Unbiased Simulation of Stochastic Differential Equations

Monte Carlo Project

Nathaniel Cogneaux - Paul-Emile Galine

PSL Research University, Paris-Dauphine

19/01/2024

Introduction

We aim to estimate $V_0 = \mathbb{E}[g(X_{t_1}, \ldots, X_{t_n})]$, where X is the solution to the SDE $dX_t = \mu(t, X_t)dt + \sigma(t, X_t)dW_t$, with μ, σ as the drift and diffusion coefficients (with standard assumptions).

- Traditional Monte Carlo methods approximate V_0 by simulating multiple paths of X (e.g., using the Euler-Scheme) and approximating the expectation over many simulations.
- These methods suffer from bias and is computationally expensive.
- The approach we studied:
    - Provides an unbiased and thus more accurate estimation of V_0.
    - Presents computational time advantage over standard methods in some cases.

Plan

Agenda :

I. Markovian Case
   a. Quick review of the algorithm and main theorem
   b. Proof in simplest case
   c. Comparison with Euler-Scheme

II. Path-dependent case
   a. Quick review of the algorithm
   b. Illustration for case =2
   c. Comparison with Euler-Scheme

III. General case
   a. Lamperti's transformation
   b. General case and its limits

IV. Conclusion

Review of the Algo in the Markovian Case (Part 1)

Goal: Approximate $V_0 = E[g(X_T)]$ (n=1 for $t_0 = 0, .., t_n = T$)

Assumption: Constant non-degenerate σ.

Time Grid Definition:

Discrete time grid on [0, T].

Random variables τ_i follow $\mathcal{E}(\beta)$ exponential distribution.

$T_k := \left(\sum_{i=1}^{k} \tau_i\right) \wedge T$.

$N_T := \max\{k : T_k < T\}$.

Euler Scheme:

Define $\hat{X}$ as the Euler scheme solution on the grid $(T_k)_{k \geq 0}$.

$\hat{X}_{T_{k+1}} = \hat{X}_{T_k} + \mu(T_k, \hat{X}_{T_k})\Delta T_{k+1} + \sigma \Delta W_{T_{k+1}}$.

Review of the Algo in the Markovian Case (Part 2)

Estimator ψ:

$$\hat{\psi} = e^{\beta T} \left[g(\hat{X}_T) - g(\hat{X}_{T_{N_T}}) \mathbf{1}_{\{N_T > 0\}} \right] \beta^{-N_T} \prod_{k=1}^{N_T} \mathcal{W}_k^1.$$

Weight Function $\mathcal{W}_k^1$:

$$\mathcal{W}_k^1 = \frac{(\mu(T_k, \hat{X}_{T_k}) - \mu(T_{k-1}, \hat{X}_{T_{k-1}})) \cdot (\sigma^T)^{-1} \Delta W_{T_{k+1}}}{\Delta T_{k+1}}.$$

Theorem (Unbiasedness and Finite Variance)

Let V_0 be defined as above, with the function g assumed to be globally Lipschitz. Then it is unbiased: $V_0 = \mathbb{E}[\hat{\psi}]$, with finite variance, i.e., $\mathbb{E}[\hat{\psi}^2] < \infty$.

Approximation of V_0 with MC methods:

$V_0 \approx \hat{V}_0^N := \frac{1}{N} \sum_{n=1}^{N} \hat{\psi}_n$, where $(\hat{\psi}_n)_{n=1}^{N} \overset{\text{i.i.d.}}{\sim} \hat{\psi}$, MC estimator with finite variance thanks to law of large numbers.

Sketch of proof on a toy example

Context:

- One-dimensional SDE: $X_t = x_0 + \int_0^t \mu(s, X_s)ds + W_t$ and $\hat{X}_t = x_0 + bt + W_t$.
- Estimator ψ: $\psi = e^{\beta T} g(\hat{X}_T) \prod_{k=1}^{N_T} \frac{(\mu(T_k, \hat{X}_{T_k}) - b)\Delta W_{T_{k+1}}}{\beta \Delta T_{k+1}}$.
- Goal: Show $E[\psi] = E[g(X_T)]$ (it also has finite variance, is integrable but not the point here)

Introduce a sequence ψ_n:

- Defined as: $\psi_n = e^{\beta T_{n+1}} \left(\prod_{k=1}^{\min(N_T, n)} \frac{(\mu(T_k, \hat{X}_{T_k}) - b)\Delta W_{T_{k+1}}}{\beta \Delta T_{k+1}} \right) \times \left(g(\hat{X}_T) \mathbf{1}_{\{N_T \leq n\}} + \left(\frac{\mu - b}{\beta} \partial_x u \right) (T_{n+1}, \hat{X}_{T_{n+1}}) \mathbf{1}_{\{N_T > n\}} \right)$
- One has: $E[\psi_n] = E[g(X_T)]$ (We will see why!)
- Uniform integrability for $(\psi_n)_{n>0}$ using technical lemmas.
- Apply dominated convergence to get the result.

