Chapter 6: Stochastic Differential Equations

Lecturer: Kenneth Ng

Preview

This chapter introduces the theory and techniques in solving linear stochastic differential equations (SDEs). We will also cover methods to simulate the solutions of SDEs.

Key topics in this chapter:

1. General theory of SDEs;
2. Arithmetic and geometric SDEs;
3. Simulations of SDEs.

1 Theory of Stochastic Differential Equations

A process X satisfies a stochastic integral equation if it admits the following representation:

$$X_t = \xi + \int_0^t b(s, X_s)\, ds + \int_0^t \sigma(s, X_s)\, dB_s,\ 0 \le t \le T,$$

where $T > 0$, ξ is a random variable, and $b, \sigma : [0, \infty) \times \mathbb{R} \to \mathbb{R}$. We can also write the above in differential form:

$$dX_t = b(t, X_t)\, dt + \sigma(t, X_t)\, dB_t,\ X_0 = \xi,\ 0 \le t \le T. \tag{1}$$

We call (1) a ***stochastic differential equation*** (***SDE***). Given a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\ge 0}, \mathbb{P})$, we want to find a process X that satisfies the SDE (1). Such a process is called a ***(strong) solution*** of the SDE (1).

Before attempting to solve a SDE, a natural question is whether a solution exists, and if so, whether it is unique. The existence and uniqueness of a solution depend on the regularity of the coefficients b and σ. Even in the simpler case of ordinary differential equations (ODEs), where $\sigma = 0$, uniqueness may fail if b lacks sufficient regularity.

Example 1.1 Let $\alpha > 0$. Consider the ODE

$$\frac{dX_t}{dt} = |X_t|^\alpha, \ X_0 = 0.$$

The equation has a unique solution for $\alpha \geq 1$, which is given by $X_t = 0$ for $t \geq 0$. If $\alpha \in (0,1)$, the equation has infinitely many solution. Indeed, for any arbitrary $s \geq 0$, the function

$$X_t = \begin{cases} 0, & \text{if } 0 \leq t \leq s; \\ \left(\dfrac{t-s}{\beta}\right)^\beta, & \text{if } t \geq s, \end{cases}$$

where $\beta := (1-\alpha)^{-1}$, is a solution of the given ODE.

To ensure the existence and uniqueness of a solution to an SDE, we introduce the following regularity conditions:

Definition 1.1 A function $f(t,x)$ is said satisfy the

1. ***global Lipschitz condition*** if there exists $K > 0$ such that, for any $x, y \in \mathbb{R}$ and $t \geq 0$,
$$|f(t,x) - f(t,y)| \leq K|x-y|;$$
2. ***linear growth condition*** if there exists $L > 0$ such that, for any $t \geq 0$ and $x \in \mathbb{R}$,
$$|f(t,x)| \leq L(1+|x|).$$

Remark 1.1. If there exists $C > 0$ such that $|f(t,0)| \leq C$ for all $t \geq 0$, then f being globally Lipschitz implies that f is of linear growth. To see this, for any $t \geq 0$ and $x \in \mathbb{R}$,

$$|f(t,x)| \leq |f(t,x) - f(t,0)| + |f(t,0)| \leq K|x| + |f(t,0)| \leq C + K|x| \leq L(1+|x|),$$

where $L := \max\{C, K\}$.

We now introduce the general theory of SDEs, whose proof is out of the scope of our course:

Theorem 1.2 Suppose that the coefficients b and σ satisfy the global Lipschitz and linear growth conditions, and $\mathbb{E}[|\xi|^2] < \infty$. Then, the SDE (1) admits a unique solution, which satisfies

$$\mathbb{E}\left[\int_0^T |X_t|^2 \, dt\right] < \infty.$$

2 Arithmetic SDEs

An arithmetic SDE takes the form

$$dX_t = (a + bX_t)\,dt + \sigma dB_t,\ 0 \le t \le T, \tag{2}$$

where $a, b, \sigma \in \mathbb{R}$. The OU process

$$dX_t = \theta(\mu - X_t)\,dt + \sigma\,dB_t,\ X_0 = r,$$

is an example of an arithmetic SDE with $a = \theta\mu$ and $b = -\theta$.

