MIE1622 Assignment 4

Report

By: Qian Zhang

1000062736

April 11, 2019

1. **MATLAB Code**

clc;

clear all;

format long

% Pricing a European option using Black-Scholes formula and Monte Carlo simulations

% Pricing a Barrier option using Monte Carlo simulations

S0 = 100; % spot price of the underlying stock today

K = 105; % strike at expiry

mu = 0.05; % expected return

sigma = 0.2; % volatility

r = 0.05; % risk-free rate

T = 1.0; % years to expiry

Sb = 110; % barrier

% Define variable numSteps to be the number of steps for multi-step MC

% numPaths - number of sample paths used in simulations

numPaths = 10000;

numSteps = 252;

% Implement your Black-Scholes pricing formula

[call\_BS\_European\_Price, putBS\_European\_Price] = BS\_european\_price(S0, K, T, r, sigma);

% Implement your one-step Monte Carlo pricing procedure for European option

% numSteps = 1;

[callMC\_European\_Price\_1\_step, putMC\_European\_Price\_1\_step] = MC\_european\_price(S0, K, T, r, mu, sigma, 1, numPaths);

% Implement your multi-step Monte Carlo pricing procedure for European option

[callMC\_European\_Price\_multi\_step, putMC\_European\_Price\_multi\_step, MCpaths] = MC\_european\_price(S0, K, T, r, mu, sigma, numSteps, numPaths);

% Implement your one-step Monte Carlo pricing procedure for Barrier option

% numSteps = 1;

[callMC\_Barrier\_Knockin\_Price\_1\_step, putMC\_Barrier\_Knockin\_Price\_1\_step] = ...

MC\_barrier\_knockin\_price(S0, Sb, K, T, r, mu, sigma, 1, numPaths);

% Implement your multi-step Monte Carlo pricing procedure for Barrier option

[callMC\_Barrier\_Knockin\_Price\_multi\_step, putMC\_Barrier\_Knockin\_Price\_multi\_step] = ...

MC\_barrier\_knockin\_price(S0, Sb, K, T, r, mu, sigma, numSteps, numPaths);

disp(['Black-Scholes price of an European call option is ',num2str(call\_BS\_European\_Price)])

disp(['Black-Scholes price of an European put option is ',num2str(putBS\_European\_Price)])

disp(['One-step MC price of an European call option is ',num2str(callMC\_European\_Price\_1\_step)])

disp(['One-step MC price of an European put option is ',num2str(putMC\_European\_Price\_1\_step)])

disp(['Multi-step MC price of an European call option is ',num2str(callMC\_European\_Price\_multi\_step)])

disp(['Multi-step MC price of an European put option is ',num2str(putMC\_European\_Price\_multi\_step)])

disp(['One-step MC price of an Barrier call option is ',num2str(callMC\_Barrier\_Knockin\_Price\_1\_step)])

disp(['One-step MC price of an Barrier put option is ',num2str(putMC\_Barrier\_Knockin\_Price\_1\_step)])

disp(['Multi-step MC price of an Barrier call option is ',num2str(callMC\_Barrier\_Knockin\_Price\_multi\_step)])

disp(['Multi-step MC price of an Barrier put option is ',num2str(putMC\_Barrier\_Knockin\_Price\_multi\_step)])

% Plot results

figure(1);

%%%%%%%%%%% Insert your code here %%%%%%%%%%%%

for i=1:numPaths

plot(1:numSteps+1,MCpaths(:,i));

hold on;

end

hold off;

axis([0 numSteps+1 30 inf])

title('Underlying Stock Price Simulations');

%% Black-Scholes Equation

function [call\_BS\_European\_Price, putBS\_European\_Price] = BS\_european\_price(S0, K, T, r, sigma)

t = 0;

d1 = (1/sigma\*sqrt(T-t)) \* (log(S0/K) + (r+sigma^2/2)\*(T-t));

d2 = d1 - sigma\*sqrt(T-t);

call\_BS\_European\_Price = normcdf(d1)\*S0 - normcdf(d2)\*K\*exp(-r\*(T-t));

putBS\_European\_Price = normcdf(-d2)\*K\*exp(-r\*(T-t)) - normcdf(-d1)\*S0;

end

%% Multi-Step MC

function [callMC\_European\_Price\_multi\_step, putMC\_European\_Price\_multi\_step, paths] = MC\_european\_price(S0, K, T, r, mu, sigma, numSteps, numPaths)

paths = zeros(numSteps+1, numPaths);

dT = T/numSteps;

paths(1,:) = S0;

