Pricing Vanilla Options with Apple and Costco, Manifest the Greek Letters and Exploiting Exotic Options Pricing

Continuous-Time Finance

Final Project

Professor Arnab Bisi

Hangyu Li

March 9th, 2020

Question

For this project analysis, we intend to implement mechanical methods such as Monte Carlo Simulation and Black-Scholes Model to price vanilla options and test for certain consistencies. We expected the two methods to return similar results for the pricing of options. Furthermore, from the price engine, we want to test Vanilla options Black-Scholes Greeks with Monte Carlo Greeks, and to present the relationships between variables and alignments between methods. From the Black-Scholes model of Greeks, we expect to see consistent results from the empirical data sets and present them with graphs. We are also interested in investigating the above questions through data from Apple and Costco, reasons for selection will be presented later. Moreover, we want to price other exotic options besides the vanilla ones, to explore and to have an understanding of such financial products. With the bearish looking environment in the United States equity market, exotic options such as Barrier options can be implemented for hedging reasons. Under this outlook commodity trading with Asian options is a possibility for traders to utilize, thus, we intend to build a price engine for both exotic options.

Data Description

Our team chose to use historical daily prices and returns for Apple and Costco, in addition, we utilize both the call and the put options of Apple and Costco, in terms of varying exercise dates. We chose to use Apple and Costco is because they operate in very different sectors. Costco is viewed as a defensive stock, given its operation sector is in the consumer staples, whereas Apple is in a rather cyclical stock operating in the Technology sector. We wish to find very different results in their options trading because of their difference in operations. The two companies are both very successful financially in recent years, thus, we picked them as representatives for data selections. We cross-examined the option chain information from Yahoo Finance, Nasdaq and Bloomberg. We chose to use Bloomberg data to implement our analysis due to its data setup, cleanliness, and consistency. In our data set, we obtained the exercise dates which provides us with the Time to Maturity of the option, we also obtained Strike Price on that date, alongside with the implied volatility. We further use the strike price of 290 for the Appel options, which is closest to the current underlying stock price. We also use a strike price of 310 for Costco options, for the same reasons. Other variables are the daily adjusted close price of the underlying stock for the past 5 years and trading dates. Our data set also provides the implied volatility of the options for the option period. The risk-free rate is benchmarked on the LIBOR rate for the corresponding time to maturity period, this data was retired from Bloomberg and ICE database.

Methodologies

Topic 1: Vanilla Options in Black-Scholes World

Implementing the Monte Carlo Method to price call, put and forward prices: We used each of the annualized stock daily log-returns from the past year data to compute R and Sigma. Then we initiated the Geometric Brownian Motion with drift r-d, volatility sigma to the stock price by 10,000 times through random number generator, we found out the prices at different time T. Moving on, we take out the static strike price from the different stock prices to obtain our options payoffs, by discounting it back to time Zero, we achieved a set of product prices, result shown in figure 3, the Forward prices are shown in figure 4.

Consistency tests: We also conducted consistency tests for the Monte Carlo results, put-call parity was checked to be consistent with theory, the result showing in figure 5, taking out the forward and put price from the corresponding call price, we found out the results to be zeros. We also tested for the price of the call option is monotone decreasing with the strike price, shown in Figures 6 and 7. When the strike price is lower, the call option price goes higher. In figure 8 we tested consistency 3, our call options price is between the S and S-Ke^(-rT). We computed the upper bound of the call option, which is the Stock price, then the lower bound of the option price is stock price minus the present value of the strike price. Lastly, we tested for the relationship of the call option price with increasing time to maturity, as the result shown in figure 9, it is a positive sloped relationship.

Using Black-Scholes formulas to price call, put and forward prices: Once we constrained the strike prices for Apple and Costco, we chose 13 different times to maturity corresponding with Appel options, and 8 for Time to Maturity for Costco. The reason for choosing these varying times to maturity is due to the different contract set exercise dates. For risk-free selection, we used the corresponding LIBOR Rate for each option's time to maturity, this data is retrieved from Bloomberg, double-checked with the ICE benchmark. We then composed the sigma using the underlying stock's standard deviation of the log return for the past 5 years of data. Moving forward, we used these data as inputs for the Black-Scholes model to compute put, call and forward prices with varying time to maturity, and results are shown in figures 10, and the prices of corresponding zero-coupon bond prices are shown in figure 11.