The SDE (2) can be solved using the ***method of integrating factor***. Consider the process $Y_t := e^{-bt}X_t$. By Itô's lemma,

$$\begin{aligned} dY_t &= -bY_t\,dt + e^{-bt}\,dX_t \\ &= \left(-bY_t + ae^{-bt} + bY_t\right)\,dt + \sigma e^{-bt}\,dB_t \\ &= ae^{-bt}\,dt + \sigma e^{-bt}\,dB_t. \end{aligned}$$

Integrating both sides yields

$$e^{-bt}X_t - X_0 = Y_t - Y_0 = \int_0^t ae^{-bs}\,ds + \int_0^t \sigma e^{-bs}\,dB_s$$

Rearranging the above yields

$$X_t = \begin{cases} e^{bt}X_0 + \dfrac{a}{b}(e^{bt} - 1) + \sigma \displaystyle\int_0^t e^{b(t-s)}\,dB_s, & \text{if } b \ne 0; \\ X_0 + at + \sigma B_t, & \text{if } b = 0. \end{cases} \tag{3}$$

From (3), we see that X can take both positive and negative values. When $b = 0$, X follows an arithmetic Brownian motion, or a Brownain motion with drift parameter a.

Suppose that X_0 is deterministic, we can compute the mean and variance of X_t:

$$\mathbb{E}[X_t] = \begin{cases} e^{bt}X_0 + \dfrac{a}{b}(e^{bt} - 1), & \text{if } b \ne 0; \\ X_0 + at, & \text{if b=0,} \end{cases}$$

$$\text{Var}[X_t] = \text{Var}\left[\sigma \int_0^t e^{b(t-s)}\,dB_s\right] = \sigma^2 \int_0^t e^{2b(t-s)}\,ds = \begin{cases} \dfrac{\sigma^2}{2b}\left(e^{2bt} - 1\right), & \text{if } b \ne 0; \\ \sigma^2 t, & \text{if } b = 0, \end{cases}$$

where we have use Itô's isometry when computing the variance. By Proposition 3.3 in Chapter 5, we have

$$X_t \sim \begin{cases} \mathcal{N}\left(e^{bt}X_0 + \frac{a}{b}(e^{bt}-1), \frac{\sigma^2}{2b}\left(e^{2bt}-1\right)\right), & \text{if } b \neq 0; \\ \mathcal{N}\left(X_0 + at, \sigma^2 t\right), & \text{if } b = 0, \end{cases} \tag{4}$$

Example 2.1 Let X be an OU process which satisfies

$$dX_t = \theta(\mu - X_t)\,dt + \sigma\,dB_t,\ \ X_0 = r,$$

where $\theta > 0$. Find an explicit expression for X and deduce its distribution.

Solution. Using the method of integrating factor with $b = -\theta$ and $a = \mu\theta$, we have

$$X_t = e^{-\theta t}r + \mu(1 - e^{-\theta t}) + \sigma\int_0^t e^{-\theta(t-s)}\,dB_s,$$

and

$$X_t \sim \mathcal{N}\left(e^{-\theta t}r + \mu(1-e^{-\theta t}), \frac{\sigma^2}{2\theta}(1-e^{-2\theta t})\right).$$

□

When a, b, σ are time-dependent deterministic functions, i.e., X follows

$$dX_t = (a_t + b_t X_t)\,dt + \sigma_t\,dB_t,$$

we can solve the equation by considering

$$Y_t := e^{-\int_0^t b_s\,ds}X_t.$$

Following the above calculations, we can deduce that

$$X_t = e^{\int_0^t b_s\,ds}X_0 + \int_0^t a_s e^{\int_s^t b_u\,du}\,ds + \int_0^t \sigma_s e^{\int_s^t b_u\,du}\,dB_s.$$

3 Geometric SDEs

A geometric SDE takes the form

$$dX_t = \mu X_t\,dt + \sigma X_t\,dB_t,\ \ 0 \le t \le T, \tag{5}$$

where $\mu, \sigma \in \mathbb{R}$. Compared with arithmetic SDE (2), the diffusion term in (5) also depends linearly in X.

