Trinomial CRR Code Developer Documentation

In this article we are going to make use of Black-Scholes model, in order to price European options. In this article, the content main focus on:

- Structure of the Code
- Available classes and Function
- Main Variables
- How to extend the code by adding new payoffs
- Test runs

---

1. Structure of the Code

Code containing a total of five files, “main.cpp”, “Proj01_Options.h”, “ProjOption01.cpp”, “TriModel01.h” and “TriModel01.cpp”, where “.cpp” are source files and “.h” are header files.

- **main.cpp** - Startup of our function, that is establishment of an independent process, and the process has become the entry procedures for various other functions, method calls, so the whole process of running track was sort of like a stack.
- **Proj01_Options.h** - Header file of “ProjOption01.cpp”. Declared functions and classes used in “ProjOption01.cpp”.
- **TriModel01.h** - Header file of “TriModel01.cpp”. Declared functions and classes used in “TriModel01.cpp”.

- **TriModel01.cpp** - To build a Trinomial CRR module.
- **ProjOption01.cpp** - To calculate Option price and Greeks.

2. Available classes and Function

At this stage, we begin to illstrute core classes and function into our C++ implementation. We start from Class:

- **Trimodel** - To build a Trinomial CRR model, which containts 5 parameters(S,r,q,sigma) .
- **Euroption** - To calculate the option price and Greeks.
- **Call** - Subclass of Euroption, which used to calculate call option.
- **Put** - Subclass of Euroption, which used to calculate put option.

Functions:

- **RiskNeutProb_up** - To calculate the probability goes up.
- **RiskNeutProb_down** - To calculate the probability goes up.
- **S(n,i)** - This function represents the stock price at node i, where n is the time.
- **PayOff** - European option where the price of the option is dependent upon the price of the underlying asset at expiry.

3.Main Variables

- **S0** - Stock Price at begin.
- **r** - Free-risky interests rates
- **q** - Divided Yield

- **T** - Time Materity(Years)
- **sigma** - Stock volatility.
- **N** - Steps of the process.
- **dx** - The asset price can goes up by dx each step.
- **dt** - Time interval. $dt = T/N$
- **v** - Equal to $r - q - 0.5 * sigma^2$

4. How to extend the code by adding new payoffs

Of course, you can try to change the codes if you want. For an example, if you want to change the call payoff function to $(2S - K)^+$ instead of $(S - K)^+$.

What need to do is modify the codes in payoff function in "ProjOption01.cpp" as below:

```
double Call::Payoff(double z)
{
    if(z>K) return 2z-K;// that is z-k recently
    return 0.0;
}
```

5. Test runs

Let us try to run our codes. By clicking build and run. You will see the instructions in the terminal. Follow the instructions and input an example data as

$S0 = 100, r = 0.05, q = 0.03, sigma = 0.2, T = 1, N = 500, K = 100$

The test process runs as below:

```
Hello, welcome to Option Price Calculator, Please enter som
e basic information firstly
Please enter Underlying Price S0:  100
Please enter Free-Risk Interest Rates r:  0.05
Please enter Dividend Yield q:  0.03
Please enter Volatility sigma:  0.2
Please enter Years Until Expiration:  1
Please enter steps to expiry N:  500
Good Job. TriNOMIAL Trees Model has been build.
Now, Please also enter some Options information
Enter call option data:
Enter strike price K:100


The results about Call Option :
Call Option Price is 8.64873
Call Delta = 0.562174
Call Gamma = :0.0189944
Call Theta = :-4.49087
Enter Put option data:
Enter strike price K:100


The results about Put Option :
Put Option Price is 6.72712
Put Delta = -0.40833
Put Gamma = :0.0189944
Put Theta = :-2.64591
All This results has also been saved in Excel Files EurPric
es.xls and EurGreeks.xls

Program ended with exit code: 0
```