Lecture 17. Numerical SDEs: Advanced topics

Tiejun Li[1,2]

[1]School of Mathematical Sciences (SMS),
&
[2]Center for Machine Learning Research (CMLR),
Peking University,
Beijing 100871,
P.R. China
tieli@pku.edu.cn

Office: No. 1 Science Building, Room 1376E

Table of Contents

Implicit scheme

To overcome the stiffness issue, one can also apply implicit schemes, e.g. simplest implicit Euler:

$$X_{n+1} = X_n + b(X_{n+1})\delta t_n + \sigma(X_n)\delta W_n$$

or semi-implicit scheme

$$X_{n+1} = X_n + \Big[\alpha b(X_n) + (1-\alpha) b(X_{n+1})\Big]\delta t_n + \sigma(X_n)\delta W_n$$

for $\alpha \in (0,1)$.

Implicit scheme

- The fully implicit scheme is also considered but not very successful although one can transform the Ito SDE form into right-most endpoint form at first.

Implicit scheme

- The fully implicit scheme is also considered but not very successful although one can transform the Ito SDE form into right-most endpoint form at first.
- For example

$$X_{n+1} = X_n + \Big[b(X_{n+1}) - c(X_{n+1})\Big]\delta t_n + \sigma(X_{n+1})\delta W_n$$

where

$$c_i(x) = \sum_{jk} \frac{\partial \sigma_{ij}}{\partial x_k} \sigma_{kj}$$

is from the transformation.

Implicit scheme

- The fully implicit scheme is also considered but not very successful although one can transform the Ito SDE form into right-most endpoint form at first.
- For example

$$X_{n+1} = X_n + \Big[b(X_{n+1}) - c(X_{n+1})\Big]\delta t_n + \sigma(X_{n+1})\delta W_n$$

  where

$$c_i(x) = \sum_{jk} \frac{\partial \sigma_{ij}}{\partial x_k}\sigma_{kj}$$

  is from the transformation.
- If $b = 0, \sigma(x) = x$, the above scheme implies

$$X_{n+1} = \frac{X_n}{1 - \delta W_n}$$

Implicit scheme

- The fully implicit scheme is also considered but not very successful although one can transform the Ito SDE form into right-most endpoint form at first.
- For example

$$X_{n+1} = X_n + \Big[b(X_{n+1}) - c(X_{n+1})\Big]\delta t_n + \sigma(X_{n+1})\delta W_n$$

where

$$c_i(x) = \sum_{jk} \frac{\partial \sigma_{ij}}{\partial x_k}\sigma_{kj}$$

is from the transformation.

- If $b = 0, \sigma(x) = x$, the above scheme implies

$$X_{n+1} = \frac{X_n}{1 - \delta W_n}$$

- It is possible that $1 - \delta W_n = 0$ and indeed $\mathbb{E}|X_{n+1}| = \infty$!

Extrapolation method

Talay and Tubaro proposed the following extrapolation method based on the error expansion:

$$e(\delta) = \mathbb{E}g(X_T^{\delta}) - \mathbb{E}g(X_T) = C_{g,\beta}\delta^{\beta} + C_{g,\beta+1}\delta^{\beta+1}$$

$$e\left(\frac{\delta}{2}\right) = \mathbb{E}g(X_T^{\frac{\delta}{2}}) - \mathbb{E}g(X_T) = C_{g,\beta}(\frac{\delta}{2})^{\beta} + C_{g,\beta+1}(\frac{\delta}{2})^{\beta+1}$$

$$2^{-\beta}e(\delta) - e(\frac{\delta}{2}) = \mathbb{E}g(X_T^{\delta}) - \mathbb{E}g(X_T) = \tilde{C}_{g,\beta+1}\delta^{\beta+1}$$

See details in Stoch. Anal. Appl. 8 (1990), 483-509.

Table of Contents

Multilevel Monte Carlo method

- So far we only considered the bias error of the approximation, i.e. the error brought by the time discretization. But a real approximation also involves Monte Carlo samplings.

Multilevel Monte Carlo method

- So far we only considered the bias error of the approximation, i.e. the error brought by the time discretization. But a real approximation also involves Monte Carlo samplings.
- Since 2008, M. Giles proposed the general framework of multilevel Monte Carlo methods for SDEs, which approximates the expectation in an efficient way. This method stimulates a lot of follow-up works in different fields.

Error in the full discretization

- We have already known that the Euler-Maruyama scheme is of weak order 1 in computing $Y_E = \mathbb{E}f(X_T)$ for the SDE

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t$$

on $[0, T]$.

Error in the full discretization

- We have already known that the Euler-Maruyama scheme is of weak order 1 in computing $Y_E = \mathbb{E} f(X_T)$ for the SDE

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t$$

  on $[0, T]$.