To solve the SDE (5), we consider the process

$$Y_t := \ln X_t.$$

Applying Itô's lemma to Y, we have

$$\begin{aligned}
dY_t &= \frac{1}{X_t}\,dX_t - \frac{1}{2X_t^2} d\langle X\rangle_t \\
&= \frac{1}{X_t}\,(\mu X_t\,dt + \sigma X_t\,dB_t) - \frac{1}{2X_t^2}(\sigma X_t)^2\,dt \\
&= \left(\mu - \frac{\sigma^2}{2}\right)\,dt + \sigma\,dB_t.
\end{aligned}$$

Integrating both sides yields

$$\ln X_t - \ln X_0 = Y_t - Y_0 = \left(\mu - \frac{\sigma^2}{2}\right) t + \sigma B_t.$$

Rearranging the above yields

$$\boxed{X_t = X_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right) t + \sigma B_t\right). \qquad (6)}$$

The solution X of the SDE (5) is also called a ***geometric Brownian motion*** (***GBM***). From (6), we see that $X_t > 0$ as long as $X_0 > 0$. This makes GBM an ideal model for a stock price process. In finance, the process S that satisfies

$$dS_t = \mu S_t\,dt + \sigma S_t\,dB_t,$$

$\sigma > 0$, is called the ***Black-Scholes model*** with *rate of return* μ and *volatility* σ. If X_0 is deterministic and $X_0 > 0$, X_t follows a log-normal distribution:

$$\boxed{X_t \sim \text{lognormal}\left(\ln X_0 + \left(\mu - \frac{\sigma^2}{2}\right) t, \sigma^2 t\right).}$$

When μ and σ are time-dependent, i.e., X follows

$$dX_t = \mu_t X_t\,dt + \sigma_t X_t\,dB_t,$$

the solution of the SDE can be obtained similarly by considering $Y_t = \ln X_t$, which is given by

$$X_t = X_0 \exp\left(\int_0^t \left(\mu_s - \frac{\sigma_s^2}{2}\right)\,ds + \int_0^t \sigma_s\,dB_s\right).$$

Example 3.1 Let S be the stock price process which follows the Black-Scholes model with the rate of return μ, volatility σ, and $S_0 = 1$. For any $p > 0$, compute $\mathbb{E}[S_t^p]$.

Solution. It is known that

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

so that

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

Hence, if we let $Z \sim \mathcal{N}(0,1)$, we have

$$\begin{aligned}
\mathbb{E}[S_t^p] &= e^{p\left(\mu-\frac{\sigma^2}{2}\right)t}\mathbb{E}[e^{p\sigma B_t}] \\
&= e^{p\left(\mu-\frac{\sigma^2}{2}\right)t}\mathbb{E}[e^{p\sigma\sqrt{t}Z}] \\
&= e^{p\left(\mu-\frac{\sigma^2}{2}\right)t}e^{\frac{p^2\sigma^2 t}{2}} \\
&= e^{p\mu t+\frac{1}{2}p(p-1)\sigma^2 t}.
\end{aligned}$$

□

4 Simulations of SDEs

In this section, we introduce fundamental techniques for simulating solutions of stochastic differential equations (SDEs) using random number generators and built-in functions. `MATLAB` will be used for illustration purposes.

