Homework 3, MATH 5010

Zongyi Liu

Sun, Mar 20, 2025

1 Question 1

Suppose that the price X_t of Euro in terms of USD follows $dX_t = 0.03X_t dt + 0.1X_t dW_t$ Write the equation for Y_t the price of USD in terms of Euro.

Answer

Suppose the price of Euro in terms of USD, X_t, and it follows the stochastic differential equation $dX_t = 0.03X_t\,dt + 0.1X_t\,dW_t$

Let $Y_t = \frac{1}{X_t}$ be the price of USD in terms of Euro. To find the SDE for Y_t, we apply Ito's Lemma to the function $f(X_t) = \frac{1}{X_t}$. We compute the first and second derivatives:

$$f'(x) = -\frac{1}{x^2}, \quad f''(x) = \frac{2}{x^3}$$

Applying Ito's Lemma, we get $dY_t = f'(X_t)\,dX_t + \frac{1}{2}f''(X_t)(dX_t)^2$, then we use the given SDE for X_t:

$$dX_t = \mu X_t\,dt + \sigma X_t\,dW_t, \quad \text{with } \mu = 0.03,\ \sigma = 0.1$$
$$(dX_t)^2 = \sigma^2 X_t^2\,dt$$

Substituting into the Ito formula:

$$\begin{aligned}
dY_t &= -\frac{1}{X_t^2}(\mu X_t\,dt + \sigma X_t\,dW_t) + \frac{1}{2}\cdot\frac{2}{X_t^3}\cdot\sigma^2 X_t^2\,dt \\
&= -\mu\frac{1}{X_t}\,dt - \sigma\frac{1}{X_t}\,dW_t + \sigma^2\frac{1}{X_t}\,dt \\
&= (\sigma^2 - \mu)\frac{1}{X_t}\,dt - \sigma\frac{1}{X_t}\,dW_t \\
&= (\sigma^2 - \mu)Y_t\,dt - \sigma Y_t\,dW_t
\end{aligned}$$

Substituting the numerical values, we can get $dY_t = (0.01 - 0.03)Y_t\,dt - 0.1Y_t\,dW_t = -0.02Y_t\,dt - 0.1Y_t\,dW_t$

2 Question 2

Suppose that X_t follows the process $dX_t = 0.04X_t dt + 0.3X_t dW_t$. Using Ito's Lemma find the equation for the process for $Y_t = \ln X_t$

Answer

Suppose X_t follows the process:

$$dX_t = 0.04X_t\,dt + 0.3X_t\,dW_t$$

Let $Y_t = \ln X_t$. Using Ito's Lemma, we compute the SDE for Y_t. First, compute the derivatives:

$$f(x) = \ln x, \quad f'(x) = \frac{1}{x}, \quad f''(x) = -\frac{1}{x^2}$$

Apply Ito's Lemma:

$$dY_t = f'(X_t)\,dX_t + \frac{1}{2}f''(X_t)(dX_t)^2$$

We have:

$$dX_t = \mu X_t\,dt + \sigma X_t\,dW_t, \quad \mu = 0.04,\ \sigma = 0.3$$
$$(dX_t)^2 = \sigma^2 X_t^2\,dt = 0.09X_t^2\,dt$$

Substituting:

$$\begin{aligned} dY_t &= \frac{1}{X_t}(0.04X_t\,dt + 0.3X_t\,dW_t) + \frac{1}{2}\left(-\frac{1}{X_t^2}\right)(0.09X_t^2\,dt) \\ &= 0.04\,dt + 0.3\,dW_t - \frac{1}{2}\cdot 0.09\,dt \\ &= (0.04 - 0.045)\,dt + 0.3\,dW_t \\ &= -0.005\,dt + 0.3\,dW_t \end{aligned}$$

Therefore, the process $Y_t = \ln X_t$ satisfies:

$$dY_t = -0.005\,dt + 0.3\,dW_t$$

3 Question 3

A stock price is \$100 now. In 1 month, it can go to \$110 or \$90. The annual interest rate is 10% with continuous compounding. Using risk-free portfolios, determine the value of the one-month European put with strike price 100 and with strike price 95.

