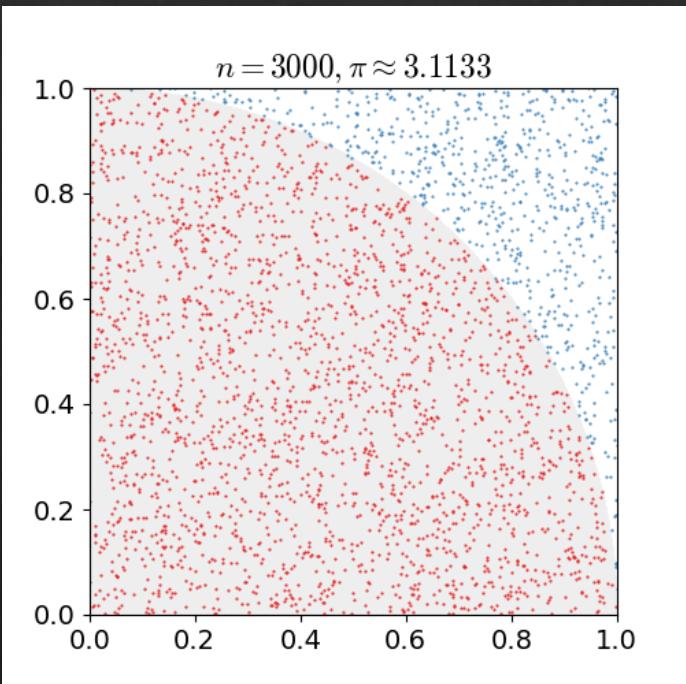
# History

- ❖ Before the Monte Carlo method was developed, simulations tested a previously understood deterministic problem, and statistical sampling was used to estimate uncertainties in the simulations. Monte Carlo simulations invert this approach, solving deterministic problems using probabilistic metaheuristics.
- ❖ An early variant of the Monte Carlo method was devised to solve the Buffon's needle problem, in which  $\pi$  can be estimated by dropping needles on a floor made of parallel equidistant strips.
- ❖ In the late 1940s, Stanislaw Ulam invented the modern version of the Markov Chain Monte Carlo method while he was working on nuclear weapons projects.
- ❖ The theory of more sophisticated mean-field type particle Monte Carlo methods had certainly started by the mid-1960s, Quantum Monte Carlo, and more specifically diffusion Monte Carlo methods can also be interpreted during that period.
- ❖ The use of Sequential Monte Carlo in advanced signal processing and Bayesian inference is more recent.
- ❖ The mathematical foundations and the first rigorous analysis of these particle algorithms were written by Pierre Del Moral in 1996.
- ❖ Now it is very widely used in physics, engineering, biology, applied statistics, artificial intelligence, design, finance, and even in climate change, rescue, and law.

# Why is it so powerful?

- ❖ The traditional empirical methods can not approximate the real process, it is difficult to get satisfactory results, while the Monte Carlo method can realistically simulate the actual process, so the solution of the problem is very consistent with the actual and can get very satisfactory results. It is a computational method based on probability and statistical theory and is a method that uses random numbers (or more commonly pseudo-random numbers) to solve many computational problems. The power of computers eliminates the need for complicated mathematical interpretations and calculations, making them understandable and approachable to most people. Also, it can save a lot of time, improve the efficiency.

# A classic example



- ❖ First, consider a quadrant inscribed in a unit square (with each length of 1).
- ❖ Then uniformly scatter a given number of points over the square (denoted by  $n$  in the GIF).
- ❖ Count the number of points which has a distance from the origin of less than 1.
- ❖ The ratio of number of points in the quadrant and total number of points is the estimation of  $\pi/4$ .
- ❖ As we can see from the GIF, when  $n$  gets bigger, the estimation is closer to  $\pi$ .

