

Black–Scholes Monte Carlo Pricer

Option Pricer Project

December 22, 2025

1 Goal

Estimate the price of options (vanilla, digital, Asian) under Black–Scholes via Monte Carlo, provide a confidence interval, and avoid storing all paths to keep memory low.

2 Model

Underlying follows a geometric Brownian motion

$$dS_t = rS_t dt + \sigma S_t dW_t, \quad S_{t_{k+1}} = S_{t_k} \exp((r - \frac{1}{2}\sigma^2)\Delta t + \sigma\sqrt{\Delta t} Z_k), \quad Z_k \sim \mathcal{N}(0, 1).$$

The time grid $\{t_k\}$ comes directly from the option (full path for an Asian).

3 Simulation algorithm

- Precompute per-step terms $(r - \frac{1}{2}\sigma^2)\Delta t_k$ and $\sigma\sqrt{\Delta t_k}$; reject non-increasing time steps.
- Antithetic variates: for each Gaussian draw Z , simulate two paths S^+ and S^- using Z and $-Z$ (variance reduction at no extra cost).
- Path construction: start both paths at S_0 , iterate over steps applying the exponential update and store S^+, S^- in reusable vectors.
- Payoff: evaluate `option->payoffPath(path)` on each completed path, then discount by $\exp(-rT)$.
- Online stats: update the running mean and $M2$ (Welford) with each discounted payoff; no need to retain past payoffs.

4 Mean and variance estimator

Online Welford updates:

$$\hat{P}_n = \hat{P}_{n-1} + \frac{X_n - \hat{P}_{n-1}}{n}, \quad M2_n = M2_{n-1} + (X_n - \hat{P}_{n-1})(X_n - \hat{P}_n).$$

Estimated variance: $\hat{\sigma}^2 = M2_n/(n - 1)$, standard error $\hat{\sigma}/\sqrt{n}$.

5 Confidence interval

$$\text{CI}_{95\%} = \hat{P}_n \pm 1.96 \times \frac{\hat{\sigma}}{\sqrt{n}},$$

available via `confidenceInterval()` after at least two paths.

6 Key implementation points (generate)

- Inputs: number of paths N ; if $N \leq 0$ the call is ignored.
- Uses two reusable vectors (`path_pos`, `path_neg`) sized to the time grid.
- For each Gaussian Z : update S^+ and S^- across all steps, then:
 - compute discounted payoff of S^+ , increment `nb_paths_`, update mean and $M2$;
 - if more paths remain, do the same with S^- (antithetic) and increment counters.
- Discount factor is $\exp(-rT)$ with T the last time step; `nb_paths_`, `estimate_`, and `M2_` track the current simulation state.
- `price()` throws if no paths were generated; `confidenceInterval()` throws if fewer than two paths are available.