Computational Finance II: Monte Carlo Methods

Author: Sven Karbach

0.0| Introduction

In the second part of our computational finance course, we explore the use of Monte Carlo methods for derivative pricing. Monte Carlo methods are highly regarded in the financial industry for their flexibility (and in some cases their effectiveness) across a broad spectrum of applications. These applications are not limited to derivative pricing but also include areas such as risk and portfolio management. A key advantage of Monte Carlo methods is their ability to efficiently manage high-dimensional settings, such as those involving large portfolios of stocks or derivatives contingent on multiple underlying assets, with relatively few model assumptions. Our emphasis on derivative pricing will lead us to operate within the risk-neutral framework. This approach is distinct from the methods used in risk and portfolio management, which often rely on the physical probability measure to reflect the actual likelihood of events and outcomes. The question which probability measure to use is not a mere theoretic consideration, but it has a profound implication on how we incorporate market data into a pricing model (through calibration) and the distributions we use for sampling (risk-neutral probabilities).

0.1| Financial Markets

We consider a continuous-time **financial market** with time-horizon $[0, T]$ and $d \in \mathbb{N}$ assets. A financial market is described by a tuple $(\Omega, \mathcal{F}, \mathbb{P}, \mathbb{F}, \mathbf{S})$, where:

- Ω represents the set of all **outcomes** $\omega \in \Omega$,
- $\mathcal{F}$ is a σ-algebra on Ω containing all **events** $A \in \mathcal{F}$,
- $\mathbb{P}$ denotes the **physical probability measure** defined on the measurable space $(\Omega, \mathcal{F})$,
- $\mathbb{F} = (\mathcal{F}_t)_{t\in[0,T]}$ is a filtration, i.e. $\mathcal{F}_t \subseteq \mathcal{F}$ for $t \in [0, T]$ being a sub-σ-algebra such that $\mathcal{F}_s \subseteq \mathcal{F}_t$ for all $0 \leq s \leq t \leq T$, and describes the flow of information in the market,
- $\mathbf{S} = (S_t)_{t\in[0,T]} = (S_t^{(1)}, \dots, S_t^{(d)})_{t\in[0,T]}$ denotes a $\mathbb{R}^d$-valued stochastic process on $[0, T]$, describing the time-evolution of some $d \in \mathbb{N}$ asset prices.

Remark on Financial Markets: When modeling time-dependent random phenomena, the natural chronological order implied by the flow of time must be respected. This means that

an event observed yesterday reveals information that we still remember today, but today we do not necessarily know what will happen tomorrow. In stochastics, this chronological ordering of information is represented by a filtration $\mathbb{F} = (\mathcal{F}_t)_{t\in[0,T]}$. Usually, we assume that:

- $\mathcal{F}_0$ is $\mathbb{P}$-trivial, i.e., $\mathbb{P}(A) \in \{0,1\}$ for all $A \in \mathcal{F}_0$, i.e., all $\mathcal{F}_0$-measurable random variables are constant $\mathbb{P}$-almost surely,
- $\mathcal{F} = \mathcal{F}_T$, i.e., all events occur in the finite time horizon.

An $\mathbb{R}^d$-valued **stochastic process** $(S_t)_{t\in[0,T]}$ is any family of $\mathbb{R}^d$-valued random variables $\{S_t : t \in [0,T]\}$ defined on $(\Omega, \mathcal{F}, \mathbb{P})$. Often, we denote a stochastic process simply by $\mathbf{S}$.

To give meaning to time in this framework, we must relate the stochastic process with the flow of information, that is, the filtration $\mathbb{F} = (\mathcal{F}_t)_{t\in[0,T]}$. A stochastic process $\mathbf{S}$ is said to be **adapted** to $\mathbb{F}$ if S_t is $\mathcal{F}_t$-measurable for all $t \in [0,T]$, i.e. given the information in $\mathcal{F}_t$ we know what the price S_t is.

0.2| Asset price processes and sample paths

Consider an asset price process $\mathbf{S} = (S_t)_{t\in[0,T]} = (S_t^{(1)}, \ldots, S_t^{(d)})_{t\in[0,T]}$. In case of a one-asset market $d = 1$, we simply write $(S_t)_{t\in[0,T]}$ instead of $(S_t^{(1)})_{t\in[0,T]}$.

We call $\omega \mapsto S_t(\omega)$ a **sample at time** t and $t \mapsto S_t(\omega)$ a **sample path** of $(S_t)_{t\in[0,T]}$.

A sample $\omega \mapsto S_t(\omega)$ is **distributed** according to the model specification. For some models, the distribution is known in closed form, e.g., in the Black-Scholes Model we have:

$$ln(S_t) \sim \mathcal{N}\big(ln(S_0) + (\mu - \frac{\sigma^2}{2})t,\ \sigma^2 t\big).$$

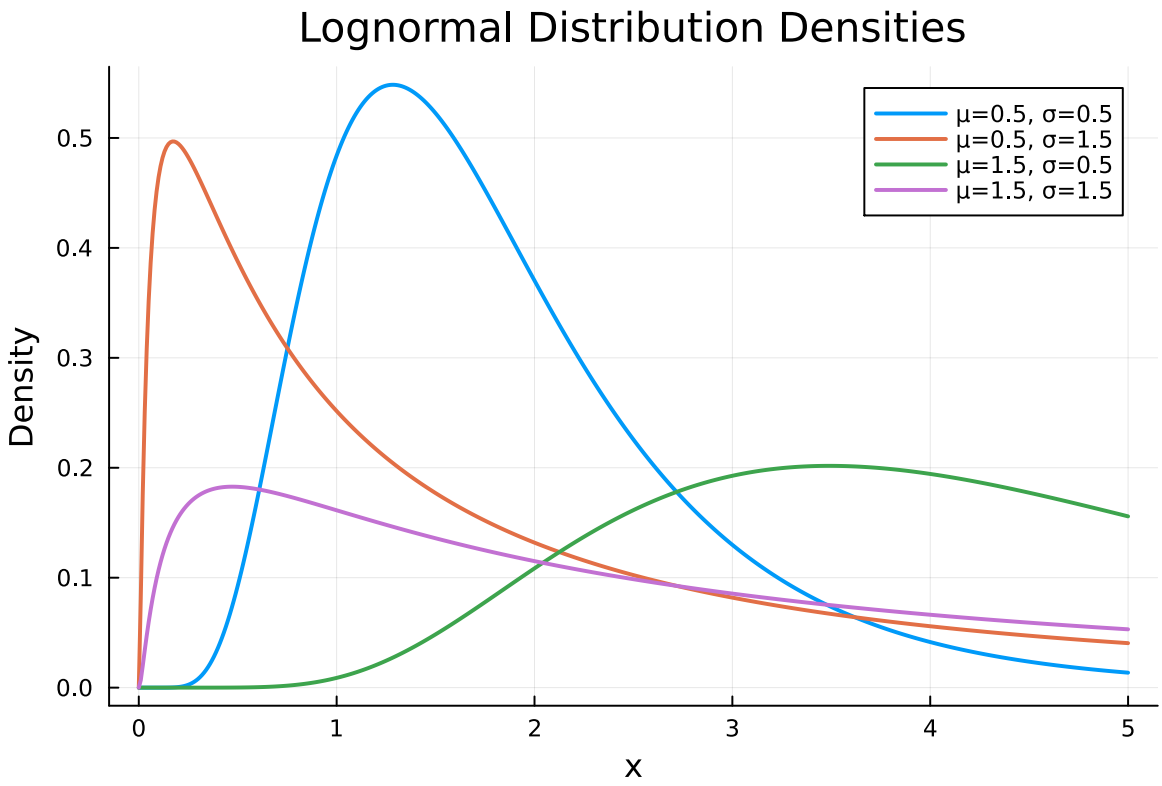
Note: How to sample from the normal distribution is assumed to be known from the course "Distributed Stochastic Simulations".

In [1]:
```
using Distributions
using Plots
# Define the range of x values for plotting
x = 0:0.01:5
# Parameters: arrays of μ (location) and σ (scale) values
mus = [0.5, 1.5]  # Location parameters
sigmas = [0.5, 1.5]  # Scale parameters
# Initialize the plot
# Generate and plot each distribution
```

```
2-element Vector{Float64}:
 0.5
 1.5
```

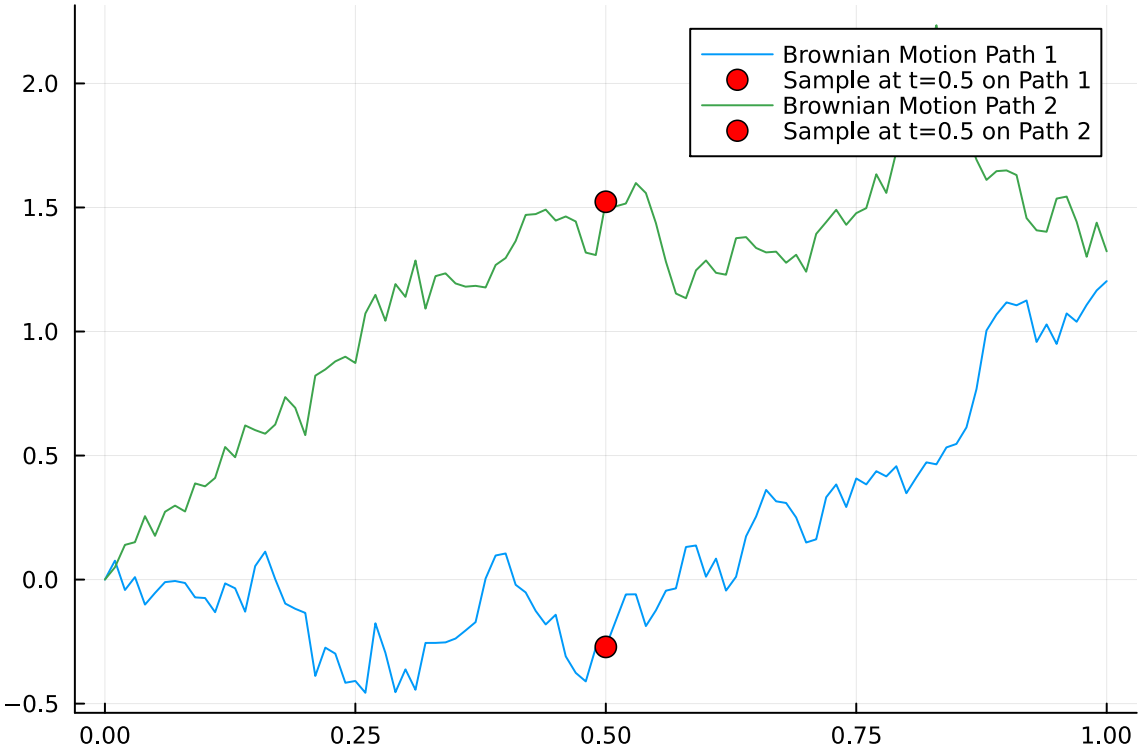
In [2]:
```
plot()
for μ in mus
```

```
    for σ in sigmas
        # Create a lognormal distribution with current μ and σ
        dist = LogNormal(μ, σ)
        # Plot the density function
        plot!(x, pdf.(dist, x), label="μ=$μ, σ=$σ", lw=2)
    end
end
plot!(title="Lognormal Distribution Densities", xlabel="x", ylabel="Density"
```



In [4]:

```
# Generate two paths of Brownian motion
W1 = generate_brownian_motion(N, dt)
W2 = generate_brownian_motion(N, dt)
# Fixed time point to sample
fixed_time = 0.5
fixed_index = Int(floor(fixed_time / dt)) + 1
plot(t, W1, label="Brownian Motion Path 1", legend=:topright)
scatter!([fixed_time], [W1[fixed_index]], label="Sample at t=0.5 on Path 1",
plot!(t, W2, label="Brownian Motion Path 2")
scatter!([fixed_time], [W2[fixed_index]], label="Sample at t=0.5 on Path 2",
```



0.3| Risk-Neutral Pricing

Our principal goal is to determine a **fair price** of an option at time $t = 0$ that pays at a future exercise date $t \in (0, T]$ an amount $H \geq 0$. The timepoint T is refered to as the **expiry date** of the option and H is called the **pay-off function**. If $t = T$, we call the option **European** and if $t \in (0, T]$ is arbitrary we call the option **American**.

Remarks:

- The pay-off function H is always non-negative. However, it is random as it is *derived* from the value of an underlying asset price process $\mathbf{S}$, which itself is uncertain. Often H depends only on the value of $\mathbf{S}$ at the exercise time t, then we migh write $H(\mathbf{S}_t)$.
- If H depends on the path of $\mathbf{S}$, i.e. on $S_{t_1}, \ldots, S_{t_n}$ for $t_1, \ldots, t_n \in [0, T]$ and $n \in \mathbb{N}$, we call the option **path dependent**.
- Throughout this lecture, we assume that the randomness of H only comes through the dependence on the underlying asset prices.

