香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Research on the Pricing of Compound Options Based on Binomial Tree Model and Monte Carlo Simulation

By

Jiang Yilu, Song Sizhe, Chen Yingxin, Song Xinjuan

220140069, 221040087, 221040028, 221040009

Abstract

Compound option refers to the option that its underlying is also a option, that is, the derivative of derivatives. The pricing method of compound option is one of the most focused issues in the research of financial derivatives. In this paper, the binomial tree model and Monte Carlo simulation method are used to price composite option underlying with stock APPL and calculate Greeks $delta(\Delta), gamma(\Gamma), theta(\theta)$. B-S formula is used to verify the relationship between strike price and implied volatility. The sensitivity analysis and variations of the model are also discussed.

We first estimate the parameters required by the model, such as volatility, risk-free interest rate and dividend yield, based on real data. Then we give the pricing of binary tree model and Monte Carlo simulation, the results show that both the pricing is very consistent. In addition, sensitivity analysis shows that as T_2 or the exercise price K_2 of C or K_1 of Y increases, the price of Y decreases. As T_1 or the price of the underlying stock increases, the price of Y increases. Remain other factor same, the European option price is lower than the American option price.

The Greeks results show that Δ increases with the increase of stock price. Γ is maximal near at-the-value and decreases when out-of-the-money or in-the-money. θ is smallest near at-the-value and becomes larger when out-of-the-money or in-the-money. Remain other factor same, the absolute value of Greeks of European option is always less than those of the American option.

Finally, we verify the various underlying variants, and the results show that the prices from high to low are American Y of underlying American option C, European Y of underlying European option C, American Y of underlying European option C and European Y of underlying European option C.

Content

1 Introduction

1.1 Introduction to call option

A call option is a contract, between the buyer and the seller of the call option, to exchange a security at a set price. The buyer of the call option has the right, but not the obligation, to buy an agreed quantity of the underlying from the seller of the option at the expiration date for the strike price. The seller is obliged to sell the underlying to the buyer if the buyer so decides. The buyer pays a fee (called a premium) for this right.

1.1.1 key terms

The strike price of a call option is the price paid by the buyer for the asset.
The exercise of a call option is the act of paying the strike price to obtain an asset.
The expiration date of an option is the day on which the option must be exercised or become worthless.
The style of option determines the exercise time of the option. A European-style option can be exercised only at expiration, while the exercise of an American-style option could occur at any time during the life of the option.

1.1.2 The calculation of payoff

The buyer is not obligated to buy the underlying, and hence will only exercise the option if the payoff is greater than zero. The algebraic expression for the payoff to a purchased call is shown as Formula 1.1.

$$\textit{Purchased call payoff} = \max\,[0,\ \textit{spot price at expiration} - \textit{strike price}] \quad (1.1)$$

1.2 Introduction to the compound option

Y is a compound option with T_1 years to expiration. The holder of derivative Y has the right of paying K_1 to buy the underlying European call option before the expiration date T_1. Define the underlying European call option be option C and the exercise time be t_{ex} ($0 \leqslant t_{ex} \leqslant T_1$).
C is a call option with T_2 year to expiration (timing from t_{ex}) and a strike price of K_2.
The payoff of compound option Y at time t_{ex} is max $(C(K_2, T_2)-K_1,0)$, where $C(K_2, T_2)$ is the price of option C.

TABLE 1 DERIVATIVES IN THE PROBLEM

Derivative	Compound option Y	European call option C
Time to expiration	T_1	T_2 (timing from t_{ex})
Right of holder	Paying K_1 to buy C within T_1	Paying K_2 to buy underlying stock at $t_{ex}+T_2$
Exercise time	t_{ex} ($0 \leqslant t_{ex} \leqslant T_1$)	$t_{ex}+T_2$
payoff	max $(C(K_2, T_2)-K_1,0)$	max $(S_{tex+T2}-K,0)$

1.3 Symbol definition

Symbols in the following article have the same meaning as Table 2.

TABLE 2 SYMBOL DEFINATION

T_1	The expiration time of compound option Y
K_1	The strike price of compound option Y
T_2	The expiration time of call option C
K_2	The strike price of call option C
$C\ (K_1, T_1)$	The price of the call option C
$Y\ (K_2, T_2)$	The price of the compound option Y
t_{ex}	The exercise time of the compound option Y
t_{ex2}	The exercise time of the call option C
n	The binomial period
σ	The volatility of stock
δ	The dividend of stock
r	The risk-free interest rate

1.4 Article Structure

The structure of the paper is as follows. In the second chapter, this article introduces the information of the company, then estimates the stock's historical volatility, implied volatility, dividend yield and the risk-free interest rate of the market from the market data.

In the third chapter, this article has constructed a n-step compound binomial tree where the underlying call option is a European option. By working backward through the binomial tree based on different stock price, this chapter estimates the price of the compound option when T1, K1, T2, K2 take different values, in case of the compound option is European style or American style. This chapter also discusses the change of the price of the compound option as certain condition changes.

The fourth chapter prices the compound option using the Monte Carlo simulation when the stock price follows geometric Brownian motion. And compare this results with binomial trees' results.

The fifth chapter introduces the $delta(\Delta), gamma(\Gamma), theta(\theta)$ of derivative. This section calculates $delta(\Delta), gamma(\Gamma), theta(\theta)$ of the compound option according to the compound binomial tree, and compares the results when the compound option is European style or American style.

Finally, this article has done a further study, that is, when the underlying call option is an American option, the implement of pricing the compound option using binomial tree.

2 Selection of the underlying stock

2.1 Introduction to the underlying stock

The underlying stock we selected is apple (NASDAQ: AAPL). The selection is based on three main reasons as followed.

1) As a leader in American science and technology, Apple has a large stock market value, good liquidity and comprehensive data available. Apple is the first technology company in human history with a market value exceeding US $1 trillion, and its market value exceeded US $2.7 trillion in November 2021.

2) Stable performance and strong profitability. Throughout fiscal year 2021, Apple's net profit was US $94.68 billion, an increase of 65% over US $57.411 billion in the same period last year; Diluted earnings per share was $5.61, up 71% from $3.28 in the same period last year.

3) Strong cash reserves and sufficient funds for capital return activities such as dividend distribution and share repurchase. From 2012 to 2020, the company has not only paid stable quarterly dividends, but also increased dividends every year.

2.2 Historical volatility

The volatility of an asset is defined as the standard deviation of continuously compound returns. Compute weekly volatility (the standard deviation of the weekly return). Suppose that the continuously compound return over week i is $r_{weekly,i}$, we can sum continuously compound returns thus obtain annual continuously compound return, as Formula (2.1).

$$r_{annual} = \sum_{i=1}^{52} r_{weekly,i} \tag{2.1}$$

The variance of the annual continuously compound return is therefore

$$Var(r_{annual}) = Var(\sum_{i=1}^{52} r_{weekly,i}) \tag{2.2}$$

It is general to suppose that returns are uncorrelated with time; i.e., the realization of the return in one period does not affect the expected returns in subsequent periods. With this assumption, the variance of a sum is the sum of the variances. Also suppose that each week has the same variance of returns. If we let σ denote the annual variance, then from equation (2.2) we have (2.3).

$$\sigma = \sqrt{52}\sigma_{weekly} \tag{2.3}$$

Measure historical volatility σ by computing the standard deviation of continuously compound historical returns. Some of the results is shown as Table 3.