Answer

Given current stock price is $S_0 = 100$, after 1 month, it will be $S_u = 110, \quad S_d = 90$. The annual risk-free interest rate is $r = 0.10$, and time to maturity is $\Delta t = \frac{1}{12}$, then the risk-neutral probability is:

$$u = \frac{S_u}{S_0} = \frac{110}{100} = 1.1, \quad d = \frac{S_d}{S_0} = \frac{90}{100} = 0.9$$

$$e^{r\Delta t} = e^{0.10 \cdot \frac{1}{12}} \approx e^{0.008333} \approx 1.00837$$

$$q = \frac{e^{r\Delta t} - d}{u - d} = \frac{1.00837 - 0.9}{1.1 - 0.9} = \frac{0.10837}{0.2} \approx 0.54185$$

Payoffs at Maturity:

Strike $K = 100$

$$\text{If } S_T = 110, \quad \text{Payoff} = \max(100 - 110, 0) = 0$$

$$\text{If } S_T = 90, \quad \text{Payoff} = \max(100 - 90, 0) = 10$$

Strike $K = 95$

$$\text{If } S_T = 110, \quad \text{Payoff} = \max(95 - 110, 0) = 0$$

$$\text{If } S_T = 90, \quad \text{Payoff} = \max(95 - 90, 0) = 5$$

Present Value under Risk-neutral Measure

$$e^{-r\Delta t} = e^{-0.008333} \approx 0.9917$$

For strike $K = 100$

$$\begin{aligned} P_{100} &= 0.9917 \cdot [q \cdot 0 + (1 - q) \cdot 10] \\ &= 0.9917 \cdot (0.45815 \cdot 10) \\ &= 0.9917 \cdot 4.5815 \approx 4.54 \end{aligned}$$

For strike $K = 95$

$$\begin{aligned} P_{95} &= 0.9917 \cdot [q \cdot 0 + (1 - q) \cdot 5] \\ &= 0.9917 \cdot (0.45815 \cdot 5) \\ &= 0.9917 \cdot 2.29075 \approx 2.27 \end{aligned}$$

So the final answers are:

- European put with strike $K = 100$: $P = \$4.54$
- European put with strike $K = 95$: $P = \$2.27$

4 Question 4

Use risk-neutral valuation to calculate the probabilities that will give you the correct put prices in problem 3.

Answer

We are given:

- $S_0 = 100$
- $S_u = 110$, $S_d = 90$
- $r = 0.10$ (annual continuous rate)
- $\Delta t = \frac{1}{12}$

$$e^{r\Delta t} = e^{0.10 \cdot \frac{1}{12}} \approx e^{0.008333} \approx 1.008368$$

We can simply calculate the rate:

- Going up: $p = \frac{R-d}{u-d} \approx \frac{1.08368-0.9}{1.1-0.9} = 0.54184$
- Going dawn: 1-p $\approx 1 - 0.54184 = 0.45816$

For strike price $K = 100$.
$E\left(P_T\right) = 0 \times p + (1-p) \times 10 = 4.582$, and the value of the put is $PV\left(E\left(P_T\right)\right) = e^{-rT} \times 4.582 \approx \$4.544$
For strike price $K = 95$.
$E\left(P_T\right) = 0 \times p + (1-p) \times 5 = 2.291$, and the value of the put $PV\left(E\left(P_T\right)\right) = e^{-rT} \times 2.291 \approx \$2.272$
Those results are the same as in Problem 3.

5 Question 5

Construct trading strategies in stock only that replicate each of the two puts of problem 3. That means construct a) synthetic long put strategy with strike price 100. b) synthetic long put strategy with strike price 95. What is the cost of each synthetic trading strategy.

Answer

To replicate a long put option using only the stock and cash (borrowing/lending), we construct a portfolio that mimics the put's payoff in both the up and down states. The strategy involves short selling shares of stock and lending cash at the risk-free rate.