% Generates paths with the corresponding steps

for iPath = 1:numPaths

for iStep = 1:numSteps

paths(iStep+1, iPath) = paths(iStep, iPath) \* exp((mu - 0.5\*sigma^2)\*dT + sigma\*sqrt(dT)\*normrnd(0,1));

end

end

call = zeros(numPaths,1);

put = zeros(numPaths,1);

% Calculates Call and Put prices for each path

for iPath = 1:numPaths

call(iPath,1) = max(paths(numSteps+1,iPath) - K, 0) \* exp(-r\*T);

put(iPath,1) = max(K - paths(numSteps+1,iPath), 0) \* exp(-r\*T);

end

% Price of the option is the average of all paths

callMC\_European\_Price\_multi\_step = mean(call);

putMC\_European\_Price\_multi\_step = mean(put);

end

%% Multi-Step MC Barrier Knock-in

% The option becomes a standard option if the barrier price was crossed

% somtime before expiration T

function [callMC\_Barrier\_Knockin\_Price\_multi\_step, putMC\_Barrier\_Knockin\_Price\_multi\_step] = ...

MC\_barrier\_knockin\_price(S0, Sb, K, T, r, mu, sigma, numSteps, numPaths)

paths = zeros(numSteps+1, numPaths);

dT = T/numSteps;

paths(1,:) = S0;

% Generates paths with the corresponding steps

for iPath = 1:numPaths

for iStep = 1:numSteps

paths(iStep+1, iPath) = paths(iStep, iPath) \* exp((mu - 0.5\*sigma^2)\*dT + sigma\*sqrt(dT)\*normrnd(0,1));

end

end

call = zeros(numPaths,1);

put = zeros(numPaths,1);

% Calculates Call and Put prices for each path based on the barrier

% Option will be valid if price at any point before expiration

% crosses the barrier price Sb

for iPath = 1:numPaths

if any(paths(:,iPath) > Sb)

call(iPath,1) = max(paths(numSteps+1,iPath) - K, 0) \* exp(-r\*T);

put(iPath,1) = max(K - paths(numSteps+1,iPath), 0) \* exp(-r\*T);

else

call(iPath,1) = 0;

put(iPath,1) = 0;

end

end

% Price of the option is the average of all paths

callMC\_Barrier\_Knockin\_Price\_multi\_step = mean(call);

putMC\_Barrier\_Knockin\_Price\_multi\_step = mean(put);

end

1. **Analyzing Results**

**2.1. MATLAB Output**

Black-Scholes price of an European call option is 8.0214

Black-Scholes price of an European put option is 7.9004

One-step MC price of an European call option is 7.9548

One-step MC price of an European put option is 7.9078

Multi-step MC price of an European call option is 8.0165

Multi-step MC price of an European put option is 7.8281

One-step MC price of an Barrier call option is 7.7939

One-step MC price of an Barrier put option is 0

Multi-step MC price of an Barrier call option is 7.9722

Multi-step MC price of an Barrier put option is 1.2884

The multi-step Monte Carlo output above is generated based on 50,000 paths and 12 steps, which are chosen for the close proximity between Monte Carlo the Black-Scholes call and put results, as well as lower computational times. The use of 12 steps is also reasonable as prices fluctuate on a monthly basis.

**2.2. MATLAB Plot of the underlying price using MC multi-step**

A close up of a map

Description automatically generated

Figure 1. M.C. multi-step simulation (underlying price Vs. steps) using 50,000 paths and 12 steps

**2.3. Compare three pricing strategies for European option and discuss their performance relative to each other.**

The Black Scholes model is based on fixed inputs (current stock price, strike price, time until expiration, volatility, risk free rate, and dividend yield), and performs well in the calculation of standard European options. However, since the model relies on fixed inputs, it does not allow for flexibility to add in non-standard features, such as barrier options. On the other hand, the Geometric Brownian Motion utilizes constant drift and volatility of the pricing model and is discretized with the Geometric Random Walk model. This model allows for the incorporation of time steps and thus allowing for more flexible input requirements, but its accuracy is dependent on the number of steps and paths used, which can be demanding on both the computational power and time. The Barrier Option in this assigned is the up-and-in option (a knock-in option with the barrier above the spot price), which is a special version of the European Option. Similar to the standard MC multi-step simulation, the Barrier option requires high computational iterations to achieve high accuracy. However, the up-and-in option is biased towards the call option in terms of option prices.

**2.4.** **Explain the difference between call and put prices obtained for European and Barrier options.**

As seen in the MATLAB outputs in section 2.1, the put price of a barrier option is much lower than that of the call price. Since this is an up-and-in barrier option, an option only becomes standard when it surpasses the barrier price $110. In the case of a put option, the higher the final price, the lower the yield of the option. If the final price is higher than the strike price, the yield of the put option is zero. As seen in Figure 1, options that have crossed the barrier are in the upper portion of the price range. This means that the price at expiry of these put options are higher if not closer to the strike price ($105) due to the existence of a barrier, resulting in yields that are either zero or close to zero. The low yield contributes to the low pricing of the put option.

