Assignment3 Report

Group12 Li Daina (3035237573), Hu Hao (3035237389), Wang Youan (3035237236)

# Contribution

* Wang Youan: Merge Code, implementation of Arithmetic Mean basket call/put options calculator and Black-Scholes Formulas for European call/put options.
* Hu Hao: implementation of closed-form formulas for geometric Asian call/put options and geometric basket call/put options, and Binomial Tree method for American call/put options, User Interface design.
* Li Daina: implementation of implied volatility calculations and arithmetic Asian call/put options, User Interface design.

# UI Introduction

Our program is a web-based option calculator. Before you start the program, we recommend to set the resolution of your computer to 1366 \* 768 so as to get the best experience. Usage as follows,

1. Run *index.exe* in the program directory
2. Typing localhost:8080 in your web browser, then you will find our index page there (Figure 1).
3. There are 7 different types of calculators, you can choose any calculator you need.
4. Type in all the required parameters
5. press calculate button
6. Show the result



Figure Calculator Usage

# Code Detail and Test Case

## European call/put option (Black-Scholes Formulas)

### Code Detail

I use four functions to calculate the Black-Scholes Formulas for European call/put option, and one class to show the calculator interface (class *EuropeanOptionHtml*). This section just follows the equation from the lecture that is . Package *scipy* are used to calculate CDF of d1 and d2.

### Test Case and Analysis

let stock price be 100, strike price be 100, volatility be 30%, risk free rate be 5%, and time to maturity be 3 years, the output for put option is 12.88, for call option is 26.81, just the expected result

## Arithmetic basket option (Monte Carlo method with control variant technique)

### Code Detail

There are three class in my code.

The first class is *Option* class, mainly used for store option information, and based on random value to generate a stock price list.

The second one the the *BasketOptions* class, used to store all kinds of option info and calculate geometric and arithmetic option price. This is the main class of this function. Using geometric basket option to act as the control variate.

The last class *ArithmeticMeanBasketOptionsHTML* is web page of this function. It will also handle some invalid input.

### Test Case and Analysis

Use the test case in the assignment 3, r = 0.03, T = 3, S1(0) = S2(0) = 100, path number m is 100000 result are as follows

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K | σ1 | σ2 | ρ | Type | Price1\* | Conf1\* | Price2\*\* | Conf2\*\* |
| 100 | **0.3** | **0.3** | **0.5** | **Put** | 10.56 | [10.46, 10.65] | 10.57 | [10.56, 10.59] |
| 100 | **0.3** | **0.3** | **0.9** | **Put** | 12.41 | [12.30, 12.51] | 12.43 | [12.43, 12.43] |
| 100 | **0.1** | **0.3** | **0.5** | **Put** | 5.50 | [5.44, 5.55] | 5.53 | [5.52, 5.54] |
| 80 | **0.3** | **0.3** | **0.5** | **Put** | 4.24 | [4.18, 4.29] | 4.26 | [4.25, 4.27] |
| 120 | **0.3** | **0.3** | **0.5** | **Put** | 19.87 | [19.73, 20.00] | 19.88 | [19.86, 19.90] |
| 100 | **0.5** | **0.5** | **0.5** | **Put** | 21.06 | [20.92, 21.21] | 21.07 | [21.05, 21.10] |
|  |  |  |  |  |  |  |  |  |
| 100 | **0.3** | **0.3** | **0.5** | **Call** | 24.43 | [24.19,24.67] | 24.50 | [24.46, 24.53] |
| 100 | **0.3** | **0.3** | **0.9** | **Call** | 26.28 | [26.01, 26.56] | 26.36 | [26.35, 26.36] |
| 100 | **0.1** | **0.3** | **0.5** | **Call** | 19.44 | [19.27, 19.61] | 19.45 | [19.43, 19.47] |
| 80 | **0.3** | **0.3** | **0.5** | **Call** | 35.33 | [35.06, 35.60] | 35.39 | [35.36, 35.43] |
| 120 | **0.3** | **0.3** | **0.5** | **Call** | 16.53 | [16.32, 16.74] | 16.59 | [16.56, 16.62] |
| 100 | **0.5** | **0.5** | **0.5** | **Call** | 34.88 | [34.40, 35.35] | 34.98 | [34.88, 35.09] |

\* represent this calculation result is without control variant

\*\* represent this calculation result is with control variant

The basket option shares some similar characteristics with normal European Put/Call options. (e.g. the changes of strike price and volatility). It also has special attributes. The increase of correlation of two stock price will increase the option price as well.