4.1 Brownian Motions

We divide the time horizon $[0, T]$ into N subintervals, with time points $\{t_0 = 0, t_1, t_2, \ldots, t_N = T\}$, where $t_i = i\Delta t$ and $\Delta t = \frac{T}{N}$. To simulate the values of a Brownian motion at these discrete times, we make use of the fact that Brownian motion has independent and Gaussian increments. Specifically, we generate N independent standard normal random variables $Z_1, \ldots, Z_N \sim \mathcal{N}(0,1)$, and define the Brownian path recursively by:

$$\begin{aligned}
B_{t_0} &:= 0, \\
B_{t_{i+1}} &:= B_{t_i} + \sqrt{\Delta t}\, Z_{i+1}, \quad \text{for } i = 1, \ldots, N-1.
\end{aligned}$$

This yields a discrete-time approximation of the Brownian motion over the interval $[0, T]$.

```
% Parameters
T = 1;            % Time horizon
N = 1000;         % Number of time steps
dt = T/N;         % Time step size
t = linspace(0, T, N+1);
B = zeros(1, N+1);      % Brownian motion
rng(1);           % Set seed to ensure reproducible output
dB = sqrt(dt)*randn(1, N);  % Increments ~ N(0, dt)

% Simulate BM
for i = 2:N+1
    B(i) = B(i-1) + dB(i-1);
end

% Plot
plot(t, B);
xlabel('Time'); ylabel('B_t');
title('Brownian Motion');
```

Listing 1: Brownian Motion Simulation

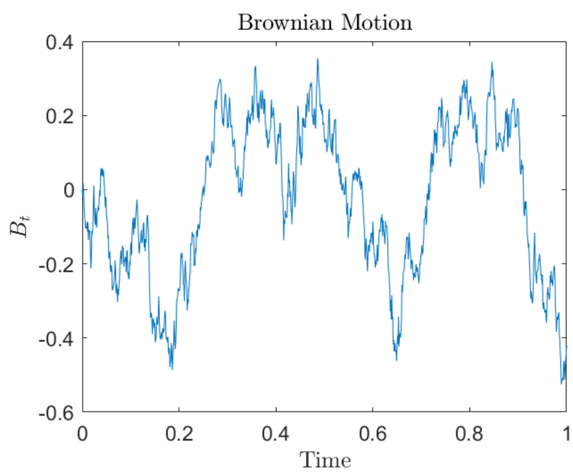

Figure 1: A simulated path of the standard Brownian motion

4.2 Arithmetic SDEs

Recall that the solution of the arithmetic SDE (2) is given, for $b \neq 0$, by

$$X_t = e^{bt} X_0 + \frac{a}{b}\left(e^{bt} - 1\right) + \sigma \int_0^t e^{b(t-s)}\, dB_s.$$

When $b = 0$, the process reduces to an arithmetic Brownian motion, which can be simulated in the same manner as a Brownian motion with drift. In this subsection, we simulate X_t for the case $b \neq 0$ using its explicit solution, where the stochastic integral is approximated by a sum of Brownian increments. Below presents the code for simulating the solution of an arithmetic SDE with initial condition $X_0 = 0.02$, $a = 0.04$, $b = -0.5$, and $\sigma = 0.1$.

```
% Parameters
a = 0.04;
b = -0.5;
sigma = 0.1;
X0 = 0.02;

T = 1;
N = 1000;
dt = T/N;
t = linspace(0, T, N+1);

% Preallocate
X = zeros(1, N+1);
X(1) = X0;

% Brownian increments
dB = sqrt(dt) * randn(1, N);

% Explicit solution path
for i = 2:N+1
    % Stochastic integral approximation:
    stoch_int = sum(exp(b * (t(i) - t(1:i-1))) .* dB(1:i-1));
    X(i) = exp(b*t(i)) * X0 + (a/b) * (exp(b*t(i)) - 1) + sigma * stoch_int;
end

% Plot
plot(t, X);
xlabel('Time');
ylabel('X_t');
title('Arithmetic SDE');
```