Consistency tests:

The result of the consistency check for the first topic indicated the correct implementation. Firstly, after obtaining the call and put option price, we let the price of a call minus the corresponding price of the put and the result was equal to the forward price perfectly, thus the computation result satisfies the put-call parity, figure 12. Secondly, we aim to test the monotone of call pricing decreasing with the strike price, at decreasing strike price K from $300 to $280, we obtained the corresponding call prices, shown in figure 13. The result does indicate once strike price decreases, the call option price increases. Thirdly, we present the fact that the call option price is indeed between S and S-Ke^(-rT) for all the inputs from our Black-Scholes Model, figure 14, showing an at the money call option, the prices are capped by the Stock prices and floored by the intrinsic value of the option. Fourth, we are testing the consistency of the call option price increases as the volatility of the underlying increases. We fixed the strike price at $290, the risk-free rate at 0.8% and Time to maturity is 12 days, it is verified that the price does indicate a monotone increasing in volatility, result shown in figure 15. Next, we test to check if the call option price is a convex function of the strike price, we fixed all other variables and set the varying strike price of K= linspace(200, 400,100), the graph shown in figure 16. The convex relationship is no doubt presented here. We further investigated the intersection of put prices and intrinsic value and presented it on graph 17. By plotting the put prices from the Black-Scholes formula and the intrinsic value of the option against the strike prices, we can obtain the intersection of the two prices at the near-at-the money position.

Topic 2 Vanilla Greeks:

We first computed the value for each Greek letter by directly implementing the formulas deduced by using the Black-Scholes model when we specify all the necessary input values. We then test our results using the finite different price method. We first ran the model 1000 times by assigning randn as the finite different price, then reran the model by assigning a smaller value (randn*0.1) to finite different prices. We then compared two methods of computing the Greek value using the Monte Carlo method; this time we directly assigned a small value (0.1) to the finite difference value. The two methods are the Monte Carlo method using different random values and the Monte Carlo method using the same random values respectively.

Graph explanation:

The graph I: We want to further investigate some of the traits of options for different stocks, we choose to implement Greeks to see what do the financial products of Apple and Costco portray. We each implemented the Greek formulas to arrive at their values and graphed the results in the corresponding variables. The range of spot prices for graph delta vs. spot price is set from$0 to $400. Figure 19 shows the Delta of Apple option against its spot price expiring in 6 months, with a static strike price of $290. Delta for call option will be a number between 0 and 1, and the delta value can be as low as 0 when the spot stock price is low enough of a value. When we gradually increase the spot stock price, the delta value first increases at an ascending rate until the delta value reaches 0.5 and then the rate of increase begins to decline. The value of delta will be infinitely close to 1 as you further increase the spot price.

Graph II: Moving forward, we graphed the Delta with increasing time to maturity for in the money, out of the money and at the money call options, the static strike price of $260(out-of-money), S0(at-the-money) and 300 (in-the-money) were used. From Figure 20, we can see a convergence trend for three moneyness options as Time increases. Deltas for in-the-money, out-of-money, and at-the-money options converge as you increase the time-to-maturity of options.

Graph III: Gamma is our next Greek of interest, by implementing Gamma against the spot prices, we see the second derivative of the option prices changes to its underlying asset price changes. Shown in figure 21, the Gamma of Apple is greatest for options that are close to the money, the same strike price of $260 was used. The rate of change in delta with respect to the change in spot stock is more obvious in this graph. The gamma value approximately 0 when the current spot price is too big or too small a value.

Graph IV: for Vega against the stock spot price, shown in figure 22, the value of Vega peaks around the Strike price of $260. Vega measures the sensitivity to volatility, so the option value sensitivity reaches its peak level around the strike price. When we plot Vega as a function of time, figure 23, the Vega value increases at a lower rate along with the increase of time to maturity.

Topic 5: Exotic options by Monte Carlo

In order to price arithmetic Asian options discrete barrier options in a Black- Scholes World, we first set a path generator. We first use Monte Carlo to simulate stock prices. We assume stock price change follows the Geometric Brownian Motion process; the stock price should have the following formula:

$$S_T = S_0 e(r - d) - \frac{1}{2}\sigma^2 T + \sigma\sqrt{t}w$$

We set the parameters s0 = 100; vol = 0.1; r = 0.05; d = 0.03; k = 103; N = 1e5; T =1. For every maturity date, such as weekly monthly annually, We generate 100000 possible stock prices. The result is a normal distribution (figure 24), with a mean close to 100.

