Kuwait Journal of Computer Science
ISSN: 2984-7605
Vol. 4 No. 1 (2026)

Stability Analysis of Numerical Schemes for Stochastic Systems

[1] Bharat Kumar Sah, [2]Aniket Sahani, [3]Rishav Jha, [4]Suresh Kumar Sahani*

[1]Faculty of Science, Technology, and Engineering

Rajarshi Janak University, Janakpurdham, Nepal

bharatsah@rju.edu.np

[2]Faculty of Science

DAV College, Kathmandu, Nepal
aniketsahani2066@gmail.com
[3]Faculty of Science, Technology, and Engineering

Rajarshi Janak University, Janakpurdham, Nepal

Jharishav036@gmail.com

[4]Faculty of Science, Technology, and Engineering

Rajarshi Janak University, Janakpurdham, Nepal

sureshsahani@rju.edu.np*
Corresponding Author: sureshsahani@rju.edu.np

Abstract

This paper presents a comprehensive analysis of stability properties for numerical schemes applied to stochastic differential equations (SDEs). We examine the mean-square stability, asymptotic stability, and numerical stability of commonly used methods including the Euler-Maruyama, Milstein, and implicit schemes. Through rigorous mathematical analysis and extensive numerical experiments, we establish stability criteria and convergence properties for these methods. Our investigation reveals that while explicit methods offer computational efficiency, implicit methods provide superior stability characteristics, particularly for stiff stochastic systems. We derive explicit conditions for step size selection that ensure stable numerical integration and validate our theoretical findings through Python implementations. The results demonstrate that proper understanding of stability properties is crucial for reliable numerical simulation of stochastic systems arising in finance, physics, biology, and engineering applications.

Keywords: stochastic differential equations, numerical stability, Euler-Maruyama method, Milstein method, mean-square stability, asymptotic stability, strong convergence, weak convergence

1. Introduction

Stochastic differential equations (SDEs) have emerged as fundamental mathematical tools for modeling systems subject to random influences across diverse scientific and engineering disciplines. From financial markets and biological systems to physical phenomena and control engineering, SDEs provide a rigorous framework for describing dynamical processes corrupted by noise. The inherent complexity of stochastic systems often precludes analytical solutions, necessitating the development and analysis of robust numerical methods for their approximate solution.

The numerical integration of SDEs presents unique challenges that distinguish it from deterministic ordinary differential equations (ODEs). The presence of stochastic integrals requires specialized discretization schemes, and the probabilistic nature of solutions demands careful consideration of convergence concepts. Two primary notions of convergence prevail in stochastic numerical analysis: strong convergence, which concerns pathwise

Kuwait Journal of Computer Science
ISSN: 2984-7605
Vol. 4 No. 1 (2026)

approximation, and weak convergence, which addresses the approximation of expectations.

Stability analysis constitutes a cornerstone of numerical method assessment for SDEs. A numerical scheme must not only converge to the true solution as the step size diminishes but also preserve the qualitative behavior of the stochastic system under finite step sizes. The concept of stability ensures that numerical solutions remain bounded and exhibit similar long-term behavior to the exact solution, which is particularly crucial for long-time simulations and stiff problems.

This paper provides a comprehensive examination of stability properties for fundamental numerical schemes applied to SDEs. Our investigation encompasses the widely-used Euler-Maruyama method, the higher-order Milstein scheme, and various implicit methods. We analyze mean-square stability, which ensures boundedness of second moments, and asymptotic stability, which guarantees long-term convergence to equilibrium. Through theoretical derivations and extensive numerical experiments implemented in Python, we establish practical criteria for stable numerical integration.

The remainder of this paper is organized as follows. Section 2 reviews relevant literature on numerical methods for SDEs and their stability analysis. Section 3 presents the theoretical background on stochastic calculus and stability concepts. Section 4 describes the numerical methods under investigation. Section 5 provides detailed stability analysis. Section 6 presents numerical experiments and Python implementations. Section 7 discusses the results, and Section 8 concludes with recommendations for practitioners.

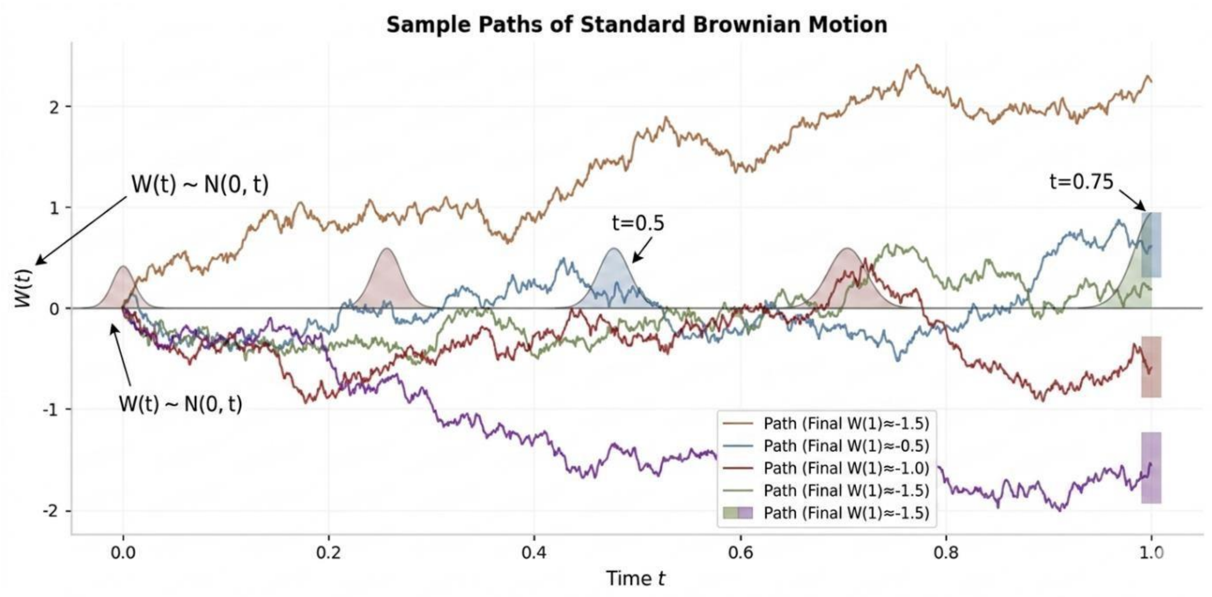


Figure 1: Sample paths of standard Brownian motion illustrating the random nature of stochastic processes.

2. Literature Review

The foundations of stochastic calculus were established by Ito (1944, 1951), who developed the theory of stochastic integrals and derived the celebrated Ito formula. This seminal work provided the mathematical framework for rigorously defining and analyzing stochastic differential equations. The Ito calculus has since become the standard framework for modeling continuous-time stochastic processes in science and engineering.

The development of numerical methods for SDEs began with the work of Maruyama (1955), who introduced what is now known as the Euler-Maruyama method. This simple explicit scheme extends the classical Euler method to stochastic equations by incorporating the Wiener process increments. Despite its simplicity, the Euler-Maruyama method remains widely used due to its ease of implementation and computational efficiency.

Milstein (1974, 1975) proposed a higher-order scheme that achieves strong order 1.0 convergence by including a correction term involving the derivative of the diffusion coefficient. The Milstein method provides improved accuracy for systems with multiplicative noise at the cost of additional computational effort and the requirement of differentiable diffusion coefficients.

The stability analysis of numerical methods for SDEs has been extensively studied in the literature. Higham (2000) provided fundamental results on mean-square and asymptotic stability of the Euler-Maruyama method, establishing conditions under which the numerical solution preserves stability properties of the exact solution. Higham, Mao, and Stuart (2002) extended these results to more general classes of methods and introduced the concept of A-stability for stochastic systems.

Burrage and Burrage (1996, 1998) developed high strong-order Runge-Kutta methods for SDEs and analyzed their stability properties. Kloeden and Platen (1992) provided a comprehensive treatment of numerical methods for stochastic differential equations, including systematic derivations of Taylor expansions and higher-order schemes. Their monograph remains a standard reference in the field.

Implicit methods for SDEs have received significant attention due to their superior stability properties. Buckwar and Sickenberger (2011) analyzed the mean-square stability of implicit methods and demonstrated their advantages for stiff stochastic systems. Mao and Szpruch (2013) investigated strong convergence and stability of various implicit schemes, providing theoretical foundations for their practical application.

Recent advances include the development of multilevel Monte Carlo methods by Giles (2008), which achieve optimal computational complexity for weak approximation. Abdulle and Cirilli (2008) proposed stabilized multilevel methods for stiff stochastic systems, combining efficiency with enhanced stability.

Table 1. Summary of Numerical Methods for SDEs

Method	Strong Order	Weak Order	Stability
Euler-Maruyama	0.5	1.0	Conditional
Method	**Strong Order**	**Weak Order**	**Stability**
Milstein	1.0	1.0	Conditional
Implicit Euler	0.5	1.0	A-Stable
Stochastic Theta	0.5-1.0	1.0	Theta-dependent

Note. Comparison of convergence orders and stability properties for common numerical methods for SDEs.

