Financial Systems Design Part II

Monte Carlo Simulation

August 2020

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Outline

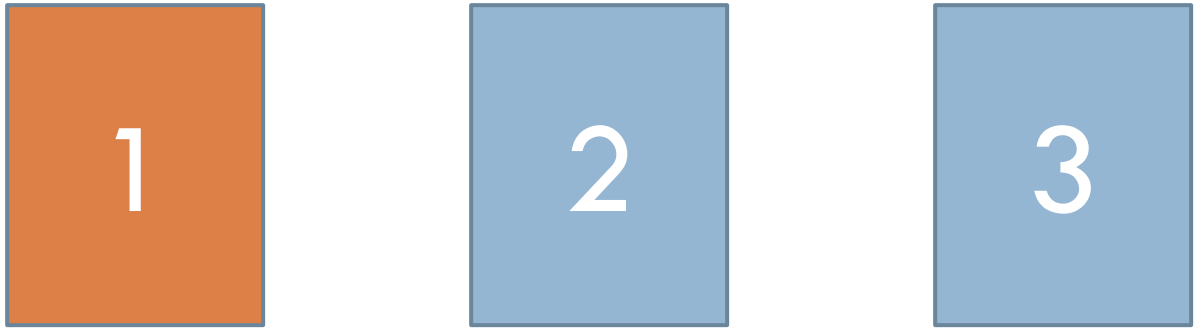
- Monty Hall Problem
- Monte Carlo Approach
- Discretization Scheme
- Variance Reduction Techniques

Simulation

- Simulation can often help us solve problems that have resisted analysis
- They also help to understand solutions that can be counterintuitive
- Take the Monty Hall problem as an example

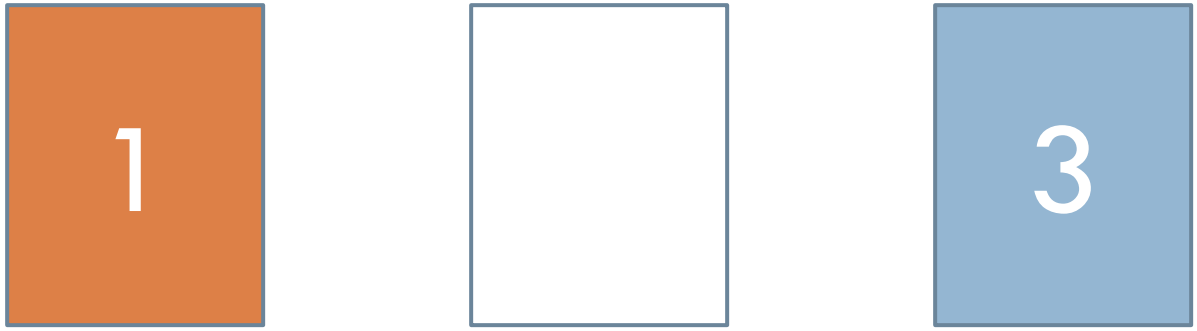
Monty Hall Problem

- You are in a game show with a chance to win a prize
- There are 3 doors and the prize is behind one of them with nothing between the other 2
- The host, who knows which door holds the prize, asks you to choose a door



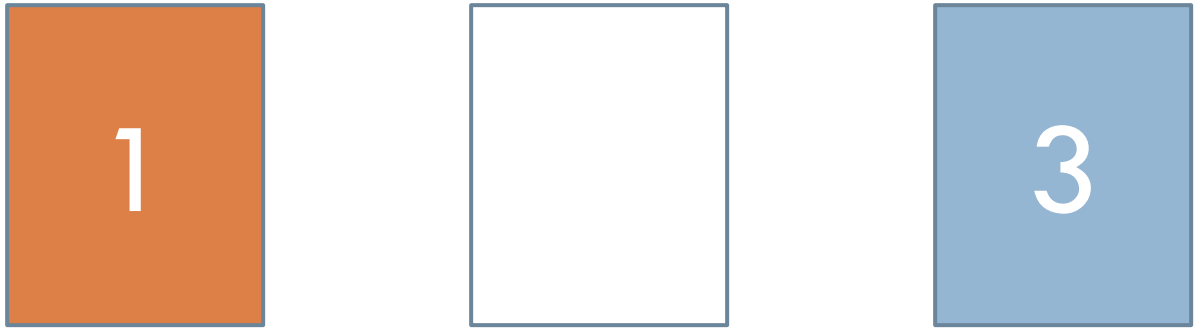
Monty Hall Problem

- Once you have chosen, the host will always open 1 of the 2 unchosen doors showing it to be empty
- Finally, he will ask if you wish to stay with the door you have chosen or switch to the remaining unopened door



Monty Hall Problem

- Should you switch?
- Would it benefit you to switch or does it make no difference?



