Introduction to Monte Carlo methods

STEVEN LI

MAIN POINTS

- Simple example
- Convergence
- Variance reduction
- Greeks
- Path-dependent options
- Multivariate Simulation and Correlation

SIMPLE EXAMPLE

European call option:

$$C = E^{RN}[e^{-rT} max(S_T - K, 0)]$$

In the Black-Scholes model, S_t follows the process

$$dS_t = rS_t dt + \sigma S_t dW_t$$

Solution of this SDE

$$S_T = S_0 \exp\left((r - \sigma^2/2)T + \sigma W_T\right)$$

Quick Facts about Brownian motion

- $W_0 = 0$
- $W_t \sim N(0,t)$
- $cov(W_t, W_{t+\Delta t} - W_t) = 0$, for $\Delta t \neq 0$
- W_t is everywhere continuous and non-differentiable
- It can also be proved from above properties that $cov(W_t, W_s) = \min(t, s)$
- $(dW_t)^2 = dt$

SIMPLE EXAMPLE

The expectation $E^{RN}[e^{-rT}max(S_T - K, 0)]$ can be calculated using Monte Carlo simulations as follows:

- Generate a random sample $S_T^1, S_T^2, \ldots, S_T^N$
- For each of the simulated values S_T^i, calculate the discounted payoff $e^{-rT}max(S_T^i - K, 0)$
- Take the average $\widehat{C} = \frac{1}{N}\sum_{i=1}^{N} e^{-rT}max(S_T^i - K, 0)$

What we get is not the exact value of the call option, it's just an "estimator"

SIMPLE EXAMPLE

- How do we generate the random S_T^i
  - We need a standard normal random number X
  - In Excel this can be obtained using Normsinv(rand()). In Python we could use np.random.randn() (the vectorized form is more efficient , we will see later in the code)
  - The random variable W_T has mean 0 and variance T, so we set $W_T = \sqrt{T} * X$
  - We use the formula for ST so we get

$$S_T = S_0 \exp\left((r - \sigma^2 2/2)T + \sigma * \sqrt{T} X) \right)$$

EXERCISE

- Use Python to simulate 1,000 Brownian Path from time 0 to T, with T=10, $\Delta T = 0.1$
- Use The random vector above to Simulate 1000 Stock prices with follows Geometric Brownian Motion, With $S_0 = 100, \sigma = 0.3, T = 10, r = 0.01$

MAIN POINTS

- Simple example
- Convergence
- Variance reduction
- Greeks
- Path-dependent options
- Multivariate Simulation and Correlation

CONVERGENCE

If you recalculate a few times you see that you get different prices

- What we get is not the exact price, it is an estimator
- The estimator is a random variable
- The estimator is called "unbiased" if its mean is equal to the true value of the option
- The difference between the estimator and the true value is the error
- A Monte Carlo scheme is good if the standard deviation of the error is small

CONVERGENCE

- The MC estimator for the call price is

$$E = \frac{1}{N}\sum e^{-rT}\mathrm{Max}(S_T{}^{(i)} - K, 0)$$

- The variables inside the sum are independent and identically distributed, in particular they have the same variance