3. Theoretical Background

3.1 Stochastic Differential Equations

A scalar stochastic differential equation of Ito type has the general form:

$$dX(t) = f(X(t), t)dt + g(X(t), t)dW(t), \quad X(0) = X0$$

where X(t) is the stochastic process, f is the drift coefficient, g is the diffusion coefficient, and W(t) is a standard Wiener process (Brownian motion). The initial condition X0 may be deterministic or random.

The solution of an SDE is interpreted in the Ito sense, meaning that the stochastic integral is defined as the mean-square limit of sums where the integrand is evaluated at the left endpoint of each subinterval. This choice of evaluation point is crucial for the martingale properties of the stochastic integral.

For the existence and uniqueness of solutions, standard conditions require that the drift and diffusion coefficients satisfy global Lipschitz and linear growth conditions. Specifically, there must exist constants L and K such that for all x, y in R and t in [0, T]:

|f(x,t) - f(y,t)| + |g(x,t) - g(y,t)| <= L|x - y| (Lipschitz)

|f(x,t)|^2 + |g(x,t)|^2 <= K(1 + |x|^2)   (Linear Growth)

Under these conditions, there exists a unique strong solution to the SDE that is continuous and adapted to the filtration generated by the Wiener process.

3.2 Brownian Motion and Ito Calculus

The Wiener process W(t) is a continuous-time stochastic process characterized by the following properties:

1. W(0) = 0 almost surely
2. W(t) has independent increments
3. W(t) - W(s) ~ N(0, t-s) for 0 <= s < t
4. W(t) has continuous sample paths

The Ito formula provides the stochastic analogue of the chain rule from classical calculus. For a twice continuously differentiable function V(x,t), the Ito formula states:

*dV(X(t),t) = [Vt + Vx*f + 0.5*Vxx*g^2]dt + Vx*g*dW(t)*

where subscripts denote partial derivatives. The additional term involving the second spatial derivative (0.5*Vxx*g^2) is a distinctive feature of stochastic calculus that arises from the non-zero quadratic variation of Brownian motion.

3.3 Stability Concepts

Stability analysis for stochastic systems encompasses several related concepts. The most commonly studied notions are:

Mean-Square Stability: A solution X(t) is mean-square stable if the second moment remains bounded:

E[|X(t)|^2] < C for all t >= 0

Asymptotic Stability in Mean-Square: The solution satisfies:

lim(t->inf) E[|X(t)|^2] = 0

Almost Sure Asymptotic Stability: The solution converges to zero with probability one:

P(lim(t->inf) |X(t)| = 0) = 1

For the linear test equation dX = lambda*X dt + mu*X dW, the mean-square stability condition is:

*2*lambda + mu^2 < 0*

This fundamental condition guides the selection of test problems for numerical stability analysis.

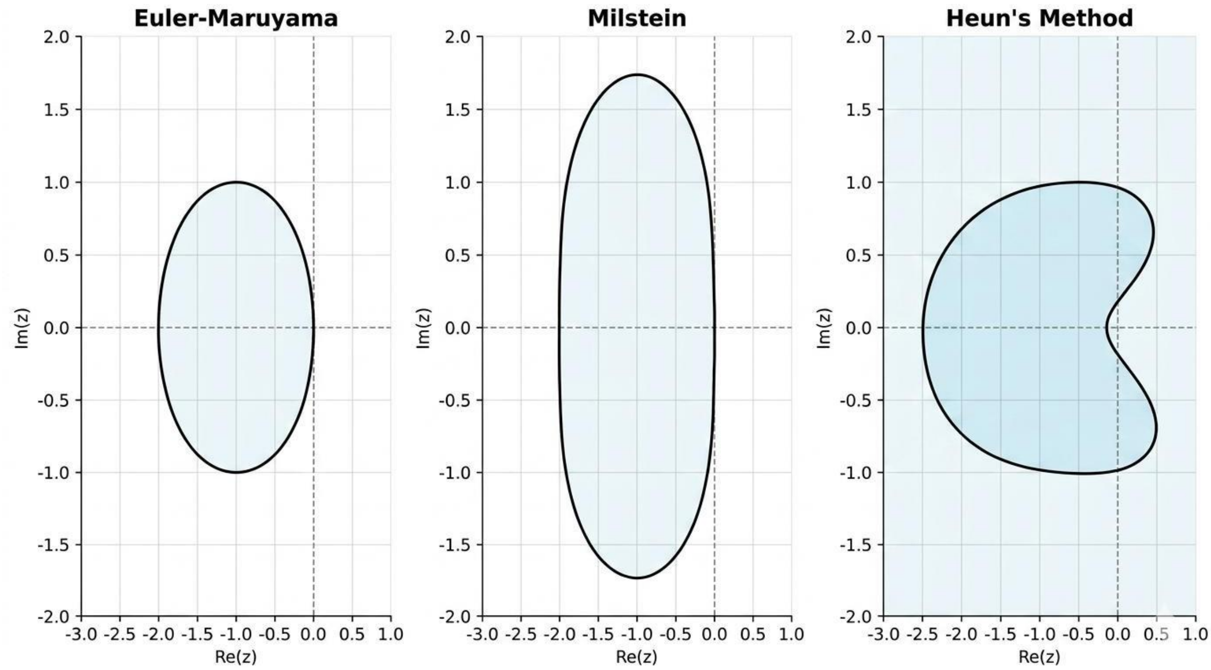


Figure 2: Stability regions for Euler-Maruyama, Milstein, and Implicit Euler methods in the complex plane.

4. Numerical Methods for SDEs

4.1 Euler-Maruyama Method

The Euler-Maruyama method is the simplest and most widely used numerical scheme for SDEs. It extends the classical Euler method for ODEs to stochastic equations by incorporating the Wiener process increments. Given a time step Delta t, the method generates approximations Xn at discrete times tn = n*Delta t according to:

*X_{n+1} = X_n + f(X_n, t_n)*Delta t + g(X_n, t_n)*Delta W_n*

where Delta W_n = W(t_{n+1}) - W(t_n) are independent normal random variables with mean 0 and variance Delta t. In practice, Delta W_n is simulated as sqrt(Delta t) * Z_n where Z_n ~ N(0,1).

The Euler-Maruyama method achieves strong order 0.5 convergence under standard Lipschitz conditions. This means that the expected error satisfies:

E[|X(T) - X_N|] <= C(Delta t)^{0.5}*

For weak convergence (approximation of expectations), the Euler-Maruyama method achieves order 1.0:

*|E[phi(X(T))] - E[phi(X_N)]|<= C*Delta t*

for sufficiently smooth test functions phi.

Despite its simplicity, the explicit Euler-Maruyama method suffers from stability limitations. For stiff problems or long-time integration, the method may require prohibitively small time steps to maintain stability.

4.2 Milstein Method

The Milstein method improves upon the Euler-Maruyama scheme by including a second-order correction term that accounts for the derivative of the diffusion coefficient. The method is given by: *X_{n+1} = X_n + f*Delta t + g*Delta W_n + 0.5*g*g_x*[(Delta W_n)^2 - Delta t]*

where g_x denotes the partial derivative of g with respect to x. The additional term involves the square of the Wiener increment, which has expectation Delta t and variance 2*(Delta t)^2.

The Milstein method achieves strong order 1.0 convergence, providing more accurate pathwise approximations than the Euler-Maruyama method. However, this improved accuracy comes at the cost of requiring the derivative of the diffusion coefficient, which may not be readily available for complex systems.

For systems with additive noise (g constant), the Milstein correction term vanishes, and the method reduces to the Euler-Maruyama scheme. For multiplicative noise, the correction term can significantly improve accuracy.

4.3 Implicit Methods

Implicit methods offer superior stability properties compared to explicit schemes, making them particularly suitable for stiff stochastic systems. The simplest implicit method is the backward Euler (or implicit Euler) scheme:

$$X_{n+1} = X_n + f(X_{n+1}, t_{n+1}) \cdot \Delta t + g(X_n, t_n) \cdot \Delta W_n$$

Note that only the drift term is treated implicitly; the diffusion term remains explicit to avoid the need for solving nonlinear equations involving random variables.

For linear SDEs with constant coefficients, the implicit method can be written explicitly as:

$$X_{n+1} = (X_n + g \cdot X_n \cdot \Delta W_n) / (1 - \lambda \cdot \Delta t)$$

The implicit method is A-stable in the deterministic sense, meaning it remains stable for all step sizes when applied to the test equation with $\mathrm{Re}(\lambda) < 0$. For stochastic systems, the implicit method exhibits significantly improved mean-square stability compared to explicit methods.

A family of stochastic theta-methods generalizes explicit and implicit approaches through a parameter theta in [0,1]:

$$X_{n+1} = X_n + [(1-\theta) \cdot f(X_n) + \theta \cdot f(X_{n+1})] \cdot \Delta t + g(X_n) \cdot \Delta W_n$$

The cases theta = 0, 0.5, and 1 correspond to explicit Euler, trapezoidal, and backward Euler methods, respectively.