TABLE 3 PARTIAL RESULTS OF HISTORICAL VOLATILITY MEASUREMENT

Week number	Price	ln (S_t / S_{t-1}) (%)
52	161.94	0.862
51	160.55	6.804
50	149.99	-0.711
	……	
3	125.88	3.413
2	121.66	0.131
1	121.50	4.740
Standard deviation × $\sqrt{52}$	21.8350%	

2.3 Implied volatility

2.3.1 Introduction to implied volatility

Implied volatility is the volatility contained in the market price of options, that is, the volatility deduced from the option price and option pricing formula. The comparison between implied volatility and historical volatility can guide investors' operation. Investors can buy and sell the volatility directly, or determine the trading opportunity with reference to the volatility.

Implied volatility is not an analytical solution, but a numerical solution. We can obtain the size of implied volatility in a theoretical way. Since the option pricing model (such as BS model) gives the quantitative relationship between the option price and five basic parameters (underlying stock price, strike price, interest rate, maturity time and volatility), as long as the first four basic parameters and the actual market price of the option are substituted into the pricing formula as known quantities, the only unknown quantity can be solved, and its size is the implied volatility.

We can write the equation of implied volatility through the option pricing formula. Assume that we observe the stock price S, strike price K, interest rate r, dividend yield δ, and time to expiration T. The implied call volatility is the $\hat{\sigma}$ that solves

$$Market\ option\ price = C\ (S(t), K, \hat{\sigma}, r, T, \delta) \tag{2.4}$$

This equation cannot be solved directly for the implied volatility $\hat{\sigma}$, because there is no closed solution. Newton iterative method and dichotomy are two common methods to calculate implied volatility. We use dichotomy method.

2.3.2 Calculation of implied volatility by dichotomy

The principle of dichotomy is that we constantly calculate the function value at the midpoint of the interval to determine the new interval. we want to find the closest value of volatility that can meet the conditions. When the accuracy we need is reached, the iteration stops. The main process is as follows:

1) Set a range for possible volatility in $[\sigma_{\min}, \sigma_{\max}]$,$\sigma_{\min}$=0,$\sigma_{\max}$=1000%

2) First calculate $C\left(\frac{1}{2(\sigma_{max}+\sigma_{min})}\right)$, compare it with the actual market option price C_{market} .

3) If $C\left(\frac{1}{2(\sigma_{max}+\sigma_{min})}\right) < C_{market}$, change the new range to $[\frac{1}{2(\sigma_{max}+\sigma_{min})}), \sigma_{max}]$. If $C\left(\frac{1}{2(\sigma_{max}+\sigma_{min})}\right) > C_{market}$, change the new range to $[\sigma_{min}, \frac{1}{2(\sigma_{max}+\sigma_{min})}]$.

4) Iteration will stop when we get a closet value of volatility, that is when $\left|C\left(\frac{1}{2(\sigma_{max}+\sigma_{min})}\right) - C_{market}\right| < 10^{-6}$.

By dichotomy, we calculate the implied call volatilities for APPLE on November 25, 2021, as is shown in Figure 1. The implement python code is attached in 8.1. The graph shows three different maturities for different strike price. Closing prices for today is 161.94. Differences across strikes for implied volatility is very large. It shows a lopsided volatility smile. For out-of-the-money calls the implied volatility is greatest, for near-the-money options and in-the-money calls, the volatility is lower and the variation range is very small.

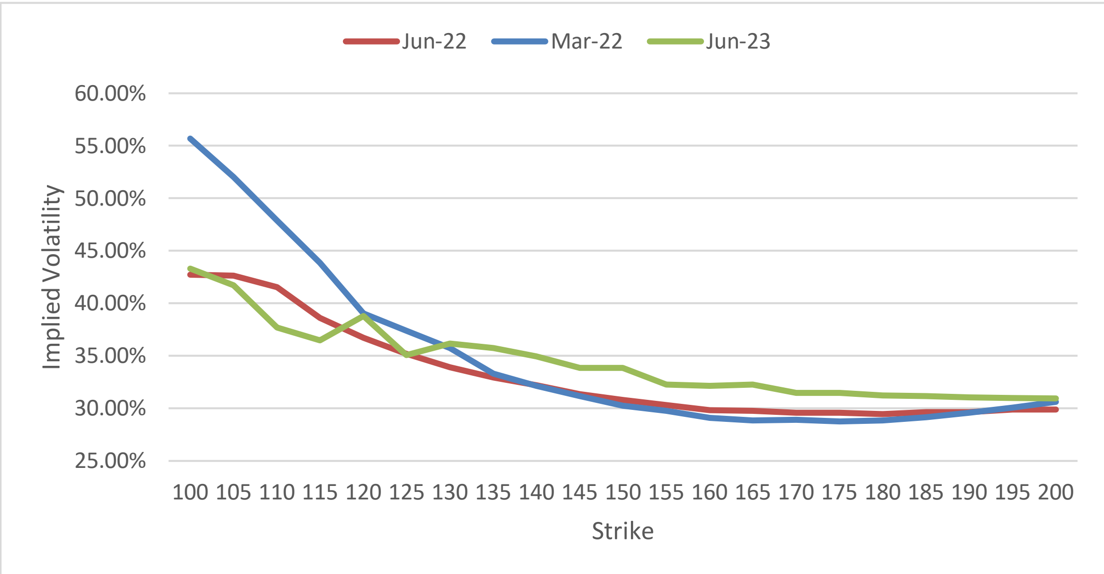

FIGURE 1 IMPLIED CALL VOLATILITIES FOR APPLE

2.4 Dividend yield

According to APPL's dividend return law, dividends are paid four times a year and each payment doesn't need to be the same. In 2021, AAPL paid dividend as Table 4. Calculate dividend yield as discrete dividends rather than continuous dividend.

Suppose there are n times of dividend payments made in one year. Each year, AAPL is expected to make dividend payments of D_{t_i} at times $t_i, i = 1,2, \dots \dots, n$, the dividend rate is r_i, it can be calculated as formula (2.5).

$$\mathrm{r_i} = {D_{t_i}}/{S_{t_i}}. \tag{2.5}$$

Denote δ as the dividend yield for one year, δ can be calculated as formula (2.6).

$$\prod_{\mathrm{i}}(1 + \mathrm{r_i}) = \mathrm{e}^{\delta} \tag{2.6}$$

TABLE 4 2021 APPL'S DIVIDEND PAYMEMT

Date payable	2021/02/05	2021/05/07	2021/08/06	2021/11/05
$\mathrm{r_i}$	0.58%	0.64%	0.60%	0.58%

According to formula (2.6) and Table 4, dividend yield $\delta = 2.3928\%$

2.5 Risk-free interest rate

The risk-free interest rate is the investor's theoretical return rate with zero risk. Because all investments contain more or less risk, the real risk-free rate can only be approximated. There are three possible ways to get risk-free interest risk in America. Method 1 is the short-term treasury bond interest rate, which always choose the interest rate on a three-month U.S. Treasury bill (T-bill). Method 2 use the cost of equity capital for the first period (year) is calculated using the historical risk premium yield of spot short-term government bonds and the market. At the same time, the forward risk-free interest rate is estimated by using the forward interest rate in the term structure as the cost of equity capital in the future. Method 3 use long-term treasury bond interest rate as risk-free interest rate.10-year bonds are the most active of all maturities of US bonds. Therefore, the yield of 10-year US bonds is widely recognized as "risk-free yield", which determines the lower limit of the yield of various assets.

The risk-free interest rate also changes with time. The interest rate of 10-year US bonds is updated every day, but we need a constant for calculation. Therefore, we choose the interest rate of 10-year US bonds of the last day settlement in every month, set r_i as the interest rate of month i, take the average of 12 months, set it as r.

$$\mathrm{r} = 1/12 \sum_{\mathrm{i}=1}^{12} \mathrm{r_i} \tag{2.7}$$

Using the 10-year US interest rate of last 12 months and equation (2.7), acquire r =1.4960%.

then convert it into the compound interest rate of the current year, set it as r_f.

$$(1 + \mathrm{r}) = \mathrm{e}^{\mathrm{r}_f} \tag{2.8}$$

Get $r_f = 1.4849\%$.