Monty Hall Problem

- There is some history behind this problem
- Marilyn Vos Savant was a child prodigy who had a column in a magazine that answers readers' questions
- This question was posed to her in 1990
- Her answer was that you should switch as it doubles your chances of getting the prize
- She received enormous responses, many from PhDs in math and statistics, that her answer was wrong

Monty Hall Problem

- But was she wrong?
- The solution Marilyn presented was simple

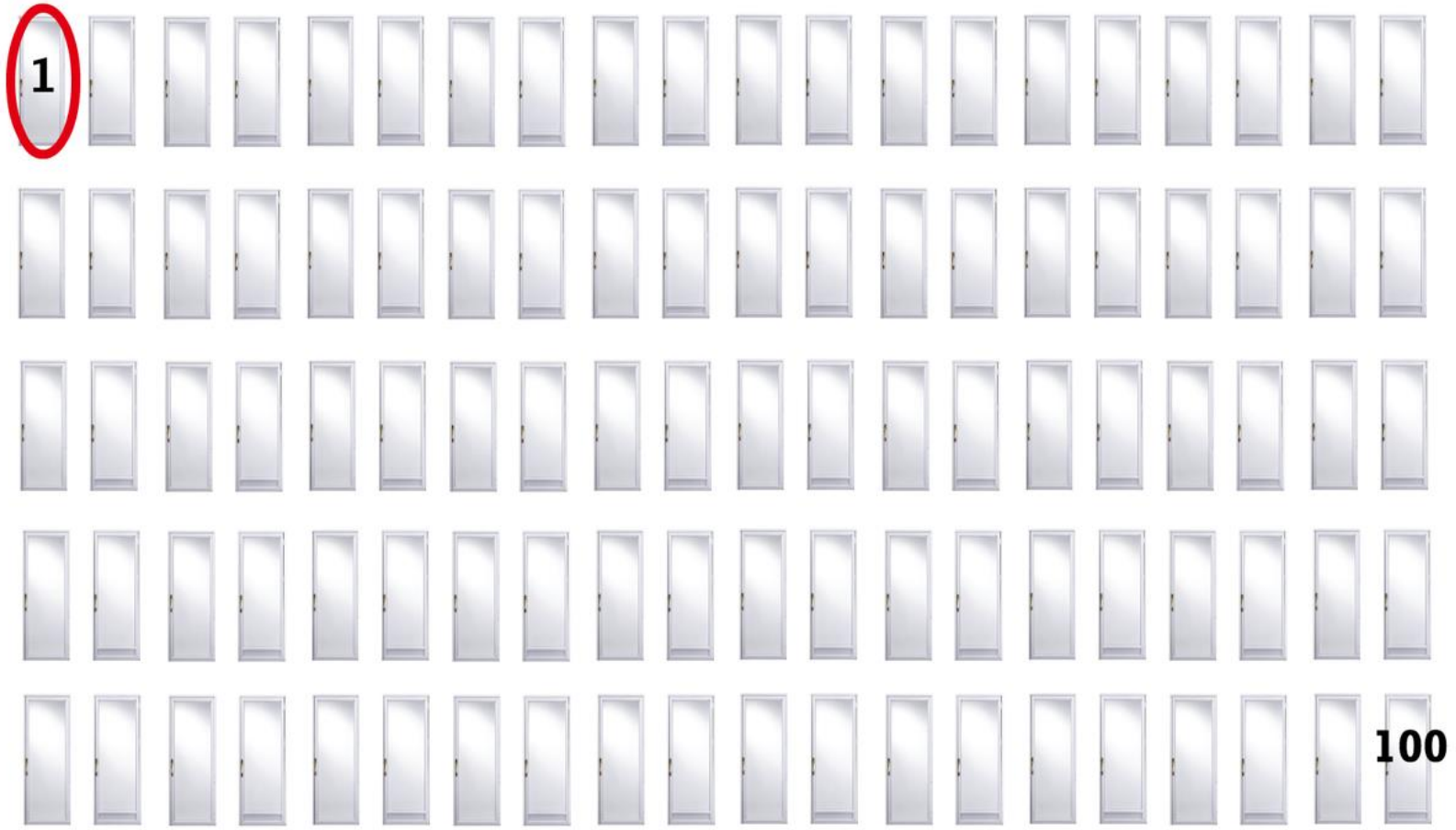
Behind door 1	Behind door 2	Behind door 3	Result if staying at door #1	Result if switching to the door offered
Empty	Empty	**Prize**	Wins nothing	**Wins prize**
Empty	**Prize**	Empty	Wins nothing	**Wins prize**
Prize	Empty	Empty	**Wins prize**	Wins nothing