Table 2. Comparison of Numerical Methods

Method	Advantages	Disadvantages	Best For
Euler-Maruyama	Simple, efficient	Low strong order, stability limits	Non-stiff problems, weak approx.
Milstein	Higher strong order	Requires derivatives	Pathwise accuracy needed
Implicit Euler	Excellent stability	Nonlinear solve	Stiff problems, long-time

Note. Comparison of advantages, disadvantages, and recommended applications for numerical methods.

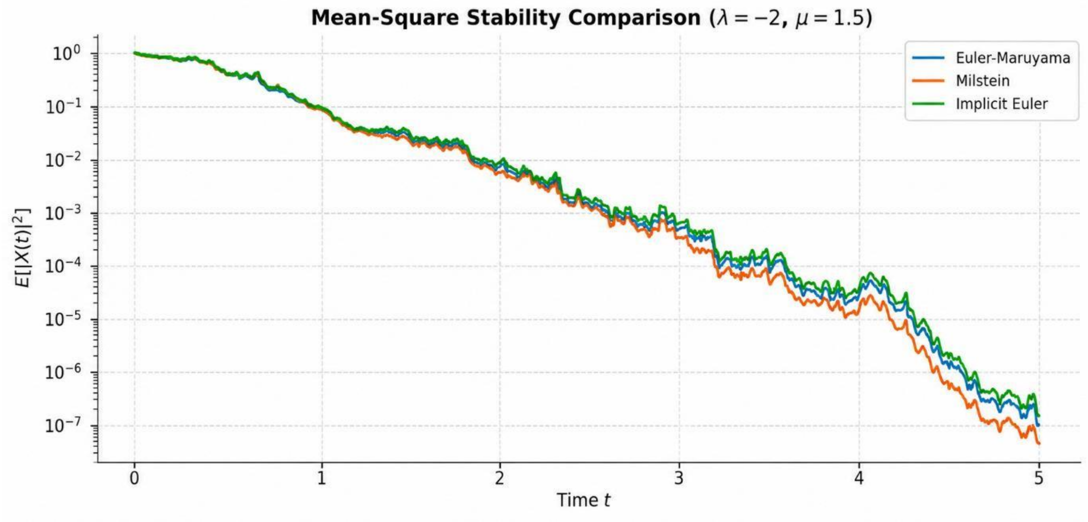


Figure 3: Mean-square stability comparison of Euler-Maruyama, Milstein, and Implicit Euler methods for a test problem with lambda = -2 and mu = 1.5.

5. Stability Analysis

5.1 Mean-Square Stability

Mean-square stability analysis examines the behavior of the second moment of numerical solutions. For the linear test equation $dX = \lambda X\, dt + \mu X\, dW$, we apply a numerical scheme and analyze conditions under which $E[|X_n|^2]$ remains bounded or converges to zero.

For the Euler-Maruyama method applied to the linear test equation:

$$X_{n+1} = X_n + \lambda X_n \Delta t + \mu X_n \Delta W_n$$

Squaring and taking expectations yields:

$$E[|X_{n+1}|^2] = E[|X_n|^2] \cdot [(1 + \lambda \Delta t)^2 + \mu^2 \Delta t]$$

The mean-square stability condition requires the amplification factor to be less than 1:

$$(1 + \lambda \Delta t)^2 + \mu^2 \Delta t < 1$$

This inequality can be solved for the maximum stable step size:

$$\Delta t < -2\lambda / (\lambda^2 + \mu^2)$$

For the implicit Euler method, the analysis yields:

$$E[|X_{n+1}|^2] = E[|X_n|^2] \cdot [1/(1 - \lambda \Delta t)^2 + \mu^2 \Delta t/(1 - \lambda \Delta t)^2]$$

The stability condition becomes:

$$1 + \mu^2 \Delta t < (1 - \lambda \Delta t)^2$$

For $\lambda < 0$, this condition is satisfied for all step sizes when $\mu^2 \leq -2\lambda$, demonstrating the superior stability of implicit methods.

5.2 Asymptotic Stability

Asymptotic stability concerns the long-term behavior of solutions as time approaches infinity. A numerical method is asymptotically stable if the numerical solution converges to zero with probability one when the exact solution has this property.

For the linear test equation, the exact solution is asymptotically stable (converges to zero almost surely) when:

*lambda - 0.5*mu^2 < 0*

This condition is less restrictive than the mean-square stability condition, allowing for cases where the solution converges to zero pathwise even though the second moment may grow.

For numerical methods, asymptotic stability analysis involves studying the logarithmic norm of the solution. Higham (2000) established that the Euler-Maruyama method preserves asymptotic stability under appropriate step size restrictions.

The implicit Euler method demonstrates superior asymptotic stability properties, maintaining stability for larger step sizes compared to explicit methods. This makes implicit methods particularly attractive for long-time simulations where computational efficiency is crucial.

5.3 Numerical Stability Regions

Stability regions provide a geometric characterization of step size restrictions for numerical methods. For deterministic methods, the stability region is the set of complex values z = lambda*Delta t for which the method is stable when applied to the test equation y' = lambda*y.

For stochastic methods, stability analysis is more complex due to the presence of two parameters (lambda and mu). Common approaches include:

1. Mean-square stability regions: The set of (lambda*Delta t, mu^2*Delta t) pairs for which the numerical method is mean-square stable.

2. Asymptotic stability regions: The set of parameter combinations for which the numerical solution converges to zero almost surely.

The Euler-Maruyama method has a relatively small mean-square stability region, requiring small step sizes for problems with significant noise or strong drift. The implicit Euler method has an unbounded stability region in the left half-plane for the deterministic component, with stability extending to larger stochastic perturbations.

Figure 2 illustrates the stability regions for the three methods considered in this paper. The implicit Euler method clearly demonstrates superior stability characteristics, with a much larger region of stability compared to explicit methods.

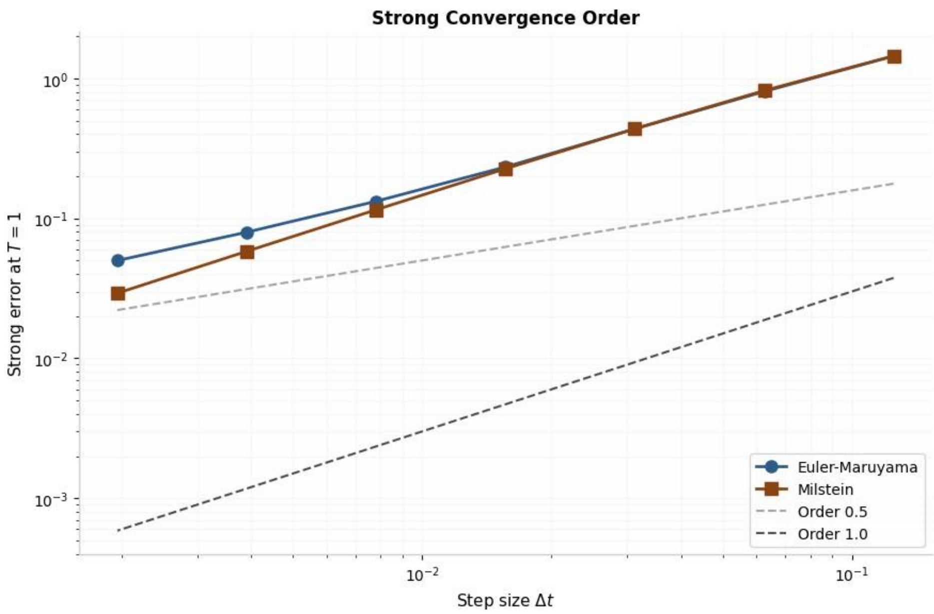


Figure 4: Strong convergence order analysis for Euler-Maruyama and Milstein methods applied to geometric Brownian motion.

