Lecture 5

Numerical schemes for SDEs

Lecture Notes
by Jan Palczewski

A Stochastic Differential Equation (SDE) is an object of the following type

$$dX_t = a(t, X_t)dt + b(t, X_t)dW_t, \quad X_0 = x.$$

A solution is a stochastic process $X(t)$, satisfying

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

Today we will study algorithms that can be used to solve SDEs.

SDE's play a prominent role in a range of applications, including biology, chemistry, epidemiology, mechanics, microelectronics and, of course, finance.

An important SDE is the one for the stock price in the Black-Scholes model

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

This equation has an analytical solution and we can apply Monte Carlo techniques without further studies.

However, for most SDE's occurring in practical applications, this is not the case. Take

$$dX_t = a(b - X_t)dt + c\sqrt{X_t}dW_t.$$

This SDE plays an important role in stochastic volatility models and in interest rate theory. It does not, in general, have an explicit solution.

Brownian motion

For computational purposes it is useful to consider discretized Brownian motion. We thus divide a time interval $[0, T]$ into N subintervals by setting $\delta t = T/N$ and

$$t_n = n \cdot \delta t = n\frac{T}{N}, \qquad n = 0, \ldots, N.$$

Further, due to properties of Brownian motion we can simulate its values at the selected points by

$$W_{t_{n+1}} = W_{t_n} + \Delta W_n, \qquad W_{t_0} = W_0 = 0,$$

where ΔW_n are independent random variables with distribution $\mathcal{N}(0, \delta t)$.

The stochastic integral

$$\int_0^T \sigma_s dW_s$$

is the limit of the approximating sum

$$\sum_{n=0}^{N} \sigma_{t_n}(W_{t_{n+1}} - W_{t_n}),$$

as the mesh of the partition $P = (t_0, t_1, \ldots, t_N)$ of the interval $[0, T]$ approaches zero, i.e. when δt goes to zero.

Therefore, if δt is small enough, the above formula is a good approximation for the stochastic integral.

- What is the algorithm to simulate the stochastic integral?
- How to measure the error of the approximation?

Let us use the definition from the previous slide.

one simulation ———- one omega (ω)

Euler-Maruyama scheme

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

The simplest numerical method for solving of SDEs is the *stochastic Euler* scheme (also called *Euler-Maruyama* scheme).
For the above stochastic differential equation the scheme has the form

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

where

$$\Delta W_n = W_{t_{n+1}} - W_{t_n}.$$

In order to obtain this scheme the following approximation is used

$$\int_{t_n}^{t_{n+1}} b(s, X_s) dW_s \approx b(t_n, X_n) \Delta W_n,$$

$$\int_{t_n}^{t_{n+1}} a(s, X_s) ds \approx a(t_n, X_n) \delta t.$$

Testing the Euler scheme

Let us reconsider the following SDE

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

We know that the exact solution is given by

$$S_t = S_0 e^{\sigma W_t + (\mu - \frac{1}{2}\sigma^2)t}.$$

Inputs: T, N, mu, sigma, S0

delta_t = T/N

Let N[0..N-1] be an array of N independent draws from N(0,1)

W[0] = 0
For n = 0, 1, ..., N-1
    W[n+1] = W[n] + sqrt (delta_t) * N[n]

S[0] = S0
For n = 0, 1, ..., N-1
    S[n+1] = S[n] + S[n] * mu * delta_t
                  + S[n] * sigma * (W[n+1] - W[n])

Output: S[N]

We can now vary the step size in order to get some idea about the approximation error.

We will measure the performance in two ways. We make M independent runs of the algorithm and compute

- $$\frac{1}{M}\sum_{m=1}^{M}\left|S_0 e^{\sigma W^m[N]+(\mu-\frac{\sigma^2}{2})T} - S^m[N]\right| \approx \mathbb{E}|S_T - S[N]|$$
- $$\left|\mathbb{E}\left(S_0 e^{\sigma W_T+(\mu-\frac{\sigma^2}{2})T}\right) - \frac{1}{M}\sum_{m=1}^{M} S^m[N]\right| \approx |\mathbb{E}S_T - \mathbb{E}S[N]|$$

Convergence of schemes for SDEs

When writing algorithms we use the following notation:

- $t_n = n \cdot \delta t$,
- X_n is the computed value at t_n.

To study convergence of numerical schemes we have to introduce a new notation:

- $X_t^{\delta t}$ denotes a process that results from connecting of the points obtained by a numerical scheme with straight lines, i.e.

$$X_t^{\delta t} = X_n + \frac{t - t_n}{t_{n+1} - t_n}(X_{n+1} - X_n), \qquad \text{if } \; t \in [t_n, t_{n+1}).$$

  δt is the step size used in the algorithm.