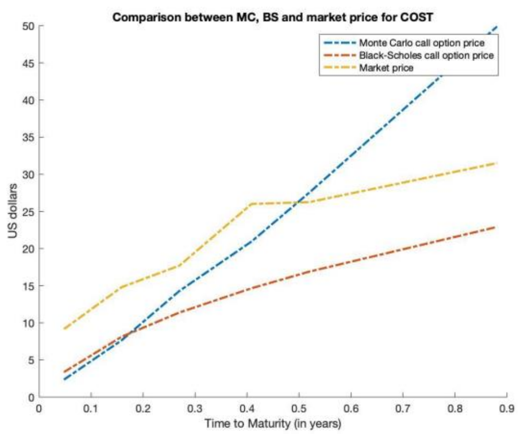
For pricing Asian call option: An Asian option is an option type where the payoff depends on the average price of the underlying asset over a certain period of time as opposed to standard options. For each possible price path (each row), take an average value and minus strike price and then take the maximum value of (St-K,0). Take the average value of the above result by the law-of-large-number, this should an unbiased estimate of option prices, which is also an expected value of an Asian call option. The expected Asian option value sheet is in figures 25 and 26.

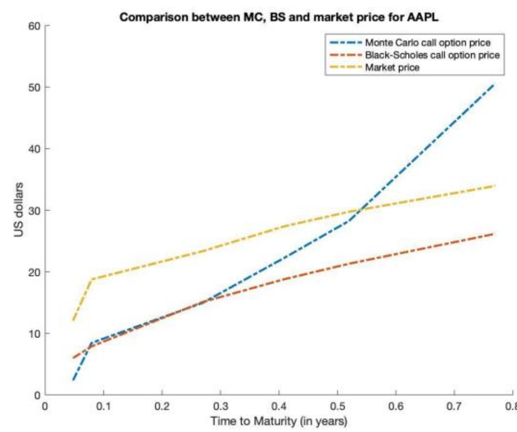
For pricing discrete barrier option: Down-and-in call is a regular call that comes into effect only if the stock price hits a barrier level from above. Use a logic term L=1 if stock price is smaller than barrier level, otherwise L=0. Then multiply the logic term by the expected option value so that only those prices dropped below the barrier would remain while others become 0. Another barrier option has a similar methodology.

Results and Conclusion

The result from topic 1: We have noticed very coherent results from the Black-Scholes and Monte Carlo Pricing of options. First, we found out the Black-Scholes model results and the Monte Carlo results have both shown consistency in all five of our consistency tests. Put-call parity was verified through the difference in Call, Put and Forward prices; the price of the call did indicate a decreasing monotone with the strike price; a usage of lower bound and upper bound presented us that the call option prices do fall within the S and S-Ke^(-rT); the consistency test also shows the call prices have a monotone increasing in volatility; lastly, the call option price is a convex function of strike. The empirical results are in alignment with our expectations at the beginning of our analyses, we have created a credible way of knowing the above intuitive relationships.

Through our comparison of models, we arrived at a very interesting result. As shown in the graph below, Figure 18. Our two models have priced very differently throughout the course of the valuation, and the results are distinctly different from the market's latest traded price. We speculate the difference from the models is caused by the fundamental different usage of data input. For the Black-Scholes Model, we applied the LIBOR rate as the risk-free rate, given the nature of risk-neutral pricing, as for the Monte Carlo simulations, we input the historical returns as R and historical volatility as Sigma. We suspect the sudden decrease in LIBOR rate may have affected the results of the Black-Scholes model, due to the global dovish monetary policy trends, the LIBOR has been reduced repeatedly in recent days. For Monte Carlo pricer, we noticed a much faster increase in prices given increases in time to maturity. We believe the high daily log returns in stock prices for both companies have caused this trend. Costco and Apple are both profitable companies with high daily returns in stock prices, as an input of the Monte Carlo pricing engine, the return has pushed the call option prices to grow faster.





The result from topic 2: The results from Vanilla Greek letter, We can actually make the conclusion that using a smaller finite difference value in calculating the Greek value could actually provide us with a more accurate result by comparing the data in the table below.