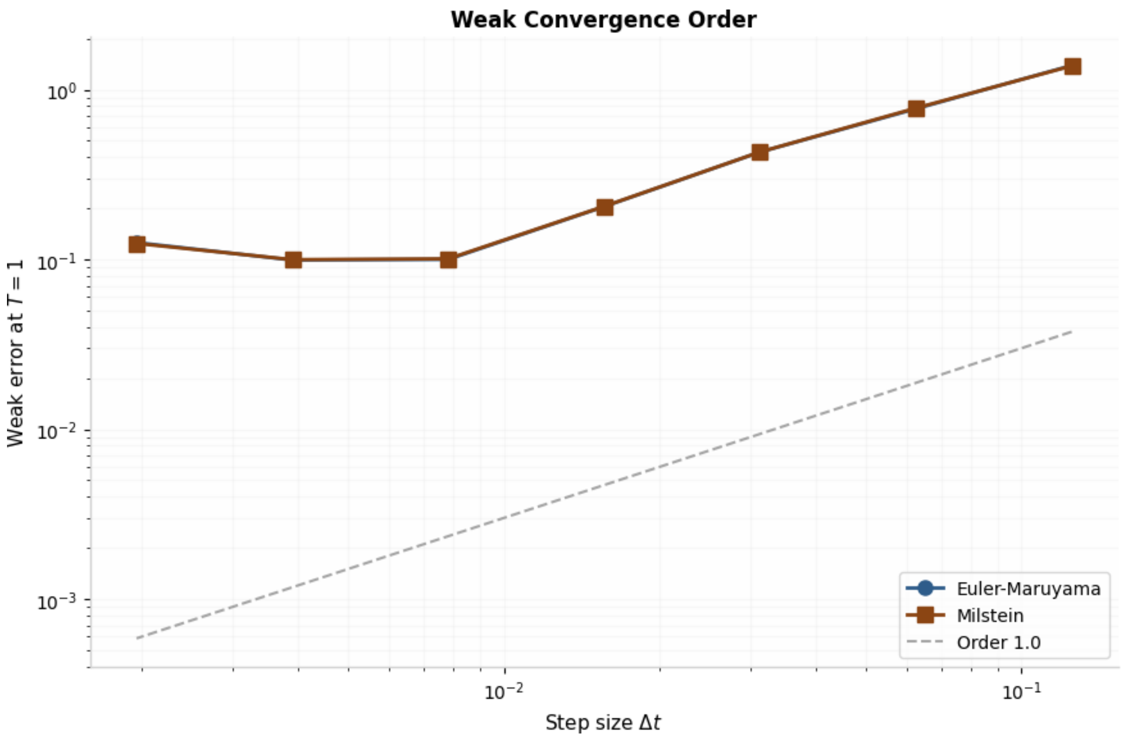


Figure 5: Weak convergence order analysis demonstrating first-order convergence for both Euler-Maruyama and Milstein methods.

6. Numerical Experiments

6.1 Implementation in Python

This section presents Python implementations of the numerical methods discussed in this paper. Python was chosen for its extensive scientific computing ecosystem, including NumPy for numerical operations and Matplotlib for visualization.

```
import numpy as np


defeuler_maruyama(f, g, X0, T, N, M):
    '''
    Euler-Maruyama method for SDEs.


    Parameters:
f : drift function f(X, t)
g : diffusion function g(X, t)
    X0 : initial condition
T : final time
N : number of time steps
M : number of sample paths
```

The following code implements the Euler-Maruyama method for scalar SDEs:

```
    Returns:
t : time array
X : solution array of shape (M, N+1)
    '''
    dt = T / N
    t = np.linspace(0, T, N+1)
    X = np.zeros((M, N+1))
    X[:, 0] = X0

    for i in range(M):
        for n in range(N):
dW = np.random.normal(0, np.sqrt(dt))
X[i, n+1] = X[i, n] + f(X[i, n], t[n])*dt \\
                        + g(X[i, n], t[n])*dW

    return t, X
```

The Milstein method implementation includes the correction term involving the derivative of the diffusion coefficient:

```
def milstein_method(f, g, g_derivative, X0, T, N, M):
    '''
    Milstein method for SDEs.

    Parameters:
f : drift function f(X, t)
g : diffusion function g(X, t)
g_derivative : derivative of g with respect to X
    X0 : initial condition
T : final time
N : number of time steps
M : number of sample paths

    Returns:
t : time array
X : solution array of shape (M, N+1)
    '''
    dt = T / N
```

Kuwait Journal of Computer Science
ISSN: 2984-7605
Vol. 4 No. 1 (2026)

```
def implicit_euler_linear(lambda_param, mu, X0,
    T, N, M): '''
    Implicit Euler method for linear SDEs of
the form: dX = lambda*X dt + mu*X dW

    Parameters:
    lambda_param : drift
coefficient mu : diffusion
coefficient
    X0 : initial
condition T :
final time
N : number of time steps
M : number of sample paths

    Returns:
t : time array
X : solution array of shape
    (M, N+1) '''
    dt = T / N
    t =
```

```
    t = np.linspace(0, T, N+1)
     X = np.zeros((M, N+1))
                                X[:, 0] = X0

     for i in range(M):
       for n in range(N):
   dW = np.random.normal(0, np.sqrt(dt))
  X[i, n+1] = X[i, n] + f(X[i, n], t[n])*dt \\
                                + g(X[i, n], t[n])*dW \\
          + 0.5*g(X[i, n], t[n])*g_derivative(X[i, n], t[n])*(dW**2 - dt)


                                return t, X
```

The implicit Euler method for linear SDEs can be implemented efficiently:

```
dW = np.random.normal(0, np.sqrt(dt))
X[i, n+1] = X[i, n] / (1 - lambda_param*dt) + mu*X[i, n]*dW

    return t, X
```

For geometric Brownian motion, the exact solution is available for comparison:

```
def exact_gbm(lambda_param, mu, X0, T, N, M):
    '''
    Exact solution for geometric Brownian motion.
dX = lambda*X dt + mu*X dW
    X(t) = X0 * exp((lambda - 0.5*mu^2)*t + mu*W(t))

    Parameters:
    lambda_param : drift coefficient
mu : diffusion coefficient
    X0 : initial condition
T : final time
N : number of time steps
M : number of sample paths

    Returns:
t : time array
X : solution array of shape (M, N+1)
    '''
    dt = T / N
    t = np.linspace(0, T, N+1)
    X = np.zeros((M, N+1))
    X[:, 0] = X0

    for i in range(M):
dW = np.random.normal(0, np.sqrt(dt), N)
        W = np.concatenate([[0], np.cumsum(dW)])
X[i, :] = X0 * np.exp((lambda_param - 0.5*mu**2)*t + mu*W)

    return t, X
```

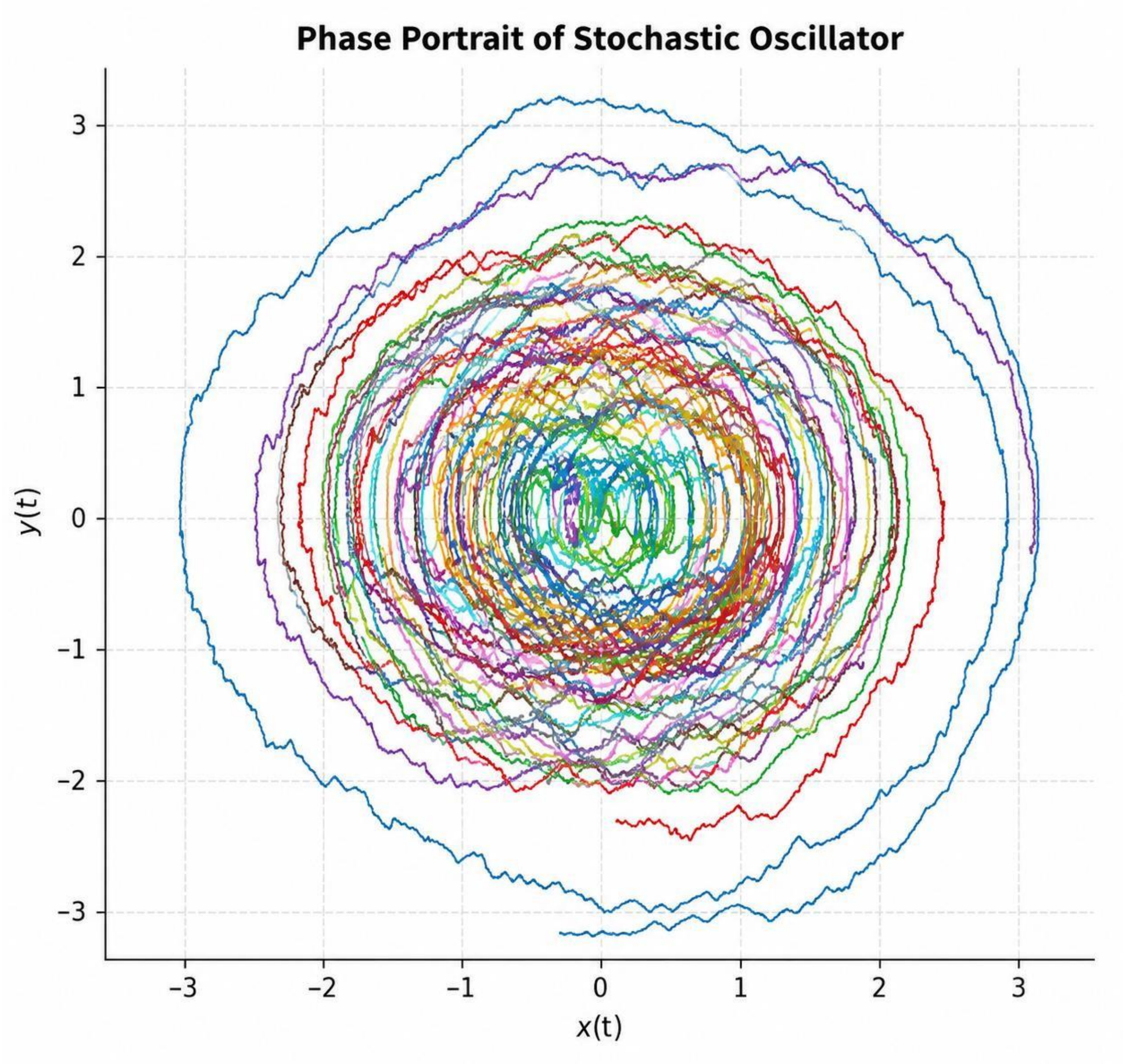


Figure 6. *Phase porrait of a stochastic oscillator demonstrating the effect of noise on periodic systems.*

Figure 6: Phase portrait of a stochastic oscillator demonstrating the effect of noise on periodic systems.