Example 0.1: Pay-off functions of European type options

- **European "Plain Vanilla" Put Option:** $H(\mathbf{S}) \triangleq H(S_T) \triangleq (K - S_T)^+$ with strike $K > 0$,
- **Asian Call Option:** $H(\mathbf{S}) \triangleq \left(\frac{1}{T} \sum_{i=1}^{M} S_{t_i} - K \right)^+$ with fixed strike $K > 0$ and discretely monitored arithemtic average,

- **Spread Call Option:** $H(\mathbf{S}) \triangleq H(S_T^1, S_T^2) \triangleq (S_T^1 - S_T^2 - K)^+$ with strike $K > 0$ written on two asset prices $(S_t^1)_{t\in[0,T]}$ anf $(S_t^2)_{t\in[0,T]}$,
- **Digital Barrier Option:** $H(\mathbf{S}) \triangleq \mathbf{1}_{\{S_t \geq B \text{ for some } t\in[0,T]\}}$ with Barrier $B > 0$.

Q1: At time T the price of an option is known to be $H(\mathbf{S})$, but what is its fair-price π_0 at $t = 0$?

It turns out (this is a deep result from mathematical finance), that in a financial market **free of arbitrage** the price of a European derivative is given as the expectation:

$$\pi_0 = \mathbb{E}_{\mathbb{Q}}[H(\mathbf{S})]$$

For an American derivative one has to take the supremum over all stopping times τ such that:

$$\pi_0 = \sup_{\tau<T} \mathbb{E}_{\mathbb{Q}}[H(\mathbf{S}_\tau)],$$

where we assume here and throughout the lecture that the discount-factor is trivial, i.e. the **risk-free rate** is zero ($r = 0$).

The expectations encountered in option pricing problems must be determined under the so called **risk-neutral measure** $\mathbb{Q}$ and we call the price at time $t = 0$ the **risk-neutral price**!

In the following, we assume that H is the pay-off of an European option and we denote it as $H(\mathbf{S})$ to emphasize its dependence on the underlying assets price process. Similarly, for the risk neutral price today we shall write:

$$\pi_0(H, \mathbf{S}) := \mathbb{E}_{\mathbb{Q}}[H(\mathbf{S})].$$

Example 0.2: European Call Option in Black-Scholes model

Consider the Black-Scholes model with drift and zero interest rate given by:

$$\mathrm{d}S_t = \mu S_t \mathrm{d}t + S_t \sigma \mathrm{d}W_t,$$

where $(W_t)_{t\geq 0}$ is a Brownian motion and $H(x) = (x - K)^+$ for some strike $K > 0$. Then we obtain:

$$\begin{aligned}\pi_0(H, \mathbf{S}) &= \mathbb{E}_{\mathbb{Q}}[(S_T - K)^+)] \\ &= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} (S_0 \exp(\sigma\sqrt{T}x - \frac{1}{2}\sigma^2 T) - K)^+ \mathrm{e}^{-x^2/2} \mathrm{d}x. \quad (1)\end{aligned}$$

Note: Here we use that we know the distribution of the stock prices in the Black-Scholes model under the risk-neutral measure explicitly!

In [5]:
```julia
# Example for Monte Carlo Integration of deterministic functions using unifo
#You might want to replace f(x)=sin(x) with the function inside the integral

# Import necessary libraries
using Random
using Statistics

# Define the function to integrate
f(x) = sin(x)

# Set up the Monte Carlo simulation
num_samples = 10000
samples = rand(num_samples)  # Generate uniform random samples in [0, 1]
mc_approximation = mean(f.(samples))  # Apply f to each sample and calculate

println("Monte Carlo approximation of integral: ", mc_approximation)
```

```
Monte Carlo approximation of integral: 0.45857450475760686
```

Q2: How can we compute the risk neutral price $\pi_0(H, \mathbf{S})$, when the exact distribution is unknown?

Assume in the following that $H(\mathbf{S})$ is integrable with respect to $\mathbb{Q}$, i.e. $\pi_0(H, \mathbf{S}) < \infty$. Then by generating a sequence $\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, \ldots$ of independent realizations of $\mathbf{S}$, the law of large numbers assures us that:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} H(\mathbf{S}^{(i)}) \overset{\mathbb{Q}}{\to} \mathbb{E}_{\mathbb{Q}}[H(\mathbf{S})] = \pi_0.$$

This is the essence of the **Monte Carlo Method** for derivatives pricing.

0.4| Monte Carlo Method for Option Pricing

1. **Simulate Price Paths**: Generate sample paths $\{(S_t(\omega_i))_{t\in[0,T]} : i = 1, \ldots, N\}$ for the underlying asset's prices over $[0, T]$ following the price model.

2. **Calculate Payoff**: At the end of each sampled path, calculate the option's payoff, e.g. for a plain Vanilla Call option $H(S_T) = (S_T - K, 0)^+$ or an Asian option $H((S_t)_{t\in[0,T]}) = \left(\frac{1}{T}\sum_{i=1}^{M} S_{t_i} - K\right)^+$ for some fixed dates $t_1, \ldots, t_M$.

3. **Average the Payoffs**: Compute the average of all simulated payoffs to estimate the expected payoff:

$$\frac{1}{N} \sum_{i=1}^{N} H(\mathbf{S}(\omega_i))$$

4. **Discount to Present Value**: Calculate the option's present value by discounting the expected payoff using the risk-free rate:

$$\hat{\pi}_0 = \exp(-rT) \cdot \frac{1}{N} \sum_{i=1}^{N} H(\mathbf{S}(\omega_i)).$$

0.5| Main Objectives in Part II

We aim to solve the following five questions:

1. Understand the stochastic nature of the Monte Carlo Methods

   $\Longrightarrow$ **1.0| Basics of Monte Carlo Methods for Derivative Pricing**

2. Quantify and enhance the speed of convergence of the Monte Carlo Methods

   $\Longrightarrow$ **2.0| Variance Reduction**

3. Simulation of solutions to stochastic differential equations

   $\Longrightarrow$ **3.0| Discretization Schemes for Stochastic Differential Equations**

4. Computing derivatives using Monte Carlo Sampling

   $\Longrightarrow$ **4.0| Sensitivity Analysis**

5. Monte Carlo Pricing of American Options (will be covered after part III)

   $\Longrightarrow$ **(5.0| Longstaff-Schwartz Method)**

1.0| Basics of Monte Carlo Methods for Derivative Pricing

Our objective in this section is to use Monte Carlo methods for computing $\pi_0 = \mathbb{E}_{\mathbb{Q}}[H(\mathbf{S})]$, i.e., the expected value under a risk-neutral probability measure $\mathbb{Q}$ of some pay-off $H(\mathbf{S})$ depending on the underling asset prices $\mathbf{S} \triangleq \mathbf{S}_{t\in[0,T]}$.

1.1| Monte Carlo Integration Error

We define the Monte Carlo integration error ε_N as:

$$\varepsilon_N := \varepsilon_N(H, \mathbf{S}) := \pi_0(H, \mathbf{S}) - \hat{\pi}_0^N(H, \mathbf{S}),$$

where $\hat{\pi}_0^N(H, \mathbf{S})$ denotes the estimate from the first N samples defined as:

$$\hat{\pi}_0^N(H, \mathbf{S}) := \frac{1}{N} \sum_{i=1}^{N} H(\mathbf{S}^{(i)}).$$

Note that $\hat{\pi}_0^N(H, \mathbf{S})$ is an unbiased estimate for $\pi_0(H, \mathbf{S})$, i.e.

$$\mathbb{E}_{\mathbb{Q}}[\hat{\pi}_0^N(H, \mathbf{S})] = \pi_0(H, \mathbf{S}), \quad \text{and thus} \quad \mathbb{E}_{\mathbb{Q}}[\varepsilon_N] = 0.$$

Furthermore, we introduce the **mean square error (MSE)** as measures of error:

$$\mathrm{MSE}(\hat{\pi}_0^N) := \mathbb{E}_{\mathbb{Q}}[\varepsilon_N(H, \mathbf{S})^2],$$

and its square root, the **root mean square error (RMSE)**:

$$\mathrm{RMSE}(\hat{\pi}_0^N) := \sqrt{\mathbb{E}_{\mathbb{Q}}[\varepsilon_N(H, \mathbf{S})^2]}.$$

Proposition 1.1: *Let $\sigma = \sigma(H, \mathbf{S}) < \infty$ denote the standard deviation of the random variable $H(\mathbf{S})$. Then the root mean square error satisfies*

$$\mathrm{RMSE}(\hat{\pi}_0^N) = \sqrt{\mathbb{E}_{\mathbb{Q}}[\varepsilon_N(H, \mathbf{S})^2]} = \frac{\sigma}{\sqrt{N}}.$$

Moreover, the random variable $\sqrt{N} \cdot \varepsilon_N$ is asymptotically normally distributed with standard deviation $\sigma(H, \mathbf{S})$, i.e. for every $x_1 < x_2 \in \mathbb{R}$, we have

$$\lim_{N \to \infty} \mathbb{Q}\left(x_1 \frac{\sigma}{\sqrt{N}} < \varepsilon_N < x_2 \frac{\sigma}{\sqrt{N}} \right) = \Phi(x_2) - \Phi(x_1),$$

where Φ denotes the cumulative distribution function of the standard normal random variable.

Proof. Using the independence of the $\mathbf{S}^{(i)}$'s and the fact that $\hat{\pi}_0^N(H, \mathbf{S})$ is an unbiased estimator of $\pi_0(H, \mathbf{S})$, we obtain:

$$\mathbb{E}_{\mathbb{Q}}[\varepsilon_N^2] = \mathrm{Var}\left(\frac{1}{N} \sum_{i=1}^{N} H(\mathbf{S}^{(i)}) \right) = \frac{1}{N^2} \sum_{i=1}^{N} \mathrm{Var}(H(\mathbf{S}^{(i)})) = \frac{\sigma^2}{N}.$$

In addition, from the central limit theorem, we know that

$$\frac{1}{\sigma\sqrt{N}} \big(\sum_{i=1}^{N} H(\mathbf{S}^{(i)}) - N\pi_0(H, \mathbf{S}) \big) \xrightarrow{d} \mathcal{N}(0, 1), \quad \text{as } N \to \infty.$$

This yields the asymptotic normality of the error. □

Implications of Proposition 1.1

Proposition 1.1 highlights two crucial aspects of the Monte Carlo simulation error:

1. **Probabilistic Nature of Error:** The simulation error lacks a deterministic bound, meaning that for any given simulation and sample size N, the error magnitude can

vary widely. However, the likelihood of encountering large errors diminishes as N increases.

2. **Convergence Rate:** The typical error, such as the root mean square error $\mathrm{MSE}(\hat{\pi}_0^N)$, diminishes at the rate of $\frac{1}{\sqrt{N}}$. To enhance the accuracy tenfold, or to gain an additional significant digit, the sample size N must be increased by a factor of 100. This demonstrates that the Monte Carlo method converges at a rate of the square-root $\frac{1}{2}$.

High-Dimensions and Monte Carlo Integration

For simplicity, consider $\mathbf{S}$ to be a d-dimensional uniform random variable. In that case the fair-price of an European Option would be:

$$\pi_0(H, U) = \int_{[0,1]^d} H(x)dx.$$

Traditional methods for numerical integration typically rely on a grid defined by $0 \leq x_1 < x_2 < \ldots < x_N \leq 1$, for an arbitrary length N. For a d-dimensional space, this grid expands to $\{x_1, \ldots, x_N\}^d$, encompassing $n = N^d$ points. The integral approximation is derived from evaluating the function f at these grid points and interpolating between them with functions whose integrals can be explicitly calculated.

For a numerical integration method of order k, the error scales as $(1/N)^k$. However, the computational effort scales with n points, making the accuracy, in terms of computational complexity n, proportional to $n^{-k/d}$. Consequently, the effective rate of convergence, considering computational costs, diminishes to k/d with increasing dimension d. This degradation in performance due to rising dimensionality is known as the *curse of dimensionality*.

1.2| Confidence Intervals of the Monte Carlo Method

In Monte Carlo simulations, controlling the error is crucial due to its inherent randomness.

Q3: How large should we choose N to ensure that the probability of the error exceeding a certain tolerance level $\varepsilon > 0$ is below a desired threshold $\delta > 0$?

Mathematically, this is represented as:

$$\mathbb{Q}\left(|\varepsilon_N(H, \mathbf{S})| > \varepsilon\right) \leq \delta.$$

This formula sets the stage for understanding how to manage errors within Monte Carlo simulations effectively.

The answer to this question is partially provided by Proposition 1.1, which states:

$$\mathbb{Q}\left(|\varepsilon_N| > \epsilon\right) = 1 - \mathbb{Q}\left(-\frac{\varepsilon}{\sigma\sqrt{N}} \leq \varepsilon_N \leq \frac{\varepsilon}{\sigma\sqrt{N}}\right) \sim 2 - 2\Phi\left(\frac{\sqrt{N}\varepsilon}{\sigma}\right).$$

Note: that the Monte Carlo error is only asymptotically normal. This implies that the equation above accurately represents the error distribution only as N approaches infinity, as indicated by the "$\sim$" symbol.