- However, many still did not agree
- Some were finally turned when a simulation proved it

In Class Exercise

- Let's try to simulate this problem to determine the chances of winning if you stay and if you switch
- The key part of the simulation is determining which is the door that is offered for switching
  - Scenario 1: The door chosen holds the prize then the host can choose any of the 2 remaining doors for switching
  - Scenario 2: The door chosen does not hold the prize then only the door with the prize can be offered as a switch
- Let's switch to Jupyter notebook

Monty Hall Problem



Monty Hall Problem

Monty Hall Problem

- Simulations are a powerful way of solving problems that resists analytical solutions or are counterintuitive
- The Monty Hall problem is one such example
- We will use simulations with the Monte Carlo method as another way of constructing VMs

Monte Carlo Simulation

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Monte Carlo Simulation

- Monte Carlo method seeks to compute the Expectation (in probability) via simulations
- The Expectation is computed using the Risk Neutral or Martingale probability measure (recall arrow securities)
- The price of the derivative is:

$$V(x_t, t) = e^{-r(T-t)} E^Q[f(x, T)|x_t]$$

- This assumes interest rate is constant
- We try to compute $E^Q[f(x, T)|x_t]$ using Monte Carlo

Monte Carlo Simulation

- Monte Carlo (MC) simulation
  1. Use random number generators to simulate asset price paths using Risk Neutral probabilities
  2. Compute final value of derivative based on the payoff of the derivative
  3. With a suitable number of simulations, compute the discounted expected value through averaging

Monte Carlo Simulation

- Imagine a digital option which pays $1 if the underlying is above the current price and $0 otherwise
- Now assume that there is equal chance that the underlying will be above or below the current price
- This is similar to a coin flip where if the coin shows heads, you get $1 and if it shows tails, you get $0

Monte Carlo Simulation

- If interest rates are not zero then the expected value will be discounted
- You can also do a simulation to determine the expected value by running the coin flip N number of times
- Expected value $= \frac{\#\ heads \times \$1 + \#\ \text{tails} \times \$0}{\text{N}}$ before discounting
- Here the assumption is each simulated coin flip is independent and has the same distribution

Monte Carlo Simulation

- If N is a small number such as 10 then you might have 3 heads and 7 tails which is not unlikely and hence a poor estimate
- If N is a large number such as 1 million then it is quite unlikely to have a large deviation from 500,000 heads
- E.g. Even if there are 505,000 heads, the number will only be 0.5% off vs the first case which is 20% off

Monte Carlo Simulation

- When it comes to pricing derivatives, it will be more involved than a coin flip
- The coin flip is replaced by a distribution model of the underlying price such as Geometric Brownian Motion
- The payoff of the option will determine the numerical result of the “coin flip”

Monte Carlo Simulation

- Monte Carlo is based on two major theorems in probability theory:
  1. Law of Large Numbers: convergence of the estimation
  2. Central Limit Theorem: distribution of the estimation
- The estimation would approach a Normal distribution
- The standard error (estimated standard deviation) of the estimation is as important as the estimated price
- Remember to report the standard error of any estimated price in this course

Standard Error

- The mean $\bar{X} = \sum_i^N X_i/N$
- Variance of $\bar{X} = Var(\sum_i^N X_i/N)$

$$= \frac{NVar(X)}{N^2}$$

$$= \frac{Var(X)}{N}$$

- Standard Deviation = $\frac{\sigma_X}{\sqrt{N}}$
- Standard Error = $\frac{\widehat{\sigma_X}}{\sqrt{N}}$
- $\widehat{\sigma_X} = \sqrt{\sum_i^N \frac{(X_i - \bar{X})^2}{(N-1)}}$

Geometric Brownian Motion

- For the Black Scholes model, the underlying stock price follows a Geometric Brownian motion
- There is an analytical solution to compute the price of an asset following Geometric Brownian Motion

$$dx = rxdt + \sigma x dW_t$$

$$x_t = x_0 e^{\left((r-\frac{\sigma^2}{2})t+\sigma Z\right)} \; where \; Z \sim N(0,t)$$

- Therefore, we can simulate the x_T quite easily using this formula where T is the maturity of the option

Risk Neutral Measure

- Note that the GBM drift of r is under the risk neutral measure
- The asset price motion under real world probability would have an unknown drift rate

$$dx = \mu x dt + \sigma x dW_t$$

- μ is replaced by r under the risk neutral measure which is inferred from the market

In Class Exercise

- Compute the price of an European up and out call option where the barrier is only observed at maturity using Monte Carlo simulations
- x_0 = 120, K = 100, B = 120, T = 1.5, σ = 0.3
- Use N = 10, 100000, 10000000
- What is the standard error for each N?

slido

Audience Q&A Session

ⓘ Start presenting to display the audience questions on this slide.

