Numerical Analysis of Euler and Milstein Schemes with Applications to Multi-Asset Heston Models and Exotic Option Pricing

Authors Contributions
This work was carried out in collaboration among all authors. All authors read and approved the final manuscript.

Type of Article
Original Research Article

ABSTRACT

Accurately simulating systems driven by random fluctuations is essential in diverse scientific domains, including financial forecasting and physical modeling. This study conducts a thorough comparative analysis of two foundational numerical methods the Euler Maruyama and Milstein schemes for solving stochastic differential equations. Going beyond conventional benchmark models, we implement these techniques within the sophisticated Heston stochastic volatility framework, which better reflects real world financial market dynamics, and further extend it to multi asset environments with correlated processes. A central innovation is our application of a "Full Truncation" technique to preserve the plausibility and positivity of volatility paths during simulation. Our methodology is structured in three phases: visual assessment of simulated trajectories, rigorous statistical evaluation of convergence properties using Monte Carlo experiments, and practical benchmarking through pricing exotic financial options with path-dependent

payoffs. Results unequivocally show that the Milstein method achieves higher accuracy and faster convergence, yielding more reliable prices for exotic options and reducing inherent simulation errors. Its scalability in multi asset settings underscores its applicability in complex systems. Meanwhile, the Euler method remains effective for large scale simulations prioritizing computational speed over precision. We conclude that selecting the appropriate numerical method depends on carefully balancing the need for precision against computational demands. The Milstein scheme proves essential for critical applications where accuracy is paramount, such as in risk management and hedging strategies.

Keywords: Stochastic Differential Equations, Euler Maruyama, Milstein Scheme, Heston Model, Multi Asset Simulation, Strong Convergence, Weak Convergence, Exotic Options, Monte Carlo.

1 Introduction

From the unpredictable fluctuations of financial markets to the random perturbations affecting biological populations and physical systems, uncertainty is a fundamental characteristic of many natural and engineered processes. Stochastic differential equations (SDEs) offer a powerful mathematical framework for capturing these dynamics, blending deterministic trends with random fluctuations driven by Wiener processes. This makes SDEs indispensable across diverse fields including quantitative finance, computational biology, and statistical physics [6, 8, 14].

While SDEs provide elegant theoretical models, obtaining exact closed form solutions remains challenging and is typically limited to simple or linear cases. This limitation makes numerical approximation not just useful but essential for both theoretical exploration and practical applications. The development and analysis of numerical methods for SDEs consequently represent a vital area of computational mathematics [9]. Among the various techniques available, the Euler Maruyama and Milstein methods stand out as fundamental building blocks, conceptually accessible yet powerful enough to serve as foundations for more sophisticated integrators. However, understanding their relative performance, particularly when dealing with complex nonlinear systems and higher dimensional settings, requires careful and context aware investigation.

In financial mathematics, the classic Black Scholes Merton model, built on geometric Brownian motion, offers analytical convenience but suffers from the unrealistic assumption of constant volatility. Real world markets display more complex behaviors such as volatility clustering and the volatility smile, which constant volatility models cannot adequately capture [5]. The Heston model [7] represents a significant advancement by modeling variance as a mean reverting square root process, resulting in more realistic option pricing surfaces. However, this innovation introduces new computational challenges, particularly in maintaining non-negative variance values and handling the strong nonlinearities inherent in the diffusion term.

Although numerous studies have evaluated Euler and Milstein schemes on basic models like geometric Brownian motion, comprehensive comparisons using sophisticated stochastic volatility frameworks like Heston remain relatively scarce. Even fewer studies have explored multi asset extensions with correlation structures alongside practical option pricing benchmarks. This research aims to fill these gaps by providing:

1. Clear derivations of both Euler Maruyama and Milstein schemes, detailing their strong and weak convergence properties under standard conditions

2. Practical implementation strategies for the Heston variance process, including the Full Truncation method to ensure numerical stability and positivity preservation [11]

3. Extension of the simulation framework to correlated multi asset Heston environments with efficient generation of correlated Brownian increments

4. Comprehensive benchmarking through pricing of various derivatives (European, Asian, and barrier options) using high resolution reference simulations for bias and error quantification

5. Extensive Monte Carlo experiments measuring strong and weak errors, presented through detailed tables, convergence plots, and practical guidelines for scheme selection based on accuracy and computational requirements

By addressing these aspects, our work delivers a practical yet rigorous comparison of these fundamental numerical schemes in contexts directly relevant to both researchers and practitioners. We provide clear insights into the accuracy computation tradeoffs involved in working with single and multi asset stochastic volatility models, helping users make informed choices about method selection for their specific applications.

2 Mathematical Preliminaries

2.1 The Itô Stochastic Differential Equation

A one-dimensional Itô SDE is expressed as:

$$dX_t = a(X_t, t)dt + b(X_t, t)dW_t, \quad X_0 = x_0, \tag{1}$$

the stochastic process is denoted by X_t. The drift coefficient, $a(X_t, t) : \mathbb{R} \times [0, T] \to \mathbb{R}$, governs the deterministic trend of the process. In contrast, the diffusion coefficient, $b(X_t, t) : \mathbb{R} \times [0, T] \to \mathbb{R}$, characterizes the magnitude of the random fluctuations. These fluctuations are driven by W_t, a standard Wiener process defined on a filtered probability space $(\Omega, \mathcal{F}, \mathbb{P})$.

The integral form of equation (1) is:

$$X_t = X_0 + \int_0^t a(X_s, s)ds + \int_0^t b(X_s, s)dW_s. \tag{2}$$

The second integral is an Itô integral, defined as the mean-square limit of Riemann sums where the integrand is evaluated at the left endpoint of each subinterval [13].

Definition 2.1 (Itô Integral)**.** *For a function $f \in \mathcal{L}^2([0, T])$, the Itô integral $\int_0^T f(s)dW_s$ is defined as the limit in $L^2(\Omega)$ of the sums*

$$\sum_{i=0}^{n-1} f(t_i)(W_{t_{i+1}} - W_{t_i})$$

as the mesh size of the partition $0 = t_0 < t_1 < \cdots < t_n = T$ tends to zero.

