Code developer documentation

February 2022

1 Introduction

This is a developer documentation for calculating prices of Asian call and put options on an underlying asset driven by a binomial model in discrete time implemented with C++. The purpose of this document is to describe the functionality and role of the various C++ scripts and functions that have been developed.

2 File structure

```
|_ project1/
    |__ bin/
    |   |__ Debug/
    |   |   |__ project1.exe
    |   |__ Release/
    |       |__ project1.exe
    |__ obj/
    |   |__ Debug/
    |   |   |__ *.o
    |   |__ Release/
    |       |__ *.o
    |__ project1.cbp
    |__ main.cpp
    |__ BinModel.cpp
    |__ BinModel.h
    |__ AvgPrices.cpp
    |__ AvgPrices.h
    |__ AsianOptions.cpp
    |__ AsianOptions.h
```

When you unzip the file you will find a number of files which are C++ scripts (.cpp and .h files), binary files, two documents; the developer documentation (this document) and end-user's instruction. The C++ scripts include the script for calculating Asian options, the script for generating and averaging individual

path prices, the script for calculating probabilities for each path, the script for building binomial models, and the main code for the application. And these are organized according to the diagram above.

3 Available functions

3.1 BinModel

There are two functions that are defined in this script.

1. RiskNeutProb
2. GetInputData

The first one aims for computing risk-neutral probability by taking the values U, D and R. The second function will taking the values of S0, U, D and R, check if the inputs are valid and modeling the binomial model.

Paramters	Conditions
S0,N,K	greater than 0.0
U	-1 < U and D < U
D	-1 < D and D < U
R	-1 < R and D < R < U

Table 1: Input parameters check conditions

Note: The code was taken from "Numerical Methods in Finance with C++" (section 1.3, p11) By Maciej J. Capinski and Tomasz Zastawniak Textbook

3.2 AvgPrices

In this file, it contains the code for generating and averaging individual path prices, calculating probabilities for each path, computing arithmetic and geometric mean.

1. GenPathByNumber: Taking a number x and the number of discrete time step (N) and pointers for the path array. This function modifies the value in the path array by encoding the number x into its binary form and stored back in the array. For example, if N=2; x=2 will be encoded to 1,0 and if N=3; x=2 will be encoded to 0,1,0 and x=3 encoded to 0,1,1.
2. GenPricesByPath: By given the values S0, U, D, the time step (N), pointers to the path and prices array. This function will generates an array of prices along the path and writes them back into the prices array. The value stored in the prices array is calculated by the formula $price(i) = price(i-1) * (1+U)^k * (1+D)^{1-k}$ where k is the value (either 1 or 0) in the path array with the corresponding index i.

3. GenProbabilityByPath: By given the values U, D, R, the time step (N) and the pointer to the path array. This function will computes the probability of the corresponding path, using the formula $(q)^i(1-q)^{N-i}$ where i is the number of growing in the path (e.g., if 1,0,1, then i=2) and q is the risk-neutral probability.

4. ArAverage: Taking the pointer to prices array and the time step (N), then calculating arithmetic average of the values in the array of prices and return the mean.

5. GeAverage: Taking the pointer to prices array and the time step (N), then computing geometric average of the values in the array of prices and return the mean.

3.3 AsianOptions

This implementation consists of the following functions

1. GetInputData: This function will taking the number of time steps (N) and the strike price (K), check if the inputs are greater than 0 and these values will be used in other functions.

2. Price: By given the values S0, U, D, R, the time step (N), the strike price (K), pointers to the selected payoff and average type functions. This function creates array to store paths and prices, then iterates over all possible 2^N paths and calculates the mean (corresponding to the given type) of each path along with the path probability. Finally, this function will return the discounted expected (of the selected) payoff; $\frac{1}{(1+R)^N}E(\text{Payoff at the path } i)$.

3. AsianCallPayoff: This function is Call Payoff function, $\max(A-K,0)$ where K is the strike price and A is the average price (either arithmetic or geometric mean).

4. AsianPutPayoff: This function is Put Payoff function, $\max(K-A,0)$ where K is the strike price and A is the average price (either arithmetic or geometric mean).