	Formula under BS	Testing Method (Big difference number)	Testing Method (Smaller ones)
Delta	0.6482	0.6448	0.6481
Gamma	0.0060	0.0060	0.0060

Viewing from the graph below, we could make the conclusion that the Monte Carlo method that using different random values will provide us with too volatile a value, a value that was far from what we calculated before. While the method that uses the same random value actually provides us a delta value closer to the previously calculated value. We can conclude that the Monte Carlo method using the same random value will be a better and more accurate way of deriving the Greek values.

	Monte Carlo method using different random values	Monte Carlo method using the same random value
Delta	11.72(too volatile)	0.6204

The result from topic 3: Having implemented the exotic pricing model, we priced Asian options with the same time to maturity but different setting times, they are monthly, three-monthly and weekly separately, given S0=100, sigma=0.1, r=0.05, d=0.03, and the strike price is 103. The computation result is shown in the table below.

Setting times	Asian options Price	Vanilla options price
three-monthly	1.3578	3.5486
monthly	1.4259	3.5549
weekly	1.4551	3.5562

From the above table, we can see that the prices of both Asian options and Vanilla options increase with the decrease of setting times, which indicates a reverse relationship between set times and options price. However, the speed of convergence varies, the convergence speed of Asian options is lower than that of Vanilla options. Then, we priced discrete barrier options for four different options, given the time to maturity is 1 year and struck at 103. The computation result is shown in the table below.

	Discrete barrier option	Discrete barrier option price	Vanilla option
(1)	**Down-and-out call with barrier at 80 and monthly barrier rates**	3.4433	(5) Vanilla call option price: 3.5549
(2)	**Down-and-in call with barrier at 80 and monthly barrier rates**	0.0412	
(3)	**Down-and-out put with barrier at 80 and monthly barrier rates**	4.3469	(6) Vanilla call option price: 4.4878
(4)	**Down-and-out put with barrier at 120 and barrier rates at 0.05,0.15, ..., 0.95**	4.1782	

From the above table we can see that (1) +(2)-(5) = -0.0704, the result is close to 0 that proves the equation: C = Cdi + Cdo valid.

Conclusion

From our extended project analyses, we managed to create two option price engines with the Monte Carlo simulation method and the classic Black-Scholes model. For our first objective topic, we tested five consistencies to reassure our model formulas were implemented correctly. Further investigation showed us that our pricing models have quite a large difference in prices compared with real market data. We conclude the possible reasons as a sudden dramatic decrease in the risk-free rate (LIBOR), and the input of historical return value for the Apple and Costco in the Monte Carlo simulation, causing further deviation from the real data. The next objective topic of Vanilla Greeks has shown us a coherent relationship of variables in the Black-Scholes model using real-world information. By further testing efforts, we have concluded that the Monte Carlo method using the same random value will be a better and more accurate way of deriving the Greek values. For the last objective topic, we were able to explore the uniqueness of Asian and Barrier options, as well as pricing them. We conclude given the restraints, the prices of these options are lower than the vanilla options, but the specialty features can be beneficial given the uncertainty in the market right now.

Appendix：

Pricing options in the real world with Apple and Costco, and exploiting exotic options trading

Hangyu Li

Note: All the figures below start with the graph of Apple, followed by the graph of Costco at the bottom

