Question 1

Let's assume there is a financial product ABC which gets traded in a market, and there are enough buyers and sellers of this product each day. Its price (let's call it A_t) is determined every day at the end of day based on demand and supply of the product in that market. So, If I were to buy this product today, I will need to pay A_t cash today or if I sell this product today then I will receive A_t cash today. In other words, the exchange of the product and cash happens on the same day on which the trade is done (or when deal is struck, let's call this date t_d (trading or dealing date) and the respective price be A_{t_d}).

We are interested in extending this market to allow market participants to trade this product right now, but settle it in a future date (let's call it t_e). The actual exchange of the product ABC and the transfer of cash happens on this future date. Let's call this new product XYZ, and it has an added feature that buyer of this has the right to purchase ABC but not the obligation on the future date. The price at which the buyer buys the product ABC is pre-agreed on the trading date itself (Let's call it B). Now we can understand that if the price of the underlying ABC rises (way above B) on the future day t_e, then it is very beneficial for the buyer of XYZ, as he can buy ABC at the cost of B on the future date and then sell it back in the market at this higher price. Whereas instead if the price falls below B on the future day t_e, then he will just not exercise his right to buy the bond at price B thus curtailing his losses. Thus the buyers payoff at the future date is $\max(A_{t_e} - B, 0)$, which is also the fair price of XYZ as of this future date.

In this problem we are trying to estimate a fair price (let's call it X_t) of this new product XYZ on the trading date t_d (i.e. estimate X_{t_d}). Following is one of the approach to estimate this

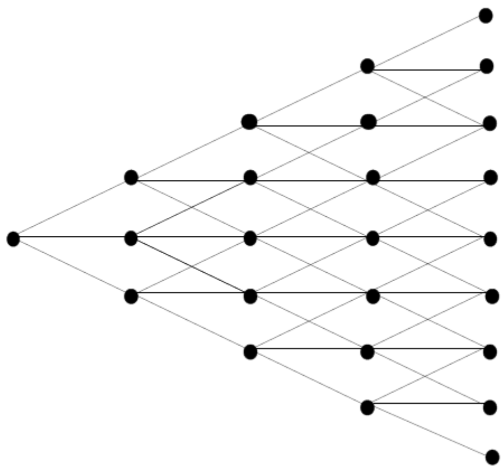
Let's assume r be the annualized rate of risk-free investment (the interest rate an investor can expect to earn on an investment that carries zero risk) and that the annualized volatility of the price A_t σ is constant and follows geometric Brownian motion. For this approach we model the price A_t as a discrete-time lattice, where at each node the price has three possible paths, an up, down and middle path. Let j_u and j_d be the jump sizes of going up and down respectively, and p_u and p_d be the respective transition probability. So at each discrete time step the next price $A_{t+\Delta t}$ is modelled as

$$A_{t+\Delta t} = \begin{cases} j_u * A_t : with\ probability\ p_u \\ A_t \quad : with\ probability\ 1 - p_u - p_d \\ j_d * A_t : with\ probability\ p_d \end{cases}$$

Δt is the length of time per step in the lattice and is simply time to maturity divided by the total number of time steps considered, also it has to satisfy $\Delta t < 2\,\frac{\sigma^2}{r^2}$. From the condition that the variance of the log of the price A_t is $\sigma^2 t$, we have $j_u = e^{\sigma\sqrt{\Delta t}}$ and $j_d = e^{-\sigma\sqrt{\Delta t}} = \frac{1}{j_u}$, and the corresponding transition probabilities given by

$$p_u = \left(\frac{e^{\frac{r\Delta t}{2}} - e^{-\sigma\sqrt{\frac{\Delta t}{2}}}}{e^{\sigma\sqrt{\frac{\Delta t}{2}}} - e^{-\sigma\sqrt{\frac{\Delta t}{2}}}} \right)^2$$

$$p_d = \left(\frac{e^{\sigma\sqrt{\frac{\Delta t}{2}}} - e^{\frac{r\Delta t}{2}}}{e^{\sigma\sqrt{\frac{\Delta t}{2}}} - e^{-\sigma\sqrt{\frac{\Delta t}{2}}}} \right)^2$$