3 Pricing the compound option using binomial model

3.1 The forward tree of the European call option C

The forward tree of the European call option C is shown as Figure 2. Let C_u and C_d represent the value of the call option C when the underlying stock goes up or down, respectively. Let N denote the period of time, when $N = n$, there are $n + 1$ possibilities of price of call option C's value. Denote node $C_{u^k d^{n-k}}$ as the value of option C if the underlying stock goes up k times and goes down $n - k$ times. At node $C_{u^k d^{n-k}}$, the stock price is calculated as formula (3.2).

$$S = C_n^k u^k d^{n-k} \times S_0. \tag{3.2}$$

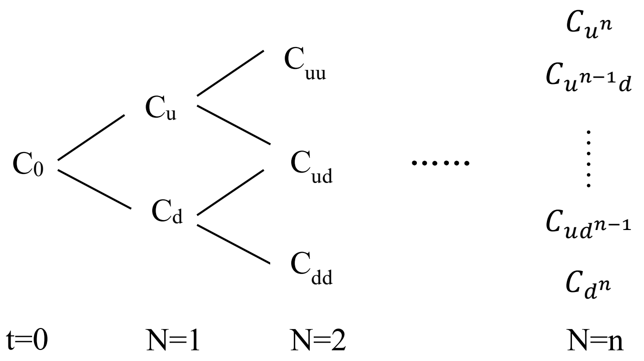

FIGURE 2 FORWARD TREE OF EUROPEAN CALL OPTION C

3.2 The forward tree of the compound option Y

The forward tree of the compound option Y is shown as Figure 3. Let Y_u and Y_d represent the value of the compound option when the underlying stock goes up or down, respectively. Let N denote the period of time, when $N = n$, there are $n + 1$ possibilities of compound option Y's value. Denote node $Y_{u^k d^{n-k}}$ as the value of option Y if the underlying stock goes up k times and goes down $n - k$ times.

If Y is a European option, Y can be exercised only at T_1; If Y is an American option, Y can be exercised at any time t_{ex} before T_1.

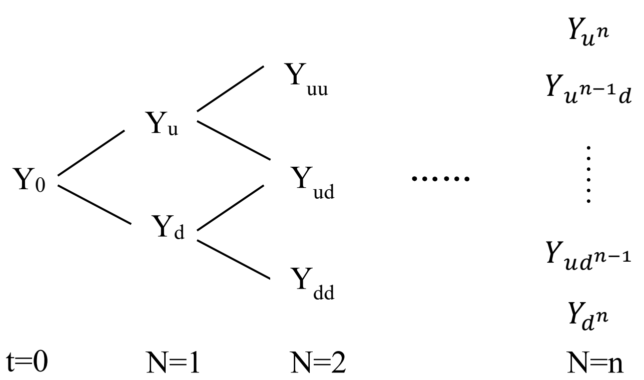

FIGURE 3 FORWARD TREE OF COMPOUND OPTION Y

3.3 Compound binomial tree

Combine the two forward trees above, construct a compound forward tree. At time t_{ex}, every node at the end of the option Y's binomial tree will be the original node of a new binomial tree of option C, as Figure 4.
If option Y is European option, $t_{ex}=T_1$; if option Y is American option, t_{ex} could be anytime within T_1. Since option C is a European call option with T_2 to expiration (timing from t_{ex}), the last node will be end at $t_{ex}+T_2$.

$T=0$	……		$T=T_1$	…… $T=t_{ex}+T_2$	European Y
$T=0$ ……	$T=t_{ex}$	……	$T=T_1$	…… $T=T_1+T_2$	American Y

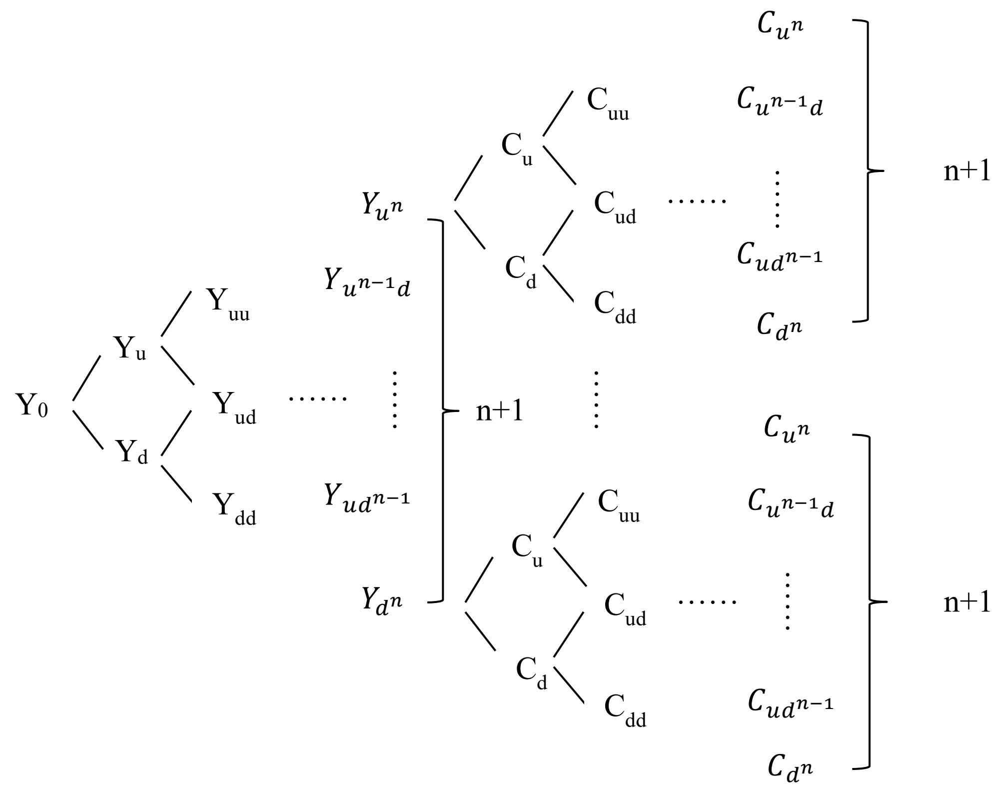

FIGURE 4 COMBINATION OF FORWARD TREES

3.4 Pricing Derivative Y

3.4.1 Pricing results

With the estimated values in the previous section, use the binomial tree of the stock as Figure 4 to price option Y. The implement python code is attached in 8.2.
When option Y is European option, $t_{ex}=T_1$; if option Y is American option, t_{ex} could be anytime within T_1. Since the duration of option contracts in the market are generally calculated by quarters, discuss T_1 and T_2 in cases of 0.25, 0.5, 0.75, 1 enables us to compute Y's value in a more realistic way.

Current stock price of AAPL (S_0) is \$161.94, hence, discuss cases when K_2 is 170, 160, 150, which stands for cases of in-the-money option, at-the-money option, and out-of-the-money option.

Discuss case by case when T_1, T_2, K_2 are different. When option Y is an at-the-money option, K_1 is close to the value of option C at t_{ex}. K_1 varies as change in T_1, T_2, K_2, as is shown in Table 5. The option value of Y, which is related to T_1, T_2, K_2, K_1, varies as change in T_1, T_2, K_2, as is shown in Table 6. Note that values with asterisk (*) are situations when option Y is an American option.