(Wikipedia)

# Common approaches

- ❖ 1. Define the domains of inputs
- ❖ 2. Construct or describe a probabilistic process for nonstochastic or stochastic process
- ❖ 3. Randomized sample from the know probabilistic process
- ❖ 4. Perform computation on the inputs to get outputs
- ❖ 5. Aggregate the outputs and make inference

Application in constructing Black-Scholes Model

# I. Background of Wiener Process and Stochastic Differential Equation (SDE)

- ❖ Random Walk: Imagine a particle  $X$  in a planar Cartesian coordinate system, with horizontal axis denoting the time and vertical axis denoting the height of the particle, this particle can either go up or go down by  $y$  magnitude with equal probability, which is  $1/2$ , let  $X(j)$  denote the change in height after  $j^{th}$  motion. Suppose the process has total time  $T$  equally divided by  $k$  interval, we can get some nice properties:

1.  $X(j) = \{y, -y\}$  with probability  $1/2$ ,  $\frac{T}{k} = \Delta t$ ,  $E[X(j)] = 0$ ,  $Var[X(j)] = y^2$

2. Let  $M(k)$  denote the net location of the particle  $X$ , such that

$$M(k) = X(0) + X(1) + \dots + X(k), \text{ and } E[M(k)] = 0, Var[M(k)] = Var[\sum X(k)] = ky^2 = T \frac{y^2}{\Delta t}$$

3. Since  $\frac{y^2}{\Delta t}$  is predefined, to make calculation simpler, we can make it equal to 1, which means that  $y = \sqrt{\Delta t}$  and  $X(j) = \{\sqrt{\Delta t}, -\sqrt{\Delta t}\}$ . And let's make the total time  $T$  finite, this random step function defines a continuous time stochastic process called Wiener Process, denoted by  $W(t)$ .

# I. Background of Wiener Process and Stochastic Differential Equation (SDE)

## ❖ Characterizations of the Wiener process

1.  $W(t)$  has independent increments,  $W(t+u) - W(t)$  ( $u > 0$ ), is independent of all past values  $W(s)$  ( $s < t$ ), such that  $W(t)$  is a martingale with  $W(0) \equiv 0$ .
2.  $[W(t+u) - W(t)]$  is normally distributed with mean 0 and variance  $u$ ,  $W(t+u) - W(t) \sim \mathcal{N}(0, u)$  by the central limit theorem.
3.  $W(t)$  is continuous in time  $t$  but nowhere differentiable in  $t$ .
4.  $dW(t) = W(t+dt) - W(t)$

$$E[dW(t)] = 0$$

$$E[dW(t)dt] = 0$$

$$\text{Var}[dW(t)] = E[dW(t)^2] = dt$$

# I. Background of Wiener Process and Stochastic Differential Equation (SDE)

## ◆ Itô's lemma

1. Since the vertical axis  $W(t)$  is nowhere differentiable in  $t$ , Riemann or Riemann-Stieltjes integral does not work, so we need Itô's lemma to help us.

2. Itô's lemma is given as follow:

$$dV(t) = \frac{\partial V(t)}{\partial t} dt + \frac{\partial V(t)}{\partial X(t)} dX(t) + \frac{1}{2} \frac{\partial^2 V(t)}{\partial^2 X(t)^2} dX(t)^2,$$

where  $V(t)$  is a twice differentiable stochastic function of  $t$  and  $X(t)$ .

3. Example of solving Geometric Brownian Motion,  $dX(t) = \mu X(t)dt + \sigma X(t)dW(t)$ , using Itô:

$$\text{Let } V(t) = \log(X(t)), \frac{\partial V(t)}{\partial t} \equiv 0, \frac{\partial V(t)}{\partial X(t)} = \frac{1}{X(t)}, \frac{\partial^2 V(t)}{\partial^2 X(t)^2} = -\frac{1}{X(t)^2},$$

$$dX(t)^2 = \mu^2 X(t)^2 dt^2 + 2\mu X(t)dt \cdot \sigma X(t)dW(t) + \sigma^2 X(t)^2 dW(t)^2 = \sigma^2 X(t)^2 dt,$$

$$dV(t) = \frac{1}{X(t)} \cdot (\mu X(t)dt + \sigma X(t)dW(t)) - \frac{1}{2X(t)^2} \sigma^2 X(t)^2 dt = \left(\mu - \frac{1}{2}\sigma^2\right) dt + \sigma dW(t).$$

# I. Background of Wiener Process and Stochastic Differential Equation (SDE)

❖ Continued:

From  $dV(t) = \left(\mu - \frac{1}{2}\sigma^2\right)dt + \sigma dW(t)$ , we can take integration of  $dV(t)$  and get

$$\int_t^T dV(u) = V(T) - V(t) = \left(\mu - \frac{1}{2}\sigma^2\right)(T-t) + \int_t^T \sigma dW(u),$$

and in general,  $\int_t^T f(u)dW(u) = \mathcal{N}[0, \int_t^T f(u)^2 dt]$ , so that  $\int_t^T \sigma dW(u) = \mathcal{N}[0, \sigma^2(T-t)]$ ,

