%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% FRE 6251 Numerical and Simulation Techniques in Finance

% Assignment #5

% Name: Surya L Gurung ID: 0449604

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Brief Reports of Assignment #5: Using Importance Sampling for Asian Call Option

(1). This simulation have five matlab files, four functions and one script files.

Function **“bisection**” gives a root of function using bisection method.

Function “**OptimalDrift**” calls fuction “bisection” root “y” of the equation

and each y gives values of from

and .

The vector Z is set to a vector **µ.** Value of S( is obtained from equation

.

Function “**MuAssetPaths**”, calls function ”**OptimalDrift**” and uses vector **µ** to get generate Z~N(µ,1) and returns **µ, Z, and asset paths matrix.**

Function “ ImportanceSampling” calls function “MuAssetPaths” and uses following Importance Sampling algorithm to get vector of option prices.

for i= 1,…. N

generate Z ~ N(µ, I)\

Yi 🡨 G(Z)

return (Y1+Y2+ ….. + YN )/N

The Matlab script file “**ImportanceSamplingMCS**” is the main file need to call from the Matlab command prompt. It calls “**ImportanceSampling**” function for different number of paths, up to 1,000,000 paths, in the multiple of 10. But I could not go beyond 10,000 paths because it gave out of memory error message.

I used example of an asset with following parameters and values:

Example asset:

So = 100;

K = 110;

r = 0.05;

sigma = 0.2;

T = 1;

cORp = 1; % this is for call option

tSteps = 20;

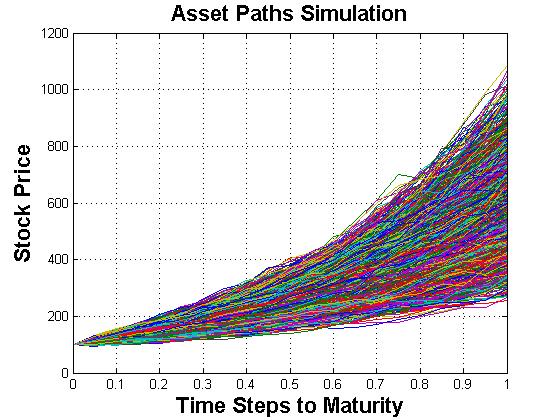
pNum = 100;

row = 1;

Conclusion:

Even though, I am able to solve the optimization equation to get µ = , I am not sure that the final result is as it should be. Please take a look at function for more detail

Results of running “ImportanceSamplingMCS” included:



**The result of running the Matlab script file “ImportanceSamplingMCS.m” for upto 1,000,000 paths.**

>> ImportanceSamplingMCS

Result Interpretation:

Paths Option Price error

---------- ----------- ------

100 0.009314 0.018251

1000 0.205050 0.333828

10000 0.852193 0.717019

??? Error using ==> mtimes

Out of memory. Type HELP MEMORY for your options.

% ImportanceSamplingMCS.m

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% FRE 6251 Numerical and Simulation Techniques in Finance

% Assignment #5

% Name: Surya L Gurung ID: 0449604

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Monte Carlo Simulation of Asian Option Importance Sampling

%-----------------------------------------------------------

% Parameter Description:

% So = initial price of asset

% K = strike price

% r = risk-free rate

% sigma = standard deviation of asset

% T = time to maturity in year

% cORp = option type, 1 for call and -1 for put put option

% tSteps = number of time steps

% pNum = number of paths

%-----------

% Example 1:

%-----------

So = 100;

K = 110;

r = 0.05;

sigma = 0.2;

T = 1;

cORp = 1; % this is for call option

tSteps = 20;

pNum = 100;

row = 1;

disp('Result Interpretation:')

disp(' ')

disp(sprintf('%s', ' Paths Option Price error' ))

disp(sprintf('%s', ' ---------- ----------- ------' ))

while pNum < 100000

[Y, error1] = ImportanceSampling(So, K, r, sigma, T, cORp, tSteps, pNum);

if row == 1

disp(sprintf(' %d %0.6f %0.6f', pNum, Y, error1))

elseif row == 2

disp(sprintf(' %d %0.6f %0.6f', pNum, Y, error1))

elseif row == 3

disp(sprintf(' %d %0.6f %0.6f', pNum, Y, error1))

elseif row == 4

disp(sprintf(' %d %0.6f %0.6f %0.6f', pNum, Y, error1))

elseif row == 5

disp(sprintf(' %d %0.6f %0.6f %0.6f ', pNum, Y, error1))

elseif row == 6

disp(sprintf(' %d %0.6f %0.6f %0.6f', pNum, Y, error1))

end

pNum = pNum \* 10;

row = row + 1;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% FRE 6251 Numerical and Simulation Techniques in Finance

