§ Numerical solutions to SDEs

Recall our SDE

$$dX(t) = a(X(t))\,dt + b(X(t))\,dW(t), \quad X(0) = X_0,$$

for some drift and dispersion functions a and b.

In this section, we aim to simulate a skeleton (trajectory) of the SDE at discrete times. More specifically, we would like to sample $X(t_1), X(t_2), \ldots X(t_T)$ for any times $0 < t_1 < t_2 < \cdots t_T$, where T stands for the number of times. For notation clarity, we may sometimes use shorthands

$$X_k := X(t_k),$$

$$X_i(t_k), \quad \text{the } i\text{-th element of vector } X(t_k) \in \mathbb{R}^d,$$

$$X_{i,k} := X_i(t_k).$$

We will also use $\Delta t_k := t_k - t_{k-1}$ and $\Delta W_k := W(t_k) - W(t_{k-1}) \sim N(0, \Delta t_k)$.

The simulation of SDEs is easy if we explicitly know their solutions. However, this is possible only for a few isolated cases, for example, the geometric Brownian motion (see, Exercise 1, Assignment 4) and linear SDEs. It is also possible to use the Girsanov-based method by Beskos, 2005 to make exact samples, but the method is restricted to very limited ~~cl~~ classes of SDEs. Hence, in this section, we formulate a few most commonly used numerical schemes to approximate the SDEs solutions.

The Euler-Maruyama is arguably the simplest approximate scheme. The idea is:

$$X(t_k) = X(t_{k-1}) + \int_{t_{k-1}}^{t_k} a(X(s))\,ds + \int_{t_{k-1}}^{t_k} b(X(s))\,dW(s)$$

$$\approx X(t_{k-1}) + a(X(t_{k-1}))(t_k - t_{k-1})$$

$$+ b(X(t_{k-1}))\boxed{\Delta W_k} \rightarrow \Delta W_k = W(t_k) - W(t_{k-1}) \sim N(0, \Delta t_k)$$

Essentially, the Euler-Maruyama method approximates the integrals above by rectangles. This works reasonably provided that the discretisation time $t_k - t_{k-1}$ is sufficiently small. Furthermore, if the dispersion coefficient is a constant b instead of a function of X, ~~viz.~~ then the Itô integral approximation is exact, viz.,

$$\int_{t_{k-1}}^{t_k} b \, dW(s) = b \, \Delta W_k .$$

abbv. Euler--Maruyama

This suggests that the EM method is better when dealing with constant dispersion than that of X-dependent ones.

Let's sumarise the algorithm.

Algorithm 9. Euler-Maruyama

Input: times $t_1, t_2, \ldots t_T$

Output: $\hat{X}(t_1), \hat{X}(t_2), \ldots \hat{X}(t_T)$ approximate samples

1. Draw $\hat{X}(0) \sim p(x_0)$
2. For $k = 1, 2, \ldots T$ do

   Draw $\Delta W_k \sim N(0, t_k - t_{k-1})$

   $\hat{X}(t_k) = \hat{X}(t_{k-1}) + a(\hat{X}(t_{k-1}))(t_k - t_{k-1}) + b(\hat{X}(t_{k-1}))\Delta W_k$.

Return $\hat{X}(t_1), \hat{X}(t_2), \ldots \hat{X}(t_T)$

Example 10. Modified Duffing-van der Pol simulation.

Consider a two-dimensional SDE:

$$d\begin{bmatrix} X_1(t) \\ X_2(t) \end{bmatrix} = \begin{bmatrix} X_2(t) \\ X_1(t)\left(\alpha - X_1(t)^2\right) - X_2(t) \end{bmatrix} dt + \begin{bmatrix} 0 \\ X_1(t) \end{bmatrix} dW(t),$$

set $\alpha = 2$, and the initial values $X_1(0) = -3$ and $X_2(0) = 0$. Simulate a path of X at times $t_1 = 0.01, t_2 = 0.02, \ldots, t_{1000} = 10$. Please see 'Lec3_van_der_pol.ipynb' for how this is done in Euler--Maruyama.

Albeit the simplicity of EM, the method ~~can be~~ is a quite crude estimator. More precisely, suppose that the drift and the dispersion functions are twice continuosly differentiable and their derivatives are bounded, we can show that the L^2-approximation error is such that

$$\mathbb{E}\left[\| X(t_k) - \underbrace{\hat{X}(t_k)}_{\text{EM approximated}} \|_2^2\right]^{\frac{1}{2}} \le \underbrace{C}_{\text{constant}} \left(t_k - t_{k-1}\right)^{\beta},$$

for $\boxed{\beta = \frac{1}{2}}$ (see, Lord etal, 2014, pp. 338 for details).

also see Kloeden and Platen, 1995, Ch. 10 for a complete proof.