Listing 2: Arithmetic SDE Simulation with Explicit Solution

Alternatively, when the initial value X_0 is deterministic, we can simulate the process $(X_t)_{t\geq 0}$ by exploiting its Gaussian distribution; see (4). In particular, for any $0 \leq t_i < t_{i+1} \leq T$, conditioning on X_{t_i}, we have

$$X_{t_{i+1}} = e^{b\Delta t} X_{t_i} + \frac{a}{b}\left(e^{b\Delta t} - 1\right) + \sigma \int_{t_i}^{t_{i+1}} e^{b(t_{i+1}-u)}\, dB_u.$$

In particular, we have

$$X_{t_{i+1}} \mid X_{t_i} \sim \mathcal{N}\left(e^{b\Delta t} X_{t_i} + \frac{a}{b}\left(e^{b\Delta t} - 1\right),\quad \frac{\sigma^2}{2b}\left(e^{2b\Delta t} - 1\right)\right).$$

Therefore, given X_{t_i}, we can simulate $X_{t_{i+1}}$ by drawing a standard normal random variable

$Z_i \sim \mathcal{N}(0,1)$ and updating via

$$X_{t_{i+1}} = e^{b\Delta t} X_{t_i} + \frac{a}{b}\left(e^{b\Delta t} - 1\right) + \sqrt{\frac{\sigma^2}{2b}\left(e^{2b\Delta t} - 1\right)}\, Z_i.$$

The following MATLAB code illustrates this simulation approach.

```
% Parameters
a = 0.04;
b = -0.5;
sigma = 0.1;
X0 = 0.02;

T = 1;
N = 1000;
dt = T / N;
t = linspace(0, T, N+1);

% Preallocate
X = zeros(1, N+1);
X(1) = X0;

% Simulate X_t iteratively using conditional distribution
for i = 2:N+1
    exp_bt = exp(b * dt);

    % Conditional mean
    mu_cond = exp_bt * X(i-1) + (a / b) * (exp_bt - 1);

    % Conditional variance of stochastic integral
    var_cond = (sigma^2 / (2 * b)) * (exp(2 * b * delta_t) - 1);

    % Sample X(i) ~ Normal(mu_cond, var_cond)
    X(i) = mu_cond + sqrt(var_cond) * randn();
end

% Plot result
plot(t, X;
xlabel('Time');
ylabel('X_t');
title('Simulation of Arithmetic SDE using conditional Gaussian increments');
```

Listing 3: Simulation of Arithmetic SDE using conditional Gaussian increments

4.3 Geometric Brownian Motions

A GBM is the solution of the SDE (5), which takes the explicit form

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

The most straightforward way to simulate a GBM is to first simulate a standard Brownian motion B_t, followed by computing X_t using the above formula. Below presents the code for simulating a GBM with initial condition $X_0 = 1$, rate of return $\mu = 0.1$ and volatility $\sigma = 0.2$.

```
% Parameters
mu = 0.1;
sigma = 0.2;
X0 = 1;

T = 1;
N = 1000;
dt = T/N;
t = linspace(0, T, N+1);
B = zeros(1, N+1);
X = zeros(1, N+1);
X(1) = X0;
dB = sqrt(dt)*randn(1, N);

% Simulate BM
for i = 2:N+1
    B(i) = B(i-1) + dB(i-1);
end

% Explicit solution
X = X0 * exp((mu - 0.5 * sigma^2) * t + sigma * B);

% Plot
plot(t, X);
xlabel('Time'); ylabel('X_t');
title('Geometric Brownian Motion');
```