The cost of the synthetic put equals the initial investment required to set up this portfolio. The synthetic put requires:

- Short position in Δ shares of stock.
- Lending B dollars at the risk-free rate.

The portfolio must satisfy:

$$\Delta S_u + Be^{rT} = P_u \quad \text{(Up state payoff)}$$
$$\Delta S_d + Be^{rT} = P_d \quad \text{(Down state payoff)}$$

Solving for Δ (hedge ratio) and B (cash position):

$$\Delta = \frac{P_d - P_u}{S_d - S_u}, \quad B = e^{-rT}(P_u - \Delta S_u)$$

5.1 Part A

Synthetic Long Put with Strike $K = \$100$

Determine Payoffs

- Up state (P_u): $\max(100 - 110, 0) = \$0$
- Down state (P_d): $\max(100 - 90, 0) = \$10$

Calculate Hedge Ratio (Δ):

$$\Delta = \frac{P_d - P_u}{S_d - S_u} = \frac{10 - 0}{90 - 110} = \frac{10}{-20} = -0.5$$

Which means short sell 0.5 shares of stock.

Calculate Cash Position (B)

$$B = e^{-rT}(P_u - \Delta S_u) = e^{-0.10 \cdot \frac{1}{12}}(0 - (-0.5 \cdot 110)) = 0.9917 \cdot 55 \approx \$54.54$$

We should lend \$54.54 at the risk-free rate.

The synthetic put's cost is the initial investment:

$$\text{Cost} = \Delta S_0 + B = (-0.5 \cdot 100) + 54.54 = -50 + 54.54 = \$4.54$$

This matches the put price calculated in Problem 3.

5.2 Part B

Synthetic Long Put with Strike $K = \$95$:

Determine Payoffs

- Up state (P_u): $\max(95 - 110, 0) = \$0$
- Down state (P_d): $\max(95 - 90, 0) = \$5$

Calculate Hedge Ratio (Δ):

$$\Delta = \frac{P_d - P_u}{S_d - S_u} = \frac{5 - 0}{90 - 110} = \frac{5}{-20} = -0.25$$

Short sell 0.25 shares of stock.

Calculate Cash Position (B):

$$B = e^{-rT}(P_u - \Delta S_u) = e^{-0.10 \cdot \frac{1}{12}}(0 - (-0.25 \cdot 110)) = 0.9917 \cdot 27.5 \approx \$27.27$$

This means to lend \$27.27 at the risk-free rate.

The synthetic put's cost is the initial investment:

$$\text{Cost} = \Delta S_0 + B = (-0.25 \cdot 100) + 27.27 = -25 + 27.27 = \$2.27$$

This matches the put price calculated in Problem 3.

6 Question 6

Create a spreadsheet modeling trajectories of geometric Brownian motion starting at 50 with growth rate 5 percent (it is also risk-free rate) and volatility 30 percent. Make a spreadsheet that calculates European calls maturing in 1 year with strikes 50 and 51 on non-dividend paying stock using Monte Carlo method and using 20,000 trajectories with 250 steps in each trajectory. Compare Monte-Carlo price with 20,000 trajectories to theoretical model price. Before you start this project, restart your computer and close internet and all other programs except Excel.

Answer

6.1 Theoretical Calculation

In this case, it is easy to calculate the theoretical BS price, given:

- S_0=50
- Growth rate=5%
- σ=30%
- Strike price=50, 51
- T=1

For strike 50, the price is 7.116, and for strike 51, the price is 6.646.
Based on BS model, we have strike price=50:

$$d_1 = \frac{\ln(50/50) + (0.05 + 0.5 \cdot 0.3^2) \cdot 1}{0.3 \cdot \sqrt{1}} = \frac{0 + 0.095}{0.3} = 0.3167$$

$$d_2 = 0.3167 - 0.3 = 0.0167$$

$$C = 50 \cdot N(0.3167) - 50 \cdot e^{-0.05} \cdot N(0.0167)$$

- $N(0.3167) \approx 0.6246$
- $N(0.0167) \approx 0.5067$
- $e^{-0.05} \approx 0.9512$

$$C \approx 50 \cdot 0.6246 - 50 \cdot 0.9512 \cdot 0.5067 \approx 31.23 - 24.11 = 7.116$$

