

Report

Option Pricer by Group 19

Program

OPTION PRICER BY GROUP 19

Exercise Style: American

Option Type: ☒ Call ☐ Put

Asset1

Stock Price: (\$) 100

Strike Price: (\$) 100

Interest Rate: (%) 5

Volatility: (%) 30

Maturity: (years) 3

Step: 50

ZHAN Hui 3035249227

LIU Jiahe 3035237365

Calculate

Result: 26.707123658

Overview

The program is written in Python 2.7 and several math related packages are included. We use Qt5 + Pyqt5 for user interface design. The executable file for windows platform is compiled with Pyinstaller.

The program has included all the required functionalities including:

- Implied volatility calculation
- European option pricing
- American option pricing
- Arithmetic/ Geometric Asian options and basket options pricing

Features

Tested environment:

Microsoft Windows 10 (64-bit)

Python 2.7

Pyqt 5.6.0 + PyQt5

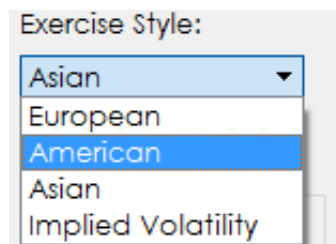
Install

The option pricer has been packed as a windows runnable program. After downloaded, unzip the a3.zip file, you can then find the executable file 'a3.exe' under the unzipped folder.

Simply run the a3.exe and hopefully the program will be popped up.

Usage

All the functionalities are navigated by a dropdown list where you can switch between different functions.



Implied Volatility

1. Click the dropdown menu and select 'Implied Volatility'
2. Input all parameters: (from top to bottom) and click 'Calculate' button
 - o Stock Price (\$)
 - o Strike Price (\$)
 - o Interest Rate (%)
 - o Repo Rate (%)
 - o Maturity (in years)
 - o Premium (\$) : Option Premium

American Option

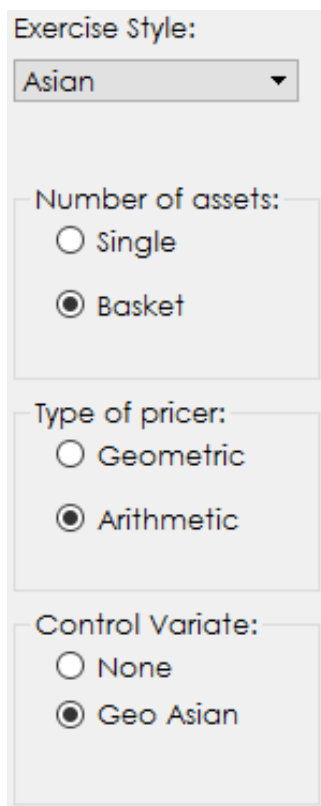
1. Click the dropdown menu and select 'American'.
2. Input all parameters, besides the common parameters, you also need to specify the steps of the binomial tree (step).

European Option

1. Click the dropdown menu and select 'European '.
2. Input all parameters, and click 'Calculate' button.

Asian Option

1. Click the dropdown menu and select 'Asian'. After that, the additional parameters and options will be visible.



The screenshot shows a configuration panel for an Asian Option. It contains four sections: 'Exercise Style' with a dropdown menu set to 'Asian'; 'Number of assets' with radio buttons for 'Single' and 'Basket' (selected); 'Type of pricer' with radio buttons for 'Geometric' and 'Arithmetic' (selected); and 'Control Variate' with radio buttons for 'None' and 'Geo Asian' (selected).

2. Number of Assets: 'Single' should be selected if you want to calculate one asset only. As 'Basket' indicates the basket option pricing (supports two assets pricing).
3. Type of pricer: 'Geometric' and 'Arithmetic' options are provided, which conducts closed form calculation and 'Monte Carlo' method respectively.
4. You may also need to specify the control variate method (no control variate, or geometric Asian option).
5. Input all parameters, besides the common parameters, you also need to specify the correlation between two assets if you are in need of the basket option pricing.

Functionalities

Implied Volatilities:

```
def implied_volatility(ty, S, K, T, r, price_true, q=0)
```

Special Input Parameters:

Name	Info
price_true (float)	Option premium
q (float)	Repo rate

The Newton's method is used which provides rapid convergence. Tolerance is set to $1.0e-8$, and the maximum number of iterations are 100.

Before the iteration, we also checked the validity of the Option premium by checking if it exceeds the upper or lower bound. Function as below (for call option):

```
if ty == 'C':          if (price_true < max((S * exp(-q * T) - K *  
exp(-r * T)), 0)) \      or (price_true > S * exp(-q *  
T)):                    return 'NaN'
```

American Options

```
def binomial_option(cp_type, exp_t, s, k, r, sigma, n)
```

Special Input Parameters:

Name	Info
n	Number of steps

We calculate the American option price with a binomial price tree.

The price or payoffs of each node is stored in a list. While it backtracks the tree from leaf nodes to the root, eg..

```
p_rec.append((pay[m] * p + pay[m + 1] * (1 - p)) * exp(-r *  
delta_t))
```

Finally, it returns the price `pay[0]` which is at the root of the binomial tree.