Listing 4: Geometric SDE (Black-Scholes) Simulation

Alternatively, `MATLAB` has a built-in function `gbm` to directly simulate a geometric Brownian motion.

```
% Parameters
mu = 0.1;           % rate of return
sigma = 0.2;        % volatility
X0 = 1;             % initial condition

T = 1;              % Time horizon
N = 1000;           % Number of steps
dt = T/N;           % Time step
nPaths = 1;         % Number of simulated paths

% Create GBM model
bm_model = gbm(0, 1, 'StartState', X0);

% Simulate paths
```

```
[paths, time] = simulate(bm_model, N, 'DeltaTime', dt, 'nTrials', nPaths);

% Extract BM (GBM returns levels; here they are just Brownian paths)
B = paths;

% Plot
plot(time, B);
xlabel('Time');
ylabel('B_t');
title('Geometric Brownian Motion');
```

Listing 5: Simulating a standard Brownian motion using the `gbm` object

4.4 Euler-Maruyama Method

Given a generic SDE (1) with no closed-form solution, we can simulate the solution using the ***Euler-Maruyama method***. Given the initial condition $X_0 = \xi$ and the grid points $\{0 = t_0, t_1, \ldots, t_N = T\}$ with $t_i = i\Delta t$ and $\Delta t = T/N$, we generate N independent standard normal variables $Z_1, \ldots, Z_N \sim \mathcal{N}(0,1)$, and define $\{X_{t_i}\}_{i=0}^N$ recursively by:

$$\begin{aligned} X_{t_0} &:= \xi, \\ X_{t_{i+1}} &:= X_{t_i} + b(t_i, X_{t_i})\Delta t + \sigma(t_i, X_{t_i})\sqrt{\Delta t} Z_{i+1}, \text{ for } i = 0, \ldots, N-1. \end{aligned}$$

The following code illustrates the simulation of the solution of the equation

$$dX_t = (\cos t + 0.1X_t)\, dt + 0.2(1 + X_t)\, dB_t,\ X_0 = 0.$$

```
% Euler-Maruyama method for SDE: dX = b(t,X) dt + sigma(t,X) dW
% Inputs:
b = @(t,x) cos(t) + 0.1*x   % function handle for drift, b(t,x)
sigma = @(t,x) 0.2*(1+x);   % function handle for diffusion, sigma(t,x)
X0 = 0;                     % initial condition
T = 1;                      % final time
N = 1000                    % number of time steps
M = 1;                      % number of sample paths
dt = T / N;                 % step size

t = linspace(0, T, N+1);
X = zeros(M, N+1);
X(:,1) = X0;

for i = 1:N
    dB = sqrt(dt) * randn(M, 1);  % Brownian increments
    X(:,i+1) = X(:,i) + b(t(i), X(:,i)) * dt + sigma(t(i), X(:,i)) .* dB;
end

plot(t, X(1,:))             % plot Path 1 of the SDE
```

```
xlabel('Time');
ylable('B_t');
title('Euler-Maruyama');
```

Alternatively, `MATLAB` offers a built-in function `sde` in the Financial Toolbox to simulate the solution of a SDE given the coefficients b, σ.

```
% Simulation of SDE using MATLAB's built-in sde function
% Requires Financial Toolbox

% Drift and diffusion functions
b = @(t,x) cos(t) + 0.1*x;
sigma = @(t,x) 0.2*(1 + x);

% SDE object definition
SDEobj = sde(b, sigma, 'StartState', 0);

% Simulation parameters
T = 1;           % Final time
N = 1000;        % Number of time steps
M = 1;           % Number of sample paths
dt = T / N;      % Step size

% Simulate paths
[Paths, Time] = simulate(SDEobj, N, 'DeltaTime', dt, 'nTrials', M);

% Plot the first path
plot(Time, Paths);
xlabel('Time');
ylabel('X_t');
title('SDE Simulation using sde function');
```

The Euler-Maruyama method is the default simulation method of the `sde` object. One can also choose alternative simulation methods by specifying in the `Method` argument in the function `simulate`; see https://www.mathworks.com/help/finance/sde.simulate.html for details.