- In real computations, we take the weak approximator

$$Y_{h,N} = \frac{1}{N}\sum_{k=1}^{N} f(X_n^{(k)}), \qquad n = T/h \in \mathbb{N}$$

  with stepsize h and N independent samples, where X_n is obtained by the Euler-Maruyama scheme.

Error in the full discretization

- We have already known that the Euler-Maruyama scheme is of weak order 1 in computing $Y_E = \mathbb{E}f(X_T)$ for the SDE

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t$$

on $[0, T]$.

- In real computations, we take the weak approximator

$$Y_{h,N} = \frac{1}{N}\sum_{k=1}^{N} f(X_n^{(k)}), \qquad n = T/h \in \mathbb{N}$$

with stepsize h and N independent samples, where X_n is obtained by the Euler-Maruyama scheme.

- The mean square error has the bias-variance decomposition

$$\begin{aligned}
\text{MSE} &= \mathbb{E}(Y_E - Y_{h,N})^2 \\
&\le 2|Y_E - \mathbb{E}f(X_n)|^2 + 2\mathbb{E}|\mathbb{E}f(X_n) - Y_{h,N}|^2 \\
&\le C_1 h^2 + C_2 N^{-1}.
\end{aligned}$$

by the weak order 1 convergence and Monte Carlo estimate.

Overall computational cost

- The above computation has the cost $C_3 N h^{-1}$.

Overall computational cost

- The above computation has the cost $C_3 N h^{-1}$.
- The cost-accuracy tradeoff

$$\min_{h,N} \mathsf{MSE} \qquad \text{subject to a given cost } K = C_3 N h^{-1} \gg 1$$

gives the optimal choice

$$N \sim O(Kh), \quad h \sim O(K^{-\frac{1}{3}}) \quad \text{and} \quad \mathsf{MSE} \sim O(K^{-\frac{2}{3}}).$$

Overall computational cost

- The above computation has the cost $C_3 N h^{-1}$.
- The cost-accuracy tradeoff

$$\min_{h,N} \text{MSE} \qquad \text{subject to a given cost } K = C_3 N h^{-1} \gg 1$$

  gives the optimal choice

$$N \sim O(Kh), \quad h \sim O(K^{-\frac{1}{3}}) \quad \text{and} \quad \text{MSE} \sim O(K^{-\frac{2}{3}}).$$

- This means that if we require the accuracy MSE $\sim O(\varepsilon^2)$, we must have $h \sim O(\varepsilon)$, $N \sim O(\varepsilon^{-2})$ and thus the cost $K \sim O(\varepsilon^{-3})$.

Overall computational cost

- The above computation has the cost $C_3 N h^{-1}$.
- The cost-accuracy tradeoff

$$\min_{h,N} \text{MSE} \qquad \text{subject to a given cost } K = C_3 N h^{-1} \gg 1$$

  gives the optimal choice

$$N \sim O(Kh), \quad h \sim O(K^{-\frac{1}{3}}) \quad \text{and} \quad \text{MSE} \sim O(K^{-\frac{2}{3}}).$$

- This means that if we require the accuracy MSE $\sim O(\varepsilon^2)$, we must have $h \sim O(\varepsilon)$, $N \sim O(\varepsilon^{-2})$ and thus the cost $K \sim O(\varepsilon^{-3})$.
- The multilevel Monte Carlo method achieves the same accuracy with cost $K \sim O(\varepsilon^{-2}(\ln \varepsilon)^2)$, which is a typical fast algorithm.

Construction of multilevel Monte Carlo method

- Define the L-level grids with time stepsize $h_l = M^{-l}T$ for $l = 0, 1, \ldots, L$.

Construction of multilevel Monte Carlo method

- Define the L-level grids with time stepsize $h_l = M^{-l}T$ for $l = 0, 1, \ldots, L$.
- Denote by $F_l = f(X_{l,M^l})$ the approximation of $f(X_T)$ at the level l, where X_{l,M^l} is the approximation of X_T with stepsize h_l.

Construction of multilevel Monte Carlo method

- Define the L-level grids with time stepsize $h_l = M^{-l}T$ for $l = 0, 1, \ldots, L$.
- Denote by $F_l = f(X_{l,M^l})$ the approximation of $f(X_T)$ at the level l, where X_{l,M^l} is the approximation of X_T with stepsize h_l.
- We have

$$\mathbb{E}F_L = \sum_{l=0}^{L} \mathbb{E}(F_l - F_{l-1}) \qquad \text{where } F_{-1} := 0.$$

Construction of multilevel Monte Carlo method

- Take N_l realizations for each summand in the equation above, and define

$$Y_l = \frac{1}{N_l} \sum_{k=1}^{N_l} \left(F_l^{(k)} - F_{l-1}^{(k)}\right), \qquad l = 0, 1, \ldots, L.$$