The Monte Carlo Method with Control Variant apparently has the much narrower confidence interval and a little higher price than that without Control Variant.

## Implied volatility calculator

### Code Detail

Mainly, two functions in the *impliedVol* class, the *blackschole*() function used to calculate the call/put price and vega Greek. The second function is *impliedVol* (), which is used to calculate the volatility according to Newton method. The *ImpliedVolHtml* class is used to map between logical output and html presentation.

### Test Case and Analysis

## Test cases were written in the testVol.py file and following is the following table (risk free rate is 4%, repo rate is 20%, Maturity time is 8/365 = 0.022:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Strike Price | S(0) | Option premium | Option type | Implied Volatility | |
| **Bid** | **Ask** |
| 1.8 | **1.96** | **0.15505** | **Call** | 0.370531 | 0.388385 |
| 1.8 | **1.96** | **0.0031** | **Put** | 0.35686 | 0.378679 |
| 2.6 | **1.96** | **0.00025** | **Call** | 0.703173 | 0.732629 |
| 2.6 | **1.96** | **0.65385** | **Put** | N/A | 2.20045 |
| 2.15 | **1.96** | **0.0026** | **Call** | 0.393748 | 0.406336 |
| 2.15 | **1.96** | **0.2023** | **Put** | N/A | 0.544185 |

## Arithmetic Asian option (Monte Carlo method with control variant technique)

### Code Detail

There are two classes and one main function in this file.

The first class is *arithmeticOption* which is used to calculate the arithmetic Asian option and its confidence interval. Inside of the class, *arithmeticOptPricer*() function is used to calculate the result according to the Monte Carlo method. Instead of using loops inside of Monte Carlo, matrix is used for random number storage and calculations. Besides, geometric Asian option variant method control method is used to reduce the error.

The second class is *ArithmeticAsianOptionPricerHtml* which is used to handle the html web page presentation.

The main function is used to test the program.

### Test Case and Analysis

Test cases used geometric option variant control

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| K | n | σ | Type | Price1\* | Conf1\* | Price2\*\* | Conf2\*\* |
| 100 | **50** | **0.3** | **Put** | 7.77 | [7.69, 7.84] | 7.80 | [7.08, 7.81] |
| 100 | **100** | **0.3** | **Put** | 7.71 | [7.64, 7.78] | 7.75 | [7.75, 7.76] |
| 100 | **50** | **0.4** | **Put** | 11.23 | [11.14, 11.32] | 11.28 | [11.28, 11.29] |
|  |  |  |  |  |  |  |  |
| 100 | **50** | **0.3** | **Call** | 14.76 | [14.62,14.90] | 14.73 | [14.72, 14.74] |
| 100 | **100** | **0.3** | **Call** | 14.66 | [14.52, 14.81] | 14.62 | [14.61, 14.63] |
| 100 | **50** | **0.4** | **Call** | 18.24 | [18.03, 18.44] | 18.21 | [18.19, 18.23] |

\* represent this calculation result is without control variant

\*\* represent this calculation result is with control variant

## Geometric Asian option and Geometric basket option (Closed-form)

### Code Detail

One function in the code file “geometricOptions.py” is used for calculating each of the two geometric options. *geometricAsian* is for calculating Geometric Asian option. The closed-form formula for the option is much like the Black-Scholes Formula. Differences exist since the underlying price is the geometric mean price of all the observations. The formula is:

, ,

,

,

where

The underlying price for Geometric basket option is the geometric mean price of all the assets. The closed-form formula is:

, ,

,

,

where

And there are two classes *GeometricOptionHtml* and *GeometricBasketHtml* to control the interfaces of the two calculators.