- X_t is the exact solution.

Denote by $(X_t^{\delta t})$ the result of a simulation with the step δt.

The numerical scheme is **strongly convergent** if

$$\lim_{\delta t \to 0} \mathbb{E}\Big(|X_T - X_T^{\delta t}| \Big) = 0.$$

The numerical scheme is **weakly convergent** if

$$\lim_{\delta t \to 0} \Big| \mathbb{E} g(X_T) - \mathbb{E} g(X_T^{\delta t}) \Big| = 0$$

for every polynomial g.

The Euler-Maruyama scheme is strongly and weakly convergent. Unfortunately, it is not unconditional. For example the following conditions are sufficient:

1) functions a( ) and b( ) are four times continuously differentiable and their first derivatives are bounded, and
2) they do not grow too fast with parameters.

Since the Euler-Maruyama scheme proved to be an approximation of the solution to the SDE, we may ask how good this approximation is. This gives rise to the notion of convergence order.

Convergence orders

The numerical scheme is **strongly convergent with order** γ if

$$\mathbb{E}\Big(|X_T - X_T^{\delta t}|\Big) \leq K_T(\delta t)^\gamma.$$

The constant K_T depends on T and the considered SDE.

The numerical scheme is **weakly convergent with order** γ if

$$\Big|\mathbb{E}g(X_T) - \mathbb{E}g(X_T^{\delta t})\Big| \leq K_T^g(\delta t)^\gamma$$

for every polynomial g. The constant K_T^g depends on T, g, and the considered SDE.

Convergence order explained

If a numerical scheme is convergent with order γ, and we make the step k times smaller then the approximation error will decrease by a factor of k^γ.

Therefore, the order equal 1 means that if we want to decrease error 100 times, we have to make the step 100 times smaller.

The order equal to $1/2$ means that if we want to decrease error 100 times, we have to make the step $100^2 = 10\,000$ times smaller. And the computation time grows by the same factor.

Euler-Maruyama scheme is

- weakly convergent with order 1,
- strongly convergent with order 0.5.

Milstein scheme, which we shall study in a moment, is

- weakly convergent with order 1,
- strongly convergent with order 1.

Weak convergence:

- distribution at moment T only,
- useful for pricing of European path-independent options.

Strong convergence:

- pathwise property,
- should be used if whole paths play a role!
- good for path-dependent options.

Consequently, Euler-Maruyama scheme can be successfully applied to pricing of path-independent options (options with payoffs depending only the stock price at the moment of exercise) i.e. of the form $h(S_T)$.

This is caused by its poor strong convergence order. If you make the step 100 times smaller, the approximation improves only by a factor of 10. Remember that the step cannot be too small because of:

1. (computational errors) errors caused by computations in a computer with the use of finite precision numbers (with finite number of digits),
2. computation time.

Milstein scheme

We will limit ourselves to the following type of SDEs

$$dX_t = a(X_t)dt + b(X_t)dW_t, \qquad X_0 = x.$$

$$\begin{aligned} X_0 &= x, \\ X_{n+1} &= X_n + a(X_n)\delta t + b(X_n)\Delta W_n \\ &\quad + \frac{1}{2}b'(X_n)b(X_n)\big((\Delta W_n)^2 - \delta t\big) \end{aligned}$$

Here b' is the first derivative of b.

Derivation of Milstein scheme

Milstein scheme is obtained as a result of application of stochastic Taylor expansion, or more easily, by Ito formula.

Consider the homogeneous scalar stochastic differential equation

$$dX_t = a(X_t)dt + b(X_t)dW_t$$

and let t_i, t_{i+1} be two consecutive points in our time discretization.

The Ito formula says, that for a given function f which is two times continuously differentiable, we can write

$$f(X_s) = f(X_{t_i}) + \int_{t_i}^{s} \left(f'(X_u)a(X_u) + \frac{1}{2} f''(X_u)b(X_u)^2 \right) du + \int_{t_i}^{s} f'(X_u)b(X_u)dW$$

We can apply the Ito formula on the expressions $a(X_s)$ and $b(X_s)$, which are the coefficients in our SDE.