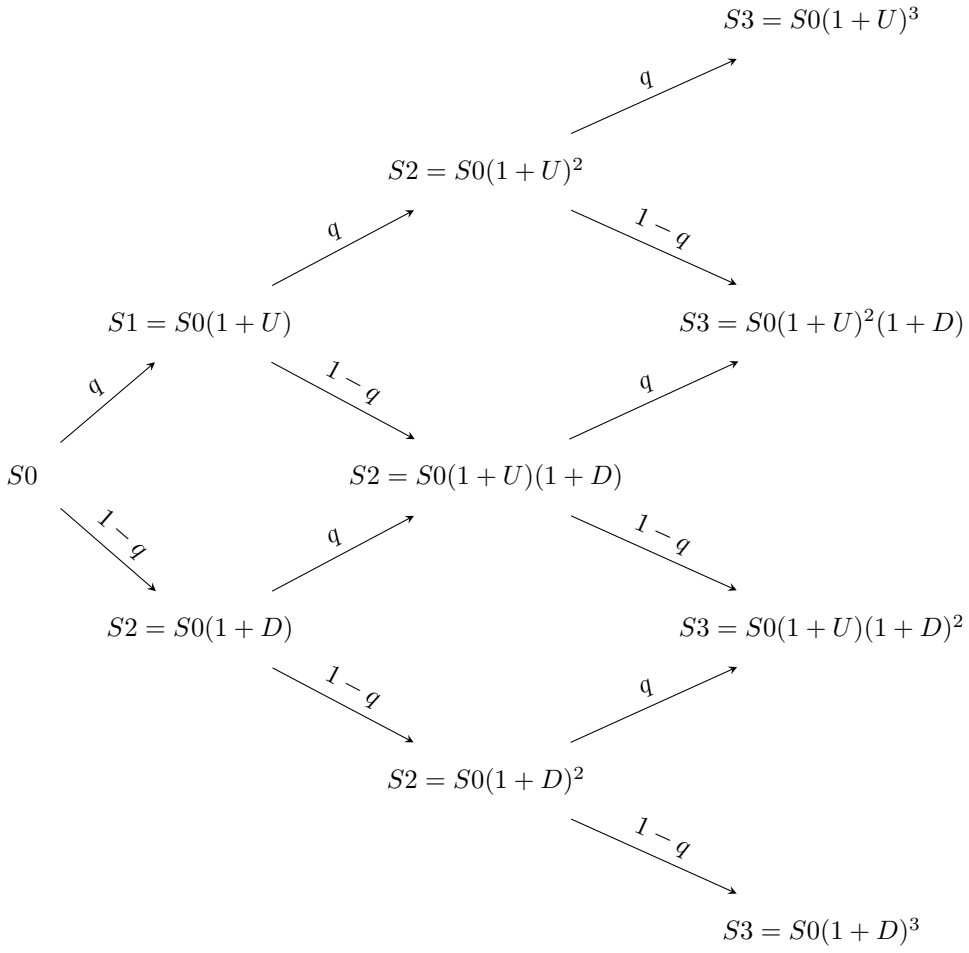
3.4 What can be further developed

We can increase the functionality of the code by adding scripts to calculate the European and American options prices, which will need to develop functions for calculating prices and calculating payoff like in AsianOptions.cpp. And it still works with BinModel.cpp.

Another way to develop this code is to writing the code in OOP style. In this way we create classes of options (Asian, European, American) instead of developing separate functions for each file, so reducing the duplication of the code.

4 Code examples

In order to demonstrate how the code works We will consider the specific case of a binomial model with N=3 of an asset with a starting price of S0=1, the risk-free growth factor R=0.5, the growing factor U=1, the falling factor D=0 and the strike price K=2 as follows:



By using $q = \dfrac{R-D}{U-D}$, we have $q = \frac{1}{2}$. Since $N = 3$, then there are $2^3 = 8$ possible paths which we can summarize as in the table below:

path	S0	S1	S2	S3	p
1	1	1	1	1	0.125
2	1	2	2	2	0.125
3	1	1	2	2	0.125
4	1	1	1	2	0.125
5	1	2	4	4	0.125
6	1	2	2	4	0.125
7	1	1	2	4	0.125
8	1	2	4	8	0.125

In the table, each row represents the stock prices at the i-th path and the last column shows the probability corresponding to the i-th path. This situation will be handled by using the GenPricesByPath and GenProbabilityByPath functions. Note: x = i-1 refers to the i-th path, e.g., to generate the prices at 8-th path, one will use GenPathByNumber with x=7.

To compute the Asian call/put options, we now have to calculate the average along the path prices which is either arithmetic and geometric averaging. Then we can calculate the expected call/put payoff by using the formula: $E(f(A_i, K))$ which can be concluded as in tables below:

path	ArAvg	CallPayoff	p*payoff	PutPayoff	p*payoff
1	1	0	0	1	0.125
2	2	0	0	0	0
3	1.666666667	0	0	0.333333333	0.041666667
4	1.333333333	0	0	0.666666667	0.083333333
5	3.333333333	1.333333333	0.166666667	0	0
6	2.666666667	0.666666667	0.083333333	0	0
7	2.333333333	0.333333333	0.041666667	0	0
8	4.666666667	2.666666667	0.333333333	0	0

Table 2: Summary table for arithmetic averaging case.

path	GeAvg	CallPayoff	p*payoff	PutPayoff	p*payoff
1	1	0	0	1	0.125
2	2	0	0	0	0
3	1.587401052	0	0	0.412598948	0.051574869
4	1.25992105	0	0	0.74007895	0.092509869
5	3.174802104	1.174802104	0.146850263	0	0
6	2.5198421	0.5198421	0.064980262	0	0
7	2	0	0	0	0
8	4	2	0.25	0	0

Table 3: Summary table for geometric averaging case.

Note: The Call Payoff function is $f(A, K) = \max(A - K, 0)$ and the Put Payoff function, $f(A, K) = \max(K - A, 0)$ where K is the strike price and A is

the average price.

Finally, one can compute the discounted expected (either call or put) payoff by using the formula:

1. Asian call options: $\frac{1}{(1+R)^N} E\left(\max(A-K,0)\right)$
2. Asian put options: $\frac{1}{(1+R)^N} E\left(\max(K-A,0)\right)$

which are summarised as follows:

E(CallPayoff)	E(PutPayoff)	E(CallPayoff)	E(PutPayoff)
0.185185185	0.074074074	0.136838674	0.079728811

Table 4: the discounted expected call/put payoff.

Note: $\frac{1}{(1+R)^N} = 3.375$.

The above calculation can be done by using our application which will prompt you to fill in the terminal screen and the final result will be calculated as shown below.

```
Enter S0: 1
Enter U: 1
Enter D: 0
Enter R: 0.5

Input data checked
There is no arbitrage

Enter N: 3
Enter K: 2

In case of arithmetic mean
Asian call option price = 0.185185

Asian put option price = 0.0740741

In case of geometric mean
Asian call option price = 0.136839

Asian put option price = 0.0797288
```