Pricing American Options by Simulation: A Comparison of Methods

Joel Braun, James Ly, Avinash Sharma

DATA 301
California Polytechnic State University, San Luis Obispo

June 8, 2016

Abstract

As quantitative American options pricing is a field of both heavy research and analysis, numerous pricing models have been and continue to be developed and refined. The intent of this research paper is to quantitatively compare some of the many options-pricing methods based off the classic Black-Scholes-Merton and the Binomial options pricing models. Results and accuracy vis-a-vis real world options pricing data are discussed, as well as trends realized as part of the analysis.

Table of contents

1 Overview **2**

1.1 A Brief Introduction to Stock Options 2

1.2 Simulation of Options 3

1.2.1 Volatility 4

1.3 Intent of Analysis 5

2 Simulation Design **6**

2.1 Tools 6

2.2 Simulation Method 6

2.3 Simulation Framework 7

2.3.1 Notes on Computational Performance 8

3 Results Analysis **9**

3.1 Overview 9

3.1.1 A note on analysis 9

3.1.2 Second subsection in the first section 9

3.1.3 Third subsection in the first section 9

3.2 Second section of the third chapter 10

3.3 The layout of formal tables 10

4 Conclusions **12**

4.1 Overview 12

4.1.1 A note on analysis 12

4.1.2 Second subsection in the first section 12

4.1.3 Third subsection in the first section 12

4.2 Second section of the third chapter 13

4.3 The layout of formal tables 13

Chapter 1

Overview

1.1 A Brief Introduction to Stock Options

In finance, an option is a contract which gives the buyer (the owner or holder of the option) the right, but not the obligation, to buy or sell an underlying asset or instrument at a specified strike price on a specified date, depending on the form of the option. (**??**). For the intent of this paper, an underlying asset or instrument will be considered to be a simple equity (or "stock", "share", etc.) in a firm.

There are two types of options that will be priced as part of this simulation. The first, the *Call* option, allows an investor to secure purchase of an asset at a certain prespecified price. The second, the *Put* option, allows an investor to secure sale of an asset at a certain prespecified price. These prespecified prices are referred to as the 'strike' prices of the option. An option is considered worth exercising (buying or selling) in circumstances where it is *in the money.* An option is in-the-money for a call when the strike price is below the underlying asset's value. It is in-the-money for a put when the strike price is above the market price of the asset. Typically, in options trading, a call option is used when investor sentiment on a stock is positive, while a put option is used when sentiment on a stock is negative.

The style of option being bought or sold adds additional complexity to the situation. Each stock option being bought or sold is given an expiry date. An *expiry date* is the date at which an option expires and a final decision must be made as to whether to exercise. There are two commonly traded types of options, however- American-Style and European-Style. American-Style stock options may be exercised at any point prior to the expiry. European-Style stock options may only be exercised at the time of expiry. This paper will solely explore American-Style options pricing as it relates to actual US market values.

1.2 Simulation of Options

With the advent of the modern financial services industry, high frequency trading, and computational data analytics it is now common practice to determine via computer simulation the estimated future prices of stock options. Numerous methods have been developed which address the idea of option estimation, the most notable being the **Black-Scholes** (sometimes Black-Scholes-Merton) method. The Black-Scholes method, introduced in 1983, is largely considered the standard for options pricing. A deep discussion of Black-Scholes is beyond the scope of this paper, but there are several points worth covering.

The Black-Scholes model makes several key assumptions. The first is that there are two possible asset categories, the stock itself and an alternate lower-risk asset. The second is that the return of the stock can be estimated by a random walk with drift. This random walk is based on geometric Brownian motion, and is most simply a random path up and down the value of an asset might take. The random walk for equities is dictated by volatility, a concept which will be discussed shortly.

It's also worth noting the original design of the Black-Scholes pricing model focused on pricing of European-Style options, but future refinements and iterations provided additional methods for the pricing of American-Style options. Two of these will be explored in the simulation, the Barone-Adesi and Whaley method and the Bjerksund-Stensland method. Both of these are approximation methods.

In order to model an option price, the Black-Scholes model requires several set variables. These are as follows:

Current underlying price: The current market price of the equity.

Option strike price: The price at which the shares will be bought or sold.

Time to expiry: Time left until the option expires.

Implied Volatility: The size of the shifts the asset may make during the random walk.

Risk-free interest rate: The rate the asset would appreciate at in a lower-risk investment.

Another commonly used model for estimating stock option prices is the binomial model. A binomial model formulates a random option pricing curve based on a number of time intervals. The stock price is simulated to move up or down at each step by an amount relating to volatility and time to expiry. It is somewhat analogous to the Black-Scholes model, but is a discrete-time model as opposed to a continuous one. The result of binomial method options calculations resembles a tree, as the decision is made to exercise or hold at each point.

Luckily, the binomial method requires exactly the same inputs as the Black-Scholes model. As such, we can luckily maintain a similar testing framework for both. We will explore the following binomial methods as part of this simulation:

- Cox-Ross-Rubenstein
- Jarrow-Rudd
- Equal Probabilities
- Trigeorgis
- Tian
- Leisen-Reimer

Binomial methods are more commonly used in professional options simulation**??**. Its ability to progressively price-in the value of an early option exercise makes it ideal for American options.