TABLE 5 VALUE OF K_1 WHEN T_1, T_2, K_2 VARY (UNIT: \$)

	T_1 (years)											
T_2 (year)	**0.25**			**0.5**			**0.75**			**1**		
	K_2			**K_2**			**K_2**			**K_2**		
	170	**160**	**150**	**170**	**160**	**150**	**170**	**160**	**150**	**170**	**160**	**150**
0.25	**7**	11	17	11	15	21	14	18	23	18	22	28
0.5	11	15	21	14	18	23	18	22	28	19	23	29
0.75	14	18	23	18	22	28	19	23	29	21	26	31
1	18	22	28	19	23	29	21	26	31	24	28	33

TABLE 6 OPTION VALUE OF Y WHEN T_1, T_2, K_2 VARY (UNIT: \$)

	T_1 (years)											
T_2 (year)	**0.25**			**0.5**			**0.75**			**1**		
	K_2			**K_2**			**K_2**			**K_2**		
	170	**160**	**150**	**170**	**160**	**150**	**170**	**160**	**150**	**170**	**160**	**150**
0.25	2.66	3.81	4.78	3.64	4.94	5.95	4.59	5.89	7.24	5.12	6.28	7.34
	2.68*	3.82*	4.80*	3.65*	4.95*	5.97*	4.60*	5.92*	7.29*	5.15*	6.33*	7.41*
0.50	2.14	3.02	3.64	3.30	4.32	5.41	3.94	4.91	5.74	5.03	6.11	7.00
	2.14*	3.02*	3.65*	3.30*	4.32*	5.42*	3.95*	4.93*	5.77*	5.05*	6.14*	7.05*
0.75	1.86	2.50	3.26	2.71	3.49	4.02	3.93	4.80	5.51	4.76	5.47	6.44
	1.86*	2.50*	3.27*	2.72*	3.49*	4.02*	3.93*	4.82*	5.53*	4.77*	5.49*	6.48*
1.00	1.30	1.82	2.12	2.75	3.43	3.87	3.71	4.27	5.02	4.28	5.11	5.90
	1.30*	1.82*	2.12*	2.75*	3.44*	3.87*	3.71*	4.28*	5.03*	4.29*	5.13*	5.93*

(*) means the value of American option Y

As can be seen from the data, when other conditions remain unchanged, when the exercise price of C, namely K_2, rises, the price of at-the-money option Y decreases gradually. As T_2 increases, the price of at-the-money option Y decreases; As T_1 goes up, the price of at-the-money option Y keeps going up; And under the same circumstances, the European option price is less than the American option price.

3.4.2 Discussion when controlling variables

Let's assume a case with following condition: T_1=0.75, K_1=\$23, T_2=0.25, K_2=\$150, n=50, S_0=\$161.94, discuss when one variable varies while others controlled, the change in option Y's value.

1) T_1 varies while other variables remain unchanged.

As is shown in Table 7, option Y's price increases as T_1 increases. In general, the farther the expiration date of an option, the longer the duration of the option, so the option price of Y increases with the increase of T_1.

TABLE 7 OPTION Y'S PRICE AS T_1 CHANGES

T_1	0.25	0.50	0.75	1.00	1.25
Option Y's price	2.88	5.33	7.24	8.80	10.15

2) K_1 varies while other variables remain unchanged.

As is shown in Table 8, option Y's price decreases as K_1 increases. It's obvious that the higher the strike price of an option is, the lower the premium of this option will be.

TABLE 8 OPTION Y'S PRICE AS K_1 CHANGES

K_1	19	21	23	25	27
Option Y's price	8.57	7.83	7.24	6.64	6.04

3) T_2 varies while other variables remain unchanged.

As is shown in Table 9, option Y's price does not change in a pattern as T_2 increases. Generally, when T_2 increases, the price of the underlying call option C will go up (but not absolutely). However, a decrease of the price of the underlying call option C will lead to a decrease of the price of option Y.

TABLE 9 OPTION Y'S PRICE AS T_2 CHANGES

T_2	0.10	0.20	0.25	0.50	0.75
Option Y's price	7.30	7.25	7.24	7.24	7.33

4) K_2 varies while other variables remain unchanged.

As is shown in Table 10, option Y's price decreases as K_2 increases. It's obvious that the higher the strike price of the call option C is, the lower the value of option C will be. Since option C is the underlying asset of option Y, the price of option Y decreases as well.

TABLE 10 OPTION Y'S PRICE AS K_2 CHANGES

K_2	130.00	140.00	150.00	160.00	170.00
Option Y's price	15.65	10.84	7.24	4.66	2.91

5) n varies while other variables remain unchanged.

As is shown in Table 11, as binomial period n increases, the price of option Y converges to a definite value. Dividing the time to expiration into more periods will generate a more realistic tree. Hence, when n increases, the pricing of option Y will be more accurate.

TABLE 11 OPTION Y'S PRICE AS N CHANGES

n	1.00	5.00	10.00	50.00	100.00
Option Y's price	9.28	6.93	7.44	7.24	7.20

6) S_0 varies while other variables remain unchanged.

As is shown in Table 12, option Y's price increases as S_0 increases. Knowing the dividend and the volatility of a stock, the greater the stock price at time 0, the greater the expectation of stock price at time t. When the strike price of call option C remains the same, option C's value increases as the increase of S_t. Since option C is the underlying asset of option Y, the price of option Y increases as well.

TABLE 12 OPTION Y'S PRICE AS S_0 CHANGES

S_0	141.94	151.94	161.94	171.94	181.94
Option Y's price	2.03	4.10	6.55	11.55	17.04

4 Pricing the compound option using Monte Carlo simulation

4.1 Geometric Brownian motion

Write both the drift and volatility as functions of X:

$$dX(t) = \alpha[X(t)]dt + \sigma[X(t)]dZ(t) \tag{4.1}$$

This equation, in which the drift, α, and volatility, σ, depend on X(t), is called an Itô process. Suppose we modify arithmetic Brownian motion to make the instantaneous mean and standard deviation proportional to X(t):

$$\frac{dX(t)}{X(t)} = \alpha \mathrm{d}t + \sigma \mathrm{d}Z(t) \tag{4.2}$$

This equation says that the dollar mean and standard deviation of $X(t)$ are $\alpha X(t)$ and $\sigma X(t)$, proportional to the level of $X(t)$. Thus, the percentage change in $X(t)$ is normally distributed with instantaneous mean α and instantaneous variance σ^2. The process in equation (4.2) is known as geometric Brownian motion.

A variable that follows geometric Brownian motion is lognormally distributed. While X is not normal, $\ln[X(t)]$ is normally distributed:

$$X(t) = X(0)e^{(\alpha - 0.5\sigma^2)t + \sigma\sqrt{t}Z} \tag{4.3}$$

where $Z \sim N(0,1)$.

4.2 Introduction to Monte Carlo simulation

Let n represent the number of binomial steps and i the number of stock price down moves. We can value a European call option by computing the expected option payoff at the final node of the binomial tree and then discounting at the risk-free rate. Hence, European call price can be calculated as (4.4).

$$e^{-rT} \sum_{i=0}^{n} \max\,[0, Su^{n-i}d^{i} - K](p^*)^{n-i}(1-p^*)^{i} \frac{n!}{(n-i)!\,i!} \tag{4.4}$$

In Monte Carlo valuation, we perform a calculation similar to that in equation (4.4). The option payoff at time T is a function of the stock price, S_T represent this payoff as $V\,(S_T\,, T\,)$. The time-0 Monte Carlo price, $V\,(S_0\,, 0)$, is as (4.5)

$$V\,(S_0\,, 0) = \frac{1}{n} e^{-rT} \sum_{i=1}^{n} V\left(S_T^i\,, T\right) \tag{4.5}$$

where $S_T^1, \cdots\cdots, S_T^n$ are n randomly drawn time-T stock prices.