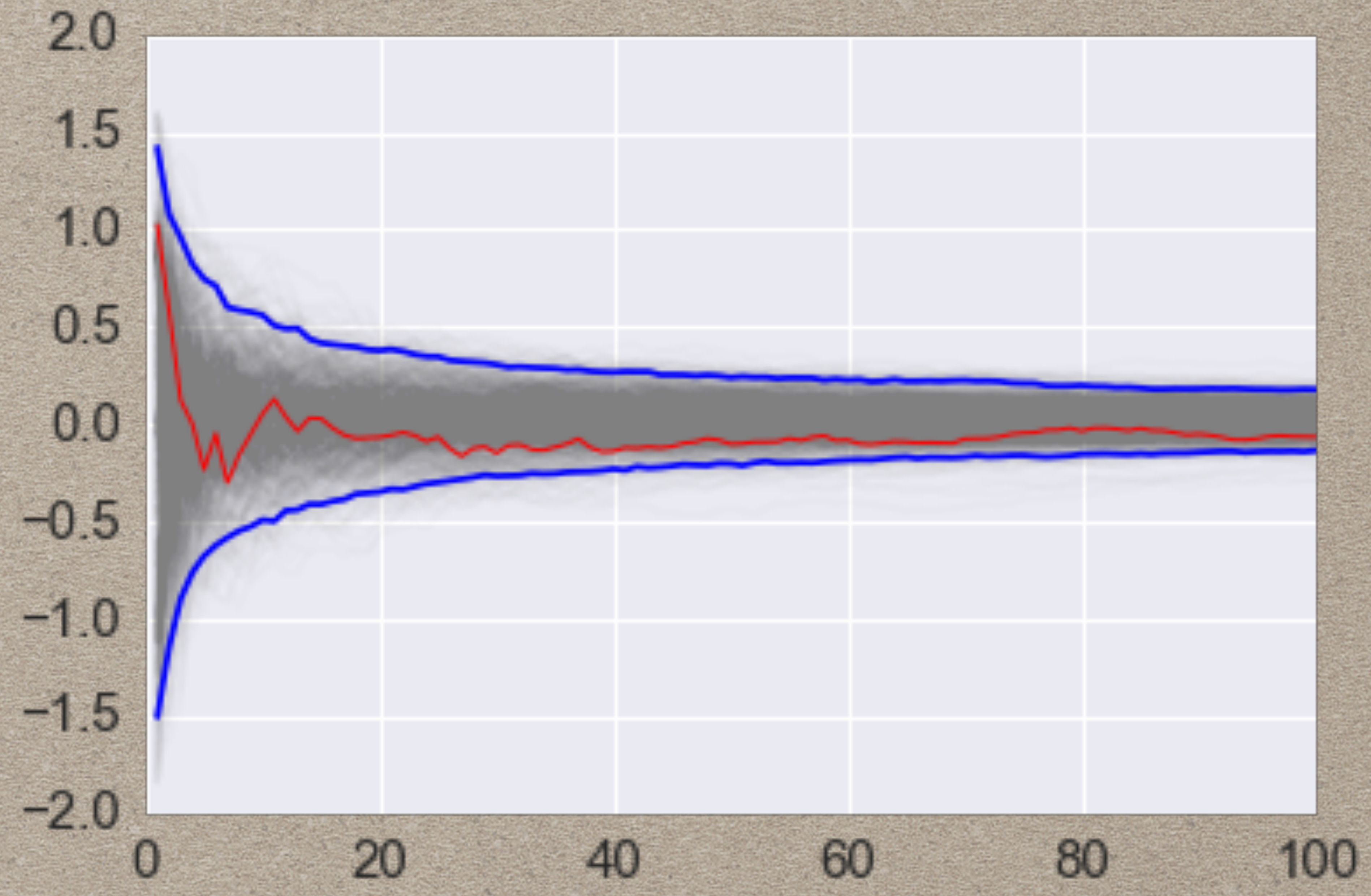
$$\begin{aligned} Var[E] &= \frac{1}{N^2} Var\left[\sum e^{-rT}\mathrm{Max}(S_T{}^{(i)} - K, 0)\right] \\ &= \frac{1}{N} Var\left[e^{-rT}\mathrm{Max}(S_T{}^{(i)} - K, 0)\right] \end{aligned}$$

- The standard deviation of the estimator is the square root of its variance, so we have

$$\sigma[E] = \frac{C}{\sqrt{N}}$$

CONVERGENCE

- This means that the error is inverse proportional with the square root of the number of paths
- If we want to increase the accuracy of a Monte Carlo scheme by a factor of 10 we need to run 100 times more paths
- Put it differently, if your MC scheme gives you correct results up to one dollar and you want correct results up to one cent, you need to run 10000 times more paths (which generally takes 10000 times more time)