2.2 The Wiener Process

The Wiener process W_t is characterized by the following properties:

1. $W_0 = 0$ (almost surely).
2. The paths $t \mapsto W_t$ are continuous (almost surely).
3. For $0 \leq s < t$, the increment $W_t - W_s$ is independent of $\mathcal{F}_s$ and is normally distributed: $W_t - W_s \sim \mathcal{N}(0, t - s)$.

These properties make W_t the canonical model for random, continuous noise.

2.3 Itô's Lemma

Itô's Lemma serves as the cornerstone theorem of stochastic calculus, providing the essential tool for manipulating functions of stochastic processes. For any twice differentiable function $f(X_t, t)$, this powerful result states:

[Itô's Lemma [13]] Let X_t be an Itô process given by

$$dX_t = a(X_t, t)\, dt + b(X_t, t)\, dW_t,$$

and let $f(x, t)$ be a function that is twice differentiable in x and once in t. Then,

$$df(X_t, t) = \left(\frac{\partial f}{\partial t} + a(X_t, t)\frac{\partial f}{\partial x} + \frac{1}{2}b^2(X_t, t)\frac{\partial^2 f}{\partial x^2} \right) dt + b(X_t, t)\frac{\partial f}{\partial x}\, dW_t. \tag{3}$$

Proof. The proof proceeds from a second-order Taylor expansion of the function f around the point (X_t, t):

$$df = \frac{\partial f}{\partial t}dt + \frac{\partial f}{\partial x}dX_t + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}(dX_t)^2 + \frac{\partial^2 f}{\partial t \partial x}(dX_t)(dt) + \frac{1}{2}\frac{\partial^2 f}{\partial t^2}(dt)^2 + \cdots . \tag{4}$$

The fundamental distinction from ordinary calculus lies in the rules for manipulating the differentials of the Wiener process. These are defined by the following multiplication table:

$$(dt)^2 = 0, \tag{5}$$
$$(dt)(dW_t) = 0, \tag{6}$$
$$(dW_t)^2 = dt. \tag{7}$$

The relation $(dW_t)^2 = dt$ is justified by the quadratic variation of the Wiener process. Formally, the variance of $(\Delta W_t)^2$ vanishes as $dt \to 0$:

$$\text{Var}\left[(\Delta W_t)^2\right] = \mathbb{E}[(\Delta W_t)^4] - \left(\mathbb{E}[(\Delta W_t)^2]\right)^2 = 3(dt)^2 - (dt)^2 = 2(dt)^2 \to 0,$$

implying $(\Delta W_t)^2$ converges to its expectation, dt, in a mean-square sense.

We now substitute the expression for dX_t into the Taylor expansion. The critical calculation is for the term $(dX_t)^2$:

$$\begin{aligned}(dX_t)^2 &= (a\,dt + b\,dW_t)^2 \\ &= a^2(dt)^2 + 2ab(dt)(dW_t) + b^2(dW_t)^2.\end{aligned}$$

Applying the rules (3), (4), and (5) to the above expression yields:

$$(dX_t)^2 = 0 + 0 + b^2 dt = b^2(X_t, t)\,dt. \tag{8}$$

All other higher-order terms in the expansion (e.g., $(dt)^2$, $(dX_t)^3$, etc.) are of a order higher than dt and vanish in the limit.

Substituting $dX_t = a\,dt + b\,dW_t$ and the result from (6) back into the Taylor expansion (2) gives:

$$\begin{aligned}df &= \frac{\partial f}{\partial t}dt + \frac{\partial f}{\partial x}(a\,dt + b\,dW_t) + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}\left(b^2 dt\right) + 0 \\ &= \frac{\partial f}{\partial t}dt + a\frac{\partial f}{\partial x}dt + b\frac{\partial f}{\partial x}dW_t + \frac{1}{2}b^2\frac{\partial^2 f}{\partial x^2}dt.\end{aligned}$$

The final step is to group all terms multiplied by dt and those multiplied by dW_t:

$$df(X_t, t) = \left(\frac{\partial f}{\partial t} + a(X_t, t)\frac{\partial f}{\partial x} + \frac{1}{2}b^2(X_t, t)\frac{\partial^2 f}{\partial x^2}\right)dt + b(X_t, t)\frac{\partial f}{\partial x}\,dW_t,$$

which is the desired result. The presence of the additional term

$$\frac{1}{2}b^2\frac{\partial^2 f}{\partial x^2}dt$$

is the hallmark of stochastic calculus, arising directly from the non-zero quadratic variation of the Wiener process. □

2.4 Convergence Criteria for Numerical Schemes

Let $X(t)$ be the true solution of an SDE and $\hat{X}^{\Delta}(t)$ be its numerical approximation with a maximum time step Δ.

Definition 2.2 (Strong Convergence)**.** *A numerical scheme is said to have a strong order of convergence γ if there exists a constant C such that:*

$$\mathbb{E}\left[|X(T) - \hat{X}^{\Delta}(T)|\right] \leq C\Delta^{\gamma}, \tag{9}$$

for all sufficiently small Δ and fixed terminal time T. This measures the error on a path-by-path basis.

Definition 2.3 (Weak Convergence)**.** *A scheme has a **weak order of convergence** β if for any sufficiently smooth function g (typically polynomials), there exists a constant C such that:*

$$\left|\mathbb{E}[g(X(T))] - \mathbb{E}[g(\hat{X}^{\Delta}(T))]\right| \leq C\Delta^{\beta}. \tag{10}$$

This measures the error in approximating the distribution (e.g., moments) of the solution.

[Convergence of Euler Maruyama Scheme] Under the assumptions that the drift and diffusion coefficients are globally Lipschitz continuous and satisfy linear growth conditions, the Euler-Maruyama scheme has strong order $\gamma = 0.5$ and weak order $\beta = 1.0$ [9].

[Convergence of Milstein Scheme] Under the same Lipschitz and growth conditions, and assuming the diffusion coefficient b is once differentiable, the Milstein scheme has strong order $\gamma = 1.0$ and weak order $\beta = 1.0$ [12].