Topic 1 Monte Carlo Simulation

Figure 1

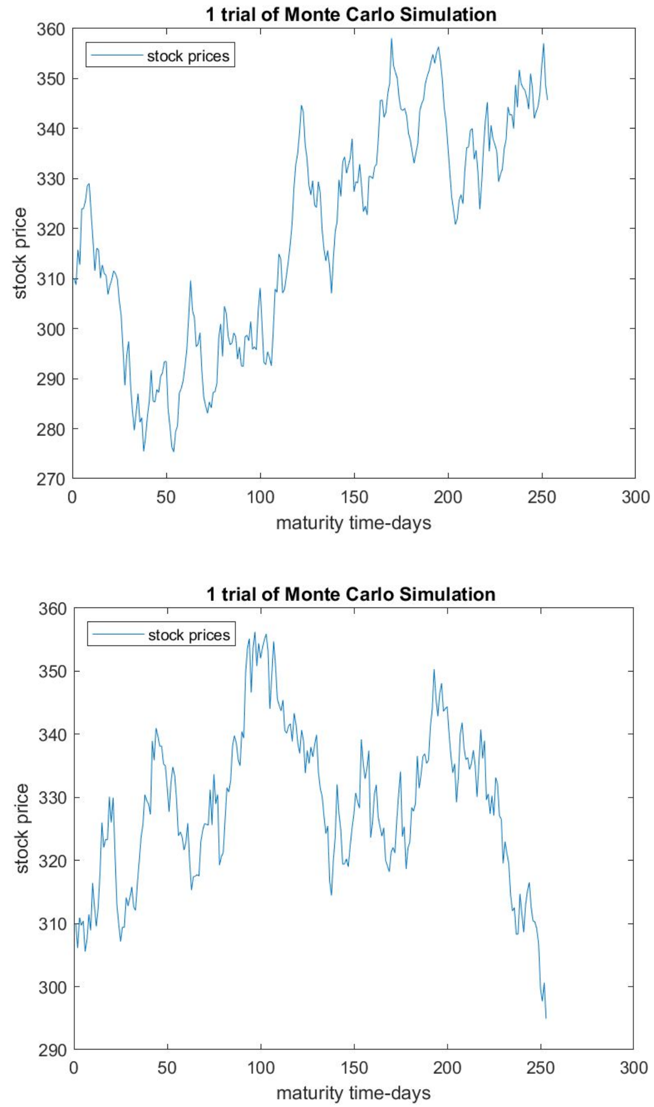


Figure 2: Stock Price Path

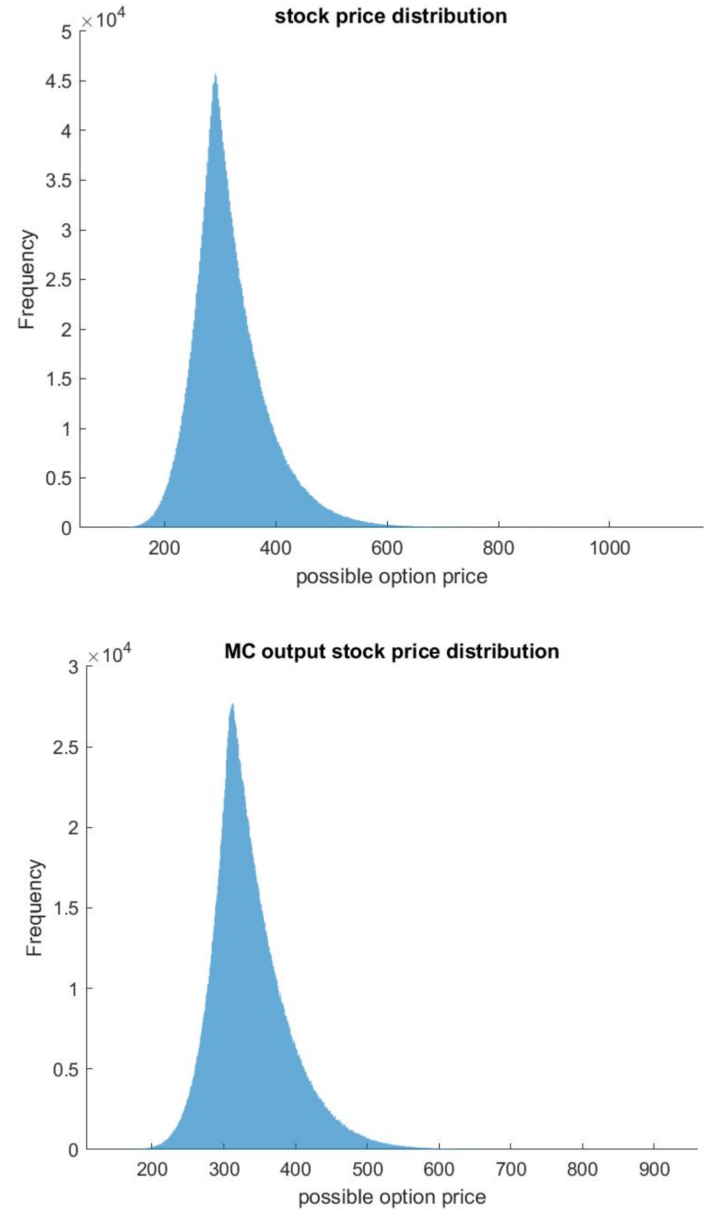


Figure 3: Option Prices (consistency 2)