Random Variable Generation

- A key component in Monte Carlo simulation is naturally the generation of random numbers
- This is a big topic with many different techniques
- This topic will be skipped due to the time constraint but for here are interesting techniques to read up on:
  - Probability Integral Transform (which we will use)
  - Alias Algorithm (for discrete distributions)
  - Rejection Algorithm (when the inverse CDF is unknown)
  - Copula Models (for multivariate distributions)

Random Variable Generation

- We will simply use algorithms in numpy and scipy for this course
- These packages have algorithms that can automatically generate normal random variables
- However, it is useful in some circumstances to generate a Uniform RV in (0,1) and use the inverse CDF of the required distribution to transform the uniform RV
- This is the Probability Integral Transform method

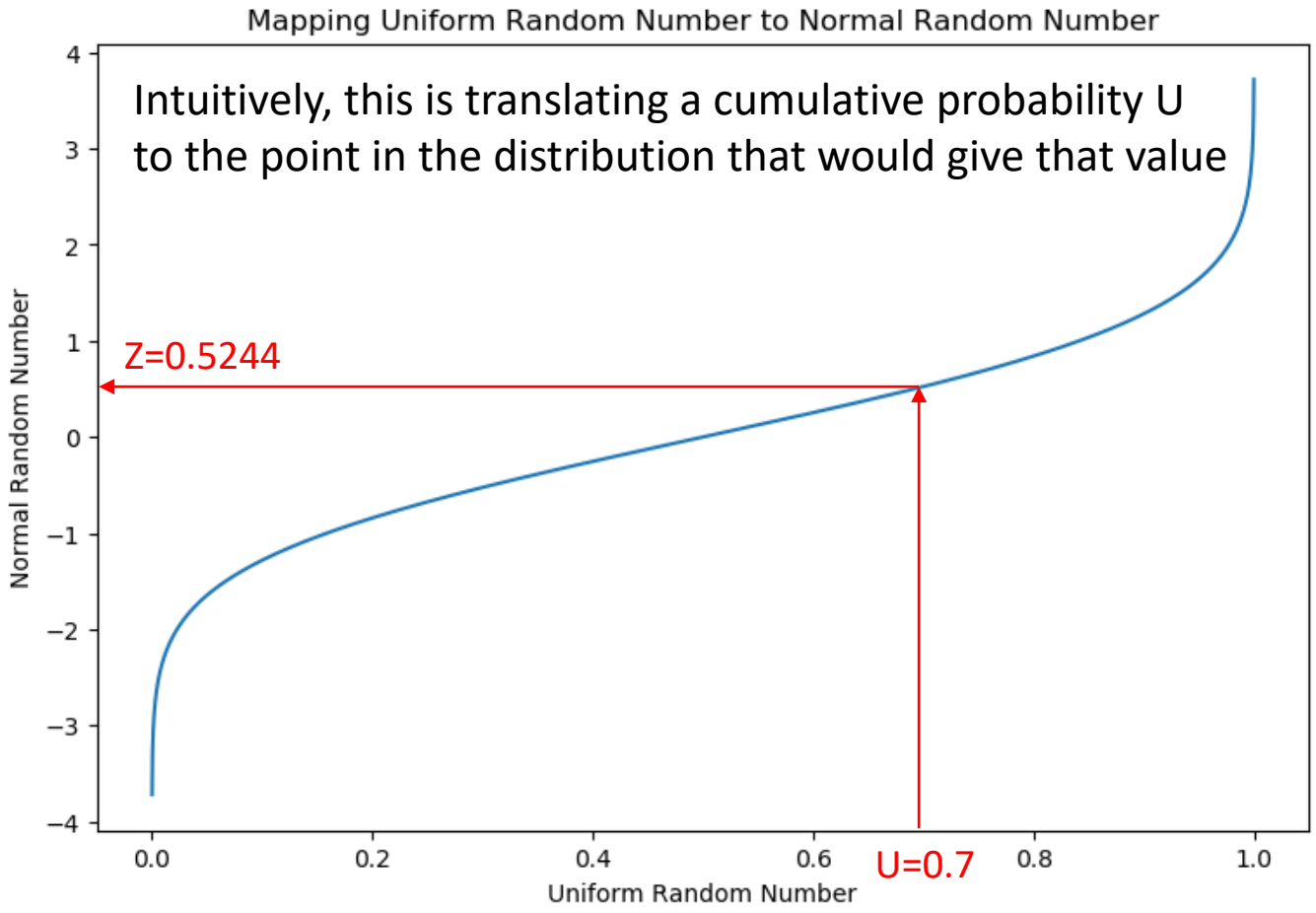
Probability Integral Transform

The method can be briefly summarised:

1. Generate a uniform random number in [0, 1] called U
2. Use the inverse CDF of the required distribution on U
3. The output will be a random number of the required distribution

- E.g. if standard Normal random numbers are required then $F^{-1}(U) = Z$ then Z will have a standard Normal distribution
- In Scipy, each of the distribution modules will have a ppf function which is the inverse CDF

Probability Integral Transform



Problems with Monte Carlo

- Requires large sample to reduce the error
- Hence running time can be slow
- Variance reduction techniques can help to reduce the sample size as we will see later
- Another technique that works for Monte Carlo is parallel processing

Parallel Processing for Monte Carlo

- Parallel processing suits Monte Carlo as each path is intrinsically independent
- Allows you to distribute the number of paths to be simulated across available resources
- The package ipyparallel in Python can create multiple engines running on a multi-core CPU
- https://ipyparallel.readthedocs.io/en/latest/
- Let's go through some examples on Jupyter Notebook

In Class Exercise

- Compute the same price as in the previous example of an European up and out call option
- x_0 = 120, K = 100, B = 120, T = 1.5, σ = 0.3
- Use N = 10000000 and time the algorithm without parallel processing
- Perform the simulations using ipyparallel where the N simulations are divided among each engine
- Time the algorithm line by line
- How does the calculation time compare and which operations take up most of the time?

In Class Exercise

- The computation time is heaviest on returning the massive array of the simulations
- The computation of the average and standard error of the combined array on a single engine also takes up significant time
- Let's rework the option pricing function to only return 2 numbers, average and variance of the simulations instead of the entire array of simulations

Standard Error

Say there are two samples with size N_1 and N_2

The combined sample unbiased variance, $\widehat{\sigma_X^2} = \frac{\sum_1^N (X_i - \bar{X})^2}{N-1}$

where $N = N_1 + N_2 \; and \; \bar{X} = \sum_1^N \frac{X_i}{N}$

will not equal $\frac{\sum_1^{N_1} (X_i - \overline{X_1})^2 + \sum_1^{N_2} (X_i - \overline{X_2})^2}{N-1}$

where $\overline{X_j} = \sum_{i=1}^{N_j} \frac{X_i}{N_j}$

unless $\overline{X_1} = \overline{X_2}$

Standard Error

- Piecewise variance
- We have 2 independent samples for X
- X_i for $i = 1\ to\ N_1$ and X_j for $j = 1\ to\ N_2$
- Then $\widehat{\sigma_X^2} = \varphi/(N-1)$ where N = N_1+N_2 and

$$\varphi = \sum_{1}^{N_1}(X_i - \overline{X_1})^2 + \sum_{1}^{N_2}(X_j - \overline{X_2})^2 + \frac{N_1 N_2}{N}(\overline{X_2} - \overline{X_1})^2$$

Standard Error

- For multiple samples where N_i are all the same sizes which we will denote by N and there are n samples

$$\varphi_{1,2} = \sum_{i=1}^{N}(X_i - \overline{X_1})^2 + \sum_{j=1}^{N}(X_j - \overline{X_2})^2 + \frac{N}{2}(\overline{X_2} - \overline{X_1})^2$$

where $\varphi_{1,2}$ is the numerator of variance for the combined samples 1 and 2

- You can think of the red term as an adjustment factor for the difference in the mean values
- Also $\varphi_{1,2} = \sum_{1}^{2N}(X_i - \overline{X_{1,2}})^2$ $where$ $\overline{X_{1,2}}$ mean of the combined sample

Standard Error

- When we try to add a third sample to the mix

$$\varphi_{1,2,3} = \sum_{i=1}^{2N}(X_i - \overline{X_{1,2}})^2 + \sum_{j=1}^{N}(X_j - \overline{X_3})^2 + \frac{2N}{3}(\overline{X_3} - \overline{X_{1,2}})^2$$

$$= \varphi_{1,2} + \sum_{1}^{N}(X_j - \overline{X_3})^2 + \frac{2N}{3}(\overline{X_3} - \overline{X_{1,2}})^2$$

$$= \sum_{i=1}^{N}(X_i - \overline{X_1})^2 + \sum_{j=1}^{N}(X_j - \overline{X_2})^2 + \sum_{k=1}^{N}(X_k - \overline{X_3})^2 + \frac{N}{2}(\overline{X_2} - \overline{X_1})^2 + \frac{2N}{3}(\overline{X_3} - \overline{X_{1,2}})^2$$