And for strike price=51, we can do similar steps:

$$C = 50 \cdot 0.5989 - 51 \cdot e^{-0.05} \cdot 0.4803$$

$$e^{-0.05} 0.9512$$

$$C \approx 29.945 - 51 \cdot 0.9512 \cdot 0.4803 \approx 29.945 - 23.298 = 6.646$$

6.2 Utilizing Excel

First we need to set up parameters like this:

	A	B
1	Initial Stock Price (S_0)	50
2	Risk-Free Rate (r)	5%
3	Volatility (σ)	30%
4	Time to Maturity (T)	1
5	Strike Price (K_1)	50
6	Strike Price (K_2)	51
7	Number of Paths	20,000
8	Steps per Path	250
9	Time Increment	0.004
10	Drift Term	0.00002
11	Volatility Term	0.018973666

Then in B9 we get Time Increment to be =B4/B8, in B10 we get Drift Term to be =(B2-0.5*B3^2)*B9, in B11, we have Volatility Term to be =B3*SQRT(B9).

To produce trajectories, we have =D2*EXP(B10 + B11*NORM.S.INV(RAND())) to spread from ET2 (250 steps) to 20001 row. Then, we have:

EU as Terminal Stock Price to be ET2

EV as Call Payoff (K=50) to be MAX(EU2 - B5, 0)

EW as Call Payoff (K=51) to be MAX(EU2 - B6, 0)

EX as Discounted Payoff (K=50) to be EV2 * EXP(-B2 * B4)

EY as Discounted Payoff (K=51) to be EW2 * EXP(-B2 * B4)

For the results using Monte Carlo method, we have B13 as MC Price (K=50) = AVERAGE(EX2:EX20001)

B14 as MC Price (K=51) to be AVERAGE(EY2:EY20001)

Finally, for theoretical BS price, we have

B16 as d1 (K=50) to be (LN(B1/B5)+(B2+0.5*B3^2)*B4)/(B3*SQRT(B4))

B17 as d2 (K=50) to be B16 - B3*SQRT(B4)

B18 d1 (K=51) to be (LN(B1/B6)+(B2+0.5*B3^2)*B4)/(B3*SQRT(B4))

B19 as d2 (K=51) to be B18 - B3*SQRT(B4)

B20 as BS Price (K=50) to be B1*NORM.S.DIST(B16,1) - B5*EXP(-B2*B4)*NORM.S.DIST(B17,1)

B21 as BS Price (K=51) to be B1*NORM.S.DIST(B18,1) - B6*EXP(-B2*B4)*NORM.S.DIST(B19,1), the results are 7.115627393 and 6.646252533, which are the same as we previously calculated.

And the 20,000 trajectory-MC-calculated results are 7.03062133 and 6.47033162 for K 50 and 51, and differences are 0.08500605 and 0.17592091 respectively. This is reasonable as Monte-Carlo method is stochastic, with randomness incorporated in it, and the number of traces it has will impact the overall accuracy of the model. In this case, 20,000 trajectory is already a large number thus the result is fairly accurate.

7 Question 7

Calculate with 40,000 trajectories. Compare Monte-Carlo price with 40,000 trajectories to theoretical model price. Calculate with 60,000 trajectories. Compare Monte-Carlo price with 60,000 trajectories to theoretical model price.

Answer

This is similar to Problem 6, just adding more rows.

For 40,000 trajectories, we have Monte-Carlo calculated results to be `7.094032435` and `6.62401321` for K 50 and 51, and differences are `0.02159495` and `0.02223932`.

For 60,000 trajectories, we have Monte-Carlo calculated results to be `7.01352395` and `6.54471058` for K 50 and 51, and differences are `0.10210344` and `0.10154195`.