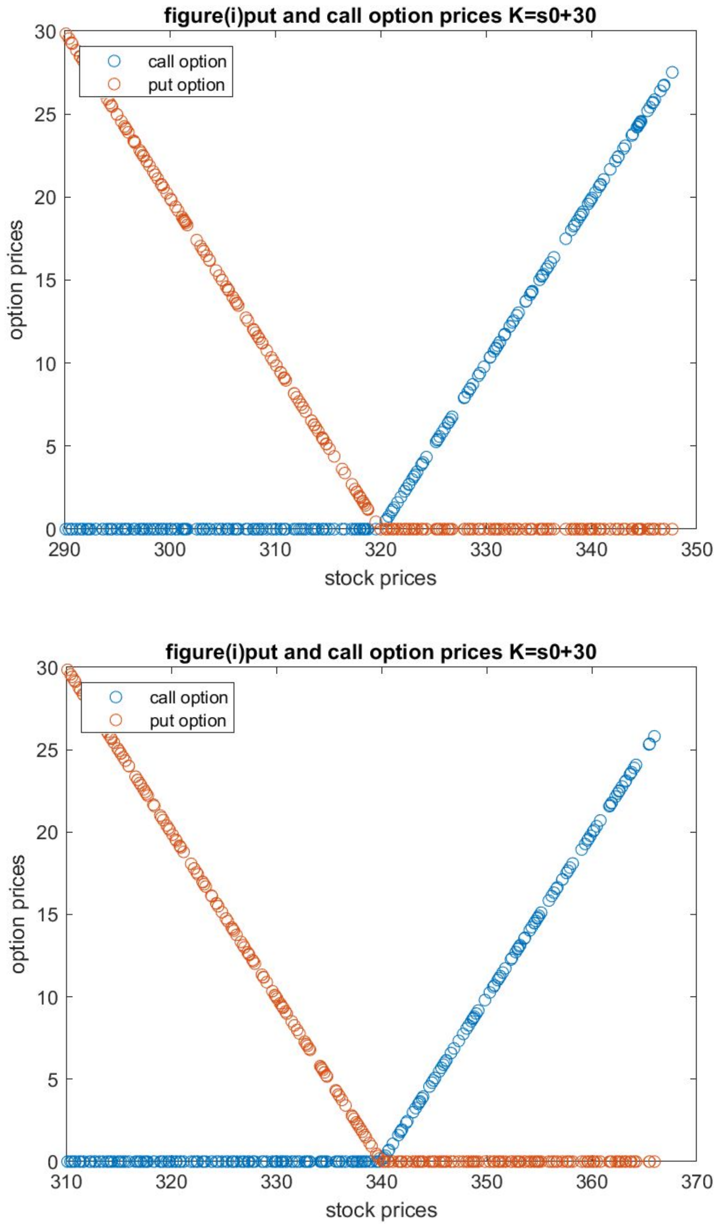


Figure 4: Forward Price(consistency 3)