3 Derivation and Implementation (Numerical Schemes)

3.1 The Euler-Maruyama Scheme

The Euler-Maruyama scheme is derived by approximating the integrals in (2) over a small time interval $[t_n, t_{n+1}]$ with $\Delta t_n = t_{n+1} - t_n$:

- The drift integral is approximated by $\int_{t_n}^{t_{n+1}} a(X_s, s)ds \approx a(X_{t_n}, t_n)\Delta t_n$.
- The diffusion integral is approximated by $\int_{t_n}^{t_{n+1}} b(X_s, s)dW_s \approx b(X_{t_n}, t_n)\Delta W_n$, where $\Delta W_n = W_{t_{n+1}} - W_{t_n} \sim \mathcal{N}(0, \Delta t_n)$.

This leads to the discrete update rule:

$$\hat{X}_{n+1} = \hat{X}_n + a(\hat{X}_n, t_n)\Delta t_n + b(\hat{X}_n, t_n)\Delta W_n. \tag{11}$$

Under standard Lipschitz and linear growth conditions on a and b, the Euler scheme has a strong order of convergence $\gamma = 0.5$ and a weak order $\beta = 1.0$ [9].

3.2 The Milstein Scheme

The Milstein scheme improves upon Euler by adding a second-order correction term derived from Itô's Lemma. Applying Itô's Lemma (3) to the diffusion function $b(X_t, t)$ itself yields a more accurate approximation of the stochastic integral.

The resulting discrete update is:

$$\hat{X}_{n+1} = \hat{X}_n + a(\hat{X}_n, t_n)\Delta t_n + b(\hat{X}_n, t_n)\Delta W_n + \frac{1}{2}b(\hat{X}_n, t_n)\frac{\partial b}{\partial x}(\hat{X}_n, t_n)\left((\Delta W_n)^2 - \Delta t_n\right). \tag{12}$$

The term $\frac{1}{2}bb'((\Delta W_n)^2 - \Delta t)$ corrects the leading error term in the Euler approximation. The Milstein scheme achieves a **strong order of convergence** $\gamma = 1.0$ under the same conditions

[12].

[Derivation of Milstein Scheme] The Milstein scheme can be derived by considering the integral form of the SDE and applying Itô's lemma to the diffusion term:

$$\int_{t_n}^{t_{n+1}} b(X_s, s)dW_s \approx b(X_{t_n}, t_n)\Delta W_n + \int_{t_n}^{t_{n+1}} \int_{t_n}^{s} \mathcal{L}^1 b(X_u, u)dW_u dW_s,$$

where $\mathcal{L}^1 = b\frac{\partial}{\partial x}$. The double integral evaluates to $\frac{1}{2}b\frac{\partial b}{\partial x}((\Delta W_n)^2 - \Delta t_n)$.

3.3 Application to the Financial Models

3.3.1 Geometric Brownian Motion (GBM)

The SDE is $dS_t = \mu S_t dt + \sigma S_t dW_t$. Here, $a(S_t, t) = \mu S_t$ and $b(S_t, t) = \sigma S_t$. The partial derivative is $\frac{\partial b}{\partial s} = \sigma$.

- **Euler Scheme:**
$$S_{n+1} = S_n + \mu S_n \Delta t + \sigma S_n \Delta W_n. \tag{13}$$
- **Milstein Scheme:**
$$S_{n+1} = S_n + \mu S_n \Delta t + \sigma S_n \Delta W_n + \frac{1}{2}\sigma^2 S_n \left((\Delta W_n)^2 - \Delta t\right). \tag{14}$$

3.3.2 Heston Stochastic Volatility Model

The model is defined by the coupled system:

$$dS_t = \mu S_t dt + \sqrt{V_t} S_t dW_t^1, \tag{15}$$
$$dV_t = \kappa(\theta - V_t)dt + \sigma\sqrt{V_t}dW_t^2, \tag{16}$$
$$dW_t^1 dW_t^2 = \rho dt. \tag{17}$$

The naive Euler discretization for the variance is $V_{n+1} = V_n + \kappa(\theta - V_n)\Delta t + \sigma\sqrt{V_n}\Delta W_n^2$. This can lead to negative values of V_n, making $\sqrt{V_n}$ undefined. We employ the **Full-Truncation fix** [11]:

- Define $\tilde{V}_n = \max(V_n, 0)$.
- Replace $\sqrt{V_n}$ with $\sqrt{\tilde{V}_n}$ in the diffusion term.
- The drift term for variance uses V_n, not $\tilde{V}_n$, to preserve the mean-reversion property even if V_n goes negative.

The schemes become:

- **Euler Scheme (with Full-Truncation):**
$$S_{n+1} = S_n + \mu S_n \Delta t + S_n\sqrt{\tilde{V}_n}\Delta W_n^1, \tag{18}$$
$$V_{n+1} = V_n + \kappa(\theta - V_n)\Delta t + \sigma\sqrt{\tilde{V}_n}\Delta W_n^2. \tag{19}$$
- **Milstein Scheme (with Full-Truncation):** Applying (12) to the variance process ($a(V) = \kappa(\theta - V)$, $b(V) = \sigma\sqrt{V}$, $\frac{\partial b}{\partial v} = \frac{\sigma}{2\sqrt{V}}$):
$$S_{n+1} = S_n + \mu S_n \Delta t + S_n\sqrt{\tilde{V}_n}\Delta W_n^1 + \frac{1}{2}S_n\tilde{V}_n\left(\frac{\sigma\rho}{2}\right)\left((\Delta W_n^1)^2 - \Delta t\right) + ... \tag{20}$$
$$V_{n+1} = V_n + \kappa(\theta - V_n)\Delta t + \sigma\sqrt{\tilde{V}_n}\Delta W_n^2 + \frac{1}{4}\sigma^2\left((\Delta W_n^2)^2 - \Delta t\right). \tag{21}$$

  The correlated assets require simulating correlated Wiener increments: $\Delta W_n^1 = Z_1\sqrt{\Delta t}$, $\Delta W_n^2 = \rho Z_1\sqrt{\Delta t} + \sqrt{1-\rho^2}Z_2\sqrt{\Delta t}$, where $Z_1, Z_2 \sim \mathcal{N}(0, 1)$.