Construction of multilevel Monte Carlo method

- Take N_l realizations for each summand in the equation above, and define

$$Y_l = \frac{1}{N_l} \sum_{k=1}^{N_l} \left(F_l^{(k)} - F_{l-1}^{(k)} \right), \qquad l = 0, 1, \ldots, L.$$

- Correspondingly define the final estimator

$$\hat{Y}_L = \sum_{l=0}^{L} Y_l.$$

Construction of multilevel Monte Carlo method

- From Monte Carlo estimate we have $\mathrm{var}(Y_l) = V_l/N_l$, where $V_l := \mathrm{var}(F_l - F_{l-1})$ for $l = 0, 1, \ldots, L$.

Construction of multilevel Monte Carlo method

- From Monte Carlo estimate we have $\mathrm{var}(Y_l) = V_l/N_l$, where $V_l := \mathrm{var}(F_l - F_{l-1})$ for $l = 0, 1, \ldots, L$.
- With independent sampling for $\hat{Y}_L$, we get

$$\mathrm{var}(\hat{Y}_L) = \sum_{l=0}^{L} \mathrm{var}(Y_l) = \sum_{l=0}^{L} \frac{V_l}{N_l}$$

with computational cost

$$K \sim O\left(\sum_{l=0}^{L} N_l h_l^{-1}\right).$$

Cost-accuracy tradeoff in multilevel Monte Carlo method

- The key point of multilevel Monte Carlo is that with the decomposition

$$\mathbb{E}F_L = \sum_{l=0}^{L} \mathbb{E}(F_l - F_{l-1}) \qquad \text{where } F_{-1} := 0,$$

the term $F_l - F_{l-1}$ has smaller fluctuations, i.e. smaller variance, at higher levels provided that the realizations of $F_l - F_{l-1}$ come from two discrete approximations with different time stepsizes but same Brownian paths.

Cost-accuracy tradeoff in multilevel Monte Carlo method

- The key point of multilevel Monte Carlo is that with the decomposition

$$\mathbb{E}F_L = \sum_{l=0}^{L} \mathbb{E}(F_l - F_{l-1}) \qquad \text{where } F_{-1} := 0,$$

the term $F_l - F_{l-1}$ has smaller fluctuations, i.e. smaller variance, at higher levels provided that the realizations of $F_l - F_{l-1}$ come from two discrete approximations with different time stepsizes but same Brownian paths.

- This property suggests that we can use less Monte Carlo simulations for higher levels, i.e. finer grids, but more simulations for lower levels, i.e. coarser grids.

Cost-accuracy tradeoff in multilevel Monte Carlo method

- The key point of multilevel Monte Carlo is that with the decomposition

$$\mathbb{E}F_L = \sum_{l=0}^{L} \mathbb{E}(F_l - F_{l-1}) \qquad \text{where } F_{-1} := 0,$$

  the term $F_l - F_{l-1}$ has smaller fluctuations, i.e. smaller variance, at higher levels provided that the realizations of $F_l - F_{l-1}$ come from two discrete approximations with different time stepsizes but same Brownian paths.
- This property suggests that we can use less Monte Carlo simulations for higher levels, i.e. finer grids, but more simulations for lower levels, i.e. coarser grids.
- This cost-accuracy tradeoff is the origin of the efficiency of multilevel Monte Carlo method.

Optimal choice

- Now let us consider the minimization

$$\min_{N_l} \mathrm{var}(\hat{Y}_L) = \sum_{l=0}^{L} \frac{V_l}{N_l} \qquad \text{subject to the cost } K = \sum_{l=0}^{L} N_l h_l^{-1} \gg 1.$$

Optimal choice

- Now let us consider the minimization

$$\min_{N_l} \operatorname{var}(\hat{Y}_L) = \sum_{l=0}^{L} \frac{V_l}{N_l} \qquad \text{subject to the cost } K = \sum_{l=0}^{L} N_l h_l^{-1} \gg 1.$$

- This is generally a very difficult problem so we relax N_l to be continuous. Upon introducing Lagrange multiplier we get the minimizer

$$N_l = \lambda \sqrt{V_l h_l}, \qquad \text{where} \quad \lambda = K \left(\sum_{l=0}^{L} \sqrt{V_l h_l^{-1}} \right)^{-1}.$$

Computational complexity analysis

- From the strong and weak convergence result of Euler-Maruyama Scheme, we have

$$|\mathbb{E}(F_l) - Y_E| = O(h_l), \quad \mathbb{E}|X_T - X_{l,M^l}|^2 = O(h_l).$$