$V(T) - V(t) = \left(\mu - \frac{1}{2}\sigma^2\right)(T-t) + (\sigma\sqrt{T-t})\mathcal{N}[0,1]$ , remember  $V(t) = \log(X(t))$ , so

$$X(T) = X(t)e^{(\left(\mu - \frac{1}{2}\sigma^2\right)(T-t) + (\sigma\sqrt{T-t})\mathcal{N}[0,1])}$$

Hence, we find out the relationship between any two arbitrary points in a stochastic process and can perform further analysis.

## II. Black-Scholes Model

- ❖ Black-Scholes model is a mathematical model for observing and predicting the dynamics of a financial market containing investment derivatives using certain assumptions. Typically, it can give theoretical estimate of the price of European options. Today we are going to only discuss Black-Scholes Model for non-dividend stock.
- ❖ Let  $C$  denote the price of a European call option,  $h$  denote the number of shares of the underlying stock,  $B$  denote the cash,  $r$  denote the risk-free rate,  $S(t)$  denote the price of underlying stock at time  $t$ . Stock process follows a geometric Brownian motion.
- ❖ By replication of risk,  $C = hS + B$ ,  $dC = hdS + dB$ , using Itô's lemma, we can get
- ❖  $dC(t) = \frac{\partial C(t)}{\partial t} dt + \frac{\partial C(t)}{\partial S(t)} dS(t) + \frac{1}{2} \frac{\partial^2 C(t)}{\partial^2 S(t)^2} dS(t)^2 = hdS + dB$ , to eliminate risk, set  $h = \frac{\partial C(t)}{\partial S(t)}$ .
- ❖ Black-Scholes PDE is given as  $\frac{\partial C(t)}{\partial t} + \frac{1}{2} \frac{\partial^2 C(t)}{\partial^2 S(t)^2} \sigma^2 S^2 + rS \frac{\partial C(t)}{\partial S(t)} - rC(t) = 0$

