Monte Carlo and Derivatives Pricing

Jacob Steen

Rensselaer Polytechnic Institute

Introduction

Financial markets are often perceived as unpredictable and chaotic, however, beneath the surface lies a complex interplay of patterns and randomness. Modeling the dynamics of asset prices is a true reflection of this randomness and is the cornerstone of understanding and predicting market behaviors, quantifying risk, pricing derivatives and optimizing portfolios. *Numerical Methods in Finance* by Paolo Brandimarte takes a dive into what is hidden below the surface of these financial securities, explaining techniques in simulation of financial data and how to use these results to price vanilla and exotic financial derivatives.

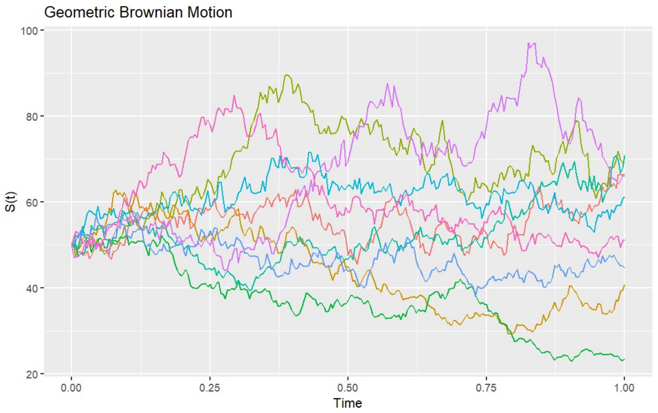
Asset Dynamic Models

Monte Carlo simulation is a cornerstone technique in quantitative finance, providing a robust framework for pricing both vanilla and exotic derivatives. This approach uses stochastic calculus to model asset prices and estimate derivative values, leveraging the inherent randomness of financial markets. At its core, Monte Carlo methods simulate numerous potential paths for the underlying asset, guided by the dynamics of random walks described by stochastic differential equations (SDEs).

The evolution of asset prices under the risk-neutral measure, Q, is typically modeled by the geometric Brownian motion (GBM), represented as:

$$dS_t = rS_t dt + \sigma S_t dW_t$$

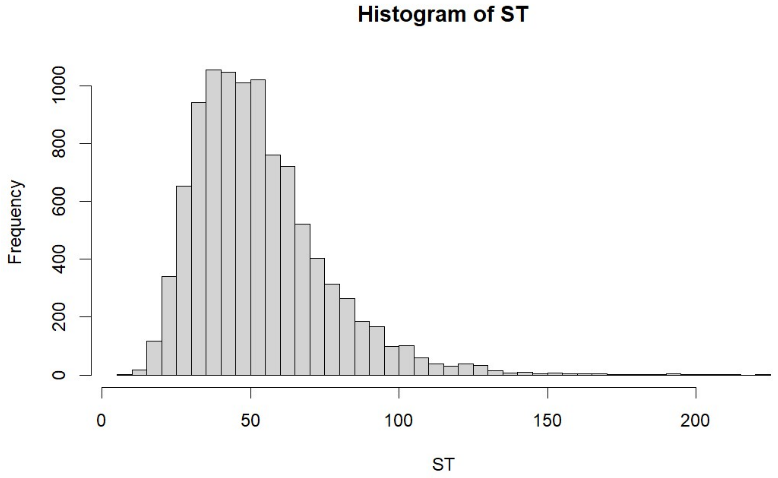
where S_t is the asset price at time t, r is the risk-free rate, σ is the volatility, and W_t denotes a Wiener process (Brownian motion). By discretizing this equation using schemes like Euler-Maruyama or Milstein, the Monte Carlo method simulates a large ensemble of price paths to estimate the expected payoff of the asset price. An example of these paths can be seen below.



We can use this Geometric Brownian Motion to price vanilla and more complicated exotic derivatives.

Vanilla Option Pricing Using Monte Carlo

Using Monte Carlo simulation to price options is straightforward but requires careful implementation to avoid complications. The process involves simulating potential asset price paths, calculating the option payoffs based on these paths, and then averaging the payoffs. This average is subsequently discounted at the risk-free rate to determine the derivative's price. In the following example, the value of a European call and put option will be computed using the simulated asset price paths described earlier. Start by assuming the call is at-the-money, the asset has a price of $50, volatility equal to 40% annually, and the risk-free rate is 5%. We can begin to price this option by simulating the terminal price of the asset. Below, a histogram displays the simulated terminal asset price.



By discounting the mean payoff at the risk-free rate, we can determine the price of a European call option. For this hypothetical asset, the fair market value of the European call is calculated to be $8.95. Using a 95% confidence level, the fair value range is bounded between $8.65 and $9.25. When compared to the Black-Scholes-Merton pricing model, the results are consistent. The Black-Scholes model assigns a value of $9.01 for the call option and $6.57 for the corresponding put option, demonstrating the alignment of Monte Carlo simulation with a more established pricing method.