1.2.1 Volatility

The asset's implied volatility is worth some discussion. Because volatility dictates the shifts of the random walk in the model, it has the potential to dictate the accuracy of the model altogether. There is some discussion as to whether historical volatility should be used in lieu of implied volatility. One of the simplest methods for calculating implied volatility is a guess-and-check approach involving repeated simulation volatility values. These values are calibrated until a result similar to the current option price is attained**??**. This is less than ideal for our analysis, as it is not quantifiable. It also does not take into account whether current options are reasonably priced or are outliers.

The GARCH Model

A model commonly used in academia for quantitative finance volatility calculations is the GARCH model. The GARCH model is an autoregressive conditional heteroskedascity method that is used to model time series**??**. In this case, we specifically will be using the GARCH(1,1) model for calculations.

1.3 Intent of Analysis

The intent of our analysis on these topics was to try and answer several key questions on the models:

Black-Scholes vs. Binomial Which options pricing class provides better accuracy?

Short term accuracy Which options pricing model is the most accurate in the short term?

Long term accuracy Which options pricing model is the most accurate in the long term?

Patterns What patterns do we see in the accuracies/inaccuracies of these models?

Our analyses of these questions will be based on several datasets. The first will be a short-dated expiry set of puts and calls expiring one calendar week from the test. The second will be a long-dated expiry set of puts and calls expiring one 30-day month from the test. The third test will be the longest-dated expiry set of puts and calls expiring six 30-day months from the test. This will provide an effective cross-section of estimation methods over both the short and long-term time periods.

Chapter 2

Simulation Design

2.1 Tools

For this simulation, we relied on several open-source libraries and publicly accessible financial data. A testing framework was developed in Python 2.x for which accessed financial data and performed simulations based on current option availability and parameters.

The cornerstone of this effort was the open source C++ library QuantLib, which provides tools for quantitative finance. QuantLib is a comparatively massive library, taking several hours to compile and measuring several GB in size. It contained a complete options-pricing simulator incorporating all pricing equations being tested. This was integrated with a Python 2.x wrapper framework that is also part of the QuantLib project.

Also used was the pip package `python-google-options-chain`, which downloads and converts Google finance data from its JSON endpoints.

To calculate the volatility values used, historical data from Yahoo! finance (provided as part of the `pandas` library) was used. This was integrated in with the `pyflux` package, which provided GARCH calculation functionality for the simulation.

2.2 Simulation Method

For the purposes of our simulation, it was determined that options prices would only be compared for the Dow Jones Industrial Average, a set of thirty shares that is a price-weighted aggregation of the largest corporations in America. This scope limitation was designed to keep data measurable- it is likely the S&P 500 would provide a better cross section of American industry.

For each of these listings on the index, simulations were run in all situations where a current option price was available. Two separate datasets were collated for the shares on the index; one for call options and another for put options. This method was used for the one week, one month, and six month tests.

2.3 Simulation Framework

The framework under which these tests were conducted was developed in Python 2.x for compatibility purposes with available QuantLib Python libraries. These were based off of samples provided by the QuantLib project for options simulation??. The testing framework itself merely served to facilitate connection between real data procured from Google Finance and the generated valuations for each option. Once an option for one of our symbols was found to have a present value, our simulation pulled additional data on underlying asset price and calculated a volatility value. This volatility value, the critical part of the simulation, was then used in conjunction with other parameters to formulate an options price.

After a price was formulated, all data was written in rows to a CSV file for later analysis. This data generation isolated puts and calls for further analysis as to the accuracy of the pricing of each. Each CSV file was then analyzed and the results prepared in Python using various additional libraries.

Column	Value
Symbol	AAPL
Tag	AAPL160610P0009950
Strike	100
Real	2.72
Barone-Adesi Whaley	2.364
Bjerksund-Stensland	2.364
FD American	2.367
Jarrow-Rudd	2.367
Cox-Ross-Rubenstein	2.367
Equal Probabilities	2.367
Trigeorgis	2.367
Tian	2.367
Leisen-Reimer	2.367

Table 2.1 **Example data row in an output CSV.**

The data in table 2.1 is representative of a single row in a CSV file. Depending on the amount of price data, individual runs for options yielded around 14,000 row entries (including both put and call valuations). From this CSV format, data could be reloaded back in to a separately designed Python analysis library for further investigation. Depending on machine speed, individual runs could take as many as several minutes to complete. Simulations were not time dependent and no performance improvements or modifications (such as multiprocessing, etc) were made to the original framework.

2.3.1 Notes on Computational Performance

While in the case of our simulations no tangible performance issues were experienced, it is important to consider that these pricing models vary in complexity. The binomial method, while boasting a higher perceived accuracy, is typically less computationally performant than a Black-Scholes implementation. Likewise, some of the models tested (such as Barone-Adesi Whaley) are actually approximation models. Thus, they are designed to be more performant and provide for quicker price calculations. Various research papers are available on this subject, but they are beyond the scope of our analysis.