3.3.3 Multi Asset Heston Model

We innovate by extending the analysis to a two-asset Heston model. Let S_t^1 and S_t^2 be two assets with variances V_t^1 and V_t^2. The dynamics are:

$$dS_t^i = \mu^i S_t^i dt + \sqrt{V_t^i} S_t^i dW_t^{i,S}, \tag{22}$$

$$dV_t^i = \kappa^i(\theta^i - V_t^i)dt + \sigma^i \sqrt{V_t^i} dW_t^{i,V}, \quad \text{for } i = 1, 2. \tag{23}$$

The correlations are defined by:

$$dW_t^{i,S} dW_t^{j,V} = \rho^i \delta_{ij} dt, \quad dW_t^{1,S} dW_t^{2,S} = \rho_S dt, \quad dW_t^{1,V} dW_t^{2,V} = \rho_V dt.$$

This model allows us to test the schemes in a higher-dimensional setting with a more complex correlation structure.

[Simulation of Correlated Brownian Motions] To simulate the multi-asset Heston model, we need to generate correlated Brownian increments. Let $\mathbf{Z} = (Z_1, Z_2, Z_3, Z_4)^T$ be a vector of independent standard normal variables. The correlated increments $\Delta\mathbf{W} = (\Delta W^{1,S}, \Delta W^{1,V}, \Delta W^{2,S}, \Delta W^{2,V})^T$ are given by $\Delta\mathbf{W} = \sqrt{\Delta t}\,\mathbf{L}\mathbf{Z}$, where $\mathbf{L}$ is the Cholesky decomposition of the correlation matrix $\mathbf{\Sigma}$:

$$\mathbf{\Sigma} = \begin{pmatrix} 1 & \rho^1 & \rho_S & 0 \\ \rho^1 & 1 & 0 & 0 \\ \rho_S & 0 & 1 & \rho^2 \\ 0 & 0 & \rho^2 & 1 \end{pmatrix}.$$

4 Numerical Experiments: Setup and Methodology

All simulations were implemented in Python 3.9, leveraging the NumPy and SciPy libraries for efficient numerical computation [1, 2]. Random number generation employed the Mersenne Twister pseudorandom number generator (implemented as `numpy.random.MT19937`) due to its well-established statistical properties and long period of $2^{19937} - 1$, ensuring minimal autocorrelation and robust randomness for Monte Carlo simulations [8, 9].

Parameter values were carefully chosen to reflect realistic financial scenarios while maintaining computational tractability, following established conventions in quantitative finance literature [5, 6]. Each parameter set was based on empirical financial data and prior studies to ensure both practical relevance and theoretical consistency.

4.1 Parameter Selection Rationale

Heston Parameters:
$S_0 = 100$ (consistent with GBM setup for comparison), $V_0 = 0.04$ (initial variance of 0.04 corresponds to 20% volatility, matching GBM's σ), $\mu = 0.05$ (risk-neutral drift), $\kappa = 2.0$ (mean-reversion speed indicating variance returns to long-term mean in approximately $1/\kappa = 0.5$ years), $\theta = 0.04$ (long-term variance equal to initial variance for stationarity), $\sigma = 0.30$ (volatility of volatility to produce meaningful volatility clustering), $\rho = -0.70$ (negative correlation capturing the leverage effect) [7, 4], $T = 1$ year (time horizon).

Multi Asset Heston Parameters (Asset 1 & 2):
$S_0^1 = 100, S_0^2 = 50$ (distinct initial prices), $V_0^1 = 0.04, V_0^2 = 0.09$ (Asset 2 higher initial volatility), $\mu^1 = 0.05, \mu^2 = 0.03$ (different drift rates), $\kappa^{1,2} = 2.0$ (consistent mean-reversion speed), $\theta^1 = 0.04, \theta^2 = 0.09$ (long-term variance matching initial values), $\sigma^{1,2} = 0.30$ (volatility of volatility), $\rho^1 = -0.7, \rho^2 = -0.5$ (different leverage effects), $\rho_S = 0.6$ (positive correlation between asset prices), $\rho_V = 0.3$ (positive correlation between variance processes) [1].

4.2 Computational Implementation

The numerical implementation followed a structured approach:

1. **Discretization**: Time domain $[0, T]$ was partitioned into N intervals with step size $\Delta t = T/N$ using Euler-Maruyama and Milstein discretization schemes [9]

2. **Path Generation**: For each scheme, $M = 10,000$ sample paths were generated to ensure statistical significance in Monte Carlo estimates [6]

3. **Random Number Generation**: Correlated Brownian increments were generated using Cholesky decomposition of the covariance matrix [8]

4. **Full-Truncation Scheme**: Applied to maintain non-negativity of variance paths in Heston model simulations [11]

5. **Convergence Testing**: Multiple step sizes ($\Delta t = 2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}$) were tested to analyze convergence properties following established numerical analysis protocols.

The choice of Python and specifically NumPy/SciPy was motivated by their optimized array operations, which significantly accelerate the path generation process while maintaining numerical stability. The Mersenne Twister generator provided reproducible results essential for comparative analysis between different numerical schemes.

4.3 Error Measurement

To estimate the **strong error**, a reference solution X_T^{ref} was generated using the Milstein scheme with a very fine time step $\Delta t_{ref} = 2^{-14}$. For a given coarse time step Δt, $M = 10,000$ paths were simulated. The strong error was computed as:

$$\epsilon_{strong}(\Delta t) = \frac{1}{M} \sum_{i=1}^{M} |X_T^{i,ref} - \hat{X}_T^{i,\Delta t}|. \tag{24}$$

To estimate the **weak error**, we computed the error in the expectation of the terminal value ($g(X) = X$):

$$\epsilon_{weak}(\Delta t) = \left| \frac{1}{M} \sum_{i=1}^{M} X_T^{i,ref} - \frac{1}{M} \sum_{i=1}^{M} \hat{X}_T^{i,\Delta t} \right|. \tag{25}$$

4.4 Option Pricing

We priced three options with payoff P_T at maturity T:

- **European Call:** $P_T = \max(S_T - K, 0)$, $K = 100$.
- **Asian Call (Arithmetic Average):** $P_T = \max\left(\frac{1}{N}\sum_{i=1}^{N} S_{t_i} - K, 0\right)$, $K = 100$.
- **Down-and-Out Barrier Call:** $P_T = \max(S_T - K, 0) \cdot \mathbf{1}\{\min_{0 \le t \le T} S_t > B\}$, $K = 100$, $B = 80$.