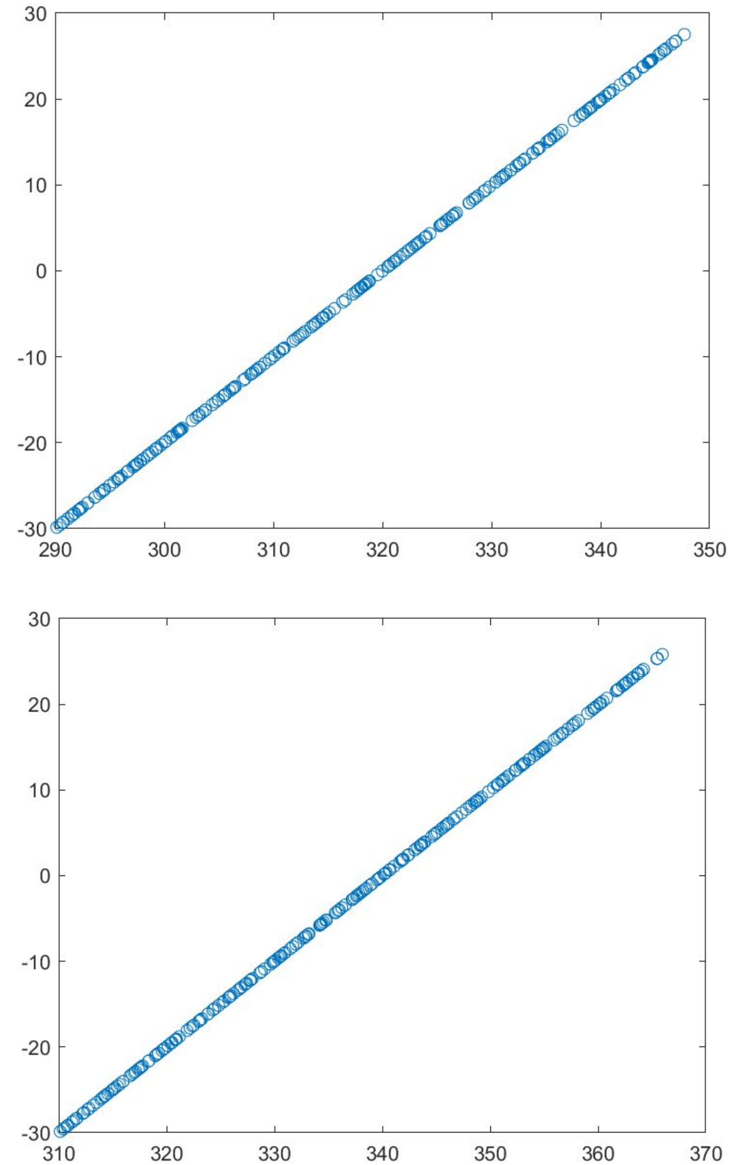


Figure 5: the put-call parity result (consistency 1)

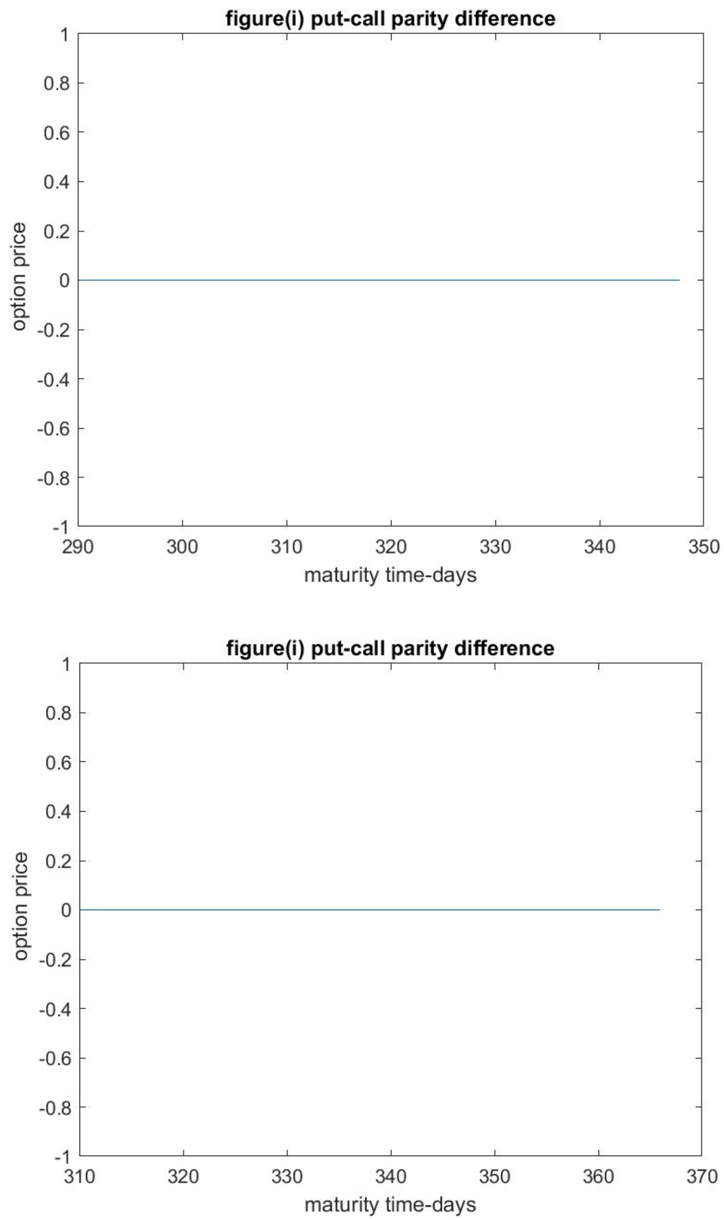


Figure 6