4.3 Application of Monte Carlo simulation

4.3.1 Monte Carlo Valuation (in the European Call case)

It can be seen from the forward tree as is shown in Figure 4 that, each node at the end of the option Y's binomial tree will be the original node of a new binomial tree of option C. For a $n-1$ step binomial tree, to estimate the value of European option Y by working backward through the binomial tree, all of the n final option values should be fathomed first. The value of option Y at T_1 is related to the value of call option C, which can also be estimated by working backward through the binomial tree. Denote $C_j (j \in (1, n))$ as the value of option C at T_1. To make sure that C_j is independent identically distributed, the estimation of C_j must be done separately. Hence, the Monte Carlo Valuation process can be divided into two separate processes.

1) Monte Carlo Valuation of the European call option C

We assume that the stock price follows the geometric Brownian motion of equation (4.2), obtained by setting $\alpha = r - q$. We generate random standard normal variables, Z, substitute them into equation (4.3), and generate many random future stock prices. Each Z creates one trial. Suppose we compute N trials. For each trial, i, we compute the value of a call as equation (4.6).

$$\max\left(0, S_{T_1+T_2}^{i} - K_2\right) = \max\left(0, S_{T_1} e^{(r-q-0.5\sigma^2)T_2 + \sigma\sqrt{T_2} Z_i} - K_2\right), i = 1 \ldots \ldots, N \tag{4.6}$$

Average the resulting values:

$$\frac{1}{N} \sum_{i=1}^{n} \max\left(0, S_{T_1+T_2}^{i} - K_2\right) \tag{4.7}$$

This expression gives us an estimate of the expected option payoff at time $T_1 + T_2$. We discount the average payoff back at the risk-free rate in order to get an estimate of the option value at T_1:

$$\bar{C} = \frac{1}{N} e^{-rT_2} \sum_{i=1}^{n} \max\left(0, S_{T_1+T_2}^{i} - K_2\right) \tag{4.8}$$

2) Monte Carlo Valuation of the compound option Y

Repeat n times of Monte Carlo Valuation of the European call option C above, then we acquire $\overline{C_j}, (j \in (1, n))$. For j, we compute the value of the compound option Y as (4.9).

$$\max\left(0, \overline{C_j} - K_1\right) \tag{4.9}$$

Average the resulting values:

$$\frac{1}{N} \sum_{i=1}^{n} \max\left(0, \overline{C_j} - K_1\right) \tag{4.10}$$

We discount the average payoff back at the risk-free rate in order to get an estimate of the option value at T_0:

$$\bar{Y} = \frac{1}{N} e^{-rT_1} \sum_{i=1}^{n} \max\left(0, \overline{C_j} - K_1\right) \tag{4.11}$$

4.3.2 Comparation to previous results

Assuming that S_t follows the geometric Brownian motion, the price of the derivative Y is estimated in the European case using Monte Carlo simulation. The implement python code is attached in 8.3.

Perform 100 times of Monte Carlo simulation to price option Y. In the previous section, we have priced option Y using binomial tree. Denote Y_1 as the option's price estimated by Monte Carlo simulation, Y_2 as the option's price estimated by binomial tree. At T_1=0.25, the estimated price of Y when using Monte Carlo simulation and binomial tree in different situations are respectively shown as Table 13. It can be seen from the table that the Monte Carlo simulation results are similar to the results obtained by binomial tree method.

TABLE 13 COMPARATION OF RESULTS (100 SIMULATIONS)

	T_2 (years)											
	0.25			0.50			0.75			1.00		
	K_2			K_2			K_2			K_2		
	170	160	150	170	160	150	170	160	150	170	160	150
Y_1	2.45	4.03	4.54	1.99	2.88	3.99	1.53	2.34	2.81	1.53	1.66	2.01
Y_2	2.66	3.81	4.78	2.14	3.02	3.64	1.86	2.50	3.26	1.30	1.82	2.12

Increase the times of simulation and compare the results to previous results, the difference between the price estimated by Monte Carlo simulation and that estimated by binomial tree changes as simulation increases is shown in Figure 5. With the increase of simulations, the pricing of option Y is more realistic. Hence, the difference of the two pricing methods is even smaller.

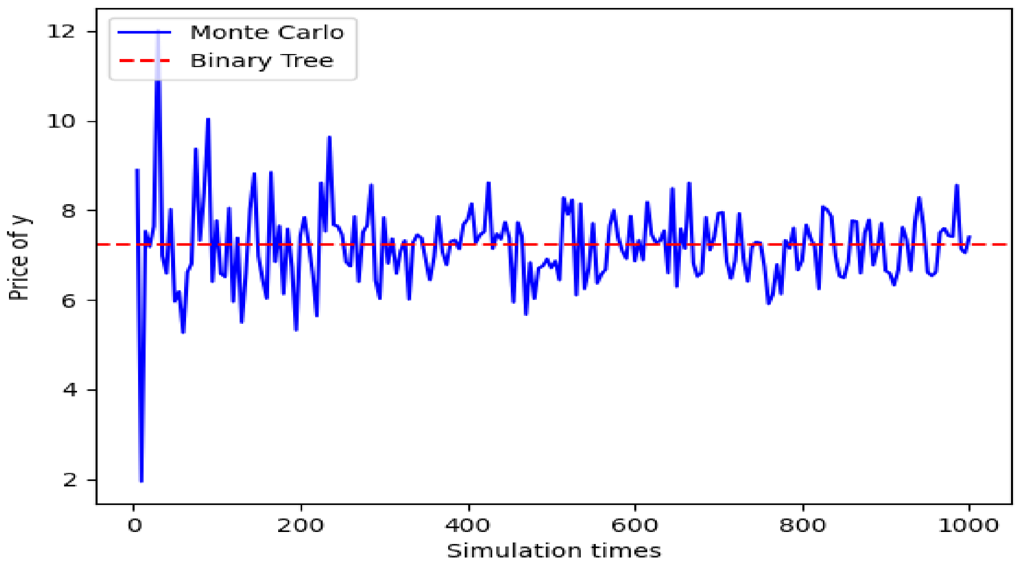

FIGURE 5 DIFFERENCE OF TWE METHODS AS SIMULATIONS INCRESE

5 Option Greeks in binomial model

5.1 Delta (Δ)

5.1.1 Calculation of delta

Delta is defined as the number of shares in the portfolio that replicates the option. It is also sensitive of the option price to a change in the shock price. The formula for delta can be written as (5.1).

$$\Delta = \frac{\partial V}{\partial S} \tag{5.1}$$

In this section, we computing delta using binomial tree. The option price and stock price can be represented as in (5.2),

$$\Delta(S, 0) = e^{-\delta h} \frac{Y_u - Y_d}{uS - dS} \tag{5.2}$$

where δ is the dividend rate of the stock, Y_u is the option value when stock price goes up, Y_d is the option value when stock price goes down, u is one plus the rate of capital gain on the stock if it goes up, and d is one plus the rate of capital loss if it goes down (same meanings in the following section).

5.1.2 Delta for Derivative Y in both the European and American cases

Estimate delta in both cases under several circumstances, the results are shown in Table 14. The implement code is attached in 8.2. Draw line charts to illustrate the change of delta as stock price changes in European case (Figure 6) and in American case (Figure 7). It can be seen that the values of delta in two cases are similar.
For both the European and American call option, delta is positive and increases as share prices rise. Because delta can be regarded as the sensitivity of the option price to a change in the stock price. The two figures illustrate that an in-the-money option will be more sensitive to the stock price than an out-of-the-money option. For deep in-the-money (i.e., the stock price larger than \$220 is high relative to the strike price of \$161), delta approaches 1, it is likely to be exercised. For deep out-of-the-money (i.e., the stock price near \$100) , it is unlikely to be exercised and the option behaves like a position with very few shares, delta approaches 0. An at-the-money option may or may not be exercised and, hence, behaves like a position with between 0 and 1 share.