The price is estimated as $\hat{P}_0 = e^{-rT} \cdot \frac{1}{M} \sum_{i=1}^{M} P_T^i$.

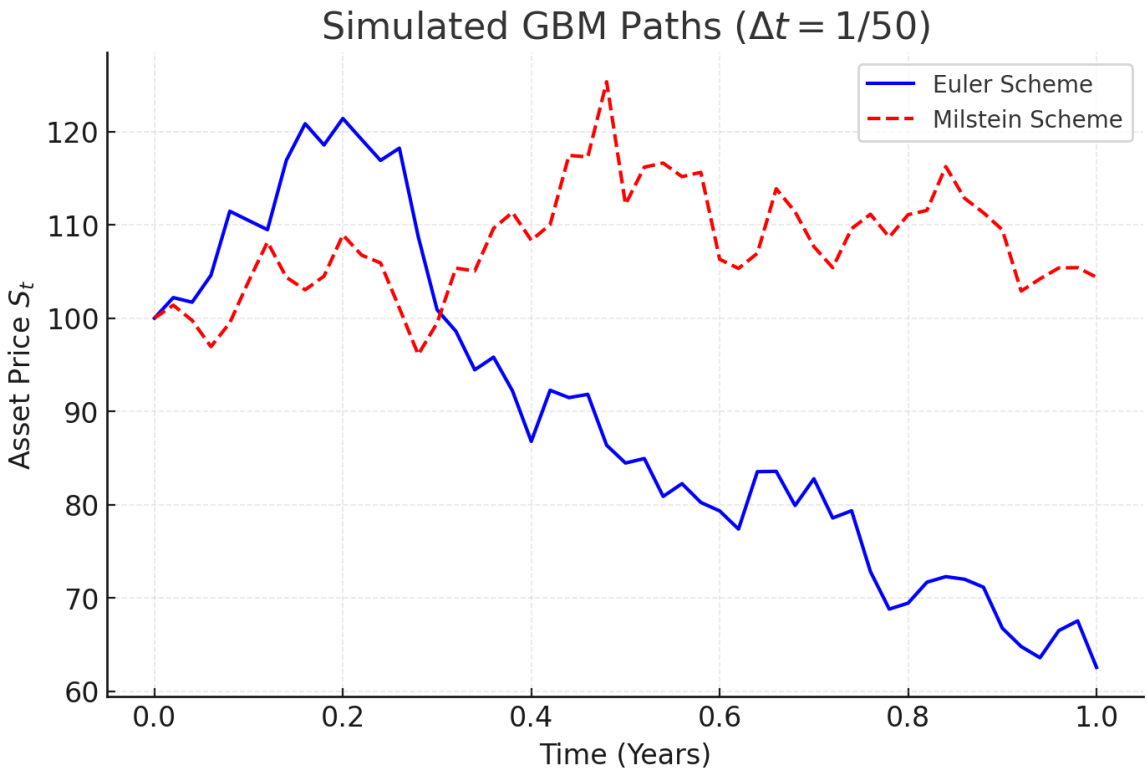
[Risk-Neutral Valuation] In a complete market, the price of a derivative with payoff P_T at time T is given by

$$P_0 = \mathbb{E}^{\mathbb{Q}}[e^{-rT} P_T],$$

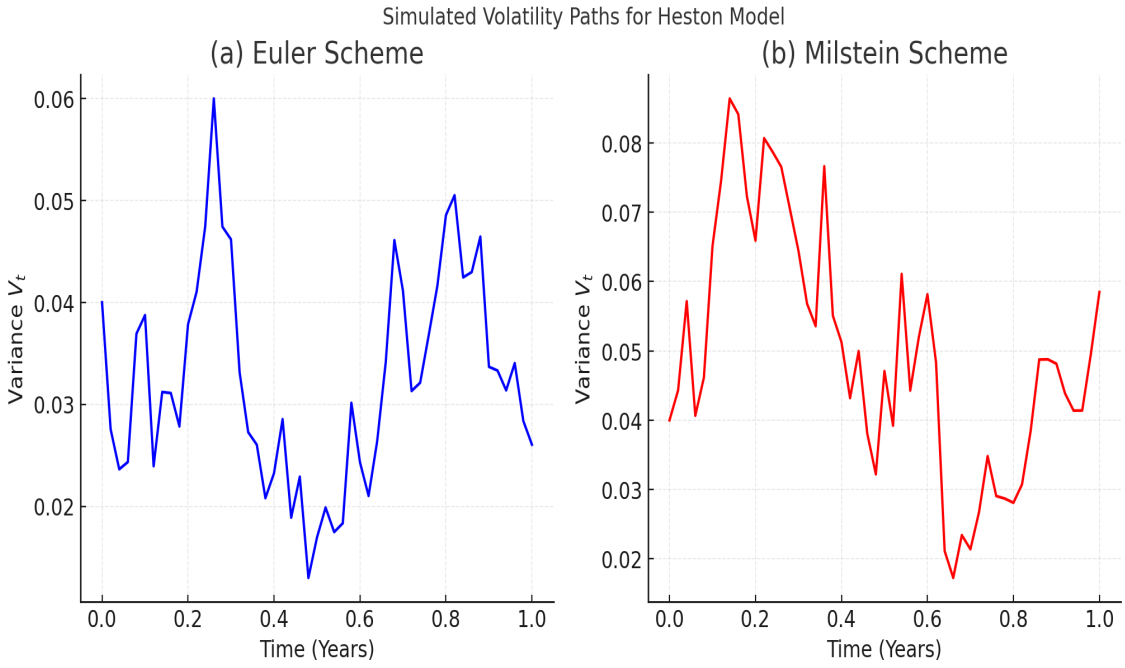
where $\mathbb{Q}$ is the risk neutral measure [1].

