Monte Carlo Stock Price Simulation: Mathematical Foundation and Implementation

Sandy Herho

March 27, 2025

1 Geometric Brownian Motion: Mathematical Derivation

We derive the stock price model directly from first principles, following the implementation in the provided code.

1.1 Basic Assumptions

Stock price modeling assumes:

1. Returns scale proportionally with the current price (multiplicative process)
2. Prices have a long-term growth trend (drift)
3. Random fluctuations are proportional to price level (volatility)

1.2 Step-by-Step Derivation

1.2.1 Relative Price Change

For a small time interval dt, we model the relative price change as:

$$\frac{dS(t)}{S(t)} = \mu dt + \sigma dW(t) \tag{1}$$

Where:

- $S(t)$ is the stock price at time t

- μ is the expected return (drift)
- σ is the volatility
- $dW(t)$ is an increment of a standard Wiener process

Rearranging to get the absolute price change:

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t) \tag{2}$$

Equation (2) is the Stochastic Differential Equation (SDE) for Geometric Brownian Motion.

1.2.2 Applying Itô's Lemma

To solve the SDE in (2), we apply Itô's lemma to the logarithm of the stock price, $X(t) = \ln(S(t))$.

For a function $f(S,t) = \ln(S)$, the partial derivatives are:

$$\frac{\partial f}{\partial t} = 0 \tag{3}$$

$$\frac{\partial f}{\partial S} = \frac{1}{S} \tag{4}$$

$$\frac{\partial^2 f}{\partial S^2} = -\frac{1}{S^2} \tag{5}$$

By Itô's lemma:

$$dX(t) = \frac{\partial f}{\partial t}dt + \frac{\partial f}{\partial S}dS(t) + \frac{1}{2}\frac{\partial^2 f}{\partial S^2}(dS(t))^2 \tag{6}$$

For the Wiener process, $(dW(t))^2 = dt$, so $(dS(t))^2 = \sigma^2 S(t)^2 dt$.

Substituting into equation (6):

$$\begin{aligned} dX(t) &= 0 + \frac{1}{S(t)}[\mu S(t)dt + \sigma S(t)dW(t)] + \frac{1}{2}\left(-\frac{1}{S(t)^2}\right)\sigma^2 S(t)^2 dt && (7) \\ &= \mu dt + \sigma dW(t) - \frac{1}{2}\sigma^2 dt && (8) \\ &= \left(\mu - \frac{1}{2}\sigma^2\right)dt + \sigma dW(t) && (9) \end{aligned}$$

1.2.3 Finding the Solution

Integrating equation (9) from 0 to t:

$$X(t) - X(0) = \left(\mu - \frac{1}{2}\sigma^2\right) t + \sigma W(t) \tag{10}$$

$$\ln\left(\frac{S(t)}{S(0)}\right) = \left(\mu - \frac{1}{2}\sigma^2\right) t + \sigma W(t) \tag{11}$$

Taking the exponential of both sides:

$$S(t) = S(0) \exp\left[\left(\mu - \frac{1}{2}\sigma^2\right) t + \sigma W(t)\right] \tag{12}$$

Equation (12) provides the analytical solution for the stock price at any future time.

1.3 Discrete Time Implementation

For practical simulation with time step Δt, we use:

$$S(t + \Delta t) = S(t) \exp\left[\left(\mu - \frac{1}{2}\sigma^2\right) \Delta t + \sigma\sqrt{\Delta t} Z\right] \tag{13}$$

Where $Z \sim \mathcal{N}(0, 1)$ is a standard normal random variable. This is the formula directly implemented in the code.

2 Monte Carlo Simulation Algorithm

The core implementation consists of generating multiple sample paths using equation (13) and analyzing the resulting distribution.

Algorithm 1 Monte Carlo Stock Price Simulation

procedure STOCKMONTECARLO(S_0, μ, σ, N, M)
    Input:
        S_0: Initial stock price
        μ: Daily expected return (drift)
        σ: Daily volatility
        N: Number of days to simulate
        M: Number of simulation paths
    Initialize results array
    for $i \leftarrow 1$ to M **do**
        Initialize prices = $[S_0]$
        for $d \leftarrow 1$ to N **do**
            $Z \leftarrow$ Random sample from $\mathcal{N}(0, 1)$
            $nextPrice \leftarrow prices[-1] \cdot \exp((\mu - 0.5\sigma^2) + \sigma \cdot Z)$
            Append $nextPrice$ to prices
        end for
        Append prices to results
    end for
    Calculate summary statistics:
    meanPath $\leftarrow$ Mean of all paths at each time point
    medianPath $\leftarrow$ Median of all paths at each time point
    for each desired percentile p **do**
        $pPath \leftarrow p$-th percentile of all paths at each time point
    end for
    return results, mean, median, percentiles
end procedure

3 Visualization Strategy

The implemented code uses a two-panel visualization approach:

Algorithm 2 Visualization of Simulation Results

procedure PLOTSIMULATIONS(simulation_data, S_0, μ, σ)
    Create figure with two subplots
    In first subplot:
    Plot all simulations with low opacity (0.02)
    Plot mean trajectory in red
    Plot median trajectory in green
    Add 90% confidence band (filling between 5th and 95th percentiles)
    Add 50% confidence band (filling between 25th and 75th percentiles)
    In second subplot:
    Plot 90% range (95th percentile - 5th percentile) over time
    Plot interquartile range (75th percentile - 25th percentile) over time
    Add appropriate labels, titles, and formatting
end procedure

This visualization approach effectively shows:

1. The entire range of possible future paths
2. The central tendency (mean and median)
3. The uncertainty band at different confidence levels
4. How uncertainty grows over time (spread visualization)

4 Key Implementation Considerations

4.1 Parameter Selection

- **Initial Price (S_0):** Starting stock price, set to 50 in the code
- **Daily Drift (μ):** Expected daily return, set to 0.0002 (about 5% annual return)
- **Daily Volatility (σ):** Set to 0.01 (about 16% annual volatility)
- **Simulation Period:** Set to 252 days (one trading year)
- **Number of Simulations:** Set to 10,000 for statistical robustness

4.2 Performance Considerations

The code includes optimizations:

- Pre-allocation of directories for output
- Efficient data structure (DataFrame) for handling simulation results
- Vectorized calculations for summary statistics
- High-resolution plotting (300 DPI) with appropriate transparency levels

4.3 Model Limitations

Important caveats about the implementation:

1. Assumes constant volatility over time
2. Assumes log-normal distribution of returns
3. Does not account for sudden market jumps or regime changes
4. Sensitive to input parameter selection
5. Cannot predict structural market changes

5 Conclusion

The implemented Monte Carlo simulation provides a robust framework for exploring the probabilistic nature of future stock prices. It generates multiple possible price paths based on the Geometric Brownian Motion model, calculates key statistical measures, and visualizes the results with appropriate uncertainty bands.

While the model has limitations, it effectively captures the essence of stock price uncertainty: returns that scale with price, long-term growth trends, and increasing uncertainty over longer time horizons. These characteristics make it suitable for various financial planning and risk assessment applications.