TABLE 14 DELTA FOR Y IN BOTH CASES

Condition	T_1=0.25 K_1=7 T_2=0.25 K_2=170	T_1=0.25 K_1=11 T_2=0.25 K_2=160	T_1=0.25 K_1=17 T_2=0.25 K_2=150
Delta in Eur. case	0.248506	0.330804	0.395873
Delta in Amer. case	0.2485440	0.331001	0.396621

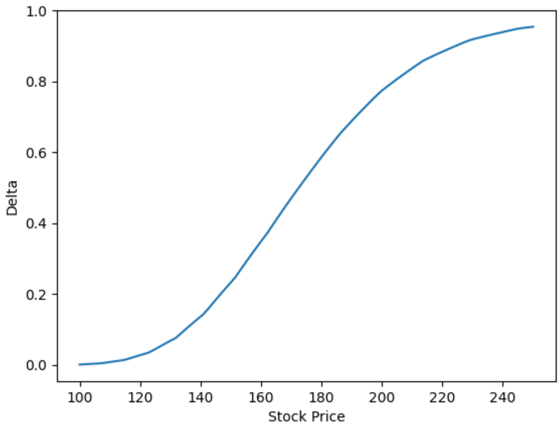

FIGURE 6 DELTA IN EUROPEAN CASE

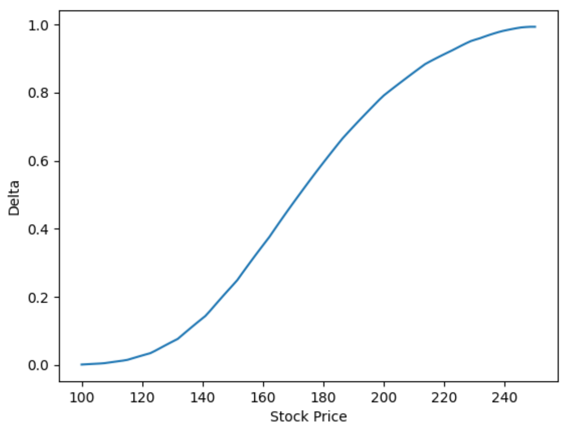

FIGURE 7 DELTA IN AMERICAN CASE

5.2 Gamma（Γ）

5.2.1 Calculation of gamma

Gamma is defined as the change in delta as the stock price changes. The calculation formula for gamma can be written as (5.3).

$$\Gamma = \frac{\partial^2 V}{\partial S^2} \tag{5.3}$$

In this section, we computing gamma using binomial tree. We cannot compute the change in delta for the time-0 delta, since only one delta at a single stock price is defined at that point. But we can compute the change in delta at time h using the two deltas that are defined there. Thus, we have (5.4).

$$\Gamma(S_h, h) = \frac{\Delta(uS,\mathrm{h}) - \Delta(dS,\mathrm{h})}{uS - dS} \tag{5.4}$$

This is an approximation since we wish to know gamma at time 0, not at time h, and at the price S_0. However, even with a small number of binomial steps, the approximation works reasonably well.

5.2.1 Gamma for Derivative Y in both the European and American cases

Estimate gamma in both cases under several circumstances, the results are shown in Table 15. The implement code is attached in 8.2. Draw line charts to illustrate the change of gamma as stock price changes in European case (Figure 8) and in American case (Figure 9). It can be seen that the values of gamma in two cases are similar.
Gamma is the change in delta as the stock price changes, so is always positive for a purchased call (both European and American).Deep in-the-money options have a delta of about 1,delta cannot change much as the stock price move, so the gamma is about 0.Similarly, deep out-of-the-money options have a delta of about 0 and, hence, a gamma of about 0.We know from above graphs that delta is always increasing, so the gamma should always be positive, thus the derivative is said to be convex.

TABLE 15 GAMMA FOR Y IN BOTH CASES

Condition	T_1=0.25 K_1=7 T_2=0.25 K_2=170	T_1=0.25 K_1=11 T_2=0.25 K_2=160	T_1=0.25 K_1=17 T_2=0.25 K_2=150
Gamma in Eur. case	0.016387	0.019467	0.021255
Gamma in Amer. case	0.016393	0.019497	0.021349

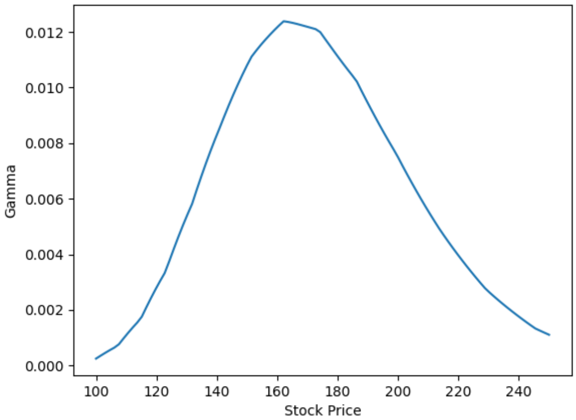

FIGURE 8 GAMMA IN EUROPEAN CASE

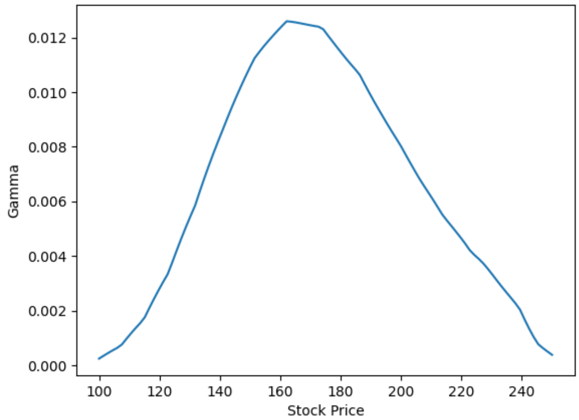

FIGURE 9 GAMMA IN AMERICAN CASE

5.3 Theta（θ）

5.3.1 Calculation of theta

Theta is defined as the change in option price when time to maturity decreases by 1 day. The calculation formula for theta can be written as (5.5).

$$\theta = \frac{\partial V(t,T)}{\partial t} \tag{5.5}$$

With theta we are interested in the pure effect on the option price of changing time. We can calculate this using delta and gamma. Define

$$\epsilon = udS - S \tag{5.6}$$

$$Y(S_{t+h},\ T-t-h) = Y(S_t,\ T-t) + \epsilon\Delta(S_t,\ T-t) + \frac{1}{2}\epsilon^2\Gamma(S_t,\ T-t) + h\theta(S_t,\ T-t) \tag{5.7}$$

Using the delta-gamma-theta approximation, equation (5.7), the option price at time $2h$ and node udS as (5.8)

$$Y(udS, 2h) = Y(S,0) + \epsilon\Delta(S,0) + \frac{1}{2}\epsilon^2\Gamma(S,0) + 2h\theta(S,0) \tag{5.8}$$

Finally, solve for $\theta(S,0)$ as (5.9).

$$\theta(S,0) = \frac{Y(udS,2h) - \epsilon\Delta(S,0) - \frac{1}{2}\epsilon^2\Gamma(S,0) - Y(S,0)}{2h} \tag{5.9}$$

5.3.1 Gamma for Derivative Y in both the European and American cases

Estimate delta in both cases under several circumstances, the results are shown in Table 16. The implement python code is attached in 8.2. Draw line charts to illustrate the change of delta as stock price changes in European case (Figure 10) and in American case (Figure 11).