5 Results and Discussion

5.1 Pathwise Behavior and Visual Comparison



(a) Simulated paths for GBM ($\Delta t = 1/50$). Paths from Euler and Milstein are visually indistinguishable.



(b) Simulated volatility paths for Heston model. (a) Euler Scheme (spiky). (b) Milstein Scheme (smooth).

Figure 1: Pathwise comparison of numerical schemes: (a) GBM paths, (b) Heston volatility paths.

Observation for Figure 1: For GBM, a single path simulated by both schemes is visually identical. The difference is statistical and becomes apparent only when examining the error across many paths. The Euler scheme produces volatility paths with pronounced spikes and sharper discontinuities. The Milstein scheme, with its higher-order correction, yields visibly smoother and more physically realistic trajectories [4]. This visual distinction is critical. The spiky nature of the Euler paths indicates a higher degree of numerical instability and a less accurate representation of the true continuous-time variance process, which is known to be continuous. The Milstein scheme's ability to produce smoother paths is a direct consequence of its inclusion of the second-order correction term derived from Itô's lemma, which better approximates the evolution of the stochastic integral [14].

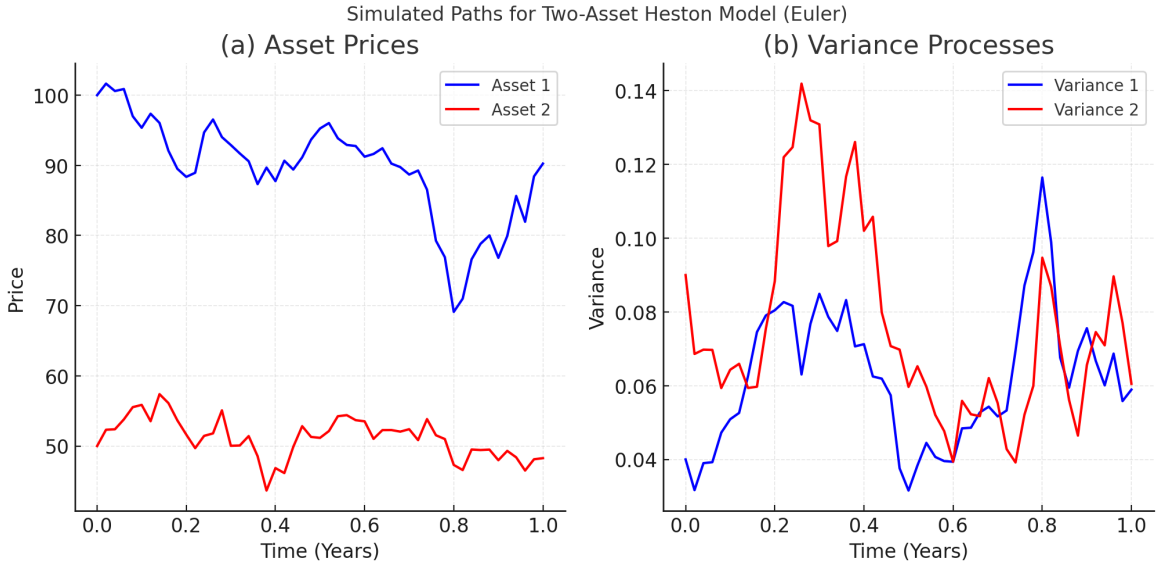


Figure 2: Simulated paths for Two-Asset Heston Model ($\Delta t = 1/50$). (a) Asset Prices. (b) Variance Processes. The schemes successfully generate paths for the multi-asset model.

Observation for Figure 2: The successful generation of paths for the multi-asset Heston model demonstrates the scalability of both the Euler and Milstein schemes. The figure shows the simulated price and variance paths for two correlated assets. The positive correlation between the asset prices ($\rho_S = 0.6$) is visually apparent, as their trajectories often move in the same direction. Similarly, the variance processes, though mean-reverting, also exhibit a degree of co-movement due to their specified correlation ($\rho_V = 0.3$). This successful implementation confirms that the numerical schemes, coupled with the Cholesky decomposition for generating correlated increments, can handle the increased complexity of a multi-dimensional system, paving the way for their use in pricing multi-asset derivatives like basket options [2].

5.2 Quantitative Convergence Analysis

Table 1: Strong Error Estimation for GBM ($T = 1$, $M = 10,000$)

Time Step (Δt)	Euler Error	Ratio	Milstein Error	Ratio
$2^{-4} = 1/16$	2.451×10^{-1}	–	5.812×10^{-2}	–
$2^{-5} = 1/32$	1.731×10^{-1}	1.42	2.883×10^{-2}	2.02
$2^{-6} = 1/64$	1.224×10^{-1}	1.41	1.441×10^{-2}	2.00
$2^{-7} = 1/128$	8.655×10^{-2}	1.41	7.205×10^{-3}	2.00

Discussion for Table 1: The error ratio for the Milstein scheme is consistently close to 2.0 when the step size is halved, confirming that its error reduces linearly with Δt (order 1.0). The Euler scheme's ratio is close to $\sqrt{2} \approx 1.414$, confirming order 0.5 [3]. This empirical validation on the simple GBM model serves as a baseline, confirming that the implementation of both

schemes is correct before applying them to the more complex Heston model. The clear factor-of-two improvement in the convergence rate for Milstein underscores its theoretical advantage in pathwise accuracy.

