Exercise Sheet: Numerical Schemes for Stochastic Differential Equations

Dr. Abhishek Chaudhary

Numerical Analysis Group, Department of Mathematics,

University of Tübingen

Assignment Overview

This assignment introduces numerical methods for solving stochastic differential equations (SDEs). You will implement and compare numerical schemes by simulating sample paths, computing errors, and analyzing convergence rates. For each exercise:

- Simulate M sample paths (trajectories) over the time interval $[0, 1]$ using time steps $\Delta t = 2^{-k}$ for $k = 4, 5, 6, 7$ (i.e., $\Delta t = 1/16, 1/32, 1/64, 1/128$).
- Create a reference solution using a very small time step $\Delta t_{\text{ref}} = 2^{-12} = 1/4096$.
- Compute the root-mean-square (RMS) strong error at the final time $T = 1$:

$$E(\Delta t) = \sqrt{\frac{1}{M}\sum_{j=1}^{M}\left|X_T^{(j),\text{scheme}} - X_T^{(j),\text{ref}}\right|^2},$$

  where $X_T^{(j),\text{scheme}}$ is the solution from the numerical scheme for the j-th path, and $X_T^{(j),\text{ref}}$ is the reference solution for the same path.
- Plot $\log(E(\Delta t))$ versus $\log(\Delta t)$ for the different Δt values. Fit a straight line to estimate the empirical order of convergence (the slope of the line).
- Use a programming language like Python, MATLAB, or R for simulations. Include your code and plots in your submission.

Hint: The empirical order of convergence is found by fitting a line to the log-log plot. If $E(\Delta t) \approx C(\Delta t)^{\gamma}$, then $\log(E(\Delta t)) \approx \log(C) + \gamma \log(\Delta t)$, and the slope γ is the order of convergence.

Getting Started

An SDE has the form $dX_t = a(X_t, t)dt + b(X_t, t)dW_t$, where a is the drift, b is the diffusion, and W_t is a Wiener process (Brownian motion). Numerical schemes approximate the solution by discretizing time into steps Δt. The Euler–Maruyama (EM) scheme is a common method, defined as:

$$X_{n+1} = X_n + a(X_n, t_n)\Delta t + b(X_n, t_n)\Delta W_n,$$

where $\Delta W_n = W_{t_{n+1}} - W_{t_n} \sim \mathcal{N}(0, \Delta t)$ is a normal random variable with mean 0 and variance Δt. For each sample path, generate a sequence of ΔW_n using random numbers.

Programming Tips:

- Use a loop to simulate M sample paths.
- For each path, generate random increments $\Delta W_n \sim \mathcal{N}(0, \Delta t)$.
- Store the final value X_T for each path to compute the RMS error.
- Use the same random seed for the reference solution and the scheme to ensure the same Brownian increments.

Exercises

Exercise 1. Ornstein–Uhlenbeck Process (Linear SDE)

Consider the SDE:

$$dX_t = \theta(\mu - X_t)dt + \sigma dW_t, \quad X_0 = 0, \quad T = 1,$$

with parameters $\theta = 2$, $\mu = 1$, $\sigma = 0.5$.

(a) The exact solution to this SDE is:

$$X_t = \mu(1 - e^{-\theta t}) + \sigma \int_0^t e^{-\theta(t-s)} dW_s.$$

Implement this incrementally for a time step Δt. For a time step from t_n to t_{n+1}, compute:

$$X_{t_{n+1}} = e^{-\theta \Delta t} X_{t_n} + \mu(1 - e^{-\theta \Delta t}) + \sigma \sqrt{\frac{1 - e^{-2\theta \Delta t}}{2\theta}} Z_n,$$

where $Z_n \sim \mathcal{N}(0, 1)$. Write code to simulate $M = 5000$ paths.

(b) Implement the Euler–Maruyama scheme:

$$X_{n+1} = X_n + \theta(\mu - X_n)\Delta t + \sigma \Delta W_n,$$

with $\Delta W_n \sim \mathcal{N}(0, \Delta t)$. Simulate $M = 5000$ paths for $\Delta t = 2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}$.

(c) Compute the RMS error $E(\Delta t)$ at $T = 1$ by comparing the EM solution to the exact solution for the same Brownian increments.

(d) Plot $\log(E(\Delta t))$ versus $\log(\Delta t)$. Fit a straight line and report the slope (empirical order). The theoretical strong order for EM is 1.0 for this SDE. Does your result match? **Hint**: Use a plotting library (e.g., Matplotlib in Python) and a linear regression function to find the slope.

Exercise 2. Geometric Brownian Motion

Consider the SDE:

$$dX_t = rX_t dt + \sigma X_t dW_t, \quad X_0 = 1, \quad T = 1,$$

with $r = 0.03$, $\sigma = 0.2$.

