MA354: Computational Finance

Name : Prem Swarup
Roll : 2003318
Branch : MnC, 2020

Q.1.

(i) Algorithm of Binomial-tree method for pricing two-asset european style options.

Step1: At first we take input for all required parameters like initial asset prices 'S_1' & 'S_2', strike price 'K', volatality corresponding to asset 1 'σ_1' & corr. to asset 2 'σ_2'. correlation 'ρ', Time to expiration 'T'. no. of steps in time 'N' & risk-free interest rate (r).

Step2: Then we calculate the factors by which assets can go up or down in a time step as.

$$u_1 = e^{\sigma_1 \sqrt{dT}} \quad , \quad u_2 = e^{\sigma_2 \sqrt{dT}}$$

$$d_1 = \frac{1}{u_1} \quad , \quad d_2 = \frac{1}{u_2}$$

Step 3: Then we calculate asset prices at maturity time for all possible ups & downs as:

$$S_1(t_n) = \{ S_{10} \times u^i \cdot d^{n-i} \} \quad \text{for } i \in \{0, 1, 2, \ldots, N\}$$

Step 4: Then we calculate pay-off corresponding various configurations of final asset prices.

<u>Step 5</u> : Then we calculate probabilities of assets going up & down corresponding to their correlation as :

Probability that price of both asset will go up $(P_{uu}) = \frac{(1+\rho)}{4}$

Probability that price of one asset will go up & price of other asset will go down $= P_{ud} = P_{du} = \frac{1}{4}(1-\rho)$

Probability that price of both asset will go down $= \frac{1}{4}(1+\rho)$

<u>Step 6</u> : Now, iterating through each time steps from final to initial time, we will calculate option price under risk-neutrality assumption as

$$V_{t_i}^{(j,k)} \cdot e^{r\,dt} = \mathbb{E}\left[V_{t_{i+1}} \,\middle|\, V_{t_i}^{(j,k)} \right]$$

$$= V_{t_i}^{(j,k)} = \mathbb{E}\left[V_{t_{i+1}} \,\middle|\, V_{t_i}^{(j,k)} \right] \cdot e^{-r\,dt}$$

(where t_i represents time step & j, k represents ups in assets

i.e, expected price at t_{i+1}th step if price at t_i step is $V_{t_i}^{(j,k)}$, multiplied by $e^{-r\,dt}$

<u>Step 7</u> : Then final price at initial time step will give the price for two-asset European style option at start.

(iii) Algorithm of backward-finite difference method for pricing two-assets european style options.

Step 1: At first we take input for all required parameters like initial assets' price 'S_1' & 'S_2', strike price 'K', volatality corr to asset 1 'σ_1', & corr to asset 2 'σ_2', correlation factor 'ρ', Time to expiration 'T', no. of iterations 'N_t' of time & risk-free interest rate 'r'.

Step 2: Then we made a meshgrid, discretising the assets' price uniformly

Step 3: Then we calculated payoffs at boundary of mesh to use them in further calculations involved in iterative steps.

Step 4: Then we calculate coefficients corresponding discretized black-scholes equations variables, do iterate through time in backward difference manner

Step 5: Then we make a sparse matrix to be able to compute payoff of whole meshgrid at any time step in a vectorized manner.

Step 6: Then we run loops for time steps and compute new payoff prices for entire meshgrid of assets at each time step using ~~cof~~ coefficients & earlier computed payoff prices.

Step 7: Then payoff corresponding to initial assets' price at time 0, will give the required price for option contract at time 0.

Scanned with CamScanner

(iii) Algorithm of Monte-Carlo method for pricing two-asset european-style options.

Step 1: At first we take input for all required parameters like initial assets' prices 'S_1' & 'S_2', strike price 'K', valatality corresponding to asset 1 'σ_1' & corr. to asset 2 'σ_2', correlation factor 'ρ', Time to expiration 'T', no. of iterations 'N' & risk-free interest rate 'r'.

Step 2: Then we compute cholesky factor L of

$$\Sigma^2 = (\sigma_i \rho_{ij} \sigma_j) \quad ; \text{where } \rho_{ij} = \begin{cases} \rho & (\text{if } i \neq j) \\ 1 & \text{otherwise} \end{cases}$$

i.e, we compute L such that $\Sigma^2 = LL^T$

Step 3: Now we will run loop for 'N' iterations in which we'll do the following:

Step 4: We will generate two standard normal random variable which are i.i.d(s)

i.e, $Z_1 \sim N(0,1)$ & $Z_2 \sim N(0,1)$, where Z_1 & Z_2 are iid

Step 5: Then we compute a potential final asset price at n^{th} iteration as

$$S_i(T) = S_{i0}\, e^{(r - \frac{1}{2}\sigma_i^2)T + \sqrt{T}\sum_{j=1}^{i} L_{ij} Z_j} \quad ; \text{where } i \in \{1,2\}$$

Step 6: Then we will calculate option price as if above obtained will be final assets price as;

$V_n = e^{-rT} \cdot V(S_1(T), S_2(T), K)$; where $V(S_1, S_2, K)$ is payoff function.

Step 7 : Then we will end the loop

Step 8 : Now we will calculate sample mean of all V_n obtained in loop as

$$\overline{V_N} = \frac{1}{N} \sum_{n=1}^{N} V_n$$

& this $\overline{V_N}$ will be option price.

Scanned with CamScanner

(IV) Algorithm of Crank-Nicolson finite difference method for pricing two-asset European style options.

<u>Step1</u>: At first we take input for all required parameters like initial assets's price 'S_1' & 'S_2', strike price 'K', volatality corr. to asset1 'σ_1' & corr. to asset 2 'σ_2', correlation factor 'ρ', Time to expiration 'T', risk-free interest rate 'r'.

<u>Step2</u>: Then we made a meshgrid, discretising the assets' price uniformly.

<u>Step3</u>: Then we calculated payoffs at boundary of mesh to use them in further calculations involved in iterative steps.

<u>Step4</u>: Then we calculate coefficients corresponding to discretized black-scholes equations' variables using crank-nicolson finite difference method, to be used to calculate assets price at next time step using assets price at previous time step.

<u>Step5</u>: Then we make a sparse matrix to be able to compute payoff of whole meshgrid at any time step in a ~~vec~~ vectorized manner.

<u>Step6</u>: Then we run loops for time steps & compute new payoff prices for entire meshgrid of assets at each time step using coefficients & earlier computed payoff prices.

<u>Step7</u>: Then payoff corresponding to initial assets' price at time 0, will give the required price for option contract at time $t = 0$