**2.5. Compute prices of Barrier options with volatility increased and decreased by 10% from the original inputs. Explain the results.**

**Barrier pricing after increasing volatility by 10% at sigma = 30%:**

One-step MC price of an Barrier call option is 11.7409

One-step MC price of an Barrier put option is 0

Multi-step MC price of an Barrier call option is 11.8712

Multi-step MC price of an Barrier put option is 3.0527

**Barrier pricing after decreasing volatility by 10% at sigma = 10%::**

One-step MC price of an Barrier call option is 3.6187

One-step MC price of an Barrier put option is 0

Multi-step MC price of an Barrier call option is 3.8162

Multi-step MC price of an Barrier put option is 0.12874

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Barrier Pricing | Volatility | | | % Difference | % Difference |
| 10% | 20% | 30% |
| One-Step Call ($) | 3.6187 | 7.7939 | 11.7409 | -53.6% | 50.6% |
| One-Step Put ($) | 0 | 0 | 0 | 0.0% | 0.0% |
| Multi-Step Call ($) | 3.8162 | 7.9722 | 11.8712 | -52.1% | 48.9% |
| Multi-Step Put ($) | 0.12874 | 1.2884 | 3.0527 | -90.0% | 136.9% |

Table 1. Barrier pricing and corresponding volatility values.

Pricing volatility is the degree of movement the option price exhibits before it is exercised. As seen in Table 1, the prices of the barrier options are directly related to changes in volatility. This is expected since a higher volatility represents higher yield potential of the option in the future, and thus a higher price premium.

1. **Discuss possible strategies to obtain the same prices from two procedures.**

As discussed in the sections above, the accuracy of the MC simulation is dependent on the number of iterations (paths and steps) performed. Essentially, the number of steps accounts for the level of discretization of time, and the number of paths represents the level of precision accounting for multiple outcomes and random errors. To design a procedure in selecting the ideal number of iterations, the procedure needs to calculate a span of steps (1 to 252), paths (1 to 100,000 or more), their corresponding prices, as well as the difference between MC and BS. This procedure would take an immense amount of computational power and time to run. However, specific combinations have been calculated and shown in the table below.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| # of Steps | # of Paths | Call (M.C.) | Put (M.C.) | Call (B.S.) | Put (B.S.) | Call Diff. | Put Diff. |
| 6 | 10,000 | $7.89 | $8.06 | $8.02 | $7.90 | $0.13 | $0.16 |
| 6 | 50,000 | $8.04 | $7.88 | $8.02 | $7.90 | $0.02 | $0.02 |
| 6 | 100,000 | $8.02 | $7.88 | $8.02 | $7.90 | $0.00 | $0.02 |
| 12 | 10,000 | $7.88 | $7.88 | $8.02 | $7.90 | $0.14 | $0.02 |
| 12 | 50,000 | $8.15 | $7.88 | $8.02 | $7.90 | $0.13 | $0.02 |
| 12 | 100,000 | $8.00 | $7.89 | $8.02 | $7.90 | $0.02 | $0.01 |
| 8 | 10,000 | $7.97 | $7.86 | $8.02 | $7.90 | $0.05 | $0.04 |
| 8 | 50,000 | $8.00 | $7.92 | $8.02 | $7.90 | $0.02 | $0.02 |
| 8 | 100,000 | $8.03 | $7.90 | $8.02 | $7.90 | $0.01 | $0.00 |
| 4 | 10,000 | $8.17 | $7.80 | $8.02 | $7.90 | $0.15 | $0.10 |
| 4 | 50,000 | $7.99 | $7.93 | $8.02 | $7.90 | $0.03 | $0.03 |
| 4 | 100,000 | $7.83 | $7.87 | $8.02 | $7.90 | $0.19 | $0.03 |
| 252 | 10,000 | $7.83 | $7.87 | $8.02 | $7.90 | $0.19 | $0.03 |
| 252 | 50,000 | $8.15 | $7.96 | $8.02 | $7.90 | $0.13 | $0.06 |
| 252 | 100,000 | $8.01 | $7.90 | $8.02 | $7.90 | $0.01 | $0.00 |

Table 2. Combinations of number of steps and paths and their corresponding prices

As seen in Table 2, a greater number of paths can significantly reduce the difference between MC pricing and BS pricing. The number of steps also help with reducing the differences, to a lesser extent. From the results in Table 2, it can be generally concluded that a number of steps higher than 6, and a number of scenarios higher than 100,000 is sufficient to produce a MC pricing model that is representative of the BS model.