A3: To align the error probability with a given δ, we solve for N, yielding:

$$N = \left(\Phi^{-1}(1 - \frac{\delta}{2})\right)^2 \frac{\sigma^2}{\varepsilon^2}.$$

This reveals that the required number of samples, N, scales inversely with the square of the tolerance, ε^2, a principle already noted in earlier discussions. This relationship underscores the increase in sample size needed to refine accuracy within the constraints of Monte Carlo simulations.

Remark 1.2: Challenges in Variance Estimation

This analysis has, until now, operated under the assumption that we know $\sigma = \sigma(H, \mathbf{S})$, the variance of the function $H(\mathbf{S})$, which is derived from the mean of $H(\mathbf{S})$. Given that the primary goal of initiating Monte Carlo simulations is to compute the mean of $H(\mathbf{S})$, it is highly improbable that the variance of $H(\mathbf{S})$ would be known a priori. Consequently, in practical applications, $\sigma(H, \mathbf{S})$ must often be substituted with a sample estimate. For guidance on deriving a sample estimator of σ.

It is critical to acknowledge the potential issues this substitution presents, notably regarding the Monte Carlo error associated with approximating $\sigma(H, \mathbf{S})$. The accuracy of Monte Carlo simulations not only depends on the estimator $\hat{\pi}_0^N(H, \mathbf{S})$ but also significantly on our estimation of σ. This introduces an additional layer of uncertainty, as the error in estimating σ directly impacts the confidence we can have in our Monte Carlo results.

Moreover, since the Monte Carlo estimator $\hat{\pi}_0^N(H, \mathbf{S})$ itself behaves as a random variable, merely reporting its value without context is insufficient. The estimator's reliability is inherently linked to the sample size N. Without knowledge of N, the precision of the estimate remains uncertain. Consequently, it is more informative to report not only the estimator's value but also a confidence interval that reflects the sample size and the variability in the estimation process. This approach provides a more comprehensive understanding of the estimator's accuracy and the confidence level of the results obtained from Monte Carlo simulations.

Definition 1.3: Let Z be a random variable and consider some level $\alpha \in (0, 1)$. The $(1 - \alpha)$-level confidence interval is defined by

$$\left(-z_{1-\frac{\alpha}{2}}, z_{1-\frac{\alpha}{2}}\right)$$

such that the critical number $z_{1-\frac{\alpha}{2}}$ satisfies:

$$\mathbb{Q}\left(|Z| \leq z_{1-\frac{\alpha}{2}}\right) = 1-\alpha.$$

The critical number $z_{1-\frac{\alpha}{2}}$ for a given level $(1-\alpha)$ can be computed from the inverse cumulative distribution function. Consider, for example, the normal distribution; then we get that

$$z_{1-\frac{\alpha}{2}} = \Phi^{-1}\left(1-\frac{\alpha}{2}\right).$$

In particular, using the inverse cdf of the normal distribution, we get that for $\alpha = 5\%$ the critical number equals 1.96, while for $\alpha = 1\%$ it equals 2.58.

Now, we can use the asymptotic normality of the Monte Carlo error ε_N to derive confidence intervals for $\hat{\pi}_0^N(H, \mathbf{S})$. Indeed, using Proposition 1.1 and denoting $\varepsilon_N = \pi_0 - \hat{\pi}_0^N$, we have

$$\begin{aligned} 1-\alpha &\approx \mathbb{Q}\left(-\frac{\sigma z_{1-\frac{\alpha}{2}}}{\sqrt{N}} \leq \varepsilon_N \leq \frac{\sigma z_{1-\frac{\alpha}{2}}}{\sqrt{N}}\right) \\ &= \mathbb{Q}\left(\hat{\pi}_0^N - \frac{\sigma z_{1-\frac{\alpha}{2}}}{\sqrt{N}} \leq \pi_0 \leq \hat{\pi}_0^N + \frac{\sigma z_{1-\frac{\alpha}{2}}}{\sqrt{N}}\right). \end{aligned}$$

Thus, the $(1-\alpha)$-level confidence interval for π_0 is

$$\mathrm{CI}_\alpha(\hat{\pi}_0^N) = \left(\hat{\pi}_0^N - \frac{\sigma z_{1-\frac{\alpha}{2}}}{\sqrt{N}}, \hat{\pi}_0^N + \frac{\sigma z_{1-\frac{\alpha}{2}}}{\sqrt{N}}\right).$$

Example 1.4: Monte Carlo Simulation in the Black–Scholes Model

We consider the Black–Scholes model:

$$\begin{cases} dS_t &= rS_t dt + \sigma S_t dW_t, \\ S_0 &= s \in \mathbb{R}^+. \end{cases}$$

where W is a standard Brownian motion, while we assume we are already under the martingale measure. We want to compute the price of a European call option with payoff function

$$H(S_T) = (S_T - K)^+,$$

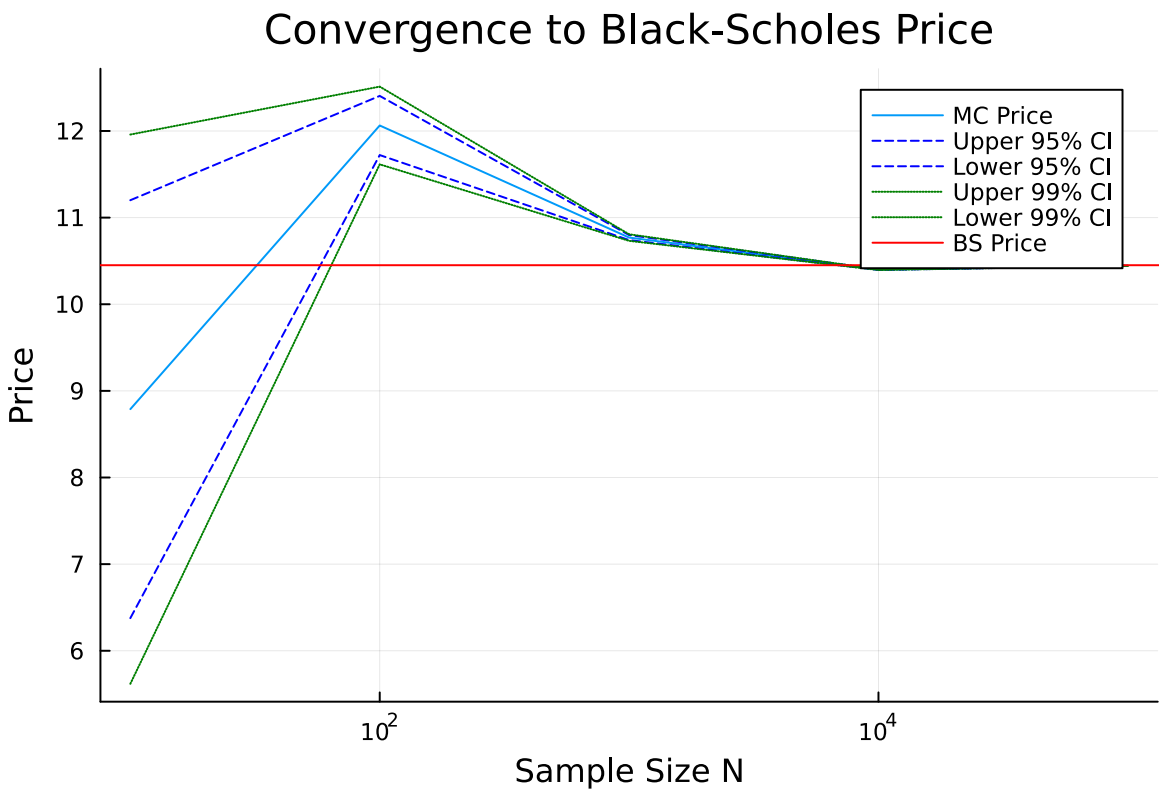
together with the 95% and 99% confidence intervals.

In [8]:
```
using Plots
```

```
# Assuming definitions for monte_carlo_bs, S0, K, T, r, sigma, and bs_price

Ms = [10, 100, 1000, 10000, 100000]  # Different sample sizes
mc_prices = []
rmse_values = []
upper_95 = []
lower_95 = []
upper_99 = []
lower_99 = []

for M in Ms
    price, RMSE = monte_carlo_bs(S0, K, T, r, sigma, M)
    push!(mc_prices, price)
    push!(rmse_values, RMSE)

    # Calculate confidence intervals
    # For 95% confidence level
    z_95 = 1.96  # z-score for 95% confidence
    push!(upper_95, price + z_95 * RMSE / sqrt(M))
    push!(lower_95, price - z_95 * RMSE / sqrt(M))

    # For 99% confidence level
    z_99 = 2.576  # z-score for 99% confidence
    push!(upper_99, price + z_99 * RMSE / sqrt(M))
    push!(lower_99, price - z_99 * RMSE / sqrt(M))
end
```

In [9]:

```
# Plotting
plot(Ms, mc_prices, label="MC Price", xlabel="Sample Size N", ylabel="Price"
plot!(Ms, upper_95, label="Upper 95% CI", linestyle=:dash, color=:blue)
plot!(Ms, lower_95, label="Lower 95% CI", linestyle=:dash, color=:blue)
plot!(Ms, upper_99, label="Upper 99% CI", linestyle=:dot, color=:green)
plot!(Ms, lower_99, label="Lower 99% CI", linestyle=:dot, color=:green)
hline!([bs_price], label="BS Price", color=:red)
```



2.0| Variance Reduction

The Idea Behind Variance Reduction:

We might be able to improve the constant factor $\sigma(H, \mathbf{S}) = \sqrt{\mathrm{Var}(H(\mathbf{S}))}$ in the RMSE

$$\sqrt{\mathbb{E}_{\mathbb{Q}}[\varepsilon_N(H, \mathbf{S})^2]} = \frac{\sigma(H, \mathbf{S})}{\sqrt{N}}.$$

The idea is to obtain, in a systematic way, random variables $\mathbf{Y}$ and functions g such that $\mathbb{E}_{\mathbb{Q}}[g(\mathbf{Y})] = \mathbb{E}_{\mathbb{Q}}[H(\mathbf{S})]$, but with smaller variance $\mathrm{Var}(g(\mathbf{Y})) < \mathrm{Var}(H(\mathbf{S}))$.

Inserting $\sigma^2(g, \mathbf{Y}) = \mathrm{Var}(g(\mathbf{Y}))$ into the equation shows that such an approach will decrease the computational work—proportional to the number of trajectories—provided that the generation of samples from $g(\mathbf{Y})$ is not prohibitively more expensive than the generation of samples from $H(\mathbf{S})$. This leads then to a faster numerical scheme, since the same error can be achieved with fewer samples. In this section we follow [3] closely.

2.1| Antithetic Variates

Whenever a random variable U has the uniform distribution or if B has the d-dimensional standard normal distribution, then so does $1-U$ and $-B$, respectively. Therefore, the transformations do not change the expected value $\mathbb{E}_{\mathbb{Q}}[H(\mathbf{S})]$, if $\mathbf{S}=U$ or $\mathbf{S}=B$!

In general, assume there exists a (simple) transformation $\tilde{\mathbf{S}}$ that has the same law as $\mathbf{S}$ and such that a realization of $\tilde{\mathbf{S}}$ can be computed from a realization of $\mathbf{S}$ by a deterministic transformation.

Then define the **antithetic variates Monte Carlo estimate** by

$$\hat{\pi}_0^{A,N}(H,\mathbf{S}) := \frac{1}{N}\sum_{i=1}^{N}\frac{H(\mathbf{S}^{(i)})+H(\tilde{\mathbf{S}}^{(i)})}{2}.$$

Since $\mathbb{E}_{\mathbb{Q}}[\frac{H(\mathbf{S})+H(\tilde{\mathbf{S}})}{2}]=\mathbb{E}_{\mathbb{Q}}[H(\mathbf{S})]$, the random variable $\hat{\pi}_0^{A,N}(H,\mathbf{S})$ is then another unbiased estimator for $\mathbb{E}_{\mathbb{Q}}[H(\mathbf{S})]$.

The application of antithetic variates makes sense if the mean square error of $\hat{\pi}_0^{A,N}(H,\mathbf{S})$ is smaller than the MSE of $\hat{\pi}_0^{2N}(H,\mathbf{S})$, i.e., if

$$\frac{1}{N}\mathrm{Var}(\frac{H(\mathbf{S})+H(\tilde{\mathbf{S}})}{2}) < \frac{1}{2N}\mathrm{Var}(H(\mathbf{S})).$$

This is equivalent to having a negative covariance between $H(\mathbf{S})$ and $H(\tilde{\mathbf{S}})$, indicating a negative dependence between the input variables $\mathbf{S}$ and $\tilde{\mathbf{S}}$ produces also negative dependence between the output variables $H(\mathbf{S})$ and $H(\tilde{\mathbf{S}})$. A simple, sufficient condition for the latter is the monotonicity of the function H that maps inputs to outputs.