Options generally becomes less valuable as time to expiration decreases, except in cases that we want to exercise the options early if possible, for example, deep in-the money call options on an asset with a high dividend yield and deep in-the-money puts. In these cases, for European options, since we cannot early exercise, the option price increases as it gets close to expiration. Theta measures the change in the option price when there is a decrease in the time to maturity of 1 day. Theta shows a trend of decreasing first and then increasing. For at-the-money option, theta reaches minimum value. For deep in-the-money and deep out of money, it approaches 0. However, there exists a slight difference. In American option, the whole curve is negative, but in European option, theta exceeds 0 for deep in-the-money price.

TABLE 16 THETA FOR Y IN BOTH CASES

Condition	T_1=0.25 K_1=7 T_2=0.25 K_2=170	T_1=0.25 K_1=11 T_2=0.25 K_2=160	T_1=0.25 K_1=17 T_2=0.25 K_2=150
Gamma in Eur. case	-9.839216	-11.627394	-12.634873
Gamma in Amer. case	-9.843362	-11.645925	-12.692523

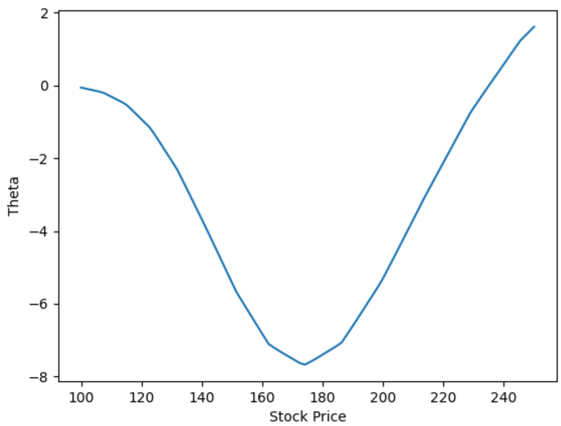

FIGURE 10 THETA IN EUROPEAN CASE

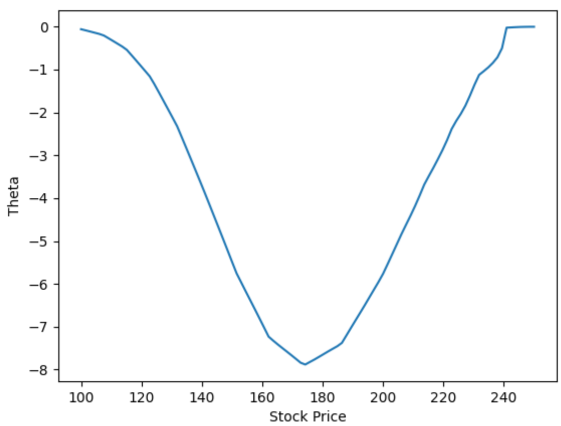

FIGURE 11 THETA IN EUROPEAN CASE

6 Further study: when underlying option is American style

In above article, we have discussed the case when option C is a European option. Now consider the case if the option Y's underlying call option C is an American option.

If option Y is European option, $t_{ex}=T_1$; if option Y is American option, t_{ex} could be anytime within T_1. Now denote the exercise time of option C as t_{ex2} (timing from t_{ex}), $0\leqslant t_{ex2}\leqslant T_2$. The binomial tree is constructed as shown in Figure 12.

$T=0$ …… $T=T_1$ …… $T=T_1+t_{ex2}$ …… $T=T_1+T_2$ European Y

$T=0$ …… $T=t_{ex}$ …… $T=T_1$ …… $T=T_1+t_{ex2}$ …… $T=T_1+T_2$ American Y

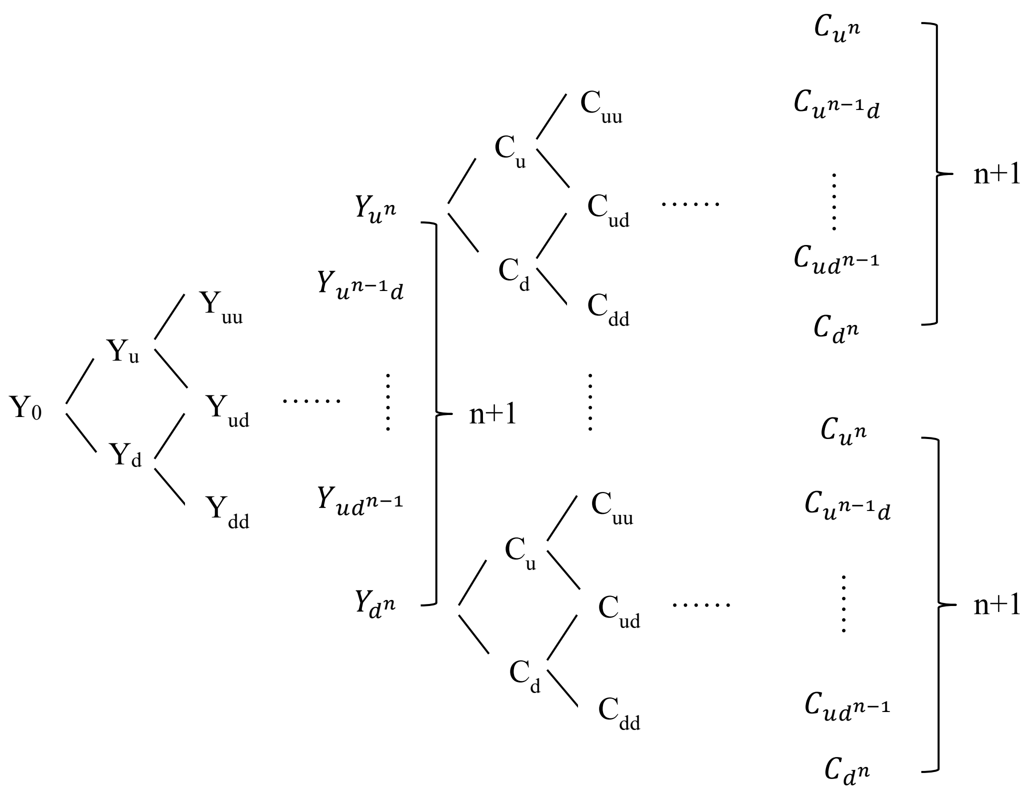

FIGURE 12 BINOMIAL TREE WHEN C IS AN AMRICAN OPTION

With the estimated values in the previous section, use the binomial tree of the stock to price option Y. This time, option Y's underlying call option C is an American option. The time when option C exercised is not $t_{ex}+T_2$ anymore, it can be exercised within a period of time from t_{ex} to $t_{ex}+T_2$. The implement python code is attached in 8.2.

When option Y is European option, $t_{ex}=T_1$; if option Y is American option, t_{ex} could be anytime within T_1. Similarly with previous study, discuss T_1 and T_2 in cases of 0.25, 0.5, 0.75, 1, discuss cases when K_2 is 170, 160, 150, which stands for cases of in-the-money option, at-the-money option, and out-of-the-money option.

When option Y is an at-the-money option, discuss case by case when T_1, T_2, K_2 are different. The option value of Y varies as change in T_1, T_2, K_2, as is shown in Table 17. Note that values with asterisk (*) are situations when option Y is an American option.