% Assignment #5

% Name: Surya L Gurung ID: 0449604

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Y, error1] = ImportanceSampling(So, K, r, sigma, T, cORp, tSteps, pNum)

% Parameter Description:

% So = initial price of asset

% K = strike price

% r = risk-free rate

% sigma = standard deviation of asset

% T = time to maturity in year

% cORp = option type, 1 for call and -1 for put put option

% tSteps = number of time steps

% pNum = number of paths

format long g

dT = T/tSteps;

syms y

%matrix of asset paths, each column represents one path

[stockPaths, mu, Z] = MuAssetPaths(So, K, r, sigma, T, tSteps, pNum);

%discounted payoff of arithmetic average asian option

Gz = exp(-r \* T) \* max(cORp \*(mean(stockPaths)- K), 0);

Yi = Gz \* exp(-mu' \* Z +0.5 \* mu' \* mu);

Y = mean(Yi);

error1 = 1.96 \* (std(Yi)/sqrt(pNum));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% FRE 6251 Numerical and Simulation Techniques in Finance

% Assignment #5

% Name: Surya L Gurung ID: 0449604

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [mu] = OptimalDrift(So, K, r, sigma, T, tSteps)

% Parameter Description:

% So = initial price of asset

% K = strike price

% r = risk-free rate

% sigma = standard deviation of asset

% T = time to maturity in year

% cORp = option type, 1 for call and -1 for put put option

% tSteps = number of time steps

% pNum = number of paths

format long g

dT = T/tSteps;

syms y

Zp = 0;

Sp = So;

sum = 0;

m = 2;

const1 = (r - 0.5 \* sigma^2) \* dT ;

const2 = sigma \* sqrt(dT);

Z1 = @(y) (sigma\*sqrt(dT)\*(y + K))/y;

f1 = @(y) (So\*exp(const1 + ((const2\*sigma\*sqrt(dT)\*(y + K))/y))) - K - y;

sPrice = @(y) Sp\*exp(const1 + const2\*y);

y = bisection(f1, 0, 100, .0000001, .0000001);

Z(1) = Z1(y);

S1 = sPrice(Z(1));

sum = sum + S1;

Sp = S1;

Zp = Z(1);

for i = 2:tSteps

m = i;

sPrice = @(y) Sp\*exp(const1 + const2\*y);

Zj = @(y) Zp - ((sigma \* sqrt(dT)\* Sp)./(m\*y));

f2 = @(y) ((sum + Sp\*exp(const1 + const2\*(Zp - (sigma\*sqrt(dT)\*Sp/(m\*y)))))/m) - K -y;

y = bisection(f2, -1, 1000, .0000001, .0000001);

Z(i) = Zj(y);

Sp = sPrice(Z(i));

Zp = Z(i);

sum = sum + Sp;

end

mu = Z;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% FRE 6251 Numerical and Simulation Techniques in Finance

% Name: Surya L Gurung ID: 0449604

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Matrix of asset prices, one column representing one path.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [paths, mu, Z] = MuAssetPaths(So,K,r,sigma, T, tSteps,pNum)

% Parameters:

% So = initial asset price

% mu = expected return

% sigma = volatility

% T = maturity time in years

% tSteps = number of time steps

% pNum = number of paths

%

dT = T/tSteps; % size of one time step

drift = r - (0.5 \* sigma^2); % drift of process

%z = randn(tSteps, pNum); % matrix of normally distributed random values

newMu = OptimalDrift(So, K, r, sigma, T, tSteps);

mu = zeros(tSteps, pNum);

one = ones(tSteps, pNum);

for i = 1: pNum

mu(:,i) = newMu';

end

Z = normrnd(mu, one);

% matrix of all paths

paths = So\*[ones(1,pNum); cumprod(exp(drift\*dT + sigma \* sqrt(dT)\* Z),1)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% FRE 6251 Numerical and Simulation Techniques in Finance

% Assignment #5

% Name: Surya L Gurung ID: 0449604

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [zero, it] = bisection(fn,a, b, hTol, vTol)

m = (a+b)/2;

xTol = 1;

yTol = 1;

iteration = 0;

if sign(fn(a)) ~= sign(fn(b))

while xTol > hTol & yTol > vTol

if sign(fn(m)) == 0

xTol = 0;

yTol = 0;

else

if sign(fn(a)) == sign(fn(m))

a = m;

else

b = m;

end

xTol = abs(b - a);

yTol = abs(fn(b) - fn(a));

end

m = (a+b)/2;

iteration = iteration + 1;

end

zero = m;

it = iteration;

else

zero = 'Limits are not appropriate.';

it = 0;

end