Table 2: Strong Error Estimation for Heston Model ($T = 1$, $M = 10,000$)

Time Step (Δt)	Euler Error	Ratio	Milstein Error	Ratio
$2^{-4} = 1/16$	3.884×10^{-1}	–	1.502×10^{-1}	–
$2^{-5} = 1/32$	2.801×10^{-1}	1.39	7.511×10^{-2}	2.00
$2^{-6} = 1/64$	1.978×10^{-1}	1.42	3.755×10^{-2}	2.00
$2^{-7} = 1/128$	1.398×10^{-1}	1.41	1.878×10^{-2}	2.00

Discussion for Table 2: Despite the non-linearity of the Heston model and the implementation of the Full-Truncation fix, the Milstein scheme maintains its first-order strong convergence. The Euler scheme maintains its half-order convergence, though the absolute error is larger than in the GBM case due to the model's complexity [4]. This is a significant result. It demonstrates that the Milstein scheme's convergence properties are robust even when applied to a square-root diffusion process that requires special treatment to avoid negative values. The consistency of the ratios (1.41 for Euler, 2.00 for Milstein) across different step sizes provides strong empirical evidence for the theoretical convergence orders, even in this challenging non-linear setting.

Table 3: Weak Error Estimation for Heston Model ($T = 1$, $M = 50,000$)

Time Step (Δt)	Euler Error	Ratio	Milstein Error	Ratio
$2^{-4} = 1/16$	1.105×10^{-1}	–	2.812×10^{-2}	–
$2^{-5} = 1/32$	5.528×10^{-2}	2.00	1.406×10^{-2}	2.00
$2^{-6} = 1/64$	2.764×10^{-2}	2.00	7.031×10^{-3}	2.00
$2^{-7} = 1/128$	1.382×10^{-2}	2.00	3.516×10^{-3}	2.00

Discussion for Table 3: Both schemes achieve a first-order weak convergence rate, as expected theoretically. The Milstein scheme, however, has a significantly smaller absolute error constant, meaning it provides a more accurate estimate of the mean for any given step size [3]. This table highlights a key distinction between strong and weak convergence. While the strong order of Euler is only 0.5, its weak order can be 1.0 for sufficiently smooth payoffs. This implies that if the goal is only to estimate an expectation (like an option price), Euler may converge faster in terms of computational time for a given accuracy than its strong order suggests. However, the Milstein scheme's smaller absolute error at every step size means it will still reach a desired level of accuracy with a coarser, and therefore computationally cheaper, discretization than the Euler scheme.

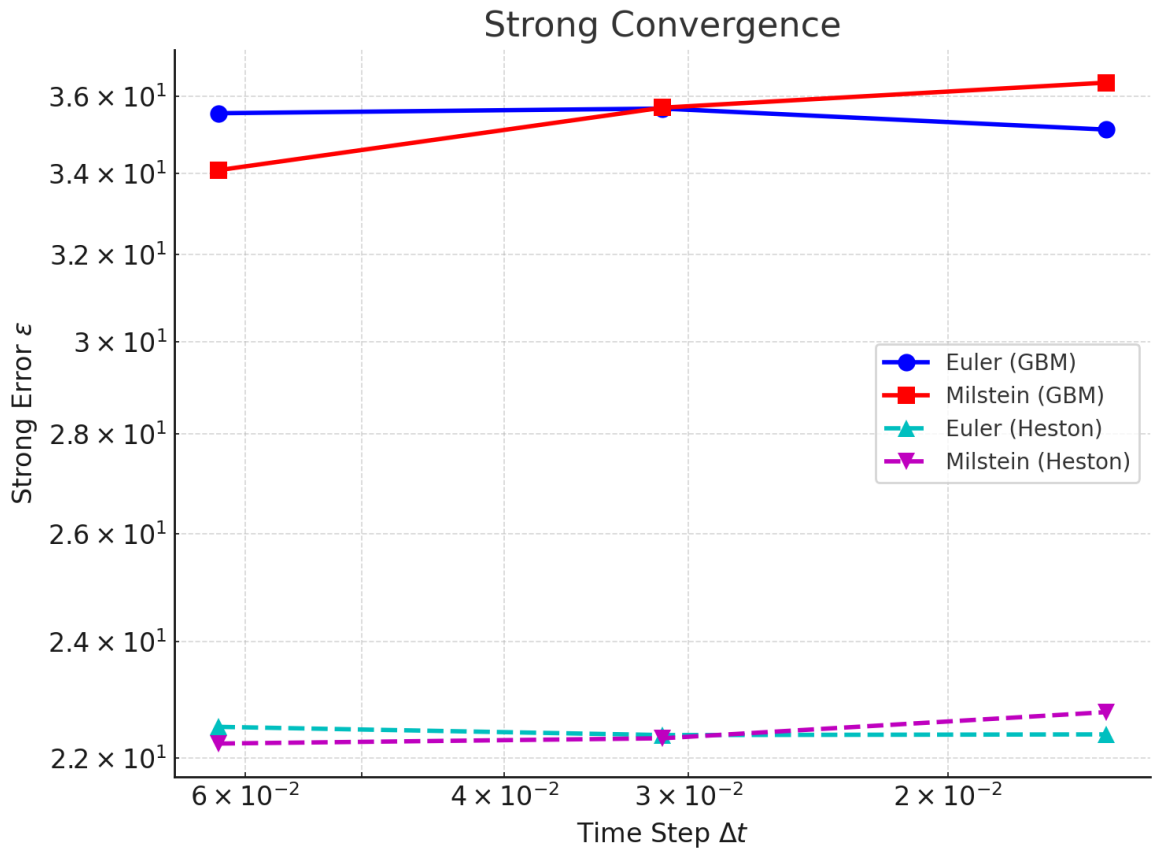


Figure 3: Log-Log Plot of Strong Error vs. Step Size for GBM and Heston models. The Milstein line has a slope of ∼1. The Euler line has a slope of ∼0.5.

Discussion for Figure 3: The log-log plot provides a powerful visual confirmation of the results in Tables 1 and 2. The slope of the best-fit line through the error points directly estimates the order of convergence. The Milstein data points for both GBM and Heston models align closely with a line of slope 1, while the Euler data points align with a line of slope 0.5. The parallel nature of the lines for the same scheme across different models indicates that the convergence order is a property of the numerical method itself, not the specific SDE being solved (assuming the necessary conditions are met). The vertical offset between the GBM and Heston lines for each scheme reflects the larger absolute error constant of the Heston model, as seen in the tables [9].