This means that the approximation error goes to zero as ~~Δt~~ $\Delta t_k \to 0$ at a speed of $\beta = \frac{1}{2}$. It is then of interests to ask: Can we develop a better approximate scheme in the way that the convergence speed $\beta > \frac{1}{2}$ is faster?

To develop such a high-order simulator, we can leverage Itô's formula. The idea is similar to using Taylor expansion for ODEs.

(intentionally left blank)

Recall the SDE

$$X(t_k) = X(t_{k-1}) + \int_{t_{k-1}}^{t_k} a(X(s))\,ds + \int_{t_{k-1}}^{t_k} b(X(s))\,dW(s),$$

and we let $X \in \mathbb{R}^d$, $a: \mathbb{R}^d \to \mathbb{R}^d$, $b: \mathbb{R}^{d\times w} \to \mathbb{R}^d$, $W \in \mathbb{R}^w$. We use notation $X_i(t_k)$ to denote the i-th element of the vector $X(t_k)$. Let us apply Itô's formula on a_i and b_{ij}:

$$a_i(X(s)) = a_i(X(t_{k-1})) + \int_{t_{k-1}}^{s} (\nabla_x a_i(X(z)))^T a(X(z)) + \frac{1}{2} tr(\Gamma(X(z)) H_x a_i(X(z)))\,dz$$

any $s > t_{k-1}$ $\quad$ $\Gamma(x) := b(x)\,b(x)^T$

$$+ \int_{t_{k-1}}^{s} (\nabla_x a_i(X(z)))^T b(X(z))\,dW(z),$$

$$b_{ij}(X(s)) = b_{ij}(X(t_{k-1})) + \int_{t_{k-1}}^{s} (\nabla_x b_{ij}(X(z)))^T a(X(z)) + \frac{1}{2} tr(\Gamma(X(z)) H_x b_{ij}(X(z)))\,dz$$

$$+ \int_{t_{k-1}}^{s} (\nabla_x b_{ij}(X(z)))^T b(X(z))\,dW(z).$$

Now substitute $a_i(X(s))$ and $b_{ij}(X(s))$ back to the SDE, we get:

$$
\begin{aligned}
X_i(t_k) = {} & X_i(t_{k-1}) + a_i(X(t_{k-1}))(t_k - t_{k-1}) \\
& + \int_{t_{k-1}}^{t_k}\int_{t_{k-1}}^{s} \left((\nabla_x a_i(X(z)))^T a(X(z)) + \frac{1}{2}\mathrm{tr}\left(\Gamma(X(z)) H_x a_i(X(z))\right)\right) dz\, ds \qquad ① \\
& + \int_{t_{k-1}}^{t_k}\int_{t_{k-1}}^{s} (\nabla_x a_i(X(z)))^T b(X(z))\, dW(z)\, ds \qquad ②
\end{aligned}
$$

$$= \int_{t_{k-1}}^{t_k} a_i(X(s))\, ds$$

$$
\begin{aligned}
& + \sum_{j=1}^{w} \Big(b_{ij}(X(t_{k-1})) \underline{\Delta W_{j,k}} = \text{the } j\text{-th element of } \Delta W_k \\
& \qquad + \int_{t_{k-1}}^{t_k}\int_{t_{k-1}}^{s} \left((\nabla_x b_{ij}(X(z)))^T a(X(z)) + \frac{1}{2}\mathrm{tr}\left(\Gamma(X(z)) H_x b_{ij}(X(z))\right)\right) dz\, dW_j(s) \qquad ③ \\
& \qquad + \int_{t_{k-1}}^{t_k}\int_{t_{k-1}}^{s} (\nabla_x b_{ij}(X(z)))^T b(X(z))\, dW(z)\, dW_j(s) \qquad ④
\end{aligned}
$$

$$\int_{t_{k-1}}^{t_k} b_{ij}(X(s))\, dW_j(s)$$

If we discard the terms ①-④, the we recover the Euler--Maruyama scheme. To do better than Euler-Maruyama, we can keep some of the terms the appy Itô's formula again on them. As an example, applying Itô's formula on ④ gives:

$$
\begin{aligned}
(\nabla_x b_{ij}(X(z)))^T b(X(z)) = {} & (\nabla_x b_{ij}(X(t_{k-1})))^T b(X(t_{k-1})) \\
& + \int_{t_{k-1}}^{t_k} \cdots (\text{omit}) \cdots d\tau + \int_{t_{k-1}}^{t_k} \cdots (\text{omit}) \cdots dW(\tau) \qquad ⑤.
\end{aligned}
$$

which we can substitute back to ④ and discard ⑤ ~~giving~~ gives:

recover the celebrated Milstein's method:

$$X_i(t_k) \approx X_i(t_{k-1}) + a_i(X(t_{k-1}))(t_k - t_{k-1}) + \sum_{j=1}^{w} \left(b_{ij}(X(t_{k-1}))\Delta W_{j,k} + \left(\nabla_x b_{ij}(X(t_{k-1}))\right)^T b(X(t_{k-1})) \int_{t_{k-1}}^{t_k} \int_{t_{k-1}}^{s} dW(z)\, dW_j(s) \right)$$

The Mistein's method, as we can see from the approximation above, is an improvement of Euler--Maruyam with an additional term

$$\left(\nabla_x b_{ij}(X(t_{k-1}))\right)^T b(X(t_{k-1})) \int_{t_{k-1}}^{t_k} \int_{t_{k-1}}^{s} dW(z)\, dW_j(s),$$

which leverages the gradient information of the dispersion function. when b is a constant, the gradient is zero, ~~we get~~ it reduces to Euler--Maruyama.

However, the iterated Itô integral $\int_{t_{k-1}}^{t_k} \int_{t_{k-1}}^{s} dW(z)\, dW_j(s)$ is a trouble maker, and is nasty to simulate. Essentially, we have to simulate a bunch of $\int_{t_{k-1}}^{t_k} \int_{t_{k-1}}^{s} dW_i(z)\, dW_j(s)$ for all $i, j = 1, 2, \dots w$, which does not scale well in the dimension of the Brownian motion W.

FYI: this is highly related to Hermite polynomial, see Kloden and Platen, 1995.

Fortunately, if the dispersion function b admits special structures, for example, diagonal, or, unidimensional, then the iterated Itô integral is cheap to compute. To see this, suppose that $b(X(t_{k-1}))$ is alway diagonal, then ~~$b_{ij}(X(t$~~ $b_{ij}(\cdot)=0$ for all $i\neq j$, hence we only need to deal with $\int_{t_{k-1}}^{t_k}\int_{t_{k-1}}^{s} dW_i(z)\,dW_i(s)$ for $i=1,2,\dots w$. Morover, doing some algebras gives:

$$\int_{t_{k-1}}^{t_k}\int_{t_{k-1}}^{s} dW_i(z)\,dW_i(s)=\int_{t_{k-1}}^{t_k}\left(W_i(s)-W_i(t_{k-1})\right)dW_i(s)$$

$$=\frac{1}{2}\left(W_i(t_k)-W_i(t_{k-1})\right)^2-\frac{1}{2}(t_k-t_{k-1}),$$

use $\int_{t_{k-1}}^{t_k} W(s)\,dW(s)=\left(W(t_k)^2-W(t_{k-1})^2-(t_k-t_{k-1})\right)\times\frac{1}{2}$ ①

So the Milstein's method for ~~this s~~ the diagonal dispersion functions simplifies to:

$$X_i(t_k)\approx X_i(t_{k-1})+a_i(X(t_{k-1}))(t_k-t_{k-1})+b_{ij}(X(t_{k-1}))\Delta W_{i,k}$$
$$+\frac{1}{2}\frac{d b_{ij}(X(t_{k-1}))}{dX_i}b_{ij}(X(t_{k-1}))\left(\Delta W_{i,k}^2-(t_k-t_{k-1})\right)$$

We can rewrite it in a more compact vector format:

①: To prove it, use Itô's formula on $\phi(W(t))=W(t)^2$.

$$X(t_k) \simeq X(t_{k-1}) + a(X(t_{k-1}))(t_k - t_{k-1}) + b(X(t_{k-1}))\Delta W_k$$
$$+ \frac{1}{2}\bar{b}(X(t_{k-1}))\, b(X(t_{k-1}))\left(\Delta W_k^2 - (t_k - t_{k-1})\right),$$

where

$$\bar{b}(X(t_{k-1})) := \begin{bmatrix} \frac{d b_{11}(X(t_{k-1}))}{d X_1} & & & \\ & \frac{d b_{22}(X(t_{k-1}))}{d X_2} & & \\ & & \ddots & \\ & & & \frac{d b_{dd}(X(t_{k-1}))}{d X_d} \end{bmatrix}$$

Let's sumarise ~~the~~ Milstein's method.