Sktech of proof on a toy example : Unbiased

Demonstrating $E[\psi_n] = E[g(X_T)]$ for all n:

1. Focus on the case $n = 0$ as a starting point for the general proof.
2. The value function $u(0, x_0)$ is defined as $E[g(X_T)]$.
3. Given μ and g smooth u solves Fokker-Planck PDE with terminal condition :

$$\begin{cases} \partial_t u(t,x) + \frac{1}{2}\partial^2_{xx} u(t,x) + \mu(t,x)\partial_x u(t,x) = 0, \\ \text{for all } (t,x) \in [0,T) \times \mathbb{R}, \\ u(T,x) = g(x). \end{cases}$$

4. Rewrite the PDE and apply Feyman-Kac to get a representation result for u(0,x_0)

Sketch of Proof on a Toy Example: Unbiased

Precisely, rewriting the Fokker-Planck Equation:

$-\partial_t u(t,x) - b\partial_x u(t,x) - \frac{1}{2}\partial^2_{xx} u(t,x) = (\mu(t,x) - b)\partial_x u(t,x)$

Feynman-Kac Representation:

$u(0,x_0) = E\left[g(\hat{X}_T) + \int_0^T (\mu(t,\hat{X}_t) - b)\partial_x u(t,\hat{X}_t)dt\right]$

Equality with ψ_0:

Knowing $T_1 = \min(T,\tau_1)$ with $\tau_1 \sim \mathcal{E}(\beta)$ and $\tau_1 \perp\!\!\!\perp W$ (thus $\hat{X}$), and using the transfer lemma, we get:

$u(0,x_0) =$

$E\left[e^{\beta T_1} g(\hat{X}_T)\mathbf{1}_{\{T_1 \geq T\}} + \frac{e^{\beta T_1}}{\beta}(\mu(T_1,\hat{X}_{T_1}) - b)\partial_x u(T_1,\hat{X}_{T_1})\mathbf{1}_{\{T_1 < T\}}\right]$

which is exactly ψ_0

End of Proof

Differentiation of u

- For bounded and continuous function ϕ and $t > 0$, we derive: $\partial_x E[\phi(\hat{X}_T)] = E\left[\frac{\phi(\hat{X_T})W_T}{T}\right]$.
- Based on this same idea and replacing with notations one has: $\partial_x u(0, x_0) =$ $E\left[\frac{e^{\beta\Delta T_1}\Delta W_{T_1}}{\Delta T_1}\left(g(\hat{X}_T)\mathbf{1}_{\{T\leq T_1\}} + \frac{\mu(T_1,\hat{X}_{T_1})-b}{\beta}\partial_x u(T_1, \hat{X}_{T_1})\mathbf{1}_{\{T_1<T\}}\right)\right]$.
- Changing initial conditions from $(0, x_0)$ to $(T_1, \hat{X}_{T_1})$, we find for $T_1 < T$: $\partial_x u(T_1, \hat{X}_{T_1}) =$ $E\left[\frac{e^{\beta\Delta T_2}\Delta W_{T_2}}{\Delta T_2}\left(g(\hat{X}_T)\mathbf{1}_{\{T\leq T_2\}} + \frac{\mu(T_2,\hat{X}_{T_2})-b}{\beta}\partial_x u(T_2, \hat{X}_{T_2})\mathbf{1}_{\{T_2<T\}}\right)\right]$. Plugged inside the representation formula we have $E[\psi_1] = u(0, x_0)$.
- This change of initial condition can be done for $(T_2, \hat{X}_{T_2})$ with $u(0, x_0) = E[\psi_2]$ until we reach $T_{N_T+1} = T$ with each time $E[g(X_T)] = u(0, x_0) = E[\psi_n]$ for all n using the representation formula.