Chapter 3

Results Analysis

3.1 Overview

And now I begin my third chapter here . . .

And now to cite some more people **? ?**]

3.1.1 A note on analysis

. . . and some more

3.1.2 Second subsection in the first section

. . . and some more . . .

First subsub section in the second subsection

. . . and some more in the first subsub section otherwise it all looks the same doesn't it? well we can add some text to it . . .

3.1.3 Third subsection in the first section

. . . and some more . . .

First subsub section in the third subsection

. . . and some more in the first subsub section otherwise it all looks the same doesn't it? well we can add some text to it and some more and some more and some more and some more and some more and some more and some more and some more and some more . . .

Second subsub section in the third subsection

. . . and some more in the first subsub section otherwise it all looks the same doesn't it? well we can add some text to it . . .

3.2 Second section of the third chapter

and here I write more . . .

3.3 The layout of formal tables

This section has been modified from "Publication quality tables in LaTeX*" by Simon Fear.

The layout of a table has been established over centuries of experience and should only be altered in extraordinary circumstances.

When formatting a table, remember two simple guidelines at all times:

1. Never, ever use vertical rules (lines).

2. Never use double rules.

These guidelines may seem extreme but I have never found a good argument in favour of breaking them. For example, if you feel that the information in the left half of a table is so different from that on the right that it needs to be separated by a vertical line, then you should use two tables instead. Not everyone follows the second guideline:

There are three further guidelines worth mentioning here as they are generally not known outside the circle of professional typesetters and subeditors:

3. Put the units in the column heading (not in the body of the table).

4. Always precede a decimal point by a digit; thus 0.1 *not* just .1.

5. Do not use 'ditto' signs or any other such convention to repeat a previous value. In many circumstances a blank will serve just as well. If it won't, then repeat the value.

A frequently seen mistake is to use '\begin{center}' . . . '\end{center}' inside a figure or table environment. This center environment can cause additional vertical space. If you want to avoid that just use '\centering'

Table 3.1 A badly formatted table

	Species I		Species II	
Dental measurement	mean	SD	mean	SD
I1MD	6.23	0.91	5.2	0.7
I1LL	7.48	0.56	8.7	0.71
I2MD	3.99	0.63	4.22	0.54
I2LL	6.81	0.02	6.66	0.01
CMD	13.47	0.09	10.55	0.05
CBL	11.88	0.05	13.11	0.04

Table 3.2 A nice looking table

Dental measurement	Species I		Species II	
	mean	SD	mean	SD
I1MD	6.23	0.91	5.2	0.7
I1LL	7.48	0.56	8.7	0.71
I2MD	3.99	0.63	4.22	0.54
I2LL	6.81	0.02	6.66	0.01
CMD	13.47	0.09	10.55	0.05
CBL	11.88	0.05	13.11	0.04

Table 3.3 Even better looking table using booktabs

Dental measurement	Species I		Species II	
	mean	SD	mean	SD
I1MD	6.23	0.91	5.2	0.7
I1LL	7.48	0.56	8.7	0.71
I2MD	3.99	0.63	4.22	0.54
I2LL	6.81	0.02	6.66	0.01
CMD	13.47	0.09	10.55	0.05
CBL	11.88	0.05	13.11	0.04

Chapter 4

Conclusions

4.1 Overview

And now I begin my third chapter here . . .

And now to cite some more people **? ?**]

4.1.1 A note on analysis

. . . and some more

4.1.2 Second subsection in the first section

. . . and some more . . .

First subsub section in the second subsection

. . . and some more in the first subsub section otherwise it all looks the same doesn't it? well we can add some text to it . . .

4.1.3 Third subsection in the first section

. . . and some more . . .

First subsub section in the third subsection

. . . and some more in the first subsub section otherwise it all looks the same doesn't it? well we can add some text to it and some more and some more and some more and some more and some more and some more and some more . . .

Second subsub section in the third subsection

. . . and some more in the first subsub section otherwise it all looks the same doesn't it? well we can add some text to it . . .

4.2 Second section of the third chapter

and here I write more . . .

4.3 The layout of formal tables

This section has been modified from "Publication quality tables in LaTeX*" by Simon Fear.

The layout of a table has been established over centuries of experience and should only be altered in extraordinary circumstances.

When formatting a table, remember two simple guidelines at all times:

1. Never, ever use vertical rules (lines).

2. Never use double rules.

These guidelines may seem extreme but I have never found a good argument in favour of breaking them. For example, if you feel that the information in the left half of a table is so different from that on the right that it needs to be separated by a vertical line, then you should use two tables instead. Not everyone follows the second guideline:

There are three further guidelines worth mentioning here as they are generally not known outside the circle of professional typesetters and subeditors:

3. Put the units in the column heading (not in the body of the table).

4. Always precede a decimal point by a digit; thus 0.1 *not* just .1.

5. Do not use 'ditto' signs or any other such convention to repeat a previous value. In many circumstances a blank will serve just as well. If it won't, then repeat the value.

A frequently seen mistake is to use '\begin{center}' . . . '\end{center}' inside a figure or table environment. This center environment can cause additional vertical space. If you want to avoid that just use '\centering'