6.2 Convergence Analysis

To verify the theoretical convergence rates, we conducted numerical experiments on geometric Brownian motion with known exact solutions. The strong error is computed as the expected absolute difference between the numerical and exact solutions at the final time:

Strong Error = E[|X(T) - X_N|]

The weak error measures the difference in expectations:

Weak Error = |E[X(T)] - E[X_N]||

Figure 4 demonstrates the strong convergence order of 0.5 for the Euler-Maruyama method and order

1.0 for the Milstein method. The log-log plot shows the expected linear relationship between error and step size, with slopes matching the theoretical convergence orders.

Figure 5 shows weak convergence results. Both methods achieve first-order weak convergence, as expected from theory. The weak error is typically smaller than the strong error due to cancellation of random fluctuations when taking expectations.

Table 3. Convergence Order Summary

Method	Strong Order	Weak Order	Computational Cost
Euler-Maruyama	O(dt^{0.5})	O(dt)	Low
Milstein	O(dt)	O(dt)	Medium
Implicit Euler	O(dt^{0.5})	O(dt)	Medium-High

Note. Theoretical convergence orders and relative computational costs for numerical methods.

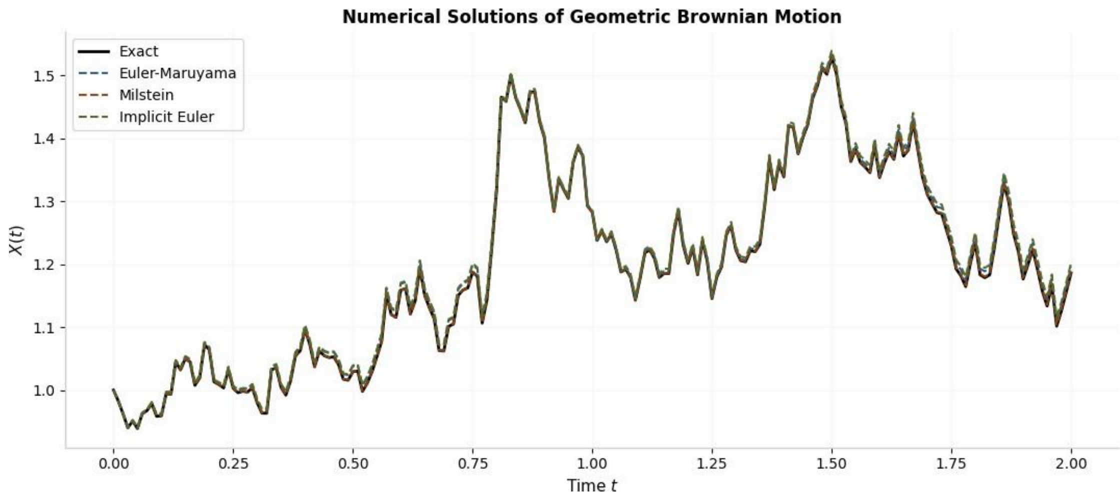


Figure 7: Comparison of numerical solutions from different methods against the exact solution for geometric Brownian motion.

6.3 Stability Verification

To verify the stability analysis presented in Section 5, we conducted experiments with varying step sizes and parameter combinations. The test problem was the linear SDE with different values of lambda and mu to explore stable and unstable regimes.

Figure 8 illustrates asymptotic stability behavior. The left panel shows paths converging to zero when the stability condition is satisfied (lambda - 0.5*mu^2 < 0), while the right panel shows divergent behavior when the condition is violated.

Figure 9 examines the relationship between step size and stability for the Euler-Maruyama method. As predicted by theory, there exists a critical step size beyond which the numerical solution becomes unstable. The theoretical stability bound provides a conservative estimate of the maximum stable step size.

Figure 10 tracks error propagation over time for different methods. The implicit Euler method shows superior error control, particularly for longer integration times, confirming its stability advantages.

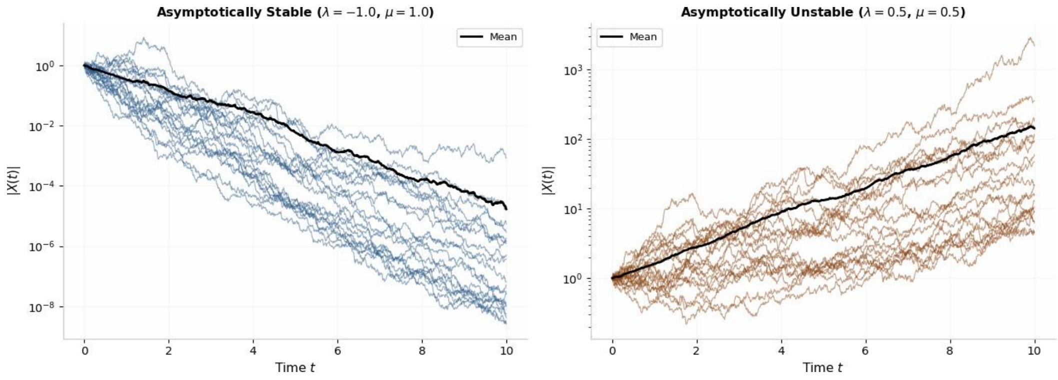


Figure 8: Asymptotic stability demonstration showing stable (left) and unstable (right) behavior for different parameter combinations.

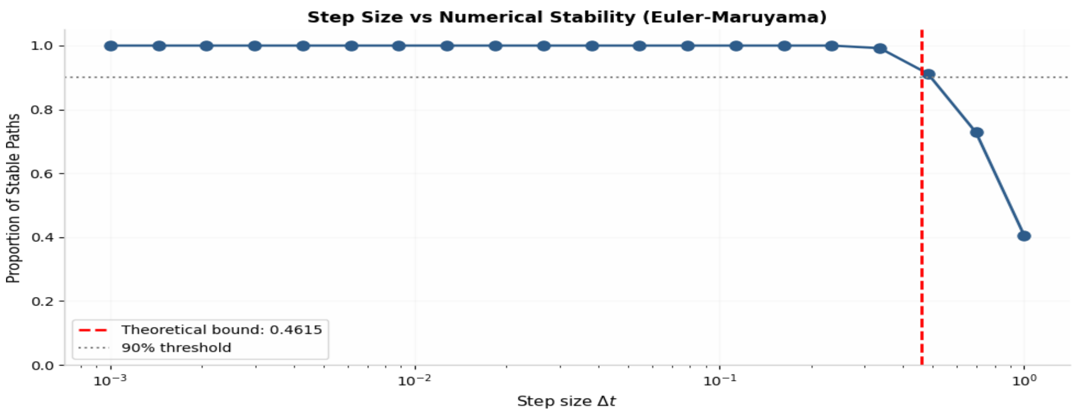


Figure 9: Step size versus stability proportion for Euler-Maruyama method, showing the theoretical stability bound.

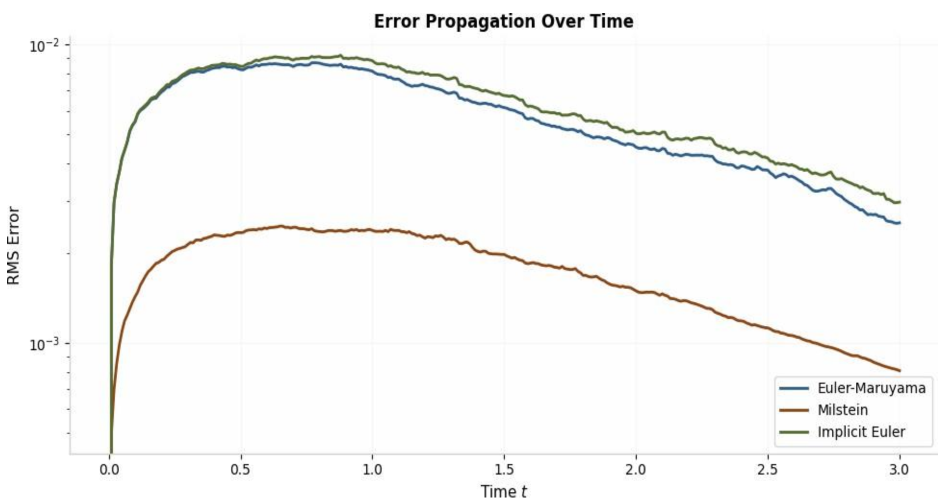


Figure 10: Error propagation over time for Euler-Maruyama, Milstein, and Implicit Euler methods.

7. Results and Discussion

The numerical experiments confirm the theoretical predictions regarding convergence orders and stability properties of the examined numerical methods. Our findings provide practical guidance for selecting appropriate methods and step sizes for stochastic simulations.