Algorithm 11. Milstein's method for diagonal dispersion

Input: Times $t_1, t_2, \ldots t_T$

output: Approximate $\hat{X}(t_1), \hat{X}(t_2), \ldots \hat{X}(t_T)$

1. Draw $\hat{X}(0) \sim P(X_0)$
2. For $k = 1, 2, \ldots T$ do

   Draw $\Delta W_k \sim N(0, t_k - t_{k-1})$

   $$\hat{X}(t_k) = \hat{X}(t_{k-1}) + a(\hat{X}(t_{k-1}))(t_k - t_{k-1}) + b(\hat{X}(t_{k-1}))\Delta W_k$$
   $$+ \frac{1}{2}\bar{b}(\hat{X}(t_{k-1}))\, b(\hat{X}(t_{k-1}))\left(\Delta W_k^2 - (t_k - t_{k-1})\right)$$

Return $\hat{X}(t_1), \hat{X}(t_2), \ldots \hat{X}(t_T)$

Example 12.

$$dX(t) = -\frac{\alpha^2}{2} X(t)\,dt + \alpha\sqrt{1-X(t)^2}\,dW(t), \quad X_0 = 0$$

Simulate a trajectory at times $t_1 = 0.01$, $t_2 = 0.02$, ... $t_{500} = 5$ using Milstein's method. then compare to the true solution

$$X(t) = \sin(\alpha W(t)),$$ see, Kloeden and Platen. 1995. PP. 121.

See ~~'tes~~ `lec3_emvs_milstein.ipynb`.

The convergence of Milstein's method is such that

$$\mathbb{E}\left[\| X(t_k) - \hat{X}(t_k) \|_2^2 \right]^{\frac{1}{2}} \leq C\,(t_k - t_{k-1})^{\beta}$$

for $\beta = 1$ which is greater that Euler-Maruyam's $\frac{1}{2}$.
(under a similar assumption on the SDE coefficients)

It is possible to do better further than Milstein's method, however, as we can see from the derivation of Milstein's method, getting higher ord schemes would be super nasty. The main challenge is really the iterated Itô integrals. If the readers are interested, please see Kloeden and Platen, 1995 for how to derive high order discretisation schemes, and they also show how to simply the schemes if the SDE coefficients have special structures (e.g., diagonal or commutative)

$\beta = 1.5$

Let us bring up a method called order 1.5 strong Taylor scheme for constant dispersion. The derivation is seen in kloeden and platen, 1995.

Algorithm 13. order 1.5 strong Taylor scheme.

Input: Times $t_1, t_2, \ldots t_T$

output: Approximate $\hat{X}(t_1), \hat{X}(t_2), \ldots \hat{X}(t_T)$

1. Draw $\hat{X}(0) \sim p(X_0)$

2. For $k = 1, 2, \ldots T$ do

   Draw ~~two~~ two independent $u_1, u_2 \sim N(0, I_w)$

   $\Delta\alpha_k = \sqrt{t_k - t_{k-1}}\, u_1$

   $\Delta\beta_k = \frac{1}{2}(t_k - t_{k-1})^{\frac{3}{2}}\left(u_1 + \frac{u_2}{\sqrt{3}}\right)$

   For $i = 1, 2, \ldots d$ do

   $e_{i,k} = tr\left(H_x a_i(X(t_{k-1}))\, \cancel{\quad}\, \Gamma\right)$

   use broadcasting to vectorise this loop or einsum

   $e_k = [e_{1,k}\ e_{2,k}\ e_{3,k} \cdots e_{d,k}]^T$

   $$\hat{X}(t_k) = \hat{X}(t_{k-1}) + a(\hat{X}(t_{k-1}))(t_k - t_{k-1}) + b\,\Delta\alpha_k + \left(Ja(\hat{X}(t_{k-1}))\, a(\hat{X}(t_{k-1})) + \frac{1}{2}e_k\right)\frac{(t_k - t_{k-1})^2}{2} + \boxed{J_a}(\hat{X}(t_{k-1}))\, b\,\Delta\beta_k$$

   Jacobian of a.