### Test Case and Analysis

Asian options:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| σ | K | n | Put option price | Call option price |
| 0.3 | 100 | 50 | 8.48270454488 | 13.2591261305 |
| 0.3 | 100 | 100 | 8.43108015568 | 13.1387791144 |
| 0.4 | 100 | 50 | 12.5587694397 | 15.7598197764 |

Volatility affects option prices more than observation times.

Basket options:

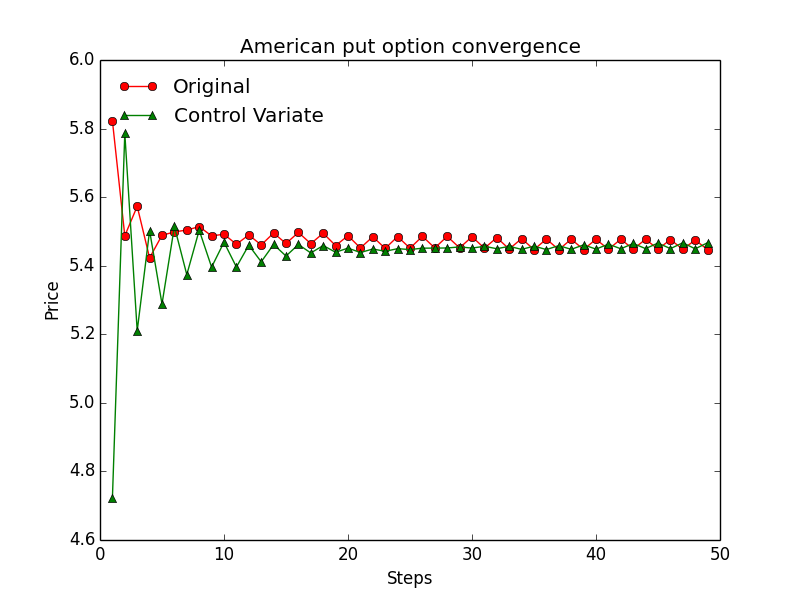
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| S1(0) | S2(0) | K | σ1 | σ2 | ρ | Put option price | Call option price |
| 100 | 100 | 100 | 0.3 | 0.3 | 0.5 | 11.491572668 | 22.1020927976 |
| 100 | 100 | 100 | 0.3 | 0.3 | 0.9 | 12.6223501611 | 25.8788255264 |
| 100 | 100 | 100 | 0.1 | 0.3 | 0.5 | 6.58638062012 | 17.9247366058 |
| 100 | 100 | 80 | 0.3 | 0.3 | 0.5 | 4.71157662855 | 32.5362562867 |
| 100 | 100 | 120 | 0.3 | 0.3 | 0.5 | 21.2891051598 | 14.6854657609 |
| 100 | 100 | 100 | 0.5 | 0.5 | 0.5 | 23.469148024 | 28.4493865195 |

## American call/put option (Binomial Tree method)

### Code Detail

The class Binomial in the code file “Binomial.py” is used to calculate the American call/put option price using Binomial Tree method. There’re three main functions in the class. *AmericanOption* is used to calculate American option price. *EuropeanOption* is used to calculate the European option price using the same steps. *execute* is used to calculate the final American option price, using control variate: Pa(n) = P(n) - Pe(n)+Black- Scholes Price. In each step, for example, in step n, there’ll be n+1 nodes in this level. The underlying price of each node equals to S0\*[und0, un-1 d1, …… u0dn].

### Test Case and Analysis

For the American put option: K = 52, maturity T = 2 years, r = 5%, σ=22.3144%, stock price = 50 

From the figure above, we can see the control variate method can converge quicker than the original method. The result converges to approximately 5.45.

# Extensions

Both geometric and arithmetic basket option calculator can calculate more than two assets