Numerical Results and Comparaison with Euler-Scheme

Method	Mean value	Statistical error	95% Confidence Interval	Computation time
US (N = 10^4)	0.20860405	0.004339791	[0.20009806, 0.21711004]	0.135535s
Euler Scheme (nSteps = 10^4)	0.20268141	0.004164094	[0.19451979, 0.21084304]	0.475454s
US (N = 10^5)	0.20422262	0.001397635	[0.20148326, 0.20696199]	1.461565s
Euler Scheme (nSteps = 10^5)	0.20521232	0.001316617	[0.20263176, 0.20779289]	4.760887s
US (N = 10^6)	0.20561571	0.000449698	[0.2047343, 0.20649712]	14.002025s
Euler Scheme (nSteps = 10^6)	0.20517708	0.000416448	[0.20436084, 0.20599332]	48.404329s
US (N = 10^7)	0.20561192	0.000142441	[0.20533273, 0.2058911]	2min 20.386787s
Euler Scheme (nSteps = 10^7)	0.20492765	0.000131292	[0.20467031, 0.20518498]	8min 4.982595s
US (N = 10^8)	0.20565379	4.5036e-05	[0.20556552, 0.20574206]	23min 30.145957s
Euler Scheme (nSteps = 10^8)	0.20478412	4.1499e-05	[0.20470278, 0.20486545]	1h 21min 39.45896s

Figure: Here, $m = 10$ for Euler-Scheme and $\beta = 0.1$ for the US method

Remark: The value of β was carefully selected to be neither too large nor too small, in order to minimize computation time and statistical error, as mentioned in our study

Comparison: Euler Scheme vs US Method

The observed results align well with our theoretical predictions...

Aspect	**Euler Scheme**	**Unbiased Method**
Statistical Error	$O(1/\sqrt{N})$	$O(1/\sqrt{N})$
Bias Error	$O(1/m)$	0
Computation Time	$O(Nm)$	$O(\beta TN)$

Table: Theoretical Order of Errors and Computation Time

Note: m represents the number of time steps in the Euler Scheme, and N the number of Monte Carlo simulations.

- Further explanations will be provided on the blackboard.
- Adjusting m to reduce the bias ?

Path-Dependent Case - Part 1

- **Goal:** Compute path-dependent $V_0 = E[g(X_{t_1}, \ldots, X_{t_n})]$, n¿1.
- **Assumption:** Constant non-degenerate σ and g Lipschitz.

Algorithm Overview:

- Recursive iteration of the Markovian Case over time subintervals $[t_k, t_{k+1}]$.
- Same notations as before for W, $(\tau_i)_{i>0}$, $N = (N_s)_{0\leq s\leq t}$, $(T_i)_{i>0}$

Subintervals $k = 1, \ldots, n$:

- Number of jump arrivals on $[t_{k-1}, t_k)$: $\tilde{N}_k := N_{t_k} - N_{t_{k-1}}$
- $\tilde{T}_0^k := t_{k-1}$ and $\tilde{T}_j^k := t_k \wedge T_{N_{t_{k-1}}+j}$.
- $\Delta\tilde{T}_j^k := \tilde{T}_j^k - \tilde{T}_{j-1}^k$.
- $\Delta\tilde{W}_j^k := W_{\tilde{T}_j^k} - W_{\tilde{T}_{j-1}^k}$ for $j = 1, \ldots, \tilde{N}_k + 1$.

Path-Dependent Case - Part 1

Illustration (n=2):

Let's do a quick illustration on the blackboard of the notations in case $n = 2$, where you have $\tilde{N}_1 = 2$ and $\tilde{N}_2 = 1$.

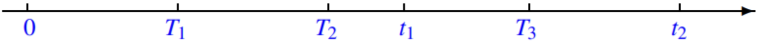

Figure: Case $n = 2$ with $\tilde{N}_1 = 2$ jumps on $[0, t_1)$, and $\tilde{N}_2 = 1$ jumps on $[t_1, t_2)$.

Path-Dependent Case - Part 2 (Summary)

- Process $(\tilde{X}_j^{k,x})$:

  Represents subintervals $k = 1, \dots, n$.
  X evolves with automatic differentiation weights $\tilde{W}_j^{k,x}$.