(a) The exact solution is:

$$X_t = X_0 \exp\left(\left(r - \frac{\sigma^2}{2}\right) t + \sigma W_t\right).$$

Implement this solution for $t = 0, \Delta t, 2\Delta t, \ldots, 1$. Generate W_t by summing increments $\Delta W_n \sim \mathcal{N}(0, \Delta t)$.

(b) Implement the EM scheme:

$$X_{n+1} = X_n + rX_n \Delta t + \sigma X_n \Delta W_n,$$

for $M = 5000$ paths and $\Delta t = 2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}$.

(c) Compute the RMS error at $T = 1$ using the exact solution as the reference.

(d) Plot $\log(E(\Delta t))$ versus $\log(\Delta t)$. Estimate the order of convergence. The theoretical strong order is 0.5. Explain any deviations. **Hint**: Ensure the same random increments are used for both the exact and EM solutions.

Exercise 3. Nonlinear Logistic SDE

Consider the SDE:

$$dX_t = \alpha X_t \left(1 - \frac{X_t}{K}\right) dt + \sigma X_t dW_t, \quad X_0 = 0.5, \quad T = 1,$$

with $\alpha = 1.5$, $K = 2$, $\sigma = 0.3$.

(a) Since no exact solution is available, use the EM scheme with $\Delta t_{\text{ref}} = 2^{-12}$ as the reference. Implement EM:

$$X_{n+1} = X_n + \alpha X_n \left(1 - \frac{X_n}{K}\right) \Delta t + \sigma X_n \Delta W_n,$$

for $M = 2000$ paths and $\Delta t = 2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}$.

(b) Compute $E(\Delta t)$ at $T = 1$ using the reference solution.

(c) Plot the convergence and estimate the order. Discuss if negative values occur in simulations and how they affect the results (since the logistic model typically expects positive values). Suggest ways to handle negative values, if observed. **Hint**: Check if X_n becomes negative and consider a modification like $X_n \leftarrow \max(X_n, 0)$.

Exercise 4. Scalar Nonlinear SDE: Euler–Maruyama vs. Milstein

Consider the SDE:

$$dX_t = (X_t - X_t^3)dt + \sigma X_t dW_t, \quad X_0 = 1, \quad \sigma = 0.5, \quad T = 1.$$

(a) Verify the EM update:

$$X_{n+1}^{\text{EM}} = X_n + (X_n - X_n^3)\Delta t + \sigma X_n \Delta W_n.$$

Implement it for $M = 2000$ paths.

(b) Verify the Milstein update:

$$X_{n+1}^{\text{Mil}} = X_n + (X_n - X_n^3)\Delta t + \sigma X_n \Delta W_n + \frac{1}{2}\sigma^2 X_n \left((\Delta W_n)^2 - \Delta t\right).$$

Implement it for $M = 2000$ paths. Note the extra term accounts for the stochastic second-order effect.

(c) Use the Milstein scheme with $\Delta t_{\text{ref}} = 2^{-12}$ as the reference.

(d) Compute $E(\Delta t)$ for both schemes and plot both errors on the same log-log plot. Report the slopes. The theoretical orders are 0.5 for EM and 1.0 for Milstein. **Hint**: The Milstein scheme includes a correction term derived from the diffusion coefficient's derivative.

Exercise 5. CIR-Type SDE with Square-Root Diffusion

Consider the SDE:

$$dX_t = \kappa(\theta - X_t)dt + \sigma\sqrt{X_t}dW_t, \quad X_0 = 1, \quad T = 1,$$

with $\kappa = 2$, $\theta = 1$, $\sigma = 0.4$.

(a) Implement the EM scheme with full truncation to ensure non-negativity:

$$X_{n+1} = X_n + \kappa(\theta - X_n)\Delta t + \sigma\sqrt{\max(X_n, 0)}\Delta W_n,$$

$$X_{n+1} \leftarrow \max(X_{n+1}, 0).$$

Simulate $M = 3000$ paths.

(b) Use the same scheme with $\Delta t_{\text{ref}} = 2^{-12}$ as the reference.

(c) Compute the RMS error at $T = 1$.

(d) Plot $\log(E(\Delta t))$ versus $\log(\Delta t)$ and estimate the convergence order. **Hint**: The square-root diffusion requires careful handling to avoid taking the square root of negative values.

Submission Instructions

Submit a report including:

- Your code (e.g., Python, MATLAB) for each exercise.
- Log-log plots for each exercise with fitted lines and reported slopes.
- Brief explanations of your results, including any issues (e.g., negative values) and how you addressed them.
- Comparisons of observed convergence orders to theoretical values.

You may discuss with classmates, but your submission must be your own work. Submit by **July 24, 2025**.