Pricing by Monte-Carlo Simulations

Introduction

Patrick Hénaff

February 20, 2023

Today's Objective

At the end of this module, you should understand:

1. How to simulate a lognormal process
2. Why it is so common to use simulation to price derivatives
3. The principle of Monte-Carlo pricing
4. The limits of MC pricing, and what can be done to improve its accuracy.

Monte-Carlo Simulations

Definition

Monte-Carlo Simulation: generation of time series of risk factors that are consistent with the observed behavior of these risk factors, or with a model.
The time series are obtained by repeated random sampling of the risk factors, according to a specified probability distribution.

Name due to von Neumann, Ulam and Metropolis (1946) while working on US defense projects.

Use of Monte-Carlo Simulations

Simulations are used for:

- Risk measurement:
  - Scenario analysis, VaR
  - Stress testing (Basel II, Solvency II)
- Derivatives pricing
  - Easy to implement (just code the payoff formula)
  - Flexible: it dissociates the dynamics of the risk factors from the evaluation of the option.

Historical vs. Risk-Neutral Probabilities

- Historical or objective probabilities:
  - The "real-world" behavior of the asset
  - Consistent with observed historical time series
  - Each asset has its own expected return
- Risk-Neutral probabilities;
  - The distribution consistent with option prices
  - The expected return of all assets is the risk-free rate

Historical vs. Risk-Neutral Probabilities

- A stock has an annualized expected return of $\mu = 5\%$ and an annualized s.d. of return $\sigma = 30\%$.
- The risk-free rate is $r = 2\%$.

Simulation of S_T: stock price at horizon T:

- Consistent with historical probabilities:

$$\begin{aligned} E\left[\frac{S_T}{S_0}\right] &= e^{\mu T} \\ V\left[\ln\left(\frac{S_T}{S_0}\right)\right] &= \sigma^2 T \end{aligned}$$

- Consistent with risk-neutral probabilities

$$\begin{aligned} E\left[\frac{S_T}{S_0}\right] &= e^{r T} \\ V\left[\ln\left(\frac{S_T}{S_0}\right)\right] &= \sigma^2 T \end{aligned}$$

Monte-Carlo Simulations: Historical vs. Risk-Neutral

- Simulations for scenario analysis and VaR are based on actual probabilities (historical VaR)
- Simulations for pricing are based on risk-neutral probabilities.
- Economic scenarii generation for Solvency II calculations use hybrid methods.
  - Market consistent
  - Use historical data when implied volatility from option market is not available

Distribution of Stock Returns

Notation:

- S_t Price of asset at time t: a Lognormal random variable
- r_t Return from $t-1$ to t: a Normal random variable

Distribution of Stock Returns: Illustration

What about $(\mu - \frac{\sigma^2}{2})$

Consider the following sequence of annual realized returns in 5 consecutive years:

Year(t)	r_t (%)	$S_t = S_{t-1}e^{r_t}$	$\frac{S_t}{S_{t-1}}$
0		100	
1	15	116.18	1.16
2	20	141.90	1.22
3	30	191.55	1.34
4	-20	156.83	0.81
5	25	201.37	1.28
Mean	14		1.167

- Expected return $E(r_t) = 14\%$
- Expected growth rate $\log(E(S_t/S_{t-1})) = \log(1.167) = 15.4\%$

Distribution of Stock Returns

$$\ln\left(\frac{S_t}{S_0}\right) \approx N((\mu - \frac{\sigma^2}{2})t, \sigma\sqrt{t})$$

How to Choose μ?

- In a risk-neutral world, for option pricing, μ is the risk-free rate:
$$E(S_t) = S_0 e^{rt}$$
- For scenario analysis, historical VaR, μ is the historical or expected rate of return.

$$E(S_t) = S_0 e^{\mu t}$$

Simulating a Log-Normal Process: Summary

Given a stock with expected continuous compounded return μ:

$$\ln(S_t) - \ln(S_0) \approx \mathcal{N}\left((\mu - \frac{1}{2}\sigma^2)t, \sigma^2 t\right)$$

or

$$S_t = S_0 e^{x_t}$$

with $x_t \approx \mathcal{N}\left((\mu - \frac{1}{2}\sigma^2)t, \sigma^2 t\right)$.

$$x_t = (\mu - \frac{1}{2}\sigma^2)t + \sigma\sqrt{t}\epsilon$$

with $\epsilon \approx \mathcal{N}(0, 1)$.