MAIN POINTS

- Simple example
- Convergence
- Variance reduction
- Greeks
- Path-dependent options
- Multivariate Simulation and Correlation

• VARIANCE REDUCTION

- To improve the accuracy of a Monte Carlo scheme we need to reduce the variance of the estimator
- So "Variance Reduction" = Accuracy Improvement
- Some Variance Reduction techniques:
  - Antithetic variates
  - Control variate
  - Importance sampling
  - Quasi-random numbers
  - Moment matching

• VARIANCE REDUCTION

- Control Variate

Want to estimate $\hat{\theta} = Y$, consider another estimator $\hat{\theta}_c = Y + c(Z - E[Z])$. Obviously $E[\hat{\theta}] = E[\hat{\theta}_c]$. But what about their variances?

$Var(\hat{\theta}_c) = Var(Y) + c^2 Var(Z) + 2c cov(Y, Z)$ (remember that $Var(\theta) = Var(Y)$)

which. Is minimized when $c = -\dfrac{cov(Y,Z)}{Var(Z)}$

minimized variance is $Var(\hat{\theta}_c) = Var(\hat{\theta}) - \dfrac{cov(Y,Z)^2}{Var(Z)}$

Therefore as long as $cov(Y,Z) \neq 0$, $Var(\hat{\theta}_c) < Var(\theta)$

Example: Take Y to be the price of an exotic option, Z can be taken as the price of an European option with otherwise identical characteristics.

VARIANCE REDUCTION

- Antithetic variates
  - whenever you use the normal random number X in the simulation, use –X as well
  - You run twice as many paths, but generally improve your accuracy by more than $\sqrt{2}$ times
- Why it works? Remember var(X+Y) decreases with corr(X,Y)

VARIANCE REDUCTION

- Quasi random numbers
    - Instead of using random numbers, use numbers that fill the space more uniformly
    - Such numbers are called low discrepancy numbers, or "quasi random numbers"
    - The error for Quasi Monte Carlo schemes decreases as 1/N as opposed to $1/\sqrt{N}$ for regular Monte Carlo
- Importance Sampling - 'shift' the density distribution in a smart way that allow rare events to be more efficiently sampled

IMPORTANCE SAMPLING

Suppose you want to sample from a distribution f, in order to calculate the exception of some function g

$$E^f[g] = \int g(x)f(x)dx$$

Suppose the support of g locates in the region where the probability density f is tiny. Therefore a lot of your samples will fall out of the region where g(x) takes material (non-zero) values.

The importance sampling method allows you to increase the sample efficiency by introducing a new density h(x) such that under h(x) the support of g lies in regions with high h(x). So

$$E^f[g] = \int g(x)\frac{f(x)}{h(x)}h(x)dx = E^h[g\frac{f}{h}] = E^h[g'] = \int g'(x)h(x)dx$$

Here $g'(x) = g(x)\dfrac{f(x)}{h(x)}$

IMPORTANCE SAMPLING EXAMPLE

Use importance sampling to estimate the probability that P(X>5), where X~N(0,1)

Here $f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}, \quad g(x) = 1_{x>5}$

Obviously under the distribution $f(x), \; g(x) = 0$ with dominate probability. Therefore naive sampling will give you g(x)=0 for almost all samples.

We need to find a new distribution under which $g(x) > 0$ with significant probability. A natural choice is $h(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-5)^2}$

So we can sample $x \sim h(x)$, and calculate the average of

$$g'(x) = g(x)\frac{f(x)}{h(x)} = 1_{x>5} e^{-5x+12.5}$$

MAIN POINTS

- Simple example
- Convergence
- Variance reduction
- Greeks
- Path-dependent options
- Multivariate Simulation and Correlation