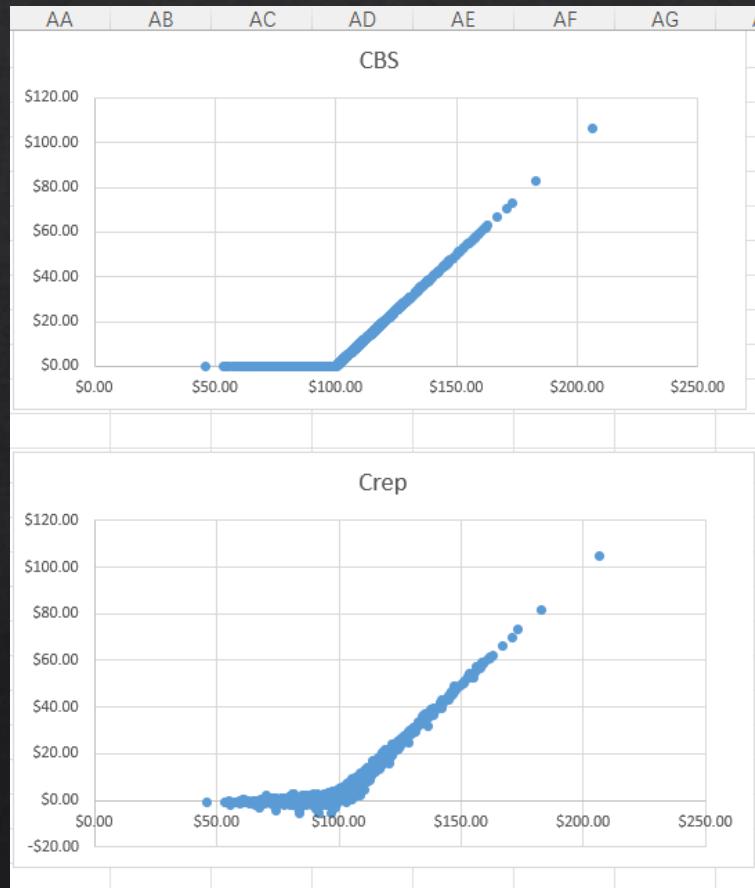
## II. Black-Scholes Model

- ❖ Solving Black-Scholes PDE, we can get the following formula:
- ❖  $C(t) = S(t)\Phi[d_1] - Ke^{-r(T-t)}\Phi[d_2], \quad d_1, d_2 = \frac{\log\left(\frac{S(t)}{K}\right) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}},$
- ❖  $h = \frac{\partial C(t)}{\partial S(t)} = \Phi[d_1]$  in the non-dividend case,  $h$  is also known as  $\Delta$  in option Greeks.
- ❖ where  $K$  is the strike price of underlying call option,  $\Phi[\cdot]$  is the cumulative distribution function (cdf) of standard normal distribution,  $\sigma$  is the volatility of the underlying stock, and  $h$  is the number of shares you are holding.
- ❖ And we can replicate the call option by borrowing cash at risk free rate and buy certain shares of the stock.
- ❖ Till now, the stochastic process is fully described, and we can define our inputs and perform Monte Carlo Simulation.

# III. Monte Carlo Simulation

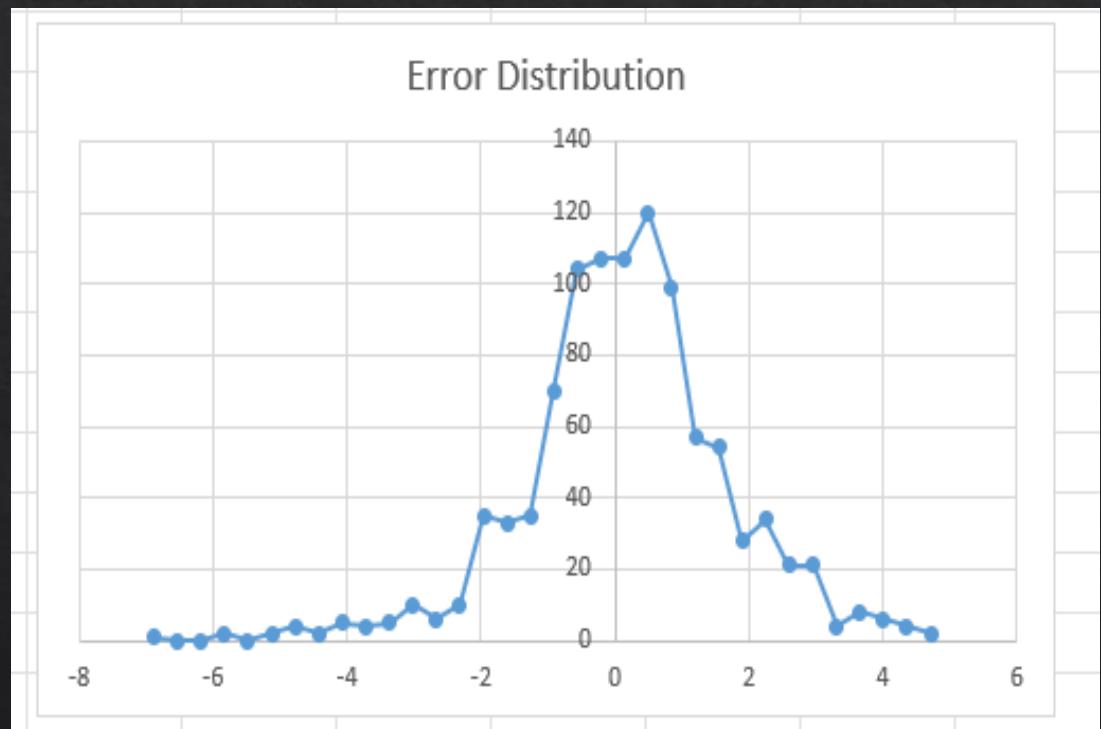
### III. Monte Carlo Simulation

- Here we have generated 1000 paths and plot the theoretical earnings and actual earnings. We can see that the earnings of our replication portfolio highly coincides with the result from theoretical model.



### III. Monte Carlo Simulation

- ❖ Here is a plot of the error distribution, vertical axis denotes the frequency and horizontal axis denotes the error between theoretical model and simulation. As we can see that, the error distribution approximately follows a normal distribution with mean 0, so we can trade more often to reduce the variance of error distribution.
- ❖ So, by using Monte Carlo Simulation, we can not only make sure that our model is convergent and implementable, but also use large number of trials to perform prediction.