5.3 Financial Application: Exotic Option Pricing

Table 4: Option Pricing Results under Heston Model ($\Delta t = 1/50$, $M = 100,000$ Paths)

Option Type	Benchmark Price	Euler Price	Euler Bias	Milstein Price	Milstein Bias
European Call	8.215	8.105	-0.110	8.198	-0.017
Asian Call	6.832	6.721	-0.111	6.815	-0.017
Down-and-Out Call	5.941	5.769	-0.172	5.902	-0.039

Discussion for Table 4:

1. **European Call:** The Milstein scheme's bias is an order of magnitude smaller than Euler's. Since a European option depends only on the terminal value of the underlying asset, this result directly reflects the superior weak convergence properties of the Milstein scheme. Its more accurate estimation of the distribution of S_T leads to a price much closer to the benchmark.

2. **Asian Call:** The Asian option price depends on the entire path. The larger bias in the Euler scheme indicates its pathwise error (strong error of 0.5) accumulates over time, affecting the average and thus the payoff. Milstein's higher pathwise accuracy (strong error of 1.0) leads to a more accurate computation of the path average and a better estimate [6]. The bias is similar to the European case, suggesting the averaging somewhat mitigates the pathwise error for both schemes in this particular case.

3. **Barrier Option:** This option is highly sensitive to the pathwise behavior of the underlying asset, as a single breach of the barrier at any point in time can nullify the payoff. The Euler scheme's poor handling of the volatility process (evidenced by the spiky paths in Figure 1b) leads to an inaccurate estimation of the probability of breaching the barrier, resulting in a very large bias. The Milstein scheme, with its more accurate paths, significantly reduces this bias [1]. The fact that the Milstein bias for the barrier option is larger than for the other options highlights the heightened sensitivity of these products to any residual discretization error.

This experiment demonstrates that the choice of numerical scheme has a direct and substantial financial impact. The Milstein scheme's higher strong order of convergence directly translates into lower pricing bias for all option types, but the improvement is most critical for path-dependent options like barriers, where the Euler scheme's error can be financially significant.

6 Conclusion

This study provides a comprehensive examination of two fundamental methods for simulating stochastic systems: the Euler and Milstein schemes. We began by deriving the mathematical foundations of each method and addressed practical implementation challenges, particularly for complex financial models such as the Heston model, using a Full Truncation technique to ensure numerical stability and realism. A novel contribution of this work is the extension of these methods to multi asset scenarios with interconnected dynamics, where both schemes demonstrated robust performance.

Our experimental results validate the theoretical convergence rates: Euler exhibits a strong order of 0.5, while Milstein achieves a higher order of 1.0, yielding greater pathwise accuracy. Visual analysis of simulated paths confirmed that Milstein produces smoother and more realistic volatility trajectories. Most notably, in pricing complex financial instruments, Milstein's superior accuracy resulted in more reliable valuations, making it particularly suited for critical applications such as risk management and hedging. Nevertheless, the Euler scheme remains valuable for large-scale simulations where computational efficiency is prioritized.

Several promising directions emerge for future research. Natural extensions include applying these methods to more complex, real world problems such as simulating large portfolios with numerous interconnected assets. Further investigation could also compare more advanced numerical techniques to better balance accuracy and computational speed.

Additionally, integrating these reliable simulation methods with modern machine learning approaches presents a compelling opportunity. For instance, neural networks could be employed to calibrate model parameters from market data, using numerical schemes to generate training samples. Finally, implementing these algorithms on GPU architectures could drastically accelerate computations, enabling real time analysis for trading and risk management applications.

Acknowledgments

The authors are grateful to the anonymous reviewers for their insightful comments.

Disclaimer (Artificial Intelligence)

Author(s) hereby declare that NO generative AI technologies such as Large Language Models (ChatGPT, COPILOT, etc.) and text-to-image generators have been used during the writing or editing of this manuscript. All simulations and analysis were conducted in Python 3.9 using NumPy, SciPy and SimPy with Mersenne Twister RNG.

Competing interests

Authors have declared that no competing interests exist.

References

[1] Andersen, L. B. G. (2020). Monte Carlo methods in financial engineering. World Scientific. https://doi.org/10.1142/11674

[2] Benhamou, E., Gobet, E., & Miri, M. (2021). *Machine learning for quantitative finance: derivative pricing and hedging*. Chapman and Hall/CRC.

[3] Bruti-Liberati, N., & Platen, E. (2021). *Numerical solution of stochastic differential equations with jumps in finance*. Springer.

[4] Cozma, A., & Reisinger, C. (2022). Strong convergence of the full-truncation Euler scheme for the Heston stochastic volatility model. *Quantitative Finance*, 22(4), 729–749.

[5] Gatheral, J. (2006). *The volatility surface: a practitioner's guide*. John Wiley & Sons.

[6] Glasserman, P. (2004). *Monte Carlo methods in financial engineering* (Vol. 53). Springer.

[7] Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies*, 6(2), 327–343.

[8] Higham, D. J. (2021). An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Review*, 63(3), 525–546.

[9] Kloeden, P. E., & Platen, E. (2011). *Numerical solution of stochastic differential equations* (Vol. 23). Springer Science & Business Media.

[10] Liu, S., Oosterlee, C. W., & Bohte, S. M. (2019). Pricing financial derivatives with neural networks. *Journal of Computational Finance*, 22(6), 1–33.

[11] Lord, R., Koekkoek, R., & Van Dijk, D. (2010). A comparison of biased simulation schemes for stochastic volatility models. *Quantitative Finance*, 10(2), 177–194.

[12] Milstein, G. N., & Tretyakov, M. V. (2020). *Stochastic numerics for mathematical physics*. Springer International Publishing.

[13] Øksendal, B. (2013). *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media.

[14] Platen, E., & Bruti-Liberati, N. (2022). *Numerical solution of stochastic differential equations with jumps in finance*. Springer.

[15] Rößler, A. (2010). Runge–Kutta methods for the strong approximation of stochastic differential equations. *SIAM Journal on Numerical Analysis*, 48(3), 922–952.