GREEKS

- Let's calculate the delta of our European call option
- Finite difference approximation: bump the stock value up by some amount dS, revalue the option, then divide the difference by dS

$$\Delta \approx \frac{C(S_0 + dS) - C(S_0)}{dS}$$

GREEKS

- There is a bad way to do this and a good way
    - The bad way: when you reprice the option for the bumped stock price you use different random numbers
    - Good way: use the same random numbers
    - Why?
- For the bad way $C(S+dS)-C(S)=O(1)$ (since both valuations are independent)
- For the good way $C(S+dS)-C(S)=O(dS)$, which ensure convergence (not blowing up) of Δ

MAIN POINTS

- Simple example
- Convergence
- Variance reduction
- Greeks
- Path-dependent options
- Multivariate Simulation and Correlation

PATH-DEPENDENT OPTIONS

- An option is path-dependent if its payoff depends not only on the terminal value of the stock, but on the whole history of the stock price
- Examples: barrier options, Asian options, etc
- To price a path-dependent option you need to
  - simulate the whole path of the stock price
  - calculate the payoff
  - take the average

PATH-DEPENDENT OPTIONS

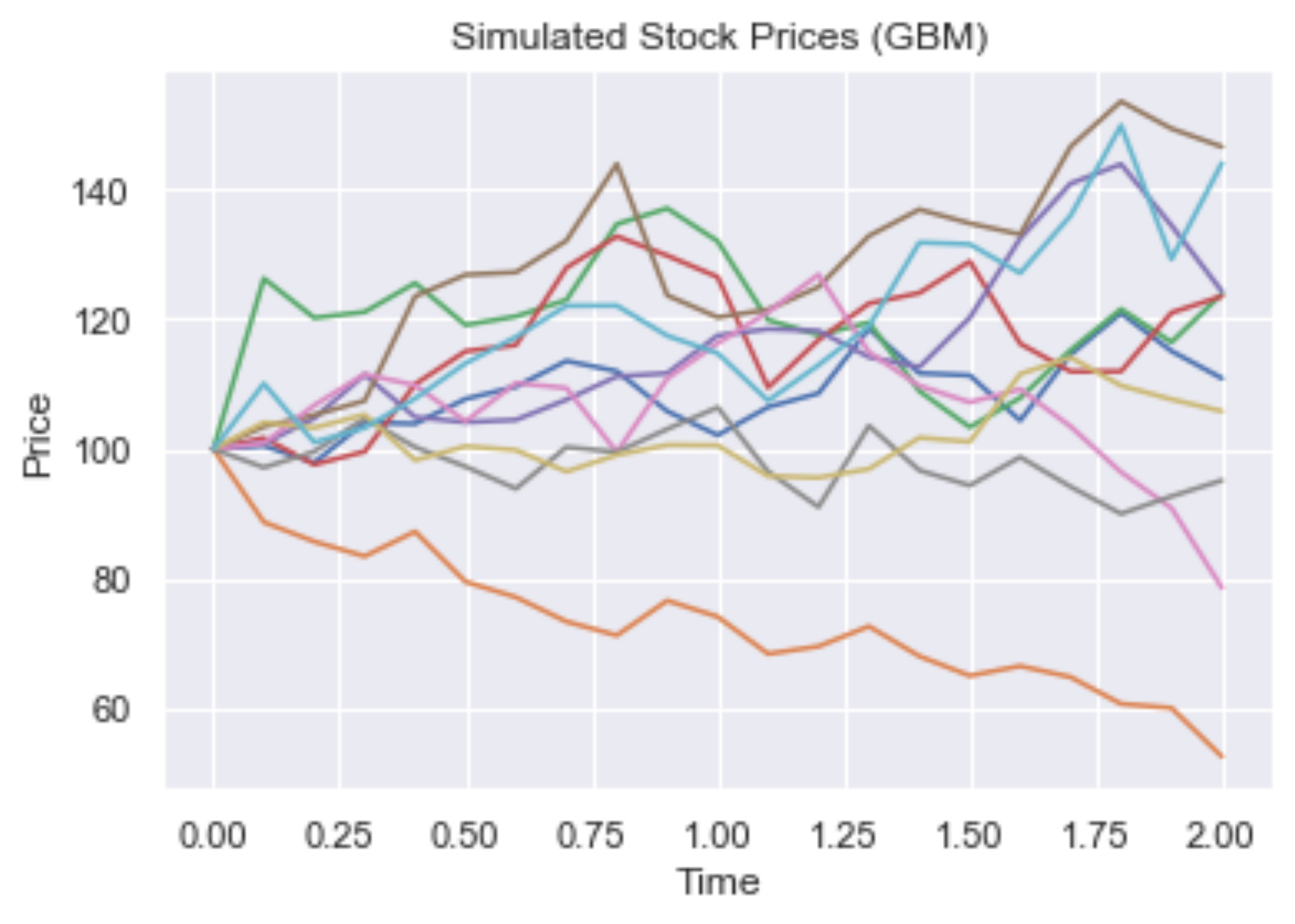
- We price an up-and-out option and an arithmetic Asian option, both with quarterly observations
- We simulate the evolution of the stock price at the times