### III. Monte Carlo Simulation

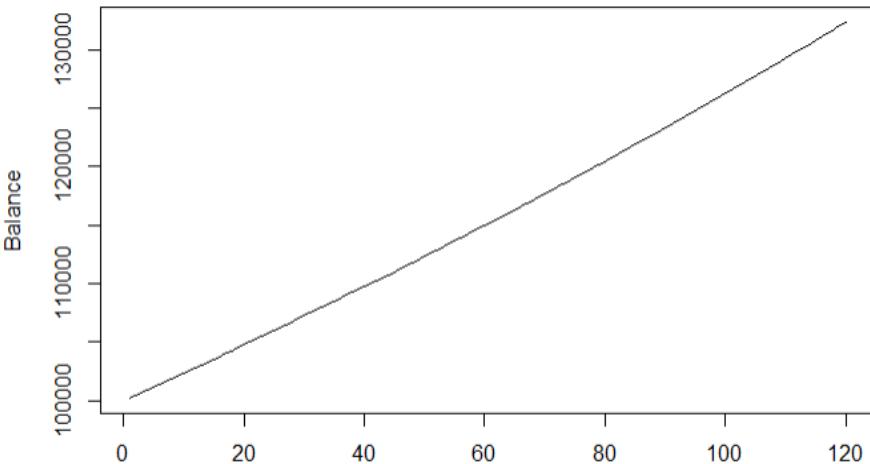
```
87
88 ```{r model3}
89 nYears <- 10
90 nScenarios <- 10000
91 nPeriods <- 12
92 initialBalance <- 100000
93 capRate <- 1
94 floorRate <- -1
95
96 # Set up arrays to hold results
97 scenarioBalances <- matrix(NA, nScenarios, nYears * nPeriods)
98 avgScenarioBalances <- rep(NA, nYears * nPeriods)
99
100 # Loop over scenarios and periods
101 for (i in 1:nScenarios) {
102   balance <- initialBalance
103   for (j in 1:(nYears * nPeriods)) {
104     # Calculate the monthly rate of return
105     r <- rnorm(1, mean = 0.0329, sd = 0.1010)
106
107     # calculate the cap and floor rates for the period
108     cap <- ifelse(r > capRate, capRate, r)
109     floor <- ifelse(r < floorRate, floorRate, r)
110
111     # calculate the monthly return
112     monthlyReturn <- (1 + cap)^(1/12) - 1
113     monthlyFloor <- (1 + floor)^(1/12) - 1
114
115     # calculate the new balance for the period
116     balance <- balance * (1 + monthlyReturn)
117
118     # Apply the floor rate if necessary
119     if (monthlyReturn < monthlyFloor) {
120       balance <- balance * (1 + monthlyFloor)
121     }
122
123     # Store the balance for the period
124     scenarioBalances[i, j] <- balance
125   }
126 }
```

# III. Monte Carlo Simulation

```
127  
128 # calculate the average scenario balance for each period  
129 for (i in 1:(nYears * nPeriods)) {  
130   avgScenarioBalances[i] <- mean(scenarioBalances[, i])  
131 }  
132 avgrate<-(avgScenarioBalances[i]/initialBalance)^(1/nYears)-1  
133 avgrate  
134 avgScenarioBalances[i]  
135 # Plot the results  
136 plot(1:(nyears * nPeriods), avgScenarioBalances, type = "l", xlab = "Time (months)", ylab = "Balance", main = "Fixed Indexed Annuity with Cap and Floor Rates")  
137 ``
```



Fixed Indexed Annuity with Cap and Floor Rates



# IV. Advantage and Disadvantage

- ❖ Disadvantages:
  - ❖ Monte Carlo simulation may require lots of iterations and the algorithm could be computationally intensive and time-consuming
  - ❖ Highly relied on the goodness of assumptions of inputs, the model could be wrong at the beginning
  - ❖ May not reach convergence
- ❖ Advantages:
  - ❖ The error has nothing to do with dimensions
  - ❖ Can easily handle continuous problem
  - ❖ Can simulate complicated situations that cannot be modeled using other methods

# References

- ❖ [https://en.wikipedia.org/wiki/Monte\\_Carlo\\_method](https://en.wikipedia.org/wiki/Monte_Carlo_method)
- ❖ [https://en.wikipedia.org/wiki/Black–Scholes\\_model](https://en.wikipedia.org/wiki/Black–Scholes_model)

# Q&A