How to simulate a log-normal process

To simulate a time series of stock prices following a log-normal process, observed at intervals Δt:

1. Start with S_0 at $t = 0$
2. for $t = \Delta t, 2\Delta t, \dots, T$:
    2.1 Simulate $\epsilon \approx \mathcal{N}(0, 1) : \epsilon_t$
    2.2 Compute

$$S_t = S_{t-\Delta t} e^{\left(r - \frac{\sigma^2}{2}\right)\Delta t + \sigma\sqrt{\Delta t}\epsilon_t}$$

How to Generate Normal Random Numbers ϵ

Using a uniform [0, 1] random generator:

1. Draw $z \approx \text{uniform}(0, 1)$
2. transform z into a normal variable $\epsilon \approx \mathcal{N}(0, 1)$

$$\epsilon = N^{-1}(z)$$

where $N()$ is the cumulative normal distribution.

Proof:

$$\begin{aligned} Pr\{\epsilon \leq x\} &= Pr\{N^{-1}(z) \leq x\} \\ &= Pr\{N(N^{-1}(z)) \leq N(x)\} \\ &= Pr\{z \leq N(x)\} \\ &= N(x) \end{aligned}$$

How to Generate Normal Random Numbers ϵ: Illustration

Lognormal Simulation Example

Year	Z_t $U(0,1)$	ϵ_t $N(0,1)$	X_t $N((r-\frac{\sigma^2}{2})\Delta t, \sigma\sqrt{\Delta t})$	e^{X_t}	S_t
0					100.00
1	0.4	-0.253	-0.021	0.979	97.92
2	0.6	0.253	0.131	1.139	111.62
3	0.7	0.524	0.212	1.236	138.03

Table: Risk-neutral log normal simulation,
$S_0 = 100, \Delta t = 1, r = 10\%, \sigma = 30\%$

Simulation of a Lognormal Process: Example

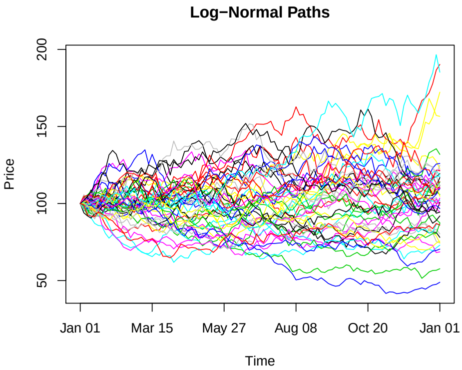
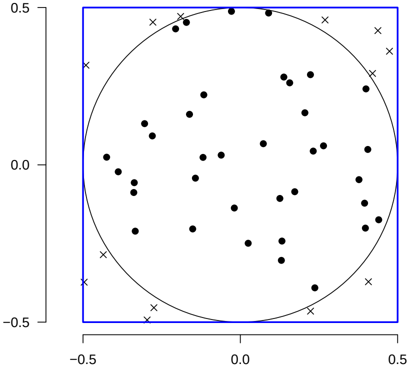


Figure: Risk-neutral log normal process,
$S_0 = 100$, $T = 1$, $r = 10\%$, $\sigma = 30\%$

Monte Carlo Simulation: Introductory Example

Estimating π by randomly throwing darts at the square.



Area of square: 1, area of circle: π

Computing circle area by simulation

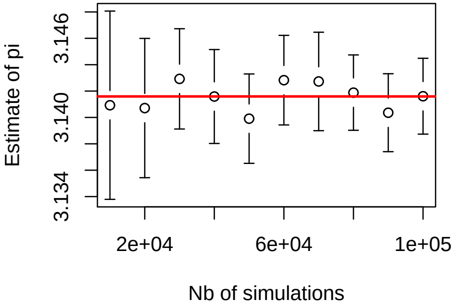
1. Simulate N throws of dart
2. Count the number of darts that land in the circle, let M be that number
3. Estimate for π is:

$$\pi = 4\frac{M}{N}$$

Estimate of π vs number of simulations

	mean	sd
10000	3.1424	0.0153
20000	3.1419	0.0122
30000	3.1389	0.0093
40000	3.1419	0.0080
50000	3.1427	0.0078
60000	3.1435	0.0068
70000	3.1415	0.0071
80000	3.1425	0.0053
90000	3.1413	0.0062
1e+05	3.1411	0.0049

Estimate of π vs. number of simulations



Estimation error

Accuracy, measured by standard deviation, improves at the rate $\sqrt{N}$.

To reduce error by a factor of 10, one needs the increase the number of simulations by a factor of 100.

This slow convergence rate has motivated research on how to improve the accuracy of the simulation by other means than increasing the number of simulations.