output: $\hat{X}(t_1), \hat{X}(t_2), \ldots \hat{X}(t_T)$

↳ Gaussian-increment based simulations

We find that the aforementioned Euler–Maruyama and order 1.5 strong Taylor methods are Gaussian based. More precisely, we can generalise them ~~and~~ under the same framework:

$$X(t_k) \approx \mathbb{E}[X(t_k) \mid X(t_{k-1})] + \underbrace{\sqrt{\mathrm{Cov}[X(t_k) \mid X(t_{k-1})]}}_{\text{↳ Cholesky decomposition}}\, \xi_k,$$

$$\xi_k \sim N(0, I_d).$$

Euler–Maruyama gives

$$\mathbb{E}[X(t_k) \mid X(t_{k-1})] \approx X(t_{k-1}) + a(X(t_{k-1}))\Delta t_k,$$

$$\mathrm{Cov}[X(t_k) \mid X(t_{k-1})] \approx \Gamma(X(t_{k-1}))\Delta t_k.$$

Order 1.5 strong Taylor gives:

$$\mathbb{E}[X(t_k) \mid X(t_{k-1})] \approx X(t_{k-1}) + a(X(t_{k-1}))\Delta t_k + \left(J_a(X(t_{k-1}))\, a(X(t_{k-1})) + \frac{1}{2} e_k\right)\frac{\Delta t_k^2}{2},$$

$$\mathrm{Cov}[X(t_k) \mid X(t_{k-1})] \approx \Gamma \Delta t_k + J_a(X(t_{k-1}))\, \Gamma\, J_a(X(t_{k-1}))^T \frac{\Delta t_k^3}{3} + \left(\Gamma J_a(X(t_{k-1}))^T + J_a(X(t_{k-1}))^T \Gamma\right)\frac{\Delta t_k^2}{2}.$$

Algorithm 14. Gaussian-increment based sampling.

Input: Times $t_1, t_2, \dots t_T$

Output: Approximate $\hat{X}(t_1), \hat{X}(t_2), \dots \hat{X}(t_T)$.

1. Draw $X(0) \sim p(x_0)$
2. For $k = 1, 2, \dots T$ do

   Draw $\xi_k \sim N(0, I_d)$

   $$\hat{X}(t_k) = \hat{\mathbb{E}}[X(t_k) \mid \hat{X}(t_{k-1})] + \sqrt{\widehat{\mathrm{Cov}}[X(t_k) \mid \hat{X}(t_{k-1})]}\, \xi_k$$

Return $\hat{X}(t_1), \hat{X}(t_2), \dots \hat{X}(t_T)$

'lec3_em_vs_ito15_vs_time.ipynb' implements the Gaussian-based simulation.

The Milstein's method, however, cannot be put into this generalisation because it has ΔW_k^2 which is not Gaussian.

The upside of such a generalisation is that we can to some extent avoid the complicated iterations of Itô's formula to derive the high order schemes. More specifically, in the next section, we will show that it can be easier and automatic to approximate the statistical quantities, such as $\mathbb{E}[X(t_k) \mid X(t_{k-1})]$ and $\mathrm{Cov}[X(t_k) \mid X(t_{k-1})]$ compared to approximate the painful iterated Itô integrals.

Another upside is that we can easily compare the methods under the same framework ~~so we can let~~ by letting the methods share the same randomness (i.e., $\xi_1, \xi_2, \ldots \xi_T$).

Example ~~#~~.15

$$dX(t) = -\theta X(t)\,dt + dW(t), \quad \theta > 0.$$

This is a linear SDE called Ornstein-Uhlenbeck process, we can compute its conditional mean and variance in closed-form:

$$\mathbb{E}[X(t_k) \mid X(t_{k-1})] = e^{-\theta(t_k - t_{k-1})} X(t_{k-1})$$

$$\mathrm{Var}[X(t_k) \mid X(t_{k-1})] = \frac{1}{2\theta}\left(1 - e^{-2\theta(t_k - t_{k-1})}\right).$$

We can compare Euler--Maruyama and ~~the~~ the order 1.5 method, see 'lec3-em-vs-ito15-vs-tme.ipynb'. In the script, there is another method called TME which we shall detail in the next section.

The downside of the Gaussian-based simulation is that it can be a poor estimator when the transition is quite non-Gaussian. Even if we can approximate $\mathbb{E}[X(t_k) \mid X(t_{k-1})]$ and $\mathrm{Cov}[X(t_k) \mid X(t_{k-1})]$ exactly, the estimator can still be poor; for some models it ~~can~~ is even ~~be~~ worse than Milstein's method, especially when the dispersion term is non-constant.