As the number of trajectories increases, the accuracy increases.

O29   fx =N29*EXP(B10 + B11*NORM.S.INV(RAND()))

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Initial Stock Price (S_0)	50	trajectory\step	step_1	step_2	step_3	step_4	step_5	step_6	step_7	step_8	step_9	step_10	step_11	step_12	step_13
2	Risk-Free Rate (r)	5%	1	50	51.384487	52.05409	51.873616	51.4892	52.239649	51.391223	51.526437	51.493682	49.974101	49.89985	51.091802	52.56921
3	Volatility (σ)	30%	2	50	50.954248	51.275235	51.228717	51.206209	51.153	52.86213	51.875792	51.907586	52.389546	52.460561	54.512646	55.92721
4	Time to Maturity (T)	1	3	50	50.16779	50.291082	50.30978	48.864498	50.565492	52.149082	53.97477	54.044662	53.028194	50.730839	49.919121	49.96924
5	Strike Price (K_1)	50	4	50	49.689971	47.173887	47.498708	46.726144	46.531922	45.395948	45.933776	46.638151	47.166897	46.797592	46.757453	44.85003
6	Strike Price (K_2)	51	5	50	49.534354	48.418429	48.371972	49.537814	48.004095	48.343748	47.920743	48.083853	47.149432	46.008914	45.575876	45.00157
7	Number of Paths	20,000	6	50	48.238642	48.350725	48.474674	47.876032	46.721525	47.779992	47.110345	49.266787	49.060788	49.168248	47.961698	48.58500
8	Steps per Path	250	7	50	48.138777	47.54388	47.293563	46.719511	46.165995	45.539548	45.08743	44.829458	44.949422	45.002344	44.977089	45.24787
9	Time Increment	0.004	8	50	49.335169	48.868232	48.880983	48.843444	48.006331	49.59753	50.11856	51.038786	49.528735	49.746172	48.82565	47.90012
10	Drift Term	0.00002	9	50	49.690843	49.307452	48.449965	48.012118	47.20992	45.930971	45.424492	46.017804	46.536801	43.46613	44.639347	43.83923
11	Volatility Term	0.018973666	10	50	50.029031	49.049315	48.857142	49.404502	49.08253	49.439181	49.086678	49.899651	51.153955	50.190867	51.579928	52.33587
12			11	50	51.447412	52.601347	52.645392	52.441885	52.58768	51.192425	52.662805	52.147027	52.813632	51.83971	52.12745	51.20783
13	MC Price (K=50)	7.013523952	12	50	49.008	47.821272	46.877222	45.217563	46.038915	45.251901	44.987752	42.699301	42.891954	43.075147	41.837599	41.90540
14	MC Price (K=51)	6.544710583	13	50	50.965385	51.525631	52.047904	51.5199	51.101707	49.578757	50.865063	50.836402	48.751979	49.189028	50.690159	51.2281
15			14	50	49.439889	49.702275	49.541485	48.809993	48.9811	49.417414	50.767715	51.73264	50.97187	49.609928	49.834773	49.3106
16	d_1 (K=50)	0.316666667	15	50	48.721854	48.368563	46.983901	48.674269	47.457881	47.103876	45.614241	46.554632	47.438867	48.751556	47.18198	46.2714
17	d_2 (K=50)	0.016666667	16	50	50.799678	49.929836	51.449775	50.698388	50.923447	50.186387	50.730164	51.221481	50.844494	51.111511	53.363657	52.98972
18	d_1 (K=51)	0.250657909	17	50	50.62653	48.846872	49.321015	47.741942	46.745885	47.384408	47.672195	48.778893	48.096648	48.439887	48.145733	49.17784
19	d_2 (K=51)	-0.04934209	18	50	51.706239	51.237139	51.038328	50.695119	51.367504	51.866208	51.387056	51.188987	49.913193	49.033354	49.627589	50.97359
20	BS Price (K=50)	7.115627393	19	50	52.270382	51.542738	52.257207	52.097256	53.549161	51.832113	52.753164	52.731629	53.044094	51.851764	52.313349	50.38790
21	BS Price (K=51)	6.646252533	20	50	50.368349	52.097606	53.2038	53.072042	53.850593	54.482938	55.259789	54.356678	54.850747	54.202522	54.342129	52.97745
22			21	50	50.919529	48.330988	49.986004	50.426315	49.778355	51.114672	51.936655	51.118282	49.808986	48.429525	48.915516	49.64924
23			22	50	49.418998	48.489418	48.714924	47.611482	47.18814	48.975026	49.67209	49.837101	48.524671	48.630893	50.380946	48.69367
24			23	50	50.185366	48.398768	48.21717	48.805346	48.773272	47.70011	46.680852	48.285976	47.775587	47.664668	46.771524	48.09696
25	Diff	0.102103441	24	50	50.499183	50.688496	51.545586	49.504153	50.173673	49.996431	50.250018	49.336554	50.670055	51.331381	51.367577	51.17937
26	Diff	0.10154195	25	50	51.932179	53.120656	52.314455	51.670378	50.539661	50.722761	51.107239	50.164752	49.305278	50.361412	49.104251	48.92560
27			26	50	50.321601	50.80097	52.117321	52.173556	52.885043	52.423997	53.126046	53.558558	53.358291	52.459429	51.840405	52.18721
28			27	50	51.252851	51.466765	52.155399	51.937238	51.521152	51.362128	50.920501	50.552418	51.847967	51.395678	49.995321	50.56391
29			28	50	50.520873	49.371219	49.160613	49.550228	49.209352	49.846577	51.094525	50.754342	50.909502	51.672838	52.022914	51.90389
30			29	50	51.159608	53.699646	53.860905	54.595572	56.275528	57.032187	55.532523	55.212017	55.439501	56.075665	54.254382	53.42391
31			30	50	49.838637	51.12451	50.889432	50.510432	50.105501	49.927163	51.112576	51.728723	49.892084	50.66276	50.050357	50.7994