- Recursive Algorithm:

  Initialize with terminal base case: $\psi_x^{n+1} := g(x_1, \dots, x_n)$.
  For $k = 1, \dots, n$:
  Update $X^{k,x}$ and $X_0^{k,x}$.
  Define $\tilde{\psi}_k^x$ recursively:

$$\tilde{\psi}_k^x := e^{\beta(t_k - t_{k-1})}(\psi_{k+1}^{X^{k,x}} - \tilde{\psi}_{k+1}^{X_0^{k,x}} \mathbf{1}_{\{\tilde{N}_k > 0\}})\beta^{-\tilde{N}_k} \prod_{j=1}^{\tilde{N}_k} \tilde{W}_j^{k,x} \tag{1}$$

Path-Dependent Case - Part 2 (Summary)

- Obtain the final result: $\tilde{\psi} := \tilde{\psi}_1^{x_0}$.
- Under added assumptions on g, we get the same theorem as before : No bias and finite variance for $\tilde{\psi}$

In other words, back to our previous example:

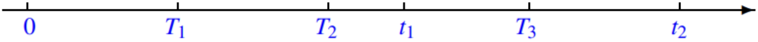

Figure: Case $n = 2$ with $\tilde{N}_1 = 2$ jumps on $[0, t_1)$, and $\tilde{N}_2 = 1$ jumps on $[t_1, t_2)$.

Path-Dependent Case - Part 2 (Summary)

- The two different processes on $[t_1, t_2]$ induce two different variables:

$$\tilde{\psi}_2^{\tilde{X}_3^1} := e^{\beta(t_2-t_1)}(g(\tilde{X}_2^1, \tilde{X}_2^{2,\tilde{X}_3^1}) - g(\tilde{X}_2^1, \tilde{X}_1^{2,\tilde{X}_3^1}))\beta^{-1}\widetilde{\mathcal{W}}_1^{2,\tilde{X}_3^1}$$

and

$$\tilde{\psi}_2^{\tilde{X}_2^1} := e^{\beta(t_2-t_1)}(g(\tilde{X}_2^1, \tilde{X}_2^{2,\tilde{X}_2^1}) - g(\tilde{X}_2^1, \tilde{X}_1^{2,\tilde{X}_2^1}))\beta^{-1}\widetilde{\mathcal{W}}_1^{2,\tilde{X}_2^1}.$$

- With $\tilde{\psi}_2^{\tilde{X}_2^1}$, $\tilde{\psi}_2^{\tilde{X}_3^1}$ and the variables on $[0, t_1]$, we obtain the variable

$$\tilde{\psi} := \tilde{\psi}_1^{x_0} = e^{\beta t_1}(\tilde{\psi}_2^{\tilde{X}_3^1} - \tilde{\psi}_2^{\tilde{X}_2^1})\beta^{-2}\widetilde{\mathcal{W}}_1^{1,x_0}\widetilde{\mathcal{W}}_2^{1,x_0}.$$

Numerical Results

We consider the same SDE as before for X and aim to compute the Asian call option payoff: $V_0 := \mathbb{E}[\left(e^{X_T} - K\right)^+]$, with parameters $K = 1$, $T = 1$, $n = 10$, and $t_k := \frac{k}{n}T$ for $k = 1, 2, \ldots, n$.

Comparison with Euler Scheme

- Of course, still more accurate
- Careful with computation time and not simple to improve compared to Euler-Scheme

Generalization to General SDEs

- Extension to SDEs with non-constant diffusion coefficients.
- Focus on computing $V_0 = \mathbb{E}[g(X_T)]$.
- Consider the SDE:

$$X_0 = x_0, \quad dX_t = \mu(t, X_t)\,dt + \sigma(t, X_t)\,dW_t.$$

- Lamperti's Transformation for simplifying SDEs:

  For $d = 1$, define $h(t, x) = \int_0^x \frac{1}{\sigma(t,y)}\,dy$.
  Transforms the SDE to a form with constant diffusion for $Y_t = h(t, X_t)$.
  For $d > 1$, a similar transformation is possible if σ is a positive definite matrix.

- The transformed SDE for Y_t can be expressed using Itô's formula.

Unbiased Simulation Algorithm and Challenges

Unbiased Simulation Algorithm:

- Adaptation for general SDEs
- Theorem: Finite expectation of $|\hat{\psi}|$, ensuring $V_0 = \mathbb{E}[\hat{\psi}]$ but infinite variance
- Driftless case: finite variance using antithetic trick

Challenges and Future Research:

- Addressing infinite variance issues
- Expanding to higher dimensions and path-dependent SDEs.

What we could have done:

- Compare with PDEs approximation
- Study the multidimentional case