The calculations above yield also the following decomposition for the MSE of the antithetic variates Monte Carlo method:

$$\mathrm{MSE}(\hat{\pi}_0^{A,N}(H,\mathbf{S})) = \mathrm{MSE}(\hat{\pi}_0^{2N}(H,\mathbf{S})) + \frac{\mathrm{Cov}\left(H(\mathbf{S}),H(\tilde{\mathbf{S}})\right)}{2N}$$

where $\mathrm{MSE}_{\hat{\pi}_0^{N}(H,\mathbf{S})}$ denotes the Mean Squared Error for the antithetic variates method with N samples, and $\mathrm{MSE}_{\hat{\pi}_0^{2N}(H,\mathbf{S})}$ is the MSE for the standard Monte Carlo method with $2N$ samples.

In other words: the improvement over the standard Monte Carlo method, if any, comes in the form of an additive factor. The larger the negative dependence between $H(\mathbf{S})$ and $H(\tilde{\mathbf{S}})$, the larger this factor is as well (for fixed N).

2.2| Control Variates

Assume there exists a random variable $\mathbf{Y}$ and a functional g such that we know the exact value of $\mathbb{E}_{\mathbb{Q}}[g(\mathbf{Y})]$. Then obviously,

$$\mathbb{E}_{\mathbb{Q}}[H(\mathbf{S})] = \mathbb{E}_{\mathbb{Q}}[H(\mathbf{S}) + \lambda(g(\mathbf{Y}) - \mathbb{E}_{\mathbb{Q}}[g(\mathbf{Y})])],$$

for any deterministic parameter λ. Thus, a Monte Carlo estimate for $\mathbb{E}_{\mathbb{Q}}[H(\mathbf{S})]$ is given by

$$\hat{\pi}_0^{C,\lambda}(H, \mathbf{S}) = \frac{1}{N} \sum_{i=1}^{N} \left(H(\mathbf{S}^{(i)}) + \lambda(g(\mathbf{Y}^{(i)}) - \mathbb{E}_{\mathbb{Q}}[g(\mathbf{Y}^{(i)})]) \right),$$

where $(\mathbf{S}^{(i)}, \mathbf{Y}^{(i)})$ are independent realizations of $(\mathbf{S}, \mathbf{Y})$. The parameter λ is chosen such that the variance $\mathrm{Var}(H(\mathbf{S}) + \lambda g(\mathbf{Y}))$ is minimized.

This is achieved when

$$\lambda = \frac{\mathrm{Cov}(H(\mathbf{S}), g(\mathbf{Y}))}{\mathrm{Var}(g(\mathbf{Y}))}.$$

Applying Proposition 1.1, we get that the mean square error for the standard and the control variates Monte Carlo simulations compare as follows:

$$\frac{\mathrm{MSE}(\hat{\pi}_0^{C,\lambda}(H, \mathbf{S}))}{N} = \frac{\mathrm{Var}(H(\mathbf{S}))}{N} \cdot (1 - \rho^2) \leq \frac{MSE(\hat{\pi}_0^N)}{N},$$

where ρ denotes the correlation coefficient between $H(\mathbf{S})$ and $g(\mathbf{Y})$.

In other words, the improvement of the control variates Monte Carlo simulation over the standard Monte Carlo method comes in the form of a multiplicative factor. Assuming that the computational work per realization is c times higher using control variates, the equation above implies that the control variates technique is $\frac{1}{c(1-\rho^2)}$-times faster than standard Monte Carlo. In particular, the improvement in speed from the use of control variates is larger as the correlation between $H(\mathbf{S})$ and $g(\mathbf{Y})$ becomes higher. If, for example, $\rho = 0.8$ and $c = 2$ the speed-up factor equals 1.38, while if $\rho = 0.95$ the speed-up factor equals 5.

2.2.1| Finding of Good Control Variates

A natural question now is how to find, or construct, good control variates. There is no general answer since these are typically specified by the problem at hand. However, in option pricing, the underlying asset provides a virtually universal source of control variates because, whenever the risk-free rate is zero, we have:

$$\mathbb{E}_{\mathbb{Q}}[S_t] = S_0, \quad \forall t \geq 0.$$

Moreover, simple options that admit a closed-form solution can be used as control variates for the pricing of more complex derivatives. In addition, simple models can be used as control variates for option pricing in more advanced models; for example, the Black–Scholes model can serve as a control variate for stochastic volatility models.

Example 2.1: Control Variate

Assume we want to compute the price of an European option with payoff $H(S_T)$ and we are given a sample $S_T^{(1)}, \dots, S_T^N$ from the law of S_T. The control variates Monte Carlo estimator takes the form

$$\hat{\pi}_0^{C,\lambda}(H, \mathbf{S}_T) = \frac{1}{N} \sum_{i=1}^{N} \left(H(S_T^{(i)}) - \hat{\lambda} S_T^{(i)} \right) + \hat{\lambda} S_0,$$

where $\hat{\lambda}$ can also be replaced by the sample estimator. If $H(S_T) = (S_T - K)^+$, i.e., we are pricing a call option, then the efficiency of the control variate depends, essentially, on the strike K.

2.4| Stratified Sampling

The main principle of stratified sampling is to partition the sample space into disjoint subsets, called strata, and to constrain the number of samples selected from each stratum. Let $A_1, \dots, A_L$ be disjoint subsets of $\mathbb{R}^d$ such that $\mathbb{P}(\mathbf{S} \in \bigcup_{l=1}^L A_l) = 1$. Then, using the law of total probability, we can estimate $H(\mathbf{S})$ as follows

$$\mathbb{E}_{\mathbb{Q}}[H(\mathbf{S})] = \sum_{l=1}^{L} \mathbb{E}_{\mathbb{Q}}[H(\mathbf{S}) | \mathbf{S} \in A_l] \mathbb{Q}(\mathbf{S} \in A_l) = \sum_{l=1}^{L} q_l \mathbb{E}_{\mathbb{Q}}[H(\mathbf{S}) | \mathbf{S} \in A_l],$$

where $q_l = \mathbb{Q}(\mathbf{S} \in A_l)$. The stratified sampling estimator takes the form

$$\hat{\pi}_0^{\mathrm{ST},N}(H, \mathbf{S}) = \sum_{l=1}^{L} q_l \frac{1}{N_l} \sum_{k=1}^{N_l} H(\mathbf{S}_{lk}),$$

where N_l denotes the fraction of observations from the stratum A_l, and $\mathbf{S}_{lk}$ are i.i.d. realizations from the distribution of $\mathbf{S}$ conditional on $\mathbf{S} \in A_l$.

Variance Reduction through Stratified Sampling

The variance of the stratified sampling estimator, using the proportional allocation $p_l = q_l$, is provided by

$$\mathrm{Var}(\hat{\pi}_0^{\mathrm{ST},N}(H, \mathbf{S})) = \sum_{l=1}^{L} \frac{q_l \sigma_l^2}{N},$$

where $\sigma_l^2 = \mathrm{Var}(H(\mathbf{S}_{lk})) = \mathrm{Var}(H(\mathbf{S})|\mathbf{S} \in A_l)$. This formulation shows that stratified sampling, especially with optimized allocations, can significantly reduce the variance of the estimator compared to standard Monte Carlo methods.

2.5| Importance Sampling

Importance sampling is related to the acceptance-rejection method and also to Girsanov's theorem. The idea is to sample more often in regions where the variance is higher, thus increasing the sampling efficiency. Assume that the underlying random variable $\mathbf{S}$ has a density q (on $\mathbb{R}^d$). Moreover, let p be another probability density. Then we can write

$$\mathbb{E}_{\mathbb{Q}}[H(\mathbf{S})] = \int_{\mathbb{R}^d} H(x)q(x)dx = \int_{\mathbb{R}^d} H(x)\frac{q(x)}{p(x)}p(x)dx = \mathbb{E}_{\mathbb{P}}\left[\frac{q(\mathbf{Y})}{p(\mathbf{Y})}H(\mathbf{Y})\right] = \pi_0(H$$

where $\mathbf{Y}$ is a d-dimensional random variable with density p. The quantity $\frac{q}{p}$ is called the likelihood ratio or the Radon–Nikodym derivative. Thus, a Monte Carlo estimate for $\mathbb{E}_{\mathbb{Q}}[H(\mathbf{S})]$ is given by

$$\hat{\pi}_0^{\mathrm{Imp},N}(H,\mathbf{S}) \triangleq \hat{\pi}_0(H\frac{p}{q},\mathbf{Y}) = \frac{1}{M}\sum_{i=1}^{M}\frac{q(\mathbf{Y}^{(i)})}{p(\mathbf{Y}^{(i)})}H(\mathbf{Y}^{(i)}).$$

The variance of $H(\mathbf{Y})\frac{q(\mathbf{Y})}{p(\mathbf{Y})}$ is crucial for the speed up, which is determined by how well p approximates q in regions where $H(x)$ contributes significantly to the integral.

Q1: How to choose the density p?

To minimize variance, ideally, we want p to be proportional to $H \cdot q$. However, p needs to be a valid probability density and hence normalized. This presents a practical challenge as it requires knowledge of the integral of $H \cdot q$, which is typically what we're trying to estimate. A good importance sampling distribution p should make the modified integrand $H \cdot \frac{q}{p}$ as flat as possible to minimize variance.

2.6| Conclusive Remark on Variance Reduction Techniques

Among the variance reduction techniques discussed, antithetic variates are straightforward to implement but offer limited speed-up. Control variates and importance sampling, in contrast, can significantly reduce variance by exploiting specific properties of the problem. However, their implementation is more complex and requires a tailored approach for each new application.

3.0| Discretization Schemes for Stochastic Differential Equations

The simulation of continuous-time financial market models necessitates their discretization into a discrete-time process. Asset price models $\mathbf{S}$ are often modeled as solutions to stochastic differential equations (SDEs):

$$\begin{cases} d\mathbf{S}_t & = \mu(t, \mathbf{S}_t)dt + \sum_{i=1}^{m} \Sigma_i(t, \mathbf{S}_t)dB_t^{(i)}, \\ S_0 & = s \in \mathbb{R}^+ \end{cases}$$

where $B^{(1)}, \ldots, B^{(m)}$ are independent Brownian motions, and the vector fields $\mu(t, \cdot), \Sigma_1(t, \cdot), \ldots, \Sigma_m(t, \cdot) : \mathbb{R}^d \to \mathbb{R}^d$ satisfy certain regularity conditions (e.g., Lipschitz continuity) ensuring the uniqueness of the solution.

Note: Numerical approximation of SDE solutions is critical when closed-form solutions or exact distributions are unknown. This section introduces two prevalent discretization schemes: Euler and Milstein, and applies them to the Black-Scholes and Heston stochastic volatility models.

Interpretation of SDEs and its parameters

The SDE

$$d\mathbf{S}_t = \mu(t, \mathbf{S}_t)dt + \sum_{i=1}^{m} \Sigma_i(t, \mathbf{S}_t)dB_s^{(i)}$$

must be interpreted as an **stochastic integral equation**:

$$\mathbf{S}_t = \mathbf{S}_0 + \int_0^t \mu(s, \mathbf{S}_s)ds + \sum_{i=1}^{m} \int_0^t \Sigma_i(s, \mathbf{S}_s)dB_s^{(i)}.$$

The stochastic integral parts have constant expectations and we obtain:

$$\begin{aligned} \mathbb{E}_{\mathbb{P}}[\mathbf{S}_{t+\Delta t} - \mathbf{S}_t \mid \mathcal{F}_t] &= \mathbb{E}_{\mathbb{Q}}\left[\int_t^{t+\Delta t} \mu(s, \mathbf{S}_s)\,\mathrm{d}s \mid \mathcal{F}_t\right] \\ &\approx \mu(t, \mathbf{S}_t)\Delta t, \quad \text{for small } \Delta t. \end{aligned}$$

The conditional variance of $\mathbf{S}_{t+\Delta t} - \mathbf{S}_t$ given $\mathcal{F}_t$ is

$$\begin{aligned} \mathbb{E}_{\mathbb{P}}\left[\left(\sum_{i=1}^{m} \int_t^{t+\Delta t} \Sigma_i(s, \mathbf{S}_s)\,\mathrm{d}B_s^{(i)}\right)^2 \mid \mathcal{F}_t\right] &= \sum_{i=1}^{m} \mathbb{E}_{\mathbb{Q}}\left[\int_t^{t+\Delta t} \Sigma_i^2(s, \mathbf{S}_s)\,\mathrm{d}s \mid \mathcal{F}_t\right] \\ &\approx \sum_{i=1}^{m} \Sigma_i^2(t, \mathbf{S}_t)\Delta t, \quad \text{for small } \Delta t. \end{aligned}$$

Hence, the coefficient μ in the SDE gives us an information on the direction of the change of $\mathbf{S}_t$ and the $\Sigma_1, \ldots, \Sigma_m$ yields information on the variance of this change.

We call μ the **drift coefficient** and $\Sigma_1, \ldots, \Sigma_m$ are called **volatiliy coefficients**.