$$t_0=0,\ t_1=0.25,\ t_2=0.5,\ \dots,\ t_{20}=5$$

PATH-DEPENDENT OPTIONS

- S_t follows the SDE. $dS_t = rS_t dt + \sigma S_t dW_t$
- We recursively simulate S at the time t_{i+1} after time t_i using the scheme:
- $S_{t_{i+1}} = S_{t_i} + rS_{t_i} + \sigma S_{t_i}\sqrt{\Delta t}Z$
- where Z is a standard normal random number

PATH-DEPENDENT OPTIONS



EXERCISE

USE MONTE CARLO TO CALCULATE THE PRICE OF

1. AN UP-AND-OUT OPTION: OPTION PAYS $max(S_T - K, 0)$ AT T IF $\max_{0 \le t \le T} S_t < B$ AND 0 OTHERWISE

3. AN UP-AND-IN OPTION: OPTION PAYS $max(S_T - K, 0)$ AT T IF $\max_{0 \le t \le T} S_t \ge B$ AND 0 OTHERWISE

5. COMPARE THE SUM OF (1) AND (2) WITH THAT OF THE EUROPEAN OPTION. WHAT DO YOU OBSERVE?

Condition:

$S_0 = 100, K = 100, B = 110, \sigma = 0.3, r = 0.05, T = 2$

MAIN POINTS

- Simple example
- Convergence
- Variance reduction
- Greeks
- Path-dependent options
- Multivariate Simulation and Correlation

HOW TO SIMULATE CORRELATED NORMALS

Suppose we want to simulate n normal random variables

$$\overrightarrow{X} = (X_1, X_2, \ldots, X_n)$$

Such that $X_i \sim N(0,1), corr(X_i, X_j) = \rho_{i,j}$

This can be achieved via Cholesky decomposition. Let P denote the nxn symmetric correlation matrix (which is guaranteed to be positive semi-definite): $P_{ij} = \rho_{i,j}$

P can be deposed into the following form $P = LL^T$, where L is a lower triangular matrix.

Suppose we have $\overrightarrow{Z} = (Z_1, Z_2, \ldots, Z_n)$ is an i.i.d. normals, i.e., $Z_i \sim N(0,1)$, $corr(Z_i, Z_j) = \delta_{i,j}$ (identity matrix)

It turns out that the transformed vector $\overrightarrow{X} = L\overrightarrow{Z}$ satisfies that $X_i \sim N(0,1), corr(X_i, X_j) = \rho_{i,j}$

EXAMPLE

$n = 2$. $corr(X_1, X_2) = \rho$. In this case $P = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$,

$$L = \begin{bmatrix} 1 & 0 \\ \rho & \sqrt{1-\rho^2} \end{bmatrix}$$

So we could simulate two i.i.d. standard normals Z_1, Z_2 and get the desired X_1, X_2 via

- $X_1 = Z_1$
- $X_2 = \rho Z_1 + \sqrt{1-\rho^2} Z_2$ (this is a well-known formula in credit risk)