**Computer Assignment 3. Price of a European Call Option**

1. **Black-Scholes Option Pricing Formula**

Use the Black-Scholes option pricing formula to calculate the price of a European call option with a strike price of $ 50 and an expiry date of 3 months. The current share price is $ 52, the interest rate is 12 % and volatility is 30 %.

Clear all objects from the workspace.

S0 <- 52 # Current share price

X <- 50 # Strike price

E <- 0.25 # Expiry date (fraction of year)

r <- 0.12 # Interest rate

sigma <- 0.3 # Volatility (standard deviation)

d1 <- (log(S0/X) + (r+sigma^2/2) \* E) / (sigma \* sqrt(E))

d2 <- d1 - sigma \* sqrt(E)

call <- S0 \* pnorm(d1) - X \* exp(1)^(-r \* E) \* pnorm(d2)

call

1. **Three Simulation of the Time Path of the Price of a Share**

Simulate the time path of the price of a share that follows a geometric Brownian motion. The initial share price is $ 52, the drift is 12 %, volatility is 30 % and the time horizon is 3 months. Divide one year into 10,000 intervals.

Clear all objects from the workspace.

S0 <- 52

X <- 50

E <- 0.25

t <- 10000 # Number of intervals within year

dt <- 1/t # Length of interval (fraction of year)

steps <- E\*t # Number of simulation steps

r <- 0.12

sigma <- 0.3

drift <- r

S <- rep(NA, steps) # Create vector with NA elements.

S[1] <- S0 # Set first element equal S0.

W <- rnorm(steps) # Create vector with N(0,1) random numbers.

for (i in 2:steps) {

 S[i] <- S[i-1]+drift\*S[i-1]\*dt+sigma\*S[i-1]\*sqrt(dt)\*W[i-1]

}

S <- ts(S, start=0, deltat = dt) # Create time series.

plot(S)

abline(h=X, lty=2)

Repeat the simulation three times and either submit three charts with the simulated time paths or put them into a single chart. What is the payoff of the option at the expiry date for each simulation?

1. **Simulation of the Call Option Price**

Conduct 5000 simulations of the time path of the share price and determine the average payoff of the option at the expiry date. The discounted average payoff is the call option price.

Clear all objects from the workspace.

S0 < - 52

X <- 50

E <- 0.25

t <- 10000

dt <- 1/t

steps <- E\*t

n <- 5000 # Number of simulations

r <- 0.12

sigma <- 0.3

drift <- r

simmx <- matrix(NA, nrow=steps, ncol=n) # Create matrix with NA.

for (j in 1:n) {

 S <- rep(NA, steps)

 S[1] <- S0

 W <- rnorm(steps)

 for (i in 2:steps) {

 S[i] <- S[i-1]+drift\*S[i-1]\*dt+sigma\*S[i-1]\*sqrt(dt)\*W[i-1]

 }

 simmx[,j] <- S # Store simulations column by column in simmx.

}

St <- simmx[steps,] # Last row of simmx

payoff <- St - X # Deviations of St from strike price

noex <- payoff <= 0 # No exercise of call option if true

payoff[noex] <- 0 # Option payoff (positive or zero)

call <- (exp(-r\*E))\*mean(payoff) # Discounted average payoff

call

Use the red button in the console window of RStudio to stop the simulation if it takes more than two minutes.

Perform the simulation three times and submit the simulated option price for each simulation. Compare the simulated option prices with that given by the Black-Scholes formula.