- This can be generalised to add all the numerators of the variance of the individual samples plus the incremental adjustment factors

Discretization Schemes

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Discretization Schemes

- There is an analytical solution for GBM which makes it easy to simulate the price at time t
- What if we use a model where the analytical formula is unknown?
- A discretization scheme would be required to simulate the path with small time steps
- Let's take a look at two such schemes
  - Euler Scheme
  - Milstein Scheme

Euler Scheme

- The same scheme as we have seen with ODEs except we are working with Stochastic Differential Equations

$$dx = rxdt + \sigma x dW_t$$

$$x_{t+\Delta t} = x_t + rx_t\Delta t + \sigma x_t\sqrt{\Delta t}Z \; where \; Z \sim N(0,1)$$

- The error is $O(\Delta t)$
- The error arises because of the assumed constancy of the drift and diffusion over $[t, t + \Delta t]$

Milstein Scheme

- The biggest component in the error comes from the constant volatility assumption
- The Milstein scheme makes a correction based on the volatility term
- The iteration is as follow:

$$x_{t+\Delta t} = x_t + r x_t \Delta t + \sigma x_t \sqrt{\Delta t} Z + \frac{1}{2}\sigma^2 x_t \Delta t (Z^2 - 1)$$

- The error is $O(\Delta t^2)$

Milstein Scheme

- CEV

$$x_{t+\Delta t} = x_t + r x_t \Delta t + \sigma x_t \sqrt{\Delta t} Z + \frac{1}{2}\beta \sigma^2 x_t^{2\beta-1} \Delta t (Z^2 - 1)$$

In Class Exercise

- Price the same option as the last exercise except we will use the CEV process for the asset where $\beta = 0.75$
- x_0 = 120, K = 100, B = 120, T = 1.5, σ = 0.3
- Use both Euler and Milstein scheme with N = 1000000
- Vary the steps at 5 / 10 / 20 and observe how the price converges with each scheme
- Perform the simulations with multiple engines using ipyparallel
- Actual price is ~4.445 (3.70% of 120)

Variance Reduction

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Variance Reduction

- Antithetic Variables
- Control Variables
- Importance Sampling
- Stratification

Antithetic Variables

- The idea behind antithetic variables is to create two estimated values that are negatively correlated
- Take the average of the two estimates to get the final estimate
- The final estimate would have a lower variance then either of the first two estimates

Antithetic Variables

$$Var\left(\frac{1}{2}\bar{X} + \frac{1}{2}\bar{X}_A\right)$$

$$= \frac{1}{4}Var(\bar{X}) + \frac{1}{4}Var(\bar{X}_A) + \frac{1}{2}Cov(\bar{X}, \bar{X}_A)$$

$$= \frac{1}{2}Var(\bar{X}) + \frac{1}{2}\frac{Cov(\bar{X}, \bar{X}_A)}{Var(\bar{X})}Var(\bar{X})$$

$$= \frac{1}{2}Var(\bar{X}) + \frac{1}{2}\rho(\bar{X}, \bar{X}_A)Var(\bar{X})$$

$$= Var(\bar{X})\left(\frac{1 + \rho(\bar{X}, \bar{X}_A)}{2}\right)$$

Antithetic Variables

- The most common method to create negatively correlated variables is to use the inverse cumulative distribution function of the variable:
  1. Create a set of uniformly distributed variables in (0,1) called U_1
  2. Create the second set of variables $U_2 = 1-U_1$
  3. Create the required variable Z_1 and Z_2 where $Z = F^{-1}(U)\ where\ F^{-1}(x)$ is the inverse CDF
  4. Use the combined set of Z_1 and Z_2 for simulation

In Class Exercise

- Price the same option in the last exercise under CEV model where $\beta = 0.75$
- We will just use the Milstein scheme but with antithetic variables
- x_0 = 120, K = 100, B = 120, T = 1.5, σ = 0.3
- Perform the simulations with multiple engines using ipyparallel

Control Variables

- The idea of control variables is to use the same set of random numbers to generate a known quantity
- The known quantity should be correlated to the value we want to estimate
- The error of the estimation for the known quantity can be calculated exactly
- We adjust the estimation of the option value using information about the correlation and the error in estimating the known quantity

Control Variables

- The method can be summarised as:
  1. Generate a set of random numbers for simulation
  2. Use the same set of random numbers to estimate the unknown option value $\bar{V}$ and the known quantity $\bar{X}$
  3. Adjust the final estimation $\bar{V}^* = \bar{V} + a(\bar{X} - E[X])$
- The quantity a is the adjustment factor and is chosen to minimise the variance of $\bar{V}^*$