We then obtain

$$\begin{aligned} X_{t_{i+1}} = X_{t_i} &+ \int_{t_i}^{t_{i+1}} \left(a(X_{t_i}) + \int_{t_i}^{s} \left(a'(X_u)a(X_u) + \frac{1}{2} a''(X_u)b^2(X_u) \right) du \right. \\ &\left. + \int_{t_i}^{s} a'(X_u)b(X_u)dW_u \right) ds \\ &+ \int_{t_i}^{t_{i+1}} \left(b(X_{t_i}) + \int_{t_i}^{s} \left(b'(X_u)a(X_u) + \frac{1}{2} b''(X_u)b^2(X_u) \right) du \right. \\ &\left. + \int_{t_i}^{s} b'(X_u)b(X_u)dW_u \right) dW_s. \end{aligned}$$

We want to achieve a method which converges strongly of order 1.

By using a time discretization, the differentials dW and dt are replaced by the corresponding discrete versions ΔW and δt. We have

$$\begin{aligned}
\delta t \cdot \delta t &= \mathcal{O}((\delta t)^2), \\
\Delta W \cdot \delta t &= \mathcal{O}((\delta t)^{3/2}), \\
\Delta W \cdot \Delta W &= \mathcal{O}(\delta t).
\end{aligned}$$

Therefore: If we are up for a method which converges strongly of order 1, we can neglect the double integrals from the previous slide, which are of type $dW_s \cdot ds$ and $ds \cdot ds$.

We then obtain

$$X_{t_{i+1}} \approx X_{t_i} + \int_{t_i}^{t_{i+1}} a(X_{t_i})ds$$
$$+ \int_{t_i}^{t_{i+1}} \left(b(X_{t_i}) + \int_{t_i}^{s} b'(X_u)b(X_u)dW_u \right) dW_s$$
$$\approx X_{t_i} + a(X_{t_i})\delta t + b(X_{t_i})\Delta W_i + \int_{t_i}^{t_{i+1}} \int_{t_i}^{s} b'(X_u)b(X_u)dW_u dW_s$$

The first two summands in the equation above are well known from the Euler-Maruyama scheme. The third one is new.

We approximate the third term above by

$$\int_{t_i}^{t_{i+1}} \int_{t_i}^{s} b'(X_u)b(X_u)dW_u dW_s \approx b'(X_{t_i})b(X_{t_i}) \int_{t_i}^{t_{i+1}} \int_{t_i}^{s} dW_u dW_s.$$

The integral on the right hand side of the last equality is well known from Continuous Time Finance. We obtain

$$b'(X_{t_i})b(X_{t_i})\int_{t_i}^{t_{i+1}}\int_{t_i}^{s} dW_u dW_s = \frac{1}{2}b'(X_{t_i})b(X_{t_i})\big((\Delta W_i)^2 - \delta t\big)$$

Substituting this in our previous approximation we finally obtain the Milstein scheme

$$\begin{aligned} X_0 &= x, \\ X_{n+1} &= X_n + a(X_n)\delta t + b(X_n)\Delta W_n \\ &\quad + \frac{1}{2}b'(X_n)b(X_n)\big((\Delta W_n)^2 - \delta t\big) \end{aligned}$$

One can prove that the Milstein scheme converges strongly with order 1 to the solution of the SDE.

Multidimensional Stochastic Differential Equations

Sometimes an SDE is multidimensional. In this case $a(t,x)$ is a vector valued function, $b(t,x)$ is a matrix valued function and W is a multidimensional Brownian motion. The solution X is then a vector valued stochastic process.

A multidimensional Brownian motion is a vector process with the following properties:

a) every coordinate is a Brownian motion

b) coordinates are pairwise independent

How to simulate multi-dimensional Brownian motion?

$$W_t = (W_t^1, W_t^2, \ldots, W_t^d).$$

How to solve a multidimensional SDE?

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

$$X_t = (X_t^1, X_t^2, \ldots, X_t^d).$$

Stochastic Volatility Models

Stochastic volatility models are asset dynamics models in which volatility is itself a random object. The following system of two SDE's represents a basic version of Heston's stochastic volatility model, but there are many other SV models used in practice.

$$dS_t = S_t \mu dt + S_t \sqrt{V_t} dW_t^1,$$
$$dV_t = (\sigma_0 - V_t)dt + \sqrt{V_t} dW_t^2.$$

S_t represents the price of a stock, V_t the volatility is the solution of an SDE.

To compute the price of a European option with payoff $h(S_T)$ in the stochastic volatility model we combine the results of last two lectures: numerical solution of SDEs and Monte Carlo. We assume, of course, that the SDE we simulate is given with respect to a risk neutral measure, so that the price is computed as an expectation.

For m=1, 2, ..., M

S[m] = price at T simulated with the Euler-Maruyama algorithm

C[m] = exp(-rT) * h(S[m])

C = C[1] + . . . + C[M]

price = C/M

B = ((C[1] - price)^2 + . . . + (C[M] - price)^2) / (M-1)

Result: the price, the confidence interval