Once the lattice of prices for A has been calculated until the future day t_e, the XYZ payoff is calculated at each node using the payoff formula of max($A_{t_e} - B, 0$) at the leaf nodes (the final rightmost nodes at t_e). After that apply the following backward induction algorithm, where $t_n = t_d + n * \Delta t$ represents the time position and j the space position in the lattice:

$$X_{t_n,j} = \left(p_u * X_{t_{n+1},j+1} + (1 - p_u - p_d) * X_{t_{n+1},j} \ + p_d * X_{t_{n+1},j-1}\right)e^{-r\Delta t}$$

By applying this we can calculate the XYZ's price at interior nodes of the lattice by considering it as a weighting of its value at the future nodes, discounted by one-time step. Thus we can calculate XYZ's price at time t_n, X_{t_n}, as the XYZ's price of an up move $p_u X_{t_{n+1}}$ plus the XYZ's price of the middle move by $(1 - \ p_u - p_d)X_{t_{n+1}}$ plus the XYZ's price of a down move by $p_d X_{t_{n+1}}$, discounted by one time step, $e^{-r\Delta t}$. We work our way backwards like this, until we reach the root node X_{t_d} .

What is the expected submission for the solution?

Three files are expected from you put together in a .zip file, for it to be qualified as a valid submission. See details for the files and their extensions. Please ensure that these files are actually zipped directly, and not any folder which contain these files.

1. Python code in a file with .py extension

This file should contain two function price_xyz and volrisk_xyz. See details below

a) You are expected to write a python function which returns X_{t_d} given other parameters. The interface of the function should be as below

def price_xyz(A_t_d, B, t_e, r, sigma, t_d, numsteps):

…

return X_{t_d}

Here A_t_d = A_{t_d} is a positive integer (e.g. 230)

B is a positive integer (e.g. 250)

t_e = t_e is the future date in a string DDMmmYYYY format (e.g. “30Jun2022”)

r is a decimal number (e.g. 0.024 for an interest rate 2.4%)

sigma = σ = is a decimal number (e.g. 0.13 for an annualized vol of 13%)

t_d = t_d is the trading date in a string DDMmmYYYY format (e.g. “25Jun2022”)

numsteps is a positive number for the number of steps in the lattice (e.g. 1000)

b) You are expected to write a python function which returns $dX_{t_d}/d\mathrm{sigma}$ given other parameters. $dX_{t_d}/d\mathrm{sigma}$ = $dX_{t_d}/d\sigma$, is the first differential of the XYZ price on the dealing date with respect to σ. You are expected to implement an analytical formula for this differentiation (**NO** bump and revalue).

def volrisk_xyz (A_t_d, B, t_e, r, sigma, t_d, numsteps):

…

return $dX_{t_d}\_dsigma$

Hint: Use chain rule.

2. A windows doc (extension .docx) file explaining your algorithm

Write a one-two pager document briefly explaining the steps in your algorithm, and or any other feature/coding practice you would like to highlight.

3. A .csv file with test outputs from the function

Write the result of the function output in cell A1, A2, A3, A4, A5 and A6 for the following inputs respectively

(1) Function = price_xyz, A_t_d = 317, B = 297, t_e = "23Sep2023", r = 0.035, sigma = 0.025, t_d = "28Jun2022", numsteps = 1000
(2) Function = price_xyz, A_t_d = 190, B = 230, t_e = "01Aug2023", r = 0.015, sigma = 0.017, t_d = "12Jan2023", numsteps = 2000
(3) Function = price_xyz, A_t_d = 100, B = 100, t_e = "12Sep2022", r = -0.015, sigma = 0.020, t_d = "26Jun2022", numsteps = 1500
(4) Function = volrisk_xyz, A_t_d = 317, B = 297, t_e = "23Sep2023", r = 0.035, sigma = 0.025, t_d = "28Jun2022", numsteps = 1000
(5) Function = volrisk_xyz, A_t_d = 190, B = 230, t_e = "01Aug2023", r = 0.015, sigma = 0.017, t_d = "12Jan2023", numsteps = 2000
(6) Function = volrisk_xyz, A_t_d = 100, B = 100, t_e = "12Sep2022", r = -0.015, sigma = 0.020, t_d = "26Jun2022", numsteps = 1500