Control Variables

$$E(\bar{V}^*) = E[\bar{V} + a(\bar{X} - E[X])]$$

$$= E(\bar{V}) + a[E(\bar{X}) - E(X)]$$

$$= E(V)$$

$$Var(\bar{V}^*) = Var(\bar{V}) + a^2 Var(\bar{X} - E[X]) + 2aCov(\bar{V}, \bar{X} - E[X])$$

$$= \frac{\sigma_V^2}{N} + \frac{a^2\sigma_X^2}{N} + 2aCov\left(\sum \frac{V}{N}, \sum \frac{X}{N}\right)$$

$$= \frac{\sigma_V^2}{N} + \frac{a^2\sigma_X^2}{N} + \frac{2a}{N^2} Cov(\sum V, \sum X)$$

$$= \frac{1}{N}(\sigma_V^2 + a^2\sigma_X^2 + 2a\sigma_{VX})$$

Adjustment Factor

- $Var(\bar{V}^*) = \frac{1}{N}(\sigma_V^2 + a^2\sigma_X^2 + 2a\sigma_{VX})$

$$\frac{d\sigma_{\bar{V}^*}^2}{da} = \frac{1}{N}(2a\sigma_X^2 + 2\sigma_{VX})$$

Find minimum by solving for a where $\frac{d\sigma_{\bar{V}^*}^2}{da} = 0$

$$i.e.\ \frac{1}{N}(2a\sigma_X^2 + 2\sigma_{VX}) = 0$$

- Minimise variance by choosing $a = -\frac{\sigma_{VX}}{\sigma_X^2}$

Adjustment Factor

- The result is $Var(\bar{V}^*) = \frac{\sigma_V^2}{N}(1-\rho^2)$ where ρ is the correlation between V and X
- The adjustment is better if there is a strong correlation (regardless of positive/negative) between X and V
- We don't know the true value of σ_{VX} and σ_X^2 so we use the same sample to estimate $\widehat{\sigma_{VX}}$ and $\widehat{\sigma_X}^2$

In Class Exercise

- We will go back to pricing the up and out European call under GBM
- x_0 = 120, K = 100, B = 120, T = 1.5, σ = 0.3
- Let's use the final spot price S(T) as the control variable since we know what is the expected value
- $E[S_T] = S_0 e^{rT}$

For Homework 9: Covariance

- Piecewise covariance
- We have 3 independent samples for X
- X_i for $i = 1\ to\ N_1, X_j$ for $j = 1\ to\ N_2$ and X_k for k $= 1\ to\ N_3$
- Similarly, we have independent samples for Y
- Unbiased estimator for $Cov(X, Y)\ = \varphi/(\sum_{i=1}^{3} N_i - 1)$

$$\varphi = \sum_{1}^{N_1}(X_i - \overline{X_1})(Y_i - \overline{Y_1}) + \sum_{1}^{N_2}(X_j - \overline{X_2})(Y_j - \overline{Y_2}) + \sum_{1}^{N_k}(X_k - \overline{X_3})(Y_k - \overline{Y_3})$$

$$+\frac{N_1 N_2}{(N_1 + N_2)}(\overline{X_2} - \overline{X_1})(\overline{Y_2} - \overline{Y_1}) + \frac{(N_1+N_2)N_3}{(N_1 + N_2 + N_3)}(\overline{X_3} - \overline{X_{12}})(\overline{Y_3} - \overline{Y_{12}})$$

Importance Sampling

- Notice that for there are large regions of possible values of X_T where the option ends up with no value?
- E.g. if $X_T < K$ then a call option value = 0
- These regions do not contribute to the evaluation of the Expected value
- What if we can restrict the simulation to only the regions which contribute to the Expected value?
- The final result would have to be weighted by the probability that X_T falls in that specific region

Importance Sampling

- Using GBM as an example
- $x_T = x_0 e^{\left((r-\frac{\sigma^2}{2})T+\sigma\sqrt{T}Z\right)}$
- If we need $x_t > K$ then that is equivalent to:
- $Z > \frac{\left[\ln\left(\frac{K}{x_0}\right)-\left(r-\frac{1}{2}\sigma^2\right)T\right]}{\sigma\sqrt{T}}$
- Let L = $\frac{\left[\ln\left(\frac{K}{x_0}\right)-\left(r-\frac{1}{2}\sigma^2\right)T\right]}{\sigma\sqrt{T}}$
- If U is the interval $(\Phi(L), 1) then\ Z = \Phi^{-1}(U) > L$ where $\Phi(x)$ is the standard Normal CDF