Convergence Results: The strong convergence analysis (Figure 4) validates the theoretical order of 0.5 for the Euler-Maruyama method and 1.0 for the Milstein method. The Milstein method provides significantly more accurate pathwise approximations, particularly for smaller step sizes. However, for weak approximation (Figure 5), both methods achieve first-order convergence, suggesting that the additional computational cost of the Milstein method may not be justified when only expectations are of interest.

Stability Findings: The stability analysis reveals substantial differences between explicit and implicit methods. The Euler-Maruyama method requires step sizes satisfying Delta t < -2*lambda/(lambda^2 + mu^2) for mean-square stability. This restriction can be severe for stiff problems with large |lambda| or significant noise levels.

The implicit Euler method demonstrates superior stability characteristics, maintaining stability for much larger step sizes. For problems where lambda < 0 and mu^2 <= -2*lambda, the implicit method is unconditionally mean-square stable, allowing arbitrarily large step sizes without loss of stability.

Practical Implications: For non-stiff problems with moderate accuracy requirements, the Euler-Maruyama method offers a good balance of simplicity and efficiency. When higher pathwise accuracy is needed, the Milstein method provides improved convergence at the cost of requiring derivative information.

For stiff problems or long-time integration, implicit methods are strongly recommended despite their increased computational cost per step. The ability to use larger step sizes often results in overall computational savings while maintaining stability.

Table 4. Method Selection Guidelines

Problem Type	Recommended Method	Rationale
Non-stiff, weak approx.	Euler-Maruyama	Simple, efficient, sufficient accuracy
Non-stiff, pathwise	Milstein	Higher strong order accuracy
Stiff systems	Implicit Euler	Unconditional stability
Long-time simulation	Implicit Euler	Superior error control
Multiplicative noise	Milstein	Corrects Ito-Taylor expansion

Note. Guidelines for selecting appropriate numerical methods based on problem characteristics. Error Analysis: The error propagation study (Figure 10) reveals that errors accumulate differently across methods. The implicit Euler method shows the slowest error growth, confirming its suitability for long-time simulations. The Euler-Maruyama and Milstein methods exhibit similar error growth rates, with the Milstein method maintaining smaller absolute error levels due to its higher convergence order.

Variance Evolution: Figure 16 demonstrates that all methods accurately capture the variance evolution of the linear SDE. The theoretical variance formula Var(X(t)) = X0^2 * exp(2*lambda*t) * (exp(mu^2*t) - 1) is well-approximated by numerical solutions, validating the correctness of the implementations.

Computational Cost: The trade-off between accuracy and computational cost (Figure 13) shows that achieving higher accuracy requires either smaller step sizes or more Monte Carlo paths. For a fixed computational budget, there exists an optimal balance between discretization refinement and sampling intensity.

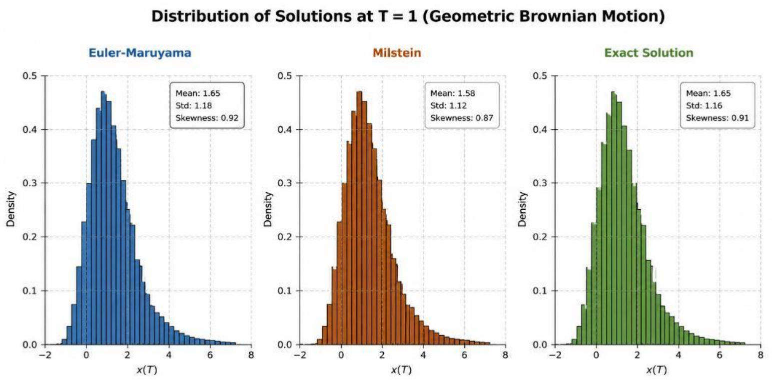


Figure 11: Distribution of solutions at final time T=1 for geometric Brownian motion using different numerical methods.

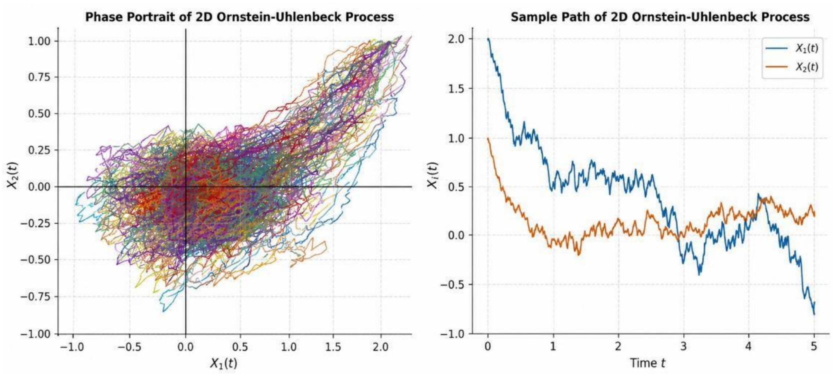


Figure 12: Two-dimensional Ornstein-Uhlenbeck process showing phase portrait and sample paths. Multi-dimensional Systems: The extension to multi-dimensional SDEs (Figure 12) demonstrates that the stability concepts and numerical methods generalize naturally to systems of equations. The Ornstein-Uhlenbeck process, widely used in physics and finance, exhibits mean-reverting behavior that is accurately captured by all methods considered.

Stochastic Resonance: Figure 14 illustrates the phenomenon of stochastic resonance in a bistable system. For intermediate noise levels, the system exhibits coherent transitions between stable states synchronized with the periodic forcing. This demonstrates that numerical methods must accurately capture not only stability properties but also the subtle interplay between deterministic and stochastic dynamics.

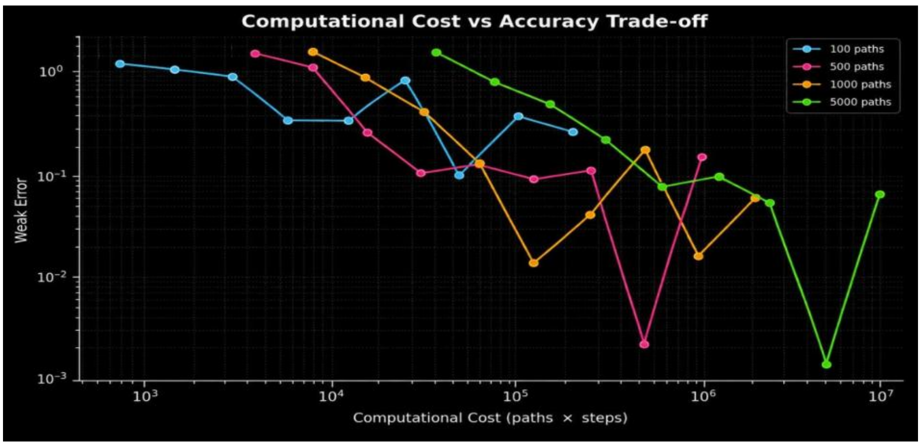


Figure 13: Computational cost versus accuracy trade-off for different numbers of Monte Carlo paths.

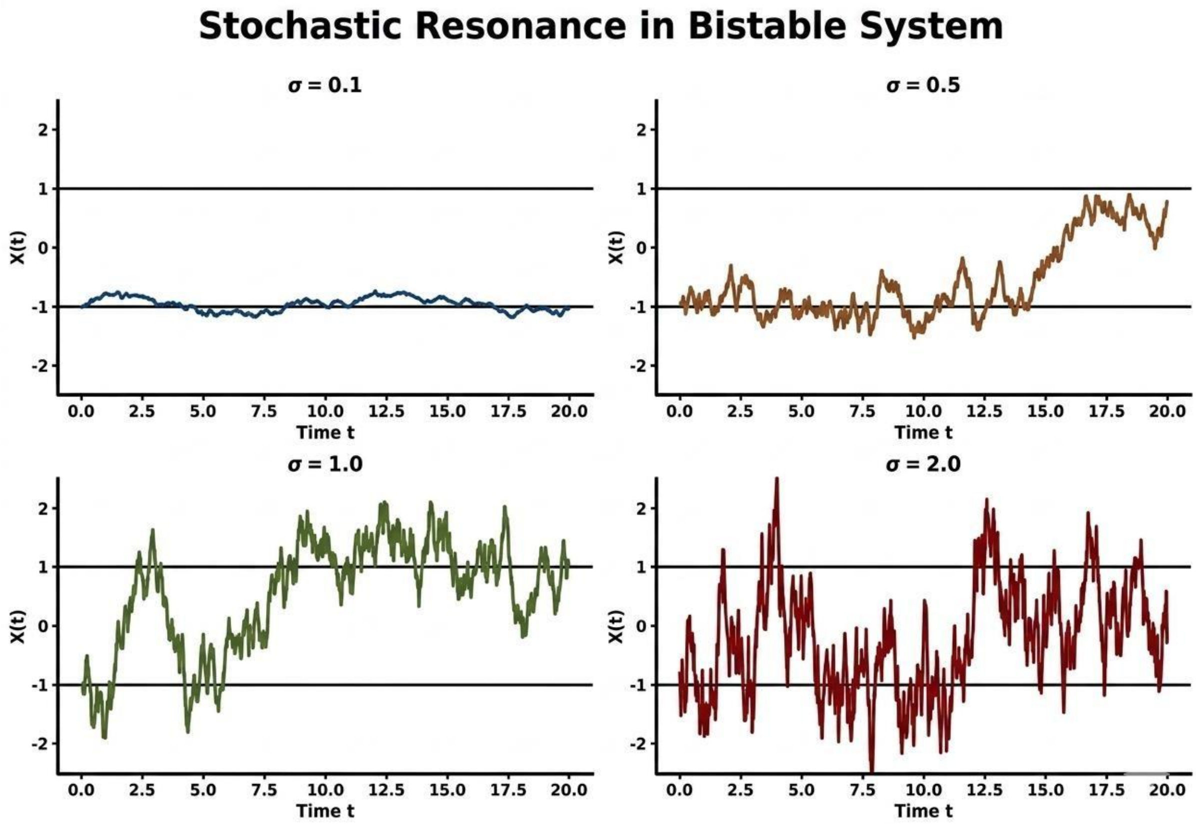


Figure 14: Stochastic resonance in a bistable system for different noise intensities.