Computational complexity analysis

- From the strong and weak convergence result of Euler-Maruyama Scheme, we have

$$|\mathbb{E}(F_l) - Y_E| = O(h_l), \quad \mathbb{E}|X_T - X_{l,M^l}|^2 = O(h_l).$$

- By assuming the Lipschitz continuity of f, we obtain

$$\mathrm{var}(F_l - f(X_T)) \leq \mathbb{E}|f(X_{l,M^l}) - f(X_T)|^2 \leq C\mathbb{E}|X_T - X_{l,M^l}|^2 = O(h_l)$$

Computational complexity analysis

- From the strong and weak convergence result of Euler-Maruyama Scheme, we have

$$|\mathbb{E}(F_l) - Y_E| = O(h_l), \quad \mathbb{E}|X_T - X_{l,M^l}|^2 = O(h_l).$$

- By assuming the Lipschitz continuity of f, we obtain

$$\mathrm{var}(F_l - f(X_T)) \leq \mathbb{E}|f(X_{l,M^l}) - f(X_T)|^2 \leq C\mathbb{E}|X_T - X_{l,M^l}|^2 = O(h_l)$$

- Thus

$$V_l = \mathrm{var}(F_l - F_{l-1}) \leq 2\mathrm{var}(F_l - f(X_T)) + 2\mathrm{var}(F_{l-1} - f(X_T)) = O(h_l)$$

since $h_{l-1} = Mh_l$ and $M \sim O(1)$.

Computational complexity analysis

- For a given tolerance $\varepsilon \ll 1$, take

$$N_l = O(\varepsilon^{-2} L h_l).$$

Computational complexity analysis

- For a given tolerance $\varepsilon \ll 1$, take

$$N_l = O(\varepsilon^{-2} L h_l).$$

- According to this choice of N_l, we get the variance estimate

$$\mathrm{var}(\hat{Y}_L) = O(\varepsilon^2)$$

  from

$$\mathrm{var}(\hat{Y}_L) = \sum_{l=0}^{L} \mathrm{var}(Y_l) = \sum_{l=0}^{L} \frac{V_l}{N_l}.$$

Computational complexity analysis

- For a given tolerance $\varepsilon \ll 1$, take

$$N_l = O(\varepsilon^{-2} L h_l).$$

- According to this choice of N_l, we get the variance estimate

$$\mathrm{var}(\hat{Y}_L) = O(\varepsilon^2)$$

from

$$\mathrm{var}(\hat{Y}_L) = \sum_{l=0}^{L} \mathrm{var}(Y_l) = \sum_{l=0}^{L} \frac{V_l}{N_l}.$$

- Further take $L = \ln \varepsilon^{-1} / \ln M$, we have

$$h_L = M^{-L} = O(\varepsilon).$$

Computational complexity analysis

- For a given tolerance $\varepsilon \ll 1$, take

$$N_l = O(\varepsilon^{-2} L h_l).$$

- According to this choice of N_l, we get the variance estimate

$$\mathrm{var}(\hat{Y}_L) = O(\varepsilon^2)$$

from

$$\mathrm{var}(\hat{Y}_L) = \sum_{l=0}^{L} \mathrm{var}(Y_l) = \sum_{l=0}^{L} \frac{V_l}{N_l}.$$

- Further take $L = \ln \varepsilon^{-1} / \ln M$, we have

$$h_L = M^{-L} = O(\varepsilon).$$

- So the bias error

$$|\mathbb{E} F_L - Y_E| = O(h_L) = O(\varepsilon).$$

Computational complexity analysis

- Combing $\mathrm{var}(\hat{Y}_L) = O(\varepsilon^2)$ and $|\mathbb{E}F_L - Y_E| = O(h_L) = O(\varepsilon)$, we obtain the overall mean square error

$$\mathsf{MSE} = \mathbb{E}(Y_E - \hat{Y}_L)^2 = O(\varepsilon^2)$$

and the computational complexity

$$K = \sum_{l=0}^{L} N_l h_l^{-1} = O(\varepsilon^{-2} L^2) = O\left(\varepsilon^{-2} (\ln \varepsilon)^2\right).$$

Computational complexity analysis

- Combing $\mathrm{var}(\hat{Y}_L) = O(\varepsilon^2)$ and $|\mathbb{E}F_L - Y_E| = O(h_L) = O(\varepsilon)$, we obtain the overall mean square error

$$\mathsf{MSE} = \mathbb{E}(Y_E - \hat{Y}_L)^2 = O(\varepsilon^2)$$

  and the computational complexity

$$K = \sum_{l=0}^{L} N_l h_l^{-1} = O(\varepsilon^{-2} L^2) = O\left(\varepsilon^{-2} (\ln \varepsilon)^2)\right).$$

- The optimal choice of M can be made by minimizing the prefactor in the estimate of the computational cost.