TABLE 17 OPTION VALUE OF Y WHEN C IS AN AMERICAN OPTION

T_2 (year)	T_1 (year)											
	0.25			0.50			0.75			1.00		
	K_2			K_2			K_2			K_2		
	170	160	150	170	160	150	170	160	150	170	160	150
0.25	2.713	3.863	4.905	3.697	5.029	6.081	4.663	6.009	7.391	5.211	6.413	7.510
	2.713*	3.865*	4.913*	3.703*	5.045*	6.110*	4.682*	6.044*	7.446*	5.247*	6.471*	7.585*
0.50	2.196	3.114	3.794	3.383	4.460	5.619	4.059	5.077	5.993	5.175	6.327	7.287
	2.196*	3.115*	3.796*	3.385*	4.467*	5.635*	4.069*	5.098*	6.029*	5.198*	6.368*	7.349*
0.75	1.927	2.628	3.455	2.825	3.652	4.284	4.081	5.045	5.840	4.953	5.720	6.816
	1.927*	2.628*	3.456*	2.826*	3.655*	4.292*	4.087*	5.058*	5.866*	4.968*	5.749*	6.864*
1.00	1.379	1.953	2.313	2.886	3.627	4.190	3.894	4.522	5.410	4.516	5.418	6.355
	1.379*	1.953*	2.314*	2.886*	3.629*	4.195*	3.897*	4.530*	5.428*	4.526*	5.439*	6.393*

(*) means the value of American option Y

7 Reference

[1] Robert L. McDonald. Derivatives Market[M]. 3rd Ed.

8 Appendix

8.1 The implement code of estimating implied volatility

```python
import math
from scipy.stats import norm

def eu_bs(s, k, t, sigma, r, q):
    """B-S formula"""
    d1 = (math.log((s * math.exp(-q * t)) / (k * math.exp(-r * t))) + 0.5 * (sigma ** 2) * t)/(sigma * math.sqrt(t))
    d2 = d1 - sigma * math.sqrt(t)
    pre = (s * math.exp(-q * t)) * norm.cdf(d1) - (k * math.exp(-r * t)) * norm.cdf(d2)
    return pre

def implied_sigma(s, k, t, r, q, c):
    """dichotomy to solve sigma reversely"""
    sigma_min = 0.00001
    sigma_max = 1000
    sigma_mid = (sigma_min + sigma_max) / 2
    call_min = eu_bs(s, k, t, sigma_min, r, q)
    call_max = eu_bs(s, k, t, sigma_max, r, q)
    call_mid = eu_bs(s, k, t, sigma_mid, r, q)
    diff = c - call_mid
    if c < call_min or c > call_max:
        print('error, the price of option is beyond the limit')
    else:
        while abs(diff) > 1e-6:
            if c > call_mid:
                sigma_min = sigma_mid
            else:
                sigma_max = sigma_mid
            sigma_mid = (sigma_min + sigma_max) / 2
            call_mid = eu_bs(s, k, t, sigma_mid, r,q)
            diff = c - eu_bs(s, k, t, sigma_mid, r,q)
    return sigma_mid
```

8.2 The implement code of constructing binomial Tree and calculating Greeks

```
import math
import pandas as pd

def binary_s(s0, u, d, node):
    """Calculate s at node(step, up, down)"""
    s = s0 * (d ** node[1]) * (u ** (node[0] - node[1]))
    return s

def forward_tree_O(n, t, r, q, sigma, s0, k, types='E', direct='C'):
    option = {}
    h = t / n
    cn = {}
    u = math.exp((r - q) * h + sigma * math.sqrt(h))
    d = math.exp((r - q) * h - sigma * math.sqrt(h))
    p = (math.exp((r - q) * h) - d) / (u - d)
    for i in range(n + 1):
        if direct == 'P':
            cn[i] = max(k - binary_s(s0, u, d, [n, i]), 0)
        else:
            cn[i] = max(binary_s(s0, u, d, [n, i]) - k, 0)
    option[n] = cn
    for i in reversed(range(n)):
        cx = {}
        for j in range(i + 1):
            cu = option[i + 1][j]
            cd = option[i + 1][j + 1]
            sx = binary_s(s0, u, d, [i, j])
            cr = math.exp(-r * h) * (p * cu + (1 - p) * cd)
            if types == 'A':
                if direct == 'P':
                    cx[j] = max(k - sx, cr)
                else:
                    cx[j] = max(sx - k, cr)
            else:
                cx[j] = cr
        option[i] = cx
    return option

def binary_tree_Y(n1, n2, t1, t2, r, q, sigma, s0, k1, k2, types='E', direct='C'):
```

```
        y = {}
        h1 = t1 / n1
        yn = {}
        u1 = math.exp((r - q) * h1 + sigma * math.sqrt(h1))
        d1 = math.exp((r - q) * h1 - sigma * math.sqrt(h1))
        p1 = (math.exp((r - q) * h1) - d1) / (u1 - d1)
        for i in range(n1 + 1):
            s = binary_s(s0, u1, d1, [n1, i])
            c = forward_tree_O(n2, t2, r, q, sigma, s, k2)[0][0]
            if direct == 'P':
                yn[i] = max(k1 - c, 0)
            else:
                yn[i] = max(c - k1, 0)
        y[n1] = yn
        for i in reversed(range(n1)):
            yx = {}
            for j in range(i + 1):
                cu = y[i + 1][j]
                cd = y[i + 1][j + 1]
                cr = math.exp(-r * h1) * (p1 * cu + (1 - p1) * cd)
                s = binary_s(s0, u1, d1, [i, j])
                c = forward_tree_O(n2, t2, r, q, sigma, s, k2)[0][0]
                if types == 'A':
                    if direct == 'P':
                        yx[j] = max(k1 - c, cr)
                    else:
                        yx[j] = max(c - k1, cr)
                else:
                    yx[j] = cr
            y[i] = yx
        return y

    def delta_y(q, n1, t1, y, u, d, s, node=None):
        if node is None:
            node = [0, 0]
        h1 = t1 / n1
        delta = math.exp(-q * h1) * (y[node[0] + 1][node[1]] - y[node[0] + 1][node[1] + 1]) / (
                binary_s(s, u, d, [node[0] + 1, node[1]]) - binary_s(s, u, d, [node[0] + 1, node[1] + 1]))
        return delta

    def gamma_y(q, n1, t1, y, u, d, s):
```

```
        gamma = (delta_y(q, n1, t1, y, u, d, s, [1, 0]) - delta_y(q, n1, t1, y, u, d, s, [1, 1])) / (
                binary_s(s, u, d, [1, 0]) - binary_s(s, u, d, [1, 1]))
        return gamma

    def theta_y(n1, t1, q, y, u, d, s):
        eps = u * d * s - s
        h = t1 / n1
        theta = (y[2][1] - eps * delta_y(q, n1, t1, y, u, d, s) - 0.5 * eps ** 2 * gamma_y(q, n1, t1, y, u, d, s) -
                 y[0][0]) / (2 * h)
        return theta
```

8.3 The implement code of Monte Carlo simulation

```
from numpy.random import randn
    import math
    import pandas as pd

    def cal_S_T(s0, sigma, r, t, q):
        """Simulation ST follows Geometric Brownean Motion """
        return s0 * math.exp((r - q - 0.5 * sigma ** 2) * t + sigma * math.sqrt(t) * randn())

    def monte_carlo(s0, sigma, r, k1, k2, t1, t2, q, simulation):
        ysum=0
        for i in range(simulation):
            csum=0
            s1 = cal_S_T(s0, sigma, r, t1, q)  # Simulate S at t1 from t0
            for j in range(simulation):
                s2=cal_S_T(s1, sigma, r, t2, q)  # Simulate S at t2 from t1
                csum+=max(s2-k2, 0)
            c=math.exp(-r *t2)*csum/simulation
            ysum+=max(c-k1, 0)
        y=math.exp(-r *t1)*ysum/simulation
        return y
```