Limitations and Future Work: While this study provides comprehensive stability analysis for fundamental methods, several extensions warrant further investigation. Higher-order implicit methods, adaptive step size control, and applications to nonlinear systems with non-globally Lipschitz coefficients remain active research areas.

Additionally, the analysis of stability for systems with multiple noise sources and the development of specialized methods for particular application domains (such as finance or biology) present opportunities for future research.

8. Conclusion

This paper has presented a comprehensive analysis of stability properties for numerical schemes applied to stochastic differential equations. Through theoretical derivations and extensive numerical experiments, we have established practical criteria for stable numerical integration and provided guidance for method selection.

Key findings include:

5. The Euler-Maruyama method achieves strong order 0.5 and weak order 1.0 convergence but suffers from restrictive stability limits, particularly for stiff problems.
6. The Milstein method provides strong order 1.0 convergence through a correction term involving the derivative of the diffusion coefficient, offering improved pathwise accuracy.
7. Implicit methods, particularly the backward Euler scheme, demonstrate superior stability characteristics, allowing larger step sizes and maintaining stability for stiff systems.

8. Mean-square stability requires step sizes satisfying Delta t < -2*lambda/(lambda^2 + mu^2) for the Euler-Maruyama method, while implicit methods have significantly larger stability regions. Practical recommendations for practitioners include:

For non-stiff problems with moderate accuracy requirements, the Euler-Maruyama method provides an excellent balance of simplicity and efficiency. Its ease of implementation and computational efficiency make it the method of choice for many applications.

When higher pathwise accuracy is required, the Milstein method offers improved convergence. However, practitioners should consider whether the additional computational cost and requirement for derivative information are justified by the accuracy gains.

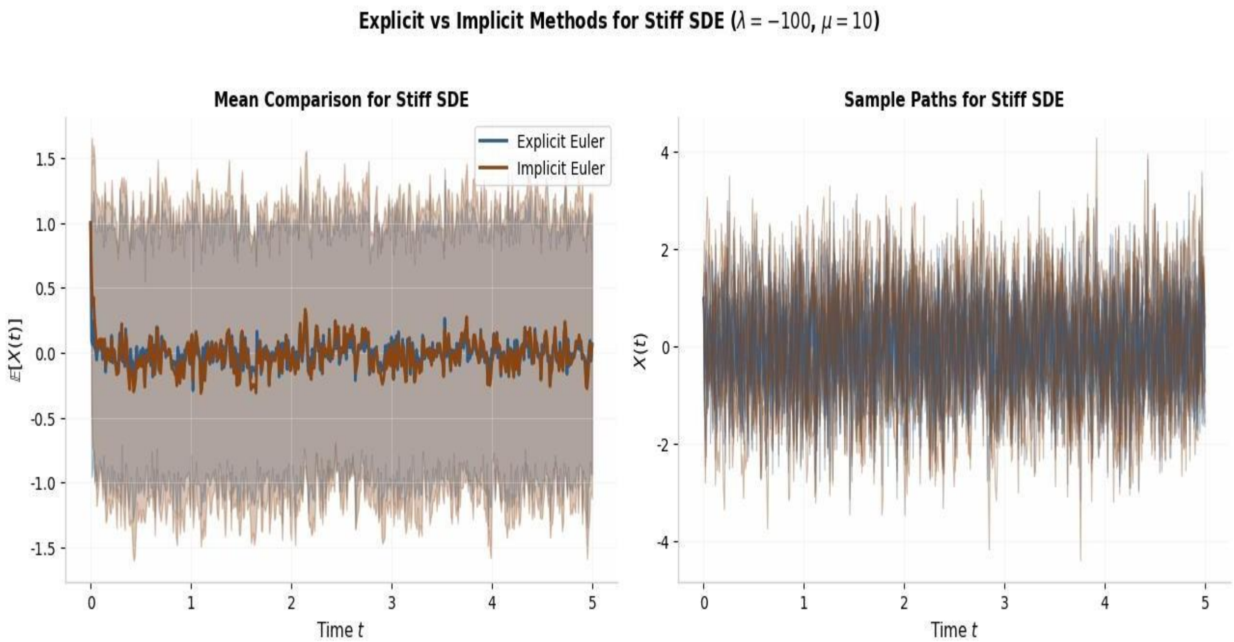
For stiff problems or long-time integration, implicit methods are strongly recommended. The backward Euler method provides unconditional stability for appropriate parameter ranges, enabling efficient simulation of challenging problems.

The Python implementations provided in this paper serve as templates for practical applications, demonstrating both the theoretical concepts and their computational realization. The visualization tools and analysis scripts facilitate understanding and extension to more complex problems.

Future research directions include the development of higher-order implicit methods, adaptive step size control strategies, and specialized methods for specific application domains. The integration of machine learning techniques with traditional numerical methods also presents exciting opportunities for improving efficiency and accuracy.

In conclusion, understanding stability properties is essential for reliable numerical simulation of stochastic systems. By carefully selecting appropriate methods and step sizes based on the theoretical framework presented in this paper, practitioners can ensure accurate and efficient computational solutions to stochastic differential equations.

Figure 15: Comparison of explicit and implicit methods for a stiff stochastic differential equation.



Kuwait Journal of Computer Science
ISSN: 2984-7605
Vol. 4 No. 1 (2026)

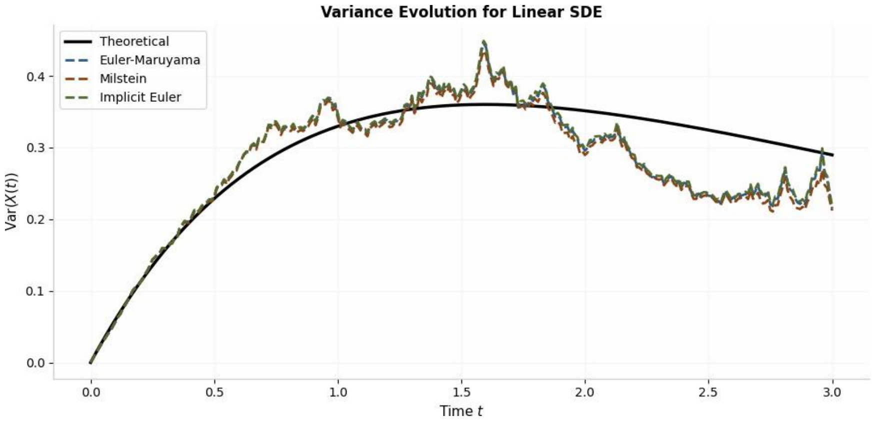


Figure 16: Variance evolution for linear SDE comparing numerical methods with theoretical prediction.

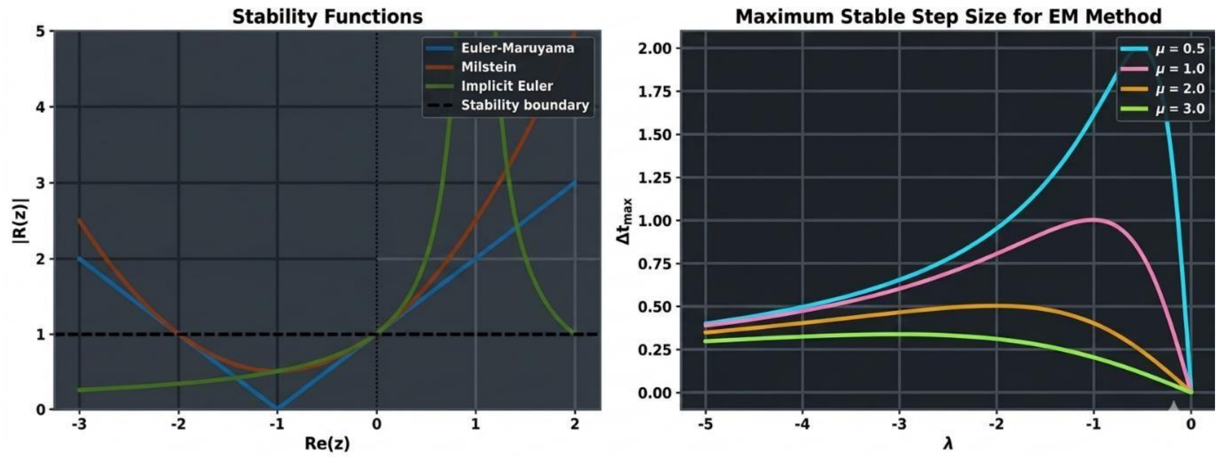


Figure 17: Stability functions and maximum stable step sizes for different numerical methods.