Importance Sampling

- Similarly, If we need $x_T < B$ then that is equivalent to:
- $Z < \frac{\left[\ln\left(\frac{\mathrm{B}}{x_0}\right) - \left(\mathrm{r} - \frac{1}{2}\sigma^2\right)\mathrm{T}\right]}{\sigma\sqrt{T}}$
- Let M = $\frac{\left[\ln\left(\frac{\mathrm{B}}{x_0}\right) - \left(\mathrm{r} - \frac{1}{2}\sigma^2\right)\mathrm{T}\right]}{\sigma\sqrt{T}}$
- If U is the interval $\left(0, \Phi(M)\right)$ then $Z = \Phi^{-1}(U) < M$
- Both results imply if U is the interval $\left(\Phi(L), \Phi(M)\right)$ then $\mathrm{L} < Z = \Phi^{-1}(U) < M$ which means $K < x_T < B$

Importance Sampling

- The process starts with us generating uniform RVs, U in the interval $\left(\Phi(L), \Phi(M)\right)$
- Then convert U using the Standard Normal inverse CDF
- This creates RVs that ensure $K < x_T < B$
- Use this set of simulations to estimate the option value
- The estimation must be weighted by the probability that the asset price falls in this range i.e. $K < x_T < B$
- $\mathrm{P}(K < x_T < B) = \Phi(M) - \Phi(L)$
- Final price = $\mathrm{P}(K < x_T < B) \times V$ where V is the estimate obtained via importance sampling

Importance Sampling

- This is essentially a change in probability measure
- The specific change can be tailored to the characteristics of the option to maximise the impact
- This technique is effective for the specific example of the Up and Out European Call option that we have been pricing

In Class Exercise

- We will go back to pricing the up and out European call under GBM
- x_0 = 120, K = 100, B = 120, T = 1.5, σ = 0.3
- The change of probability measure here would restrict the outcome of S_T to this range: $K < S_T < B$
- How does the reduction in standard error compare with the other methods?

Stratified Sampling

- Stratified sampling splits the population into subgroups or bins
- The sample within each bin is restricted to a certain segment of the population
- Estimate the option value by combining all the bins into one big sample
- The variance of the estimate will be lower if the variance in each bin is less than the population variance

Stratified Sampling

- The method can be summarised as:
  1. Create N_b bins of uniformly distributed RVs each corresponding to an interval $\left(0, \frac{1}{N_b}\right) .. (1 - \frac{1}{N_b}, 1)$
  2. Use the Probability Integral Transform method to convert the uniform RVs to the RV of required distribution e.g. standard normal
  3. Perform the standard MC procedure using each bin of RVs
  4. Get required estimate from the combined sample size

Standard Error

- The standard error cannot be calculated in the normal way as we have started with separate sample bins
- $Var(\bar{V}) = \frac{1}{N^2}\sum_1^{N_b}\frac{N}{N_b}\sigma_i^2$ (bins treated as independent) where N = total number of samples and N_b is the number of bins
- Note that variance in each bin is different
- The variance simplifies to $\frac{1}{N \times N_b}\sum_1^{N_b}\sigma_i^2$
- The variance in each bin would have to be estimated from the sample in each bin

Standard Error

$$Var(\bar{V}) = Var\left(\frac{1}{N}\sum_{i=1}^{N_b}\sum_{j=1}^{N_i} V_{ij}\right) \; where \; N_i \; is \; number \; per \; bin$$

$$= \frac{1}{N^2}\sum_{i=1}^{N_b}\sum_{j=1}^{N_i} Var(V_{ij})$$

$$= \frac{1}{N^2}\sum_{i=1}^{N_b}\sum_{j=1}^{N_i} \sigma_i^2$$

$$= \frac{1}{N^2}\sum_{1}^{N_b}\frac{N}{N_b}\sigma_i^2 \; where \; N_i = \frac{N}{N_b} \; if \; equal \; number \; per \; bin$$

$$= \frac{1}{N \times N_b}\sum_{1}^{N_b}\sigma_i^2$$

In Class Exercise

- We will go back to pricing the up and out European call under GBM
- x_0 = 120, K = 100, B = 120, T = 1.5, σ = 0.3
- Using a total sample size of N = 1,000,000
- Try using different number of bins: 100 / 1,000, / 10,000 / 100,000 / 500,000

FDM vs MC

Criteria	FDM	MC
Dimensionality	Better in lower dimensions	Easy to go to higher dimensions
Path Dependency	Depends on the type of dependency	Generally deals well with path dependency
Computing Greeks	Excellent for computing Greeks	Not efficient
Decision Points	Excellent for implementation	Difficult to implement