8 Question 8

Create Matlab code doing the same thing as problems 6 and 7. Compare Matlab and spreadsheet results.

Answer

The Matlab code is as follows:

```
% Monte Carlo simulation for European call option using GBM

% Parameters
S0 = 50;        % Initial stock price
r = 0.05;       % Risk-free rate
sigma = 0.3;    % Volatility
T = 1;          % Time to maturity in years
K1 = 50;        % Strike 1
K2 = 51;        % Strike 2
Nsteps = 250;   % Number of time steps
dt = T / Nsteps;

% Number of trajectories to simulate
N_paths_list = [500, 20000, 40000, 60000];

% Black-Scholes price function
bs_call = @(S,K,T,r,sigma) S * normcdf((log(S/K) + (r + 0.5*sigma^2)*T) / (sigma*sqrt(T)) ...
- K * exp(-r*T) * normcdf((log(S/K) + (r - 0.5*sigma^2)*T) / (sigma*sqrt(T)));

% Theoretical prices
BS_K1 = bs_call(S0, K1, T, r, sigma);
BS_K2 = bs_call(S0, K2, T, r, sigma);

fprintf('Black-Scholes Prices:\nK = 50: %.4f\nK = 51: %.4f\n\n', BS_K1, BS_K2);

% Loop over path counts
for i = 1:length(N_paths_list)
M = N_paths_list(i);

% Generate standard normal random numbers
Z = randn(M, Nsteps);

% Simulate log prices using GBM
logS = log(S0) + cumsum((r - 0.5*sigma^2)*dt + sigma*sqrt(dt).*Z, 2);
ST = exp(logS(:, end));  % Final prices

% Payoffs
payoff_K1 = max(ST - K1, 0);
payoff_K2 = max(ST - K2, 0);

% Monte Carlo prices (discounted)
MC_K1 = exp(-r*T) * mean(payoff_K1);
MC_K2 = exp(-r*T) * mean(payoff_K2);

% Print results
fprintf('Monte Carlo with %d paths:\n', M);
fprintf('  Call(K=50): %.4f | Differences: %.4f\n', MC_K1, MC_K1 - BS_K1);
fprintf('  Call(K=51): %.4f | Differences: %.4f\n\n', MC_K2, MC_K2 - BS_K2);
end
```