Pricing Path-Dependent Options Using Monte Carlo

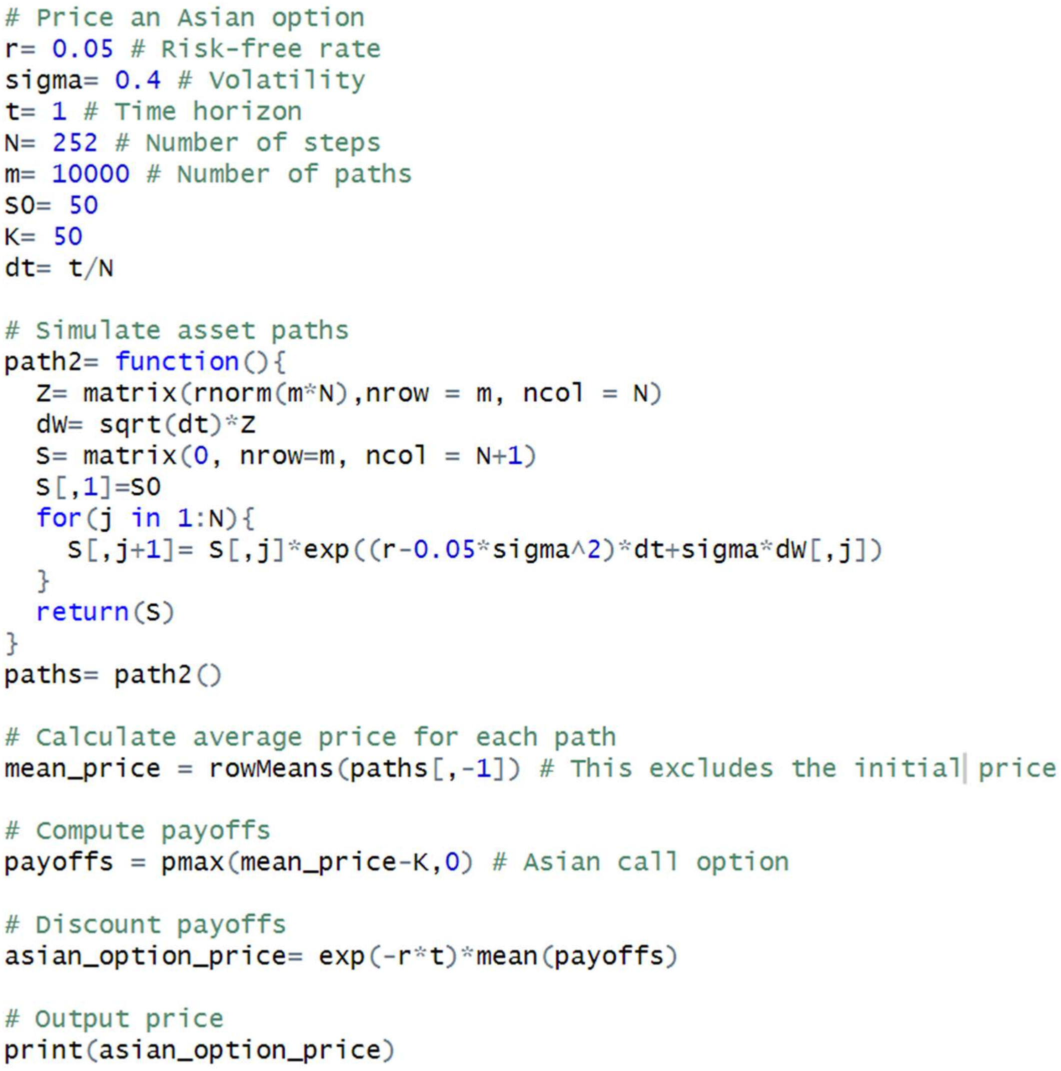
There are a few kinds of path-dependent options which are utilized by practitioners; however, the focus will be turned towards pricing Asian options using Monte Carlo methods. Asian options are an exotic derivative whose payoff depends on the mean price of an underlying asset during the life of the option. Its payoff is defined by

$$h(S) = [\bar{S} - K]^{+}$$

where,

$$\bar{S} = \frac{1}{N}\sum_{i=1}^{N} S_i$$

This shows that $\bar{S}$ is defined by the mean closing price over a specified period. For example, this could denote the average weekly closing price over the life of the contract. Using the same parameters from before, we find that an at-the-money Asian option is $6.30. This is found using the code below.



```r
# Price an Asian option
r= 0.05 # Risk-free rate
sigma= 0.4 # Volatility
t= 1 # Time horizon
N= 252 # Number of steps
m= 10000 # Number of paths
S0= 50
K= 50
dt= t/N

# Simulate asset paths
path2= function(){
  Z= matrix(rnorm(m*N),nrow = m, ncol = N)
  dW= sqrt(dt)*Z
  S= matrix(0, nrow=m, ncol = N+1)
  S[,1]=S0
  for(j in 1:N){
    S[,j+1]= S[,j]*exp((r-0.05*sigma^2)*dt+sigma*dW[,j])
  }
  return(S)
}
paths= path2()

# Calculate average price for each path
mean_price = rowMeans(paths[,-1]) # This excludes the initial price

# Compute payoffs
payoffs = pmax(mean_price-K,0) # Asian call option

# Discount payoffs
asian_option_price= exp(-r*t)*mean(payoffs)

# Output price
print(asian_option_price)
```