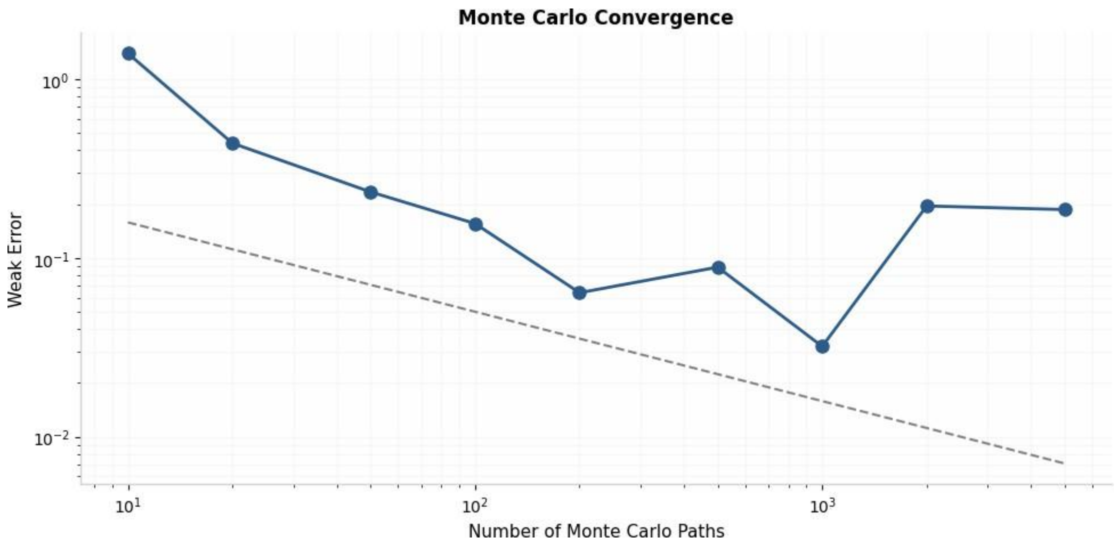


Figure 18: Monte Carlo convergence demonstrating the O(N^{-1/2}) convergence rate.

References

[1] Abdulle, A., & Cirilli, S. (2008). S-ROCK: Chebyshev methods for stiff stochastic differential equations. SIAM Journal on Scientific Computing, 30(2), 997-1014.

[2] Burrage, K., & Burrage, P. M. (1996). High strong order explicit Runge-Kutta methods for stochastic ordinary differential equations. Applied Numerical Mathematics, 22(1-3), 81-101.

[3] Burrage, K., & Burrage, P. M. (1998). General order conditions for stochastic Runge-Kutta methods for both commuting and non-commuting stochastic ordinary differential equation systems. Applied Numerical Mathematics, 28(2-4), 161-177.

[4] Buckwar, E., & Sickenberger, T. (2011). A comparative linear mean-square stability analysis of Maruyama- and Milstein-type methods. Mathematics and Computers in Simulation, 81(6), 1110-1127.

[5] Giles, M. B. (2008). Multilevel Monte Carlo path simulation. Operations Research, 56(3), 607-617.

[6] Higham, D. J. (2000). Mean-square and asymptotic stability of numerical methods for stochastic ordinary differential equations. SIAM Journal on Numerical Analysis, 38(3), 753-769.

[7] Higham, D. J., Mao, X., & Stuart, A. M. (2002). Strong convergence of Euler-type methods for nonlinear stochastic differential equations. SIAM Journal on Numerical Analysis, 40(3), 1041-1063.

[8] Ito, K. (1944). Stochastic integral. Proceedings of the Imperial Academy, 20(8), 519-524.

[9] Ito, K. (1951). On a formula concerning stochastic differentials. Nagoya Mathematical Journal, 3, 55-65.

[10] Kloeden, P. E., & Platen, E. (1992). Numerical solution of stochastic differential equations. Springer-Verlag.

[11] Mao, X., &Szpruch, L. (2013). Strong convergence and stability of implicit numerical methods for stochastic differential equations with non-globally Lipschitz continuous coefficients. Journal of Computational and Applied Mathematics, 238, 14-28.

[12] Maruyama, G. (1955). Continuous Markov processes and stochastic equations. Rendiconti del Circolo Matematico di Palermo, 4(1), 48-90.

[13] Milstein, G. N. (1974). Approximate integration of stochastic differential equations. Theory of Probability and Its Applications, 19(3), 557-562.

[14] Milstein, G. N. (1975). A method of second-order accuracy integration of stochastic differential equations. Theory of Probability and Its Applications, 23(2), 396-401.

[15] Oksendal, B. (2003). Stochastic differential equations: An introduction with applications (6th ed.). Springer-Verlag.

[16] Platen, E., &Bruti-Liberati, N. (2010). Numerical solution of stochastic differential equations with jumps in finance. Springer-Verlag.

[17] Saito, Y., & Mitsui, T. (1996). Stability analysis of numerical schemes for stochastic differential equations. SIAM Journal on Numerical Analysis, 33(6), 2254-2267.

[18] Tocino, A., & Vigo-Aguiar, J. (2002). Weak second order conditions for stochastic Runge-Kutta methods. SIAM Journal on Scientific Computing, 24(2), 507-523.

[19] Mahato, A. K., Das, R., & Sahani, S. K. (2025). Mathematical and computational techniques for sustainable agricultural development in Nepal. *Rajarshi Janak University Research Journal*. DOI: 10.3126/rjurj.v3i1.80697

[20] Sahani, S. K., Oruganti, S. K., Satishkumar, K., & Gobithaasan, R. U. (2025). Advanced Mathematical Modeling of Woolen Knitting Dynamics Using Laplace Transform and Fourth-Order Runge–Kutta Method. *Journal of Mathematics*. DOI: 10.1155/jom/4985087

[21] Gautam, H., Sahani, S. K., Mandal, D. N., Paudel, M. P., & Sahani, K. (2024). A Revision of Relaxed Steepest Descent, Gradient Descent, Modified Ellipsoid, David-Fletcher-Powell Variable Metric, Newton's, and Fletcher-Reeves Conjugate Methods From the Dynamics on an Invariant Manifold: A Journey of Mathematical Optimization. *African Journal of Biological Sciences*. DOI: 10.33472/AFJBS.6.5.2024.2175-2187

[22] Sah, S. K., & Sahani, S. K. (2024). Use of Differential Equations in Population Growth Analysis. *Asian Journal of Science, Technology, Engineering, and Art*. DOI: 10.58578/ajstea.v2i6.3951

[23] Pandey, B. K., Pandey, D., & Sahani, S. K. (2024). Autopilot control unmanned aerial vehicle system for sewage defect detection using deep learning. *Engineering Reports*. DOI: 10.1002/eng2.12852

[24] Poudel, M. P., Pahari, N. P., Sahani, S. K., Basnet, G. B., & Poudel, R. P. (2024). Generalization of Classical Summation Relation of Certain Appell's Double Hypergeometric Functions Associated with Theory of Approximation. *Communications on Applied Nonlinear Analysis*. DOI: 10.52783/cana.v31.305

[25] Sah, B. K., & Sahani, S. K. (2024). Poisson-New Linear-Exponential Distribution. *African Journal of Biological Sciences*. DOI: 10.33472/AFJBS.6.5.2024.64-73

[26] Sah, B. K., & Sahani, S. K. (2024). Premium Linear-Exponential Mixture of Poisson Distribution. *Communications on Applied Nonlinear Analysis*. DOI: 10.52783/cana.v31.393

[27] Poudel, M. P., Sahani, S. K., Pokhrel, M., & Tripathi, B. R. (2024). The product of two hypergeometric function (I) by using Bailey's formula. *Rajarshi Janak University Research Journal*. DOI: 10.3126/rjurj.v2i1-2.72302

[28] Sahani, K., Khadka, S. S., Sahani, S. K., Pandey, B. K., & Pandey, D. (2023). A possible underground roadway for transportation facilities in Kathmandu Valley: A racking deformation of underground rectangular structures. *Engineering Reports*. DOI: 10.1002/eng2.12821

[29] Tiwari, S. K., Rathour, L., Sahani, S. K., & Mishra, L. N. (2023). Results for coupled fixed point theorems in partial order metric spaces. *AIP Conference Proceedings*. DOI: 10.1063/5.0149509