And the results are:

```
Black-Scholes Prices:
K = 50: 7.1156
K = 51: 6.6463

Monte Carlo with 500 paths:
Call(K=50): 6.8674 | Differences: -0.2482
Call(K=51): 6.4115 | Differences: -0.2348

Monte Carlo with 20000 paths:
Call(K=50): 7.0254 | Differences: -0.0902
Call(K=51): 6.5610 | Differences: -0.0852

Monte Carlo with 40000 paths:
Call(K=50): 7.0489 | Differences: -0.0667
Call(K=51): 6.5827 | Differences: -0.0635

Monte Carlo with 60000 paths:
Call(K=50): 7.1175 | Differences: 0.0018
Call(K=51): 6.6471 | Differences: 0.0009
```

We can see that as the number of trajectories increases, the difference between the Monte Carlo and theoretical values will reduce.

I also used python in this problem,

```python
import numpy as np
from scipy.stats import norm

# Parameters
S0 = 50         # Initial stock price
r = 0.05        # Risk-free rate
sigma = 0.30    # Volatility
T = 1           # Time to maturity
K1 = 50         # Strike 1
K2 = 51         # Strike 2
Nsteps = 250    # Number of time steps
dt = T / Nsteps

# Function: Black-Scholes Call Price
def black_scholes_call(S, K, T, r, sigma):
d1 = (np.log(S / K) + (r + 0.5 * sigma**2)*T) / (sigma * np.sqrt(T))
d2 = d1 - sigma * np.sqrt(T)
return S * norm.cdf(d1) - K * np.exp(-r*T) * norm.cdf(d2)

# Theoretical prices
BS_K1 = black_scholes_call(S0, K1, T, r, sigma)
BS_K2 = black_scholes_call(S0, K2, T, r, sigma)

print(f"Black-Scholes Prices:\nK=50: {BS_K1:.4f}\nK=51: {BS_K2:.4f}\n")

# Monte Carlo simulation with different path counts
for M in [20000, 40000, 60000]:
# Generate standard normal random matrix
Z = np.random.normal(0, 1, (M, Nsteps))

# Simulate log-price paths
logS = np.log(S0) + np.cumsum((r - 0.5 * sigma**2) * dt + sigma * np.sqrt(dt) * Z,
    axis=1)
ST = np.exp(logS[:, -1])

# Payoffs
payoff_K1 = np.maximum(ST - K1, 0)
payoff_K2 = np.maximum(ST - K2, 0)

```

```python
    # Monte Carlo Price (discounted)
    MC_K1 = np.exp(-r * T) * np.mean(payoff_K1)
    MC_K2 = np.exp(-r * T) * np.mean(payoff_K2)

    print(f"Monte Carlo with {M} paths:")
    print(f"  Call(K=50): {MC_K1:.4f} | Differences: {MC_K1 - BS_K1:.4f}")
    print(f"  Call(K=51): {MC_K2:.4f} | Differences: {MC_K2 - BS_K2:.4f}\n")
```

And results are:

```
Monte Carlo with 20000 paths:
Call(K=50): 7.0852 | Differences: -0.0304
Call(K=51): 6.6224 | Differences: -0.0238

Monte Carlo with 40000 paths:
Call(K=50): 7.1047 | Differences: -0.0110
Call(K=51): 6.6349 | Differences: -0.0114

Monte Carlo with 60000 paths:
Call(K=50): 7.0917 | Differences: -0.0240
Call(K=51): 6.6267 | Differences: -0.0195
```

The answers are pretty similar, and the tendencies are the same: as the more trajectories, the more precise. By the Law of Large Numbers (LLN), as the number of independent simulations (N) increases, the sample average converges to the true expected value.