Basic MC pricing

Pricing an European Call option by simulation:

1. Simulate N normal random variables $\epsilon_i, i = 1, \ldots N$
2. Compute $S_T^i, i = 1, \ldots, N$
3. Evaluate the payoff:

$$V_i = \max(S_T^i - K, 0)$$

4. Compute price

$$P = e^{-rT} \frac{1}{N} \sum_{i=i}^{N} V_i$$

MC estimate of price vs. number of simulations

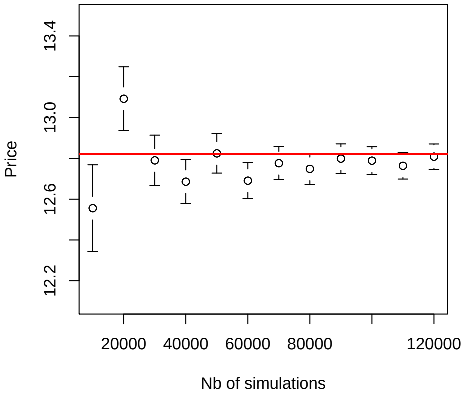
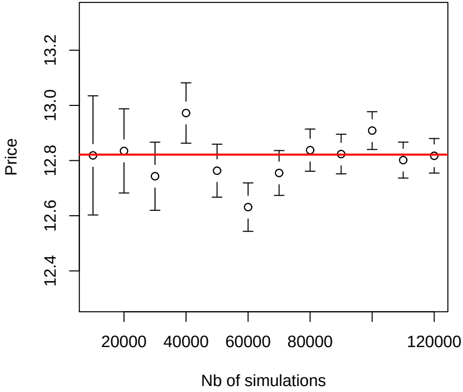


Figure: MCpricing of a call option, $S_0 = K = 100, \sigma = 30\%$

Methods for improving the convergence of Monte-Carlo simulations

- Antithetic variables: draw random numbers symetrically around the mean to ensure that the sample mean is accurate.
- Use random number generators (Sobol) that generate well-distributed samples, regardless of sample size.
- Use Control Variate methods

MC with antithetic variables. Price vs. number of simulations



Comparison of Sobol sequence and default uniform random variates

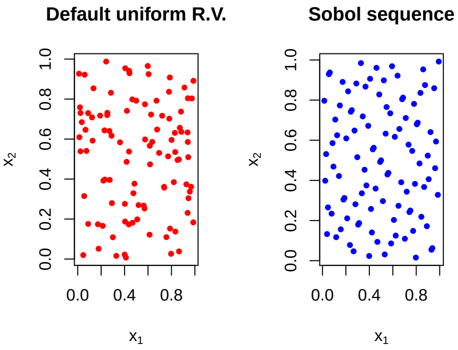
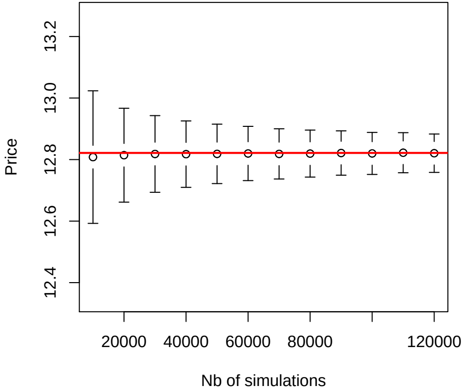


Figure: The points generated by the Sobol sequence are more evenly distributed in the rectangle

MC with Sobol sequences. Price vs. number of simulations



Summary

- Variance of MC estimator is a function of $\sqrt{\frac{1}{N}}$.
- The quality of the random number generator has a major impact on the accuracy of the MC estimation.

MC with Control Variate (Version 1)

A and B are very similar derivatives, function of the same risk factors X. We know the exact value of B: f_B.

1. Compute the values of $f_B^* = E(f_B(X))$ and $f_A^* = E(f_A(X))$ by MC simulation.
2. The quantity $e = f_B - f_B^*$ is a measure of the MC simulation bias
3. Use this measure to correct the MC price of A:

$$f_A = f_A^* + (f_B - f_B^*)$$

MC with Control Variate (Version 2)

introduce a factor α:

$$f_A = f_A^* + \alpha(f_B - f_B^*)$$

Choose α to minimize the variance of f_A.

$$V(f_A) = V(f_A^*) + \alpha^2 V(f_B^*) + 2\alpha Cov(f_A^*, f_B^*)$$

Or,

$$\alpha = \frac{Cov(f_A^*, f_B^*)}{V(f_B^*)}$$

α is the regression coefficient of f_A^* on f_B^*.