3.0.1| Short Introduction to the Heston Stochastic Volatility Model

A large class of asset price models extend the Black-Scholes framework by introducing stochastic volatility (SV) to better capture the implied volatility surface (see also section on calibration). The most prominent SV model is given by:

The Heston Model

1. **Asset Price Dynamics:**

$$dS_t = rS_t dt + \sqrt{V_t} S_t dW_t^{(1)}$$

   The evolution of the asset prices S_t is driven by a Brownian motion $(W_t^{(1)})_{t\in[0,T]}$ modulated by an **instantaneous volatility** term $\sqrt{V_t}$.

2. **Variance Dynamics:**

$$dV_t = \kappa(\theta - V_t)dt + \xi\sqrt{V_t} dW_t^{(2)}$$

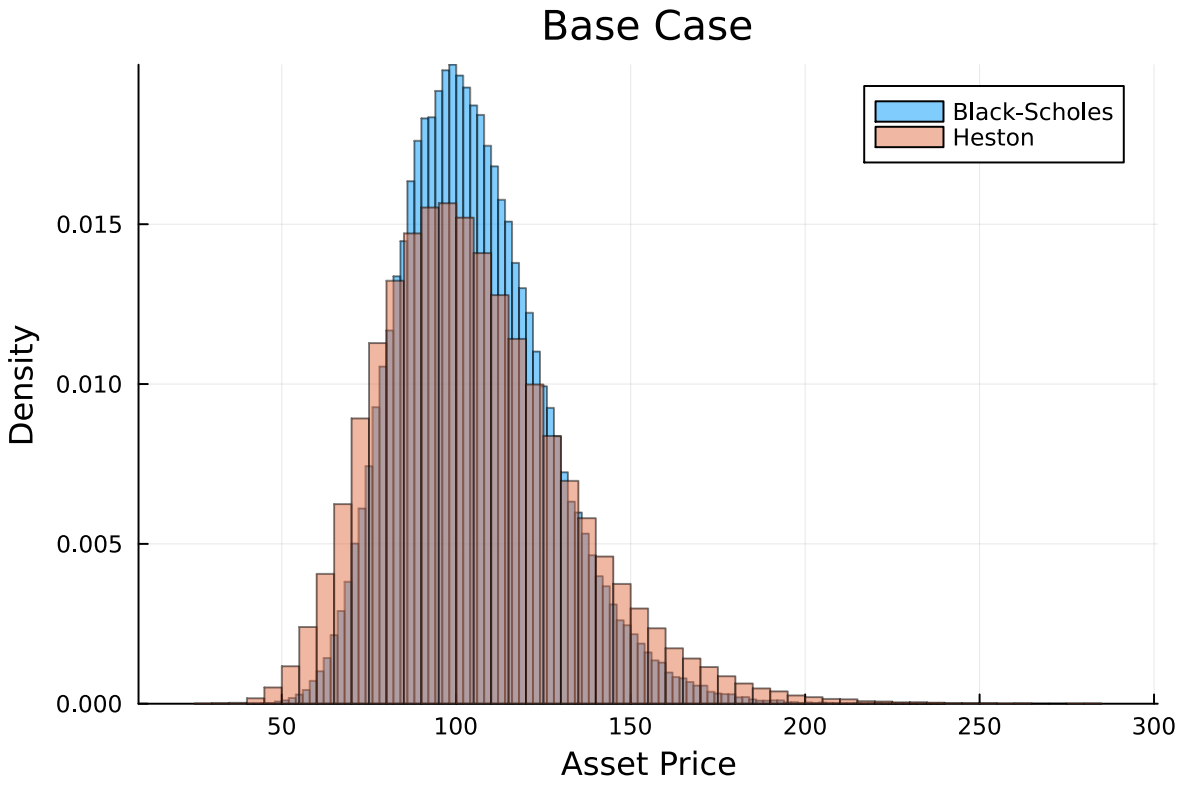
   We call V_t the **instantaneous variance** of the asset price process, with κ the **mean reversion rate**, θ the **long-term variance**, ξ the **volatility of volatility** and $(W_t^{(2)})_{t\in[0,T]}$ another Brownian motion, potentially ρ-correlated with $(W_t^{(1)})_{t\in[0,T]}$.

Some Features of the Heston Model:

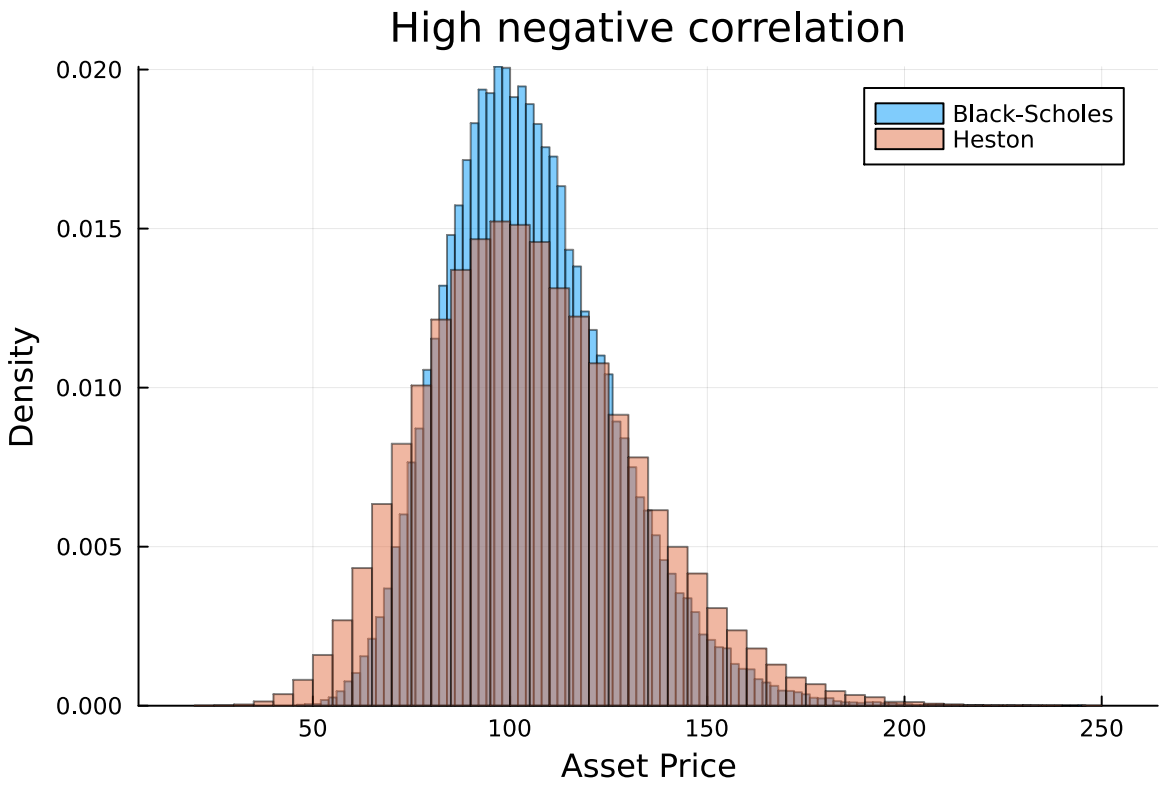
- **Stochastic Volatility:** Unlike the Black-Scholes model, which assumes constant volatility, the Heston model allows volatility to fluctuate over time according to its own stochastic process.
- **Mean Reversion:** The model incorporates mean reversion in volatility, suggesting that high levels of volatility will eventually fall back to a long-term average level, and vice versa.
- **Correlation between Asset Returns and Volatility:** The model allows for a correlation between the Brownian motions $(W_t^{(1)})_{t\in[0,T]}$ and $(W_t^{(2)})_{t\in[0,T]}$, capturing the observed phenomenon that asset returns often move inversely to changes in volatility (leverage effect).

In [64]:
```
T = 1.0; S0 = 100; r = 0.05; sigma_BS = 0.2; n = 100000; dt = 1/252
V0 = 0.1; theta = 0.04; kappa = 2.0; xi = 0.1; rho = -0.3

BS_prices = simulate_black_scholes(S0, r, sigma_BS, T, n, dt)
Heston_prices, _ = simulate_heston(S0, V0, 2.0, 0.04, 0.1, -0.1, T, n, dt)
```

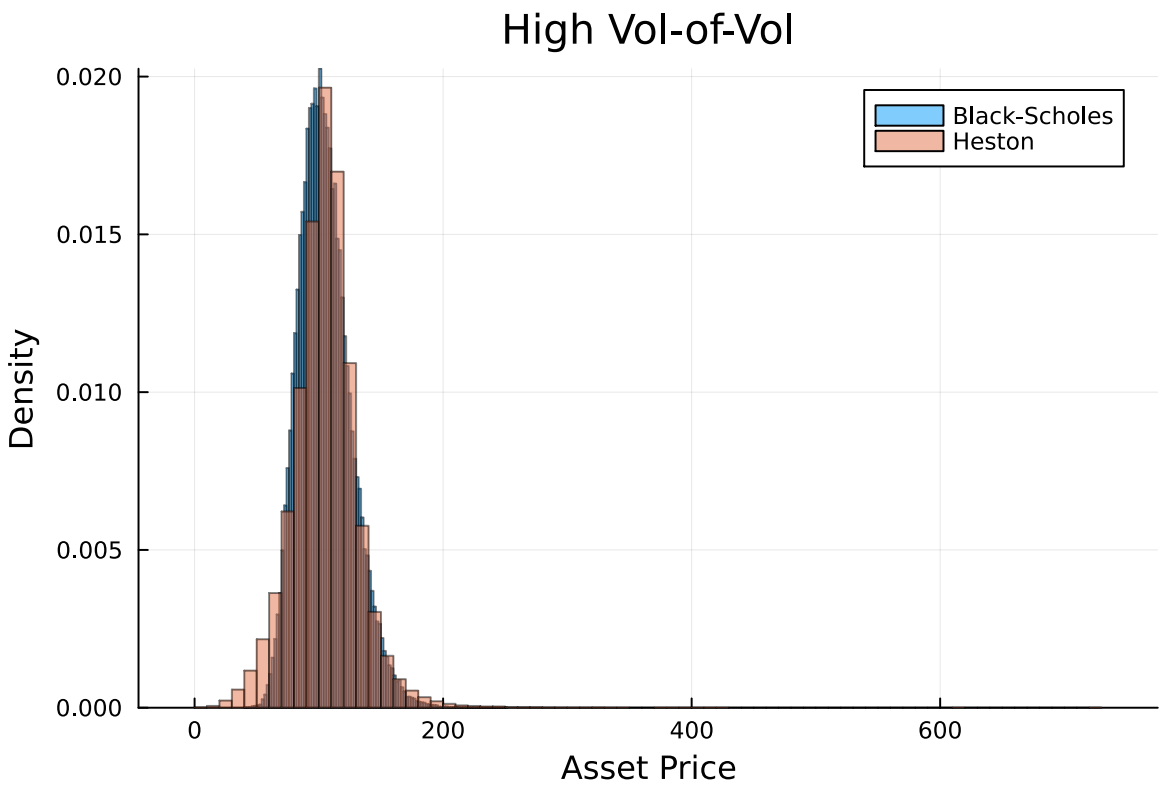
```
plot()
p = histogram(BS_prices, bins=100, alpha=0.5, label="Black-Scholes", norm=:p
    histogram!(p, Heston_prices, bins=100, alpha=0.5, label="Heston", norm=:
    xlabel!(p, "Asset Price")
    ylabel!(p, "Density")
    display(p)
```



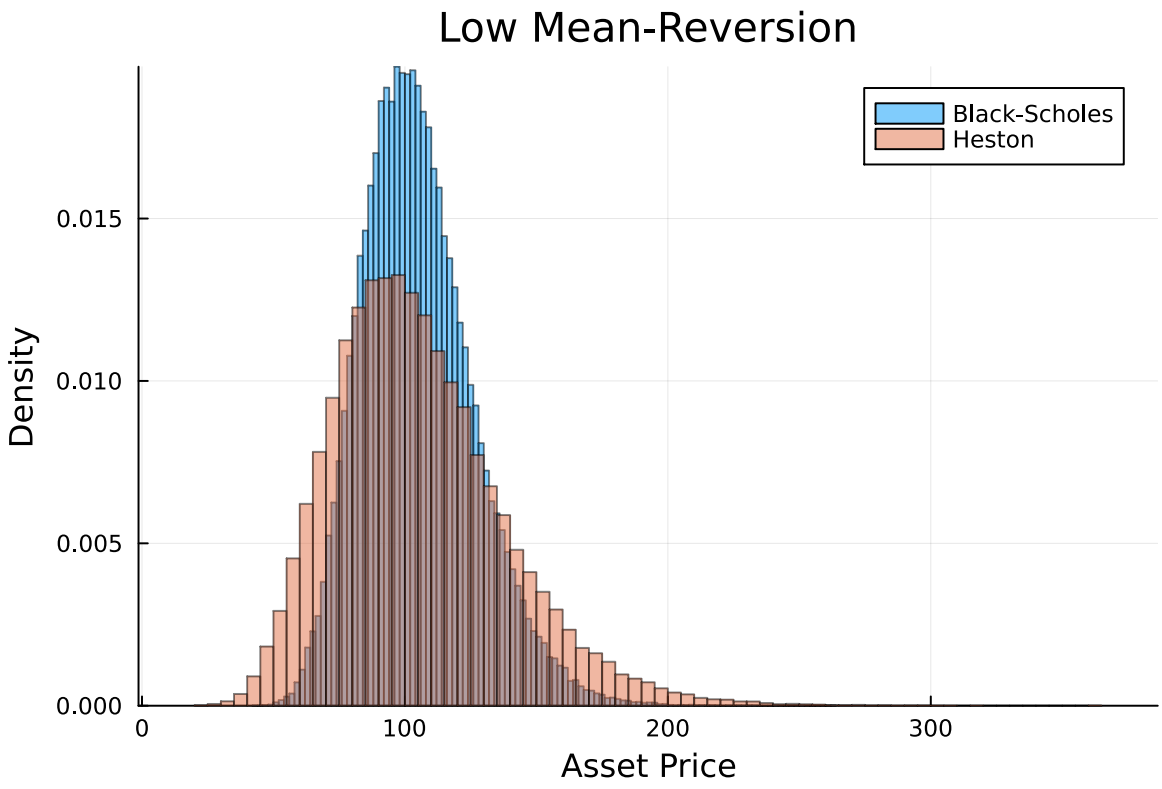
In [65]:
```
BS_prices = simulate_black_scholes(S0, r, sigma_BS, T, n, dt)
Heston_prices, _ = simulate_heston(S0, V0, 2.0, 0.04, 0.1, -0.9, T, n, dt)
plot()
p = histogram(BS_prices, bins=100, alpha=0.5, label="Black-Scholes", norm=:p
histogram!(p, Heston_prices, bins=100, alpha=0.5, label="Heston", norm=:pdf)
xlabel!(p, "Asset Price")
ylabel!(p, "Density")
display(p)
```



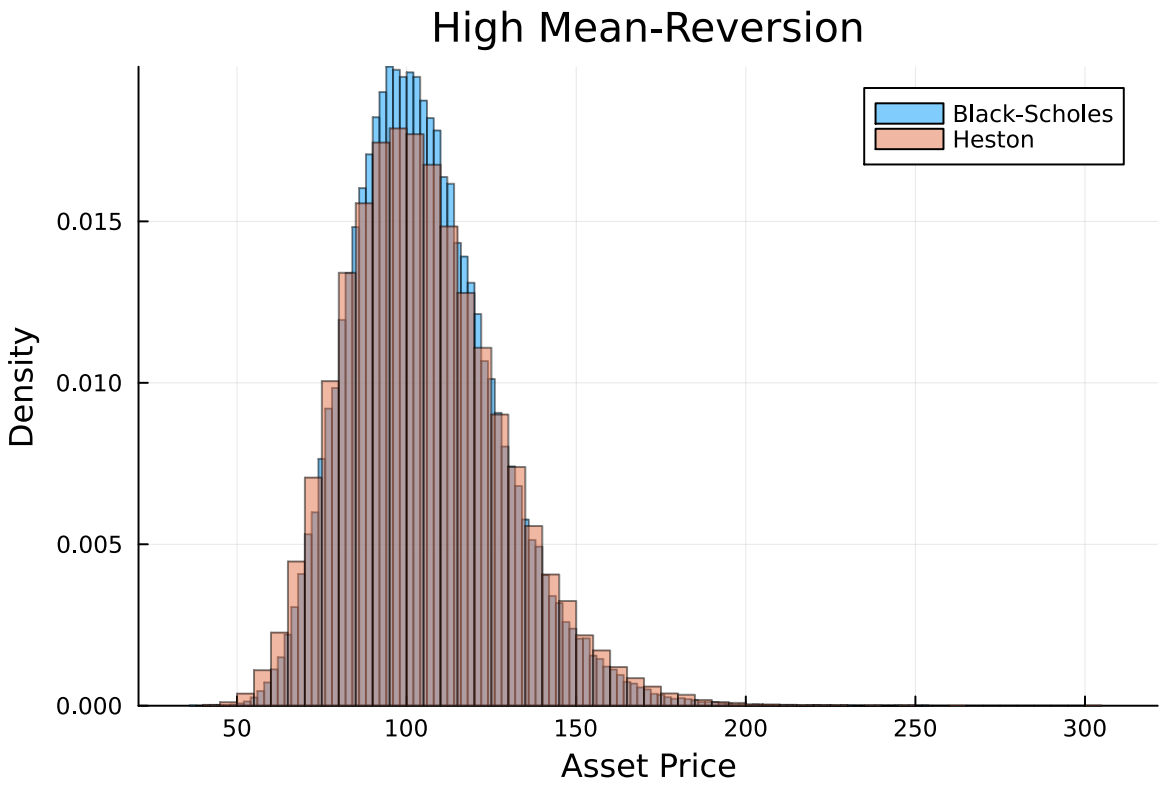
In [66]:
```
BS_prices = simulate_black_scholes(S0, r, sigma_BS, T, n, dt)
Heston_prices, _ = simulate_heston(S0, V0, 2.0, 0.04, 0.8, -0.3, T, n, dt)
plot()
p = histogram(BS_prices, bins=100, alpha=0.5, label="Black-Scholes", norm=:p
histogram!(p, Heston_prices, bins=100, alpha=0.5, label="Heston", norm=:pdf)
xlabel!(p, "Asset Price")
ylabel!(p, "Density")
display(p)
```



In [67]:
```
BS_prices = simulate_black_scholes(S0, r, sigma_BS, T, n, dt)
Heston_prices, _ = simulate_heston(S0, V0, 0.2, 0.04, 0.1, -0.3, T, n, dt)
plot()
p = histogram(BS_prices, bins=100, alpha=0.5, label="Black-Scholes", norm=:p
histogram!(p, Heston_prices, bins=100, alpha=0.5, label="Heston", norm=:pdf)
xlabel!(p, "Asset Price")
ylabel!(p, "Density")
display(p)
```



In [68]:
```
BS_prices = simulate_black_scholes(S0, r, sigma_BS, T, n, dt)
Heston_prices, _ = simulate_heston(S0, V0, 8, 0.04, 0.1, -0.3, T, n, dt)
plot()
p = histogram(BS_prices, bins=100, alpha=0.5, label="Black-Scholes", norm=:p
histogram!(p, Heston_prices, bins=100, alpha=0.5, label="Heston", norm=:pdf)
xlabel!(p, "Asset Price")
ylabel!(p, "Density")
display(p)
```



3.1| Generation of Sample Paths for Brownian Motions

Generating sample paths from a stochastic process, especially for numerical solutions to SDEs and pricing path-dependent options (like Asian or barrier options), is essential.

We first focus on generating paths for a one-dimensional Brownian motion, B, noting that the techniques apply to multi-dimensional cases as well.

Since we cannot sample the full path $(B_t)_{t\in[0,T]}$ due to its infinite-dimensional nature, we work with a finite-dimensional "skeleton" $(B_{t_1}, \ldots, B_{t_n})$ based on a partition $0 = t_0 < t_1 < \ldots < t_n = T$.

Example 3.1: In the Black-Scholes model the stock price process S can be sampled using sampled Brownian motions and transform according to:

$$S_t = S_0 \exp\left(\sigma B_t + \left(\mu - \frac{\sigma^2}{2}\right) t\right).$$

3.1.1| Discretization of Brownian Motion as Random Walk

Input: Time points $t_1, t_2, \ldots, t_n$, where $0 = t_0 < t_1 < t_2 < \ldots < t_n = T$ are the discretized time points for sampling Brownian motion.

Output: Sampled values $B_{t_1}, B_{t_2}, \ldots, B_{t_n}$ of Brownian motion at the specified time points.

1. **Initialization:**
    - Set $B_{t_0} = 0$ as the starting point of Brownian motion.
    - Calculate time increments $\Delta t_i = t_i - t_{i-1}$ for $i = 1, 2, \ldots, n$.
2. **For each time increment Δt_i:**
    - Generate an n-dimensional standard normal random variable $X = (X_1, \ldots, X_n)$.
    - Calculate the increment $\Delta B_i = \sqrt{\Delta t_i} X_i$.
3. **Construct Brownian motion path:**
    - For $i = 1$ to n: Compute $B_{t_i} = B_{t_{i-1}} + \Delta B_i$.
4. **Return** the sampled values $B_{t_1}, B_{t_2}, \ldots, B_{t_n}$.

3.1.2| Sampling from a Multivariate Normal Distribution using Cholesky Factorization

A method for generating sample paths from the finite-dimensional skeleton $(B_{t_1}, \ldots, B_{t_n})$ is based on the following property of Brownian motion:

$$(B_{t_1}, \ldots, B_{t_n}) \sim \mathcal{N}(0, \Sigma),$$

with $\Sigma_{i,j} = \min(t_i, t_j)$, for $1 \leq i, j \leq n$.

Given n independent one-dimensional normal random variables $X = (X_1, \ldots, X_n)$, we obtain an n-dimensional normal random vector with covariance matrix Σ by AX, where $\Sigma = AA^T$. In this particular case, it is easy to find the Cholesky factorization A by

$$A = \begin{pmatrix} \sqrt{t_1} & 0 & \ldots & 0 \\ \sqrt{t_2 - t_1} & \sqrt{t_2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{t_n - t_{n-1}} & \sqrt{t_n - t_{n-1}} & \ldots & \sqrt{t_n} \end{pmatrix}.$$

The matrix A transforms independent standard normal variables into a correlated Brownian motion path with covariance matrix Σ.

3.2| Discretization Schemes and Convergence

Our goal in this section is to find approximations $\hat{\mathbf{S}}$ to the solution $\mathbf{S}$ of some SDE on a fixed time interval $[0, T]$. These approximations will be based on a time grid $\Delta = \{t_0 = t_0 < t_1 < \ldots < t_N = T\}$ of size N.

Moreover, we denote the **mesh-size** of the grid by

$$|\Delta| := \max_{1 \leq i \leq N} |t_i - t_{i-1}|,$$

and we define the increments of time and of any process $\mathbf{S}$ along the grid by $\Delta t_i := t_i - t_{i-1}$ and $\Delta \mathbf{S}_i := S_{t_i} - S_{t_{i-1}}, 1 \leq i \leq N$, respectively.

Moreover, for $t \in [0, T]$, we set

$$\bar{t} = \sup\{t_i | 0 \leq i \leq N, t_i \leq t\}.$$

We will define the discrete approximation $\hat{\mathbf{S}}^{\Delta}$ along the grid as follows:

$$\hat{\mathbf{S}}_i^{\Delta} \triangleq \hat{\mathbf{S}}_{t_i} \quad \text{for } 0 \leq i \leq N.$$

Q: In what sense should $\hat{\mathbf{S}}^{\Delta}$ approximate $\mathbf{S}$?

We introduce two notions to describe the type of approximation of $\hat{\mathbf{S}}^{\Delta}$ to $\mathbf{S}$:

Definition 3.1: We say that the scheme $\hat{\mathbf{S}}^{\Delta}$ converges **strongly** to $\mathbf{S}$ if

$$\lim_{|\Delta| \to 0} \mathbb{E}_{\mathbb{Q}}[|\hat{\mathbf{S}}_T^{\Delta} - \mathbf{S}_T|] = 0.$$

Moreover, we say that the scheme $\hat{\mathbf{S}}^{\Delta}$ has strong order γ if (for $|\Delta|$ small enough)

$$\mathbb{E}_{\mathbb{Q}}[|\hat{\mathbf{S}}_T^{\Delta} - \mathbf{S}_T|] \leq C|\Delta|^{\gamma}$$

for some constant $C > 0$, which does not depend on $\gamma > 0$ or $|\Delta| > 0$.

Definition 3.2: Given a suitable class $\mathcal{G}$ of functions $f : \mathbb{R}^d \to \mathbb{C}$, we say that the scheme $\hat{\mathbf{S}}^{\Delta}$ converges **weakly** (with respect to $\mathcal{G}$) if for all $f \in \mathcal{G}$ we have:

$$\lim_{|\Delta| \to 0} \mathbb{E}_{\mathbb{Q}}[f(\hat{\mathbf{S}}_T^{\Delta})] = \mathbb{E}_{\mathbb{Q}}[f(\mathbf{S}_T)]. \quad (2)$$

Moreover, we say that $\hat{\mathbf{S}}^{\Delta}$ has weak order $\gamma > 0$ if for every $f \in \mathcal{G}$ there is a constant $C > 0$ (not depending on $|\Delta|$) such that

$$|\mathbb{E}_{\mathbb{Q}}[f(\hat{\mathbf{S}}_T^{\Delta})] - \mathbb{E}_{\mathbb{Q}}[f(\mathbf{S}_T)]| \leq C|\Delta|^{\gamma}$$

provided that $|\Delta|$ is small enough.

Remark:

1. If your main objective it the pricing of derivatives, as we placed ourself in, then weak convergence is usually enough for our purpose. Indeed, you might think of the class $\mathcal{G}$ appearing in the definition of the weak convergence as a class of pay-off functions. Then Equation (2) tells us that in order to compute $\pi_0(H, \mathbf{S}) = \mathbb{E}_{\mathbb{Q}}[H(\mathbf{S}_T)]$ then we

can first approximate $\mathbb{E}_{\mathbb{Q}}[H(\mathbf{S}_T)]$ by $\mathbb{E}_{\mathbb{Q}}[H(\hat{\mathbf{S}}_T^{\Delta})]$ for small Δ, and then apply the Monte Carlo Method to approximate $\mathbb{E}_{\mathbb{Q}}[H(\hat{\mathbf{S}}_T^{\Delta})]$ by:

$$\frac{1}{N}\sum_{i=1}^{N} H(\hat{\mathbf{S}}_T^{\Delta,(i)}),$$

where $\mathbf{S}_T^{\Delta,(1)}, \ldots, \mathbf{S}_T^{\Delta,(N)}$ are independent samples of $\mathbf{S}_T^{\Delta}$.

2. Even if the pay-off function H is not in the class $\mathcal{G}$, then weak convergence is often still sufficient for pricing purposes. Indeed, if the class $\mathcal{G}$ consists out of the Fourier basis elements, i.e. $\mathcal{G} = \{x \mapsto \mathrm{e}^{\mathrm{i}ux} \colon u \in \mathbb{R}\}$ then weak convergence means convergence of the characteristic functions and therefore we are talking about convergence in distribution. Since the price π_0 is given as the (discounted) risk-neutral expectation of the pay-off $H(\mathbf{S})$, and the expectation is only dependend on the distribution, this shows that weak convergence is enough for the pricing purpose.

In the following we assume a univariate model, where the stock price process $\mathbf{S} = (S_t)_{t\in[0,T]}$ is given by the following SDE:

$$dS_t = \mu(t, S_t)dt + \sigma(t, S_t)dW_t,$$

where $(W_t)_{t\in[0,T]}$ denotes a Brownian motion, μ is the drift term and σ the instantaneous volatility.

We simulate $(S_t)_{t\in[0,T]}$ over the time interval $[0, T]$, which we assume to be discretized as

$$0 = t_1 < t_2 < \ldots < t_m = T,$$

where the time increments are equally spaced with width Δt.

Remark: Equally-spaced time increments are primarily used for notational convenience, because it allows us to write $t_i - t_{i-1}$ as simply Δt. All the results derived with equally-spaced increments are easily generalized to unequal spacing.

Integrating $dS_t = \mu(t, S_t)dt + \sigma(t, S_t)dW_t$ from t to $t + \Delta t$ yields the following integral equation:

$$S_{t+\Delta t} = S_t + \int_t^{t+\Delta t} \mu(u, S_u)du + \int_t^{t+\Delta t} \sigma(u, S_u)dW_u,$$

which will be our starting point for any discretization scheme. We assume that at time t the value of S_t is known, and we wish to obtain the next value $S_{t+\Delta t}$!

3.3| The Euler Scheme

The simplest way to discretize the process in Equation (2) is to use Euler discretization. This is equivalent to approximating the integrals using the **left-point rule**.

Hence the first integral is approximated as the product of the integrand at time t, and the integration range Δt

$$\int_t^{t+\Delta t} \mu(S_u, u)du \approx \mu(S_t, t) \int_t^{t+\Delta t} du = \mu(S_t, t)\Delta t.$$

We use the left-point rule since at time t the value $\mu(S_t, t)$ is known.

The second integral is approximated as

$$\int_t^{t+\Delta t} \sigma(S_u, u)dW_u \approx \sigma(S_t, t)(W_{t+\Delta t} - W_t) = \sigma(S_t, t)\sqrt{\Delta t}Z,$$

where Z is a standard normal variable.

Note that $W_{t+\Delta t} - W_t$ and $\sqrt{\Delta t}Z$ are identical in distribution.

We therefore define the **Euler discretization** of the SDE takes as follows:

$$\hat{S}_{t+\Delta t} = \hat{S}_t + \mu(\hat{S}_t, t)\Delta t + \sigma(\hat{S}_t, t)\sqrt{\Delta t}Z. \quad (1)$$

Remark:

It can be shown that the Euler-Maruyama scheme has a strong convergence rate of $\mathcal{O}(\Delta t)$ in the time step size Δt. This implies that the error of the numerical solution decreases as the square root of the time step size. In the following sections, we explore opportunities for schemes to perform better than that.

3.3.1| Euler Scheme for the Black-Scholes Model

The Black-Scholes stock price dynamics under the risk neutral measure are

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

Following (1) the Euler discretization of the Black-Scholes model is thus given by:

$$\hat{S}_{t+\Delta t} = \hat{S}_t + r\hat{S}_t\Delta t + \sigma\hat{S}_t\sqrt{\Delta t}Z.$$

As an alternative, we can generate log-stock prices, and exponentiate the result.

Indeed, using Itô's lemma (see Part I of this course) $\ln S_t$ is given by:

$$d\ln S_t = \left(r - \frac{1}{2}\sigma^2\right)dt + \sigma dW_t.$$

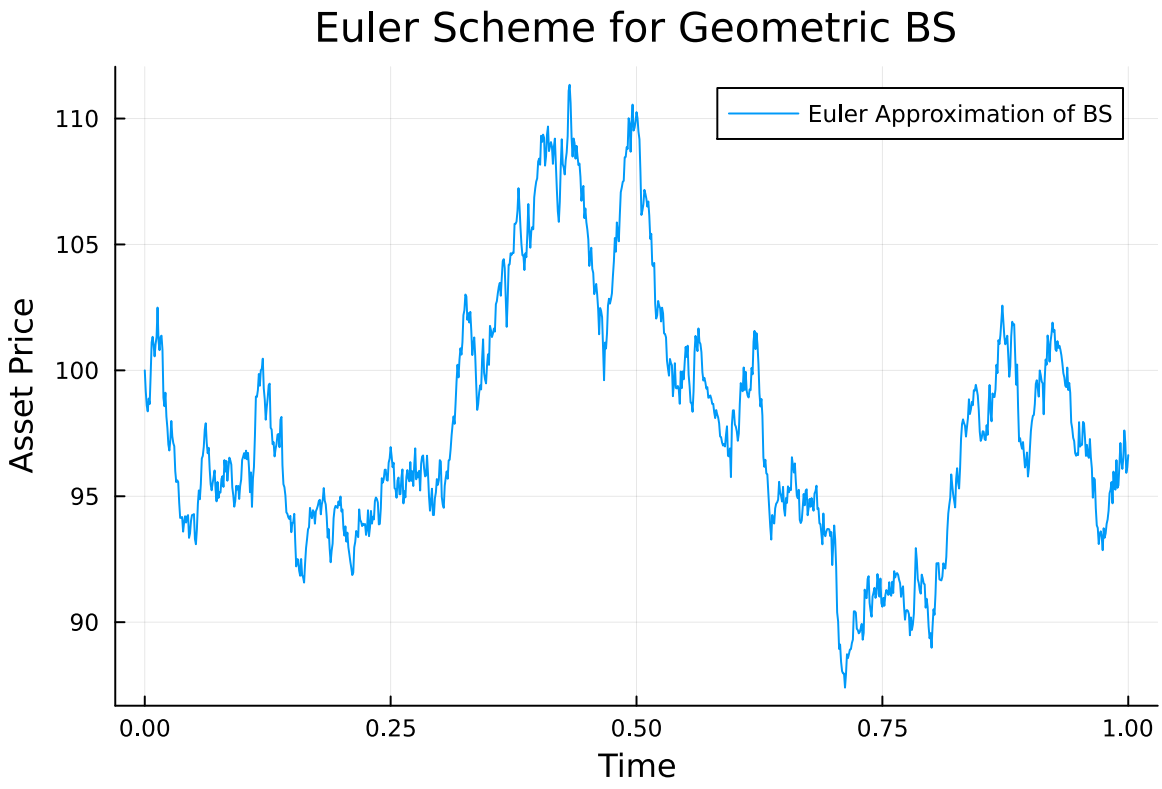
Euler discretization via Equation (1) yields:

$$\ln \hat{S}_{t+\Delta t} = \ln \hat{S}_t + \left(r - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t}Z$$

so that

$$\hat{S}_{t+\Delta t} = \hat{S}_t \exp\left(\left(r - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t}Z\right).$$

In [72]:
```
plot(time, S, label="Euler Approximation of BS", xlabel="Time", ylabel="Asse
```



3.3.2| Euler Scheme for the Heston Model

Recall that the Heston model is described by the bivariate stochastic process for the stock price S_t and its variance V_t as follows:

$$dS_t = rS_t\Delta t + \sqrt{V_t}S_t dW_t^{(1)}$$
$$dV_t = \kappa(\theta - V_t)\Delta t + \xi\sqrt{V_t}dW_t^{(2)}$$

where $\mathbb{E}_{\mathbb{Q}}[dW_t^{(1)}dW_t^{(2)}] = \rho\Delta t$ for some correlation parameter $\rho \in [-1, +1]$.

Discretization of the variance process V

In contrast to the Black-Scholes model, we also have to approximate the variance process in the Heston model:

The SDE for $(V_t)_{t\in[0,T]}$ reads in integral form as

$$V_{t+\Delta t} = V_t + \int_t^{t+\Delta t} \kappa(\theta - V_u)du + \int_t^{t+\Delta t} \xi\sqrt{V_u}dW_u^{(2)}.$$

The Euler discretization approximates the integrals using the left-point rule

$$\int_t^{t+\Delta t} \kappa(\theta - V_u)du \approx \kappa(\theta - V_t)\Delta t$$
$$\int_t^{t+\Delta t} \xi\sqrt{V_u}dW_u^{(2)} \approx \xi\sqrt{V_t}(W_{t+\Delta t} - W_t) = \xi\sqrt{V_t\Delta t}Z_V$$

where Z_V is a standard normal random variable.

Truncating negative variance

The Euler discretization of the variance is therefore given by:

$$\hat{V}_{t+\Delta t} = \hat{V}_t + \kappa(\theta - \hat{V}_t)\Delta t + \xi\sqrt{\hat{V}_t\Delta t}Z_V.$$

Note: We can not rule out that $\hat{V}_t$ assumes negative values, which invalids the square-root computation. This motivates the use of truncation schemes. In the **full truncation scheme** we replace the latter $\hat{V}_t$ by $\hat{V}_t^+ \triangleq \max(0, \hat{V}_t)$, whereas in the **reflection scheme** one replaces the latter twp $\hat{V}_t$ by its absolute value $|\hat{V}_t|$.

Discretization of the asset's price process S

The SDE for S_t is written in integral form as

$$S_{t+\Delta t} = S_t + r\int_t^{t+\Delta t} S_u du + \int_t^{t+\Delta t} \sqrt{V_u}S_u dW_u.$$

Euler discretization approximates the integrals with the left-point rule

$$\int_t^{t+\Delta t} S_u du \approx S_t\Delta t$$
$$\int_t^{t+\Delta t} \sqrt{V_u}S_u dW_{1,u} \approx \sqrt{V_t}S_t(W_{t+\Delta t} - W_t) = \sqrt{V_t\Delta t}S_t Z_S$$

where Z_S is a standard normal random variable that has correlation ρ with Z_V. We therefore end up with:

$$\hat{S}_{t+\Delta t} = \hat{S}_t + r\hat{S}_t\Delta t + \sqrt{V_t\Delta t}\hat{S}_t Z_S.$$

Discretization of the asset's log-price process $\ln S$

By Itô's lemma $\ln S_t$ follows the diffusion

$$d \ln S_t = \left(r - \frac{1}{2} V_t \right) \Delta t + \sqrt{V_t} dW_t^{(1)}$$

or in integral form

$$\ln S_{t+\Delta t} = \ln S_t + \int_0^t \left(r - \frac{1}{2} V_u \right) du + \int_0^t \sqrt{V_u} dW_u^{(1)}.$$

Euler discretization of the process for $\ln S_t$ is thus

$$\ln S_{t+\Delta t} = \ln S_t + \left(r - \frac{1}{2} V_t \right) \Delta t + \sqrt{V_t} \Delta t Z_S.$$

Hence the Euler discretization of S_t is

$$\hat{S}_{t+\Delta t} = \hat{S}_t \exp\left(\left(r - \frac{1}{2} \hat{V}_t \right) \Delta t + \sqrt{\hat{V}_t \Delta t} Z_S \right),$$

where to avoid negative variances we must apply the full truncation or reflection scheme by replacing V_t everywhere by V_t^+ or $|V_t|$.

Algorithm: Euler Scheme for (S, V) or $(\ln S, V)$ with full truncation

1. **Initialization**: Start with the initial values S_0 for the stock price and V_0 for the variance; specify all parameters.

2. **Iteration**: For each time step from t to $t + \Delta t$, perform the following steps:

   a. **Update Variance**: Compute $\hat{V}_{t+\Delta t}$ using the formula:

   $$\hat{V}_{t+\Delta t} = \hat{V}_t + \kappa(\theta - \hat{V}_t^+)\Delta t + \xi \sqrt{\hat{V}_t^+ \Delta t} Z_V,$$

   b. **Update Stock Price**: Obtain $S_{t+\Delta t}$ using either of the following formulas:

   For the arithmetic model:

   $$\hat{S}_{t+\Delta t} = \hat{S}_t + r \hat{S}_t \Delta t + \sqrt{\hat{V}_t^+ \Delta t} \hat{S}_t Z_S,$$

   or for the geometric model:

   $$\hat{S}_{t+\Delta t} = \hat{S}_t \exp\left(\left(r - \frac{1}{2} \hat{V}_t^+ \right) \Delta t + \sqrt{\hat{V}_t^+ \Delta t} Z_S \right).$$

3. **Generate Correlated Random Variables**:

   - To generate Z_V and Z_S with correlation ρ, first generate two independent standard normal variables Z_1 and Z_2.

- Then set $Z_V = Z_1$ and $Z_s = \rho Z_1 + \sqrt{1-\rho^2} Z_2$.

3.4| Milstein Scheme

The Milstein scheme works for time-homogeneous SDEs, i.e. SDEs with coefficients $\mu_t = \mu(S_t)$ and $\sigma_t = \sigma(S_t)$ depend only on S_t and not explicitly on t.

Therefore, we assume in the following that the stock price S is driven by the SDE:

$$\begin{aligned} dS_t &= \mu(S_t)dt + \sigma(S_t)dW_t \\ &= \mu_t dt + \sigma_t dW_t, \end{aligned}$$

which in integral form is equivalent to:

$$S_{t+\Delta t} = S_t + \int_t^{t+\Delta t} \mu_s ds + \int_t^{t+\Delta t} \sigma_s dW_s$$

The key idea of the Milstein scheme is based on the idea to enhace the accuracy of the discretization by expanding the coefficients $\mu_t = \mu(S_t)$ and $\sigma_t = \sigma(S_t)$ via Itô's lemma.

Assume that μ and σ are sufficiently regular. Then by Itô's formula, we have:

$$d\mu(S_t) = \left(\mu'(S_t)\mu(S_t) + \frac{1}{2}\mu''(S_t)\sigma(S_t)^2 \right) dt + (\mu'(S_t)\sigma(S_t))dW_t$$

$$d\sigma(S_t) = \left(\sigma'(S_t)\mu(S_t) + \frac{1}{2}\sigma''(S_t)\sigma(S_t)^2 \right) dt + (\sigma'(S_t)\sigma(S_t))dW_t$$

where the prime refers to differentiation in S and where the derivatives in t are zero, as we assumed that μ_t and σ_t have no direct dependence on t!

The integral form of the coefficients at time s (with $t < s < t + dt$)

$$\mu(S_s) = \mu(S_t) + \int_t^s \left(\mu'(S_u)\mu(S_u) + \frac{1}{2}\mu''(S_u)\sigma(S_u)^2 \right) du + \int_t^s (\mu'(S_u)\sigma(S_u))dW_u$$

$$\sigma(S_s) = \sigma(S_t) + \int_t^s \left(\sigma'(S_u)\mu(S_u) + \frac{1}{2}\sigma''(S_u)\sigma(S_u)^2 \right) du + \int_t^s (\sigma'(S_u)\sigma(S_u))dW_u$$

Substitute for μ_s and σ_s to obtain:

$$\begin{aligned} S_{t+\Delta t} = S_t &+ \int_t^{t+\Delta t} \Big(\mu(S_s) + \int_t^s \big(\mu'(S_u)\mu(S_u) + \frac{1}{2}\mu''(S_u)\sigma(S_u)^2 \big) du + \int_t^s (\mu'(S_u) \\ &+ \int_t^{t+\Delta t} \big(\sigma(S_s) + \int_t^s \left(\sigma'(S_u)\mu(S_u) + \frac{1}{2}\sigma''(S_u)\sigma(S_u)^2 \right) du + \int_t^s (\sigma'(S_u) \end{aligned}$$

The terms higher than order one are $dsdu = \mathcal{O}((dt)^2)$ and $dsdW_u = \mathcal{O}((dt)^{3/2})$ and will be ignored.

The term involving $dW_u dW_s$ is retained since $dW_u dW_s = \mathcal{O}(dt)$ is of order one. This leaves us with:

$$S_{t+\Delta t} \approx S_t + \int_t^{t+\Delta t} \mu(S_s)ds + \int_t^{t+\Delta t} \sigma(S_s)dW_s + \int_t^{t+\Delta t} \int_t^s (\sigma'(S_u)\sigma(S_u))dW_u dW$$

Apply Euler discretization to the last term to obtain

$$\int_t^{t+\Delta t} \int_t^s \sigma'(S_u)\sigma(S_u)dW_u dW_s \approx \sigma'_t\sigma(S_t) \int_t^{t+\Delta t} \int_t^s dW_u dW_s = \sigma'(S_t)\sigma(S_t) \int_t^{t+}$$

Now define $dY_t \triangleq W_t dW_t$. Using Itô's Lemma, it is easy to show that Y_t has solution $Y_t = \frac{1}{2}(W_t^2 - t)$ so that

$$\int_t^{t+\Delta t} W_s dW_s = Y_{t+\Delta t} - Y_t = \frac{1}{2}W_{t+\Delta t}^2 - \frac{1}{2}W_t^2 - \frac{1}{2}\Delta t.$$

Substitute this back to obtain

$$\int_t^{t+\Delta t} \int_t^s \sigma'(S_u)\sigma_u dW_u dW_s \approx \frac{1}{2}\sigma'(S_u)\sigma(S_u)\left((\Delta W_t)^2 - \Delta t\right)$$

where $\Delta W_t = W_{t+\Delta t} - W_t$, which is equal in distribution to $\sqrt{\Delta t}Z$ with Z distributed as standard normal. Combining the equations above, the general form of Milstein discretization is therefore:

$$\hat{S}_{t+\Delta t} = \hat{S}_t + \mu(\hat{S}_t)\Delta t + \sigma(\hat{S}_t)\sqrt{\Delta t}Z + \frac{1}{2}\sigma'(\hat{S}_t)\sigma(\hat{S}_t)\Delta t(Z^2 - 1).$$

Remark: Note here that the Milstein scheme includes a correction for the change in the diffusion term, i.e., the stochastic integral term in the SDE. This is particularly suitable for equations where the volatility coefficients depend on the state variable or a stochastic volatility. It can be shown that the Milstein scheme has a strong convergence rate of $\mathcal{O}(\Delta t)$ for SDEs with Lipschitz continuous coefficients. This represents a significant improvement over the Euler-Maruyama scheme which has $\mathcal{O}(\sqrt{\Delta t})$, under the condition that the diffusion coefficient's derivative with respect to the state variable is also Lipschitz continuous.

3.4.1| Milstein Scheme for the Black-Scholes Model

In the Black-Scholes model we have $\mu(S_t) = rS_t$ and $\sigma(S_t) = \sigma S_t$, so the Milstein scheme becomes:

$$\hat{S}_{t+\Delta t} = \hat{S}_t + r\hat{S}_t\Delta t + \sigma\hat{S}_t\sqrt{\Delta t}Z + \frac{1}{2}\sigma^2\hat{S}_t\Delta t(Z^2 - 1)$$

which adds the correction term $\frac{1}{2}\sigma^2\hat{S}_t\Delta t(Z^2-1)$ to the Euler scheme from before.

In the Black-Scholes model for the log-stock price, we have $\mu(S_t) = r - \frac{1}{2}\sigma^2$ and $\sigma(S_t) = \sigma$

so that $\mu'(S_t) = \sigma'(S_t) = 0$. The Milstein scheme is therefore

$$\ln \hat{S}_{t+\Delta t} = \ln \hat{S}_t + \left(r - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t}Z$$

which is identical to the Euler scheme from before. Hence, while the Milstein scheme improves the discretization of S_t in the Black-Scholes model, it does not improve the discretization of $\ln S_t$!

3.4.2| Milstein Scheme for the Heston Model

Recall the Heston Model from before, given by:

$$\begin{aligned} dS_t &= rS_t\Delta t + \sqrt{V_t}S_t\Delta W_t^{(1)} \\ dV_t &= \kappa(\theta - V_t)\Delta t + \xi\sqrt{V_t}\Delta W_t^{(2)} \end{aligned}$$

Process for V

The coefficients of the variance process are $\mu(V_t) = \kappa(\theta - V_t)$ and $\sigma(V_t) = \xi\sqrt{V_t}$ so an application of the Milstein Scheme for V_t produces

$$\hat{V}_{t+\Delta t} = \hat{V}_t + \kappa(\theta - \hat{V}_t)\Delta t + \xi\sqrt{\hat{V}_t\Delta t}Z_V + \frac{1}{4}\xi^2\Delta t(Z_V^2 - 1)$$

which can be written

$$\hat{V}_{t+\Delta t} = \left(\sqrt{\hat{V}_t} + \frac{1}{2}\xi\Delta t Z_V\right)^2 + \kappa(\theta - \hat{V}_t)\Delta t - \frac{1}{4}\xi^2\Delta t$$

The Milstein discretization of the variance process produces far fewer negative values for the variance than Euler discretization. Nevertheless, the full truncation scheme or the reflection scheme must be applied to this scheme as well!

Process for S and $\ln S$

The coefficients of the stock price process are $\mu(S_t) = rS_t$ and $\sigma(S_t) = \sqrt{V_t}S_t$ so we obtain:

$$\hat{S}_{t+\Delta t} = \hat{S}_t + r\hat{S}_t\Delta t + \sqrt{\hat{V}_t\Delta t}\hat{S}_t Z_S + \frac{1}{2}\hat{V}_t\hat{S}_t\Delta t(Z_S^2 - 1)$$

We can also discretize the log-stock process, which by Itô's lemma follows the process

$$d\ln\hat{S}_t = \left(r - \frac{1}{2}\hat{V}_t\right)\Delta t + \sqrt{\hat{V}_t}\Delta W_t^{(1)}$$

Since V_t is known at time t, we can treat it as a constant in the coefficients. An application of the Milstein Scheme yields:

$$\ln\hat{S}_{t+\Delta t} = \ln\hat{S}_t + \left(r - \frac{1}{2}\hat{V}_t\right)\Delta t + \sqrt{\hat{V}_t\Delta t}Z_S$$

which is identical to what we had before. Hence, as in the Black-Scholes model, the discretization of $\ln S_t$ rather than S_t means that there are no higher corrections to be brought to the Euler discretization. The discretization of the stock price is

$$\hat{S}_{t+dt} = \hat{S}_t\exp\left(\left(r - \frac{1}{2}\hat{V}_t\right)dt + \sqrt{\hat{V}_t\Delta t}Z_S\right).$$

Again, it is necessary to apply the full truncation or reflections schemes here!

Algorithm: Milstein Scheme for (S, V) or $(\ln S, V)$ with full truncation

Given a value for $\hat{V}_t$ at time t, the process to update to $\hat{V}_{t+dt}$ and obtain $\hat{S}_{t+dt}$ is as follows:

1. **Update Variance**: Update to $V_{t+\Delta t}$ using the equation:

   $$\hat{V}_{t+\Delta t} = \hat{V}_t + \kappa(\theta - \hat{V}_t^+)\Delta t + \xi\sqrt{\hat{V}_t^+\Delta t}Z_V + \frac{1}{4}\xi^2\Delta t(Z_V^2 - 1)$$

   where κ is the rate of mean reversion, θ is the long-term variance, ξ is the volatility of volatility, and Z_V is a random variable from the correlated standard normal distribution.

2. **Update Stock Price**: Obtain $\hat{S}_{t+\Delta t}$ using either of the following formulas:

   - For the arithmetic model:

     $$\hat{S}_{t+\Delta t} = \hat{S}_t + r\hat{S}_t\Delta t + \sqrt{\hat{V}_t^+\Delta t}\hat{S}_t Z_S + \frac{1}{2}\hat{V}_t^+\hat{S}_t\Delta t(Z_S^2 - 1)$$

     where r is the risk-free rate.

- For the geometric model:

$$\hat{S}_{t+\Delta t} = \hat{S}_t \exp\left(\left(r - \frac{1}{2}\hat{V}_t^+\right)\Delta t + \sqrt{\hat{V}_t^+ \Delta t} Z_S\right).$$

3. **Generate Correlated Random Variables**:

   - To generate Z_V and Z_S with correlation ρ, first generate two independent standard normal variables Z_1 and Z_2.
   - Set $Z_V = Z_1$ and $Z_S = \rho Z_1 + \sqrt{1-\rho^2} Z_2$.

References

1. P. E. Kloeden & E. Platen (1992). *Numerical Solution of Stochastic Differential Equations*. Springer.

2. D. J. Higham (2001). "An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations." *SIAM Review*, 43(3), 525-546.

3. Christian Bayer, Antonis Papapantoleon, Raul Tempone (2018). *Computational Finance*. Version of June 25, 2018.

4. Paul Glasserman (2004). *Monte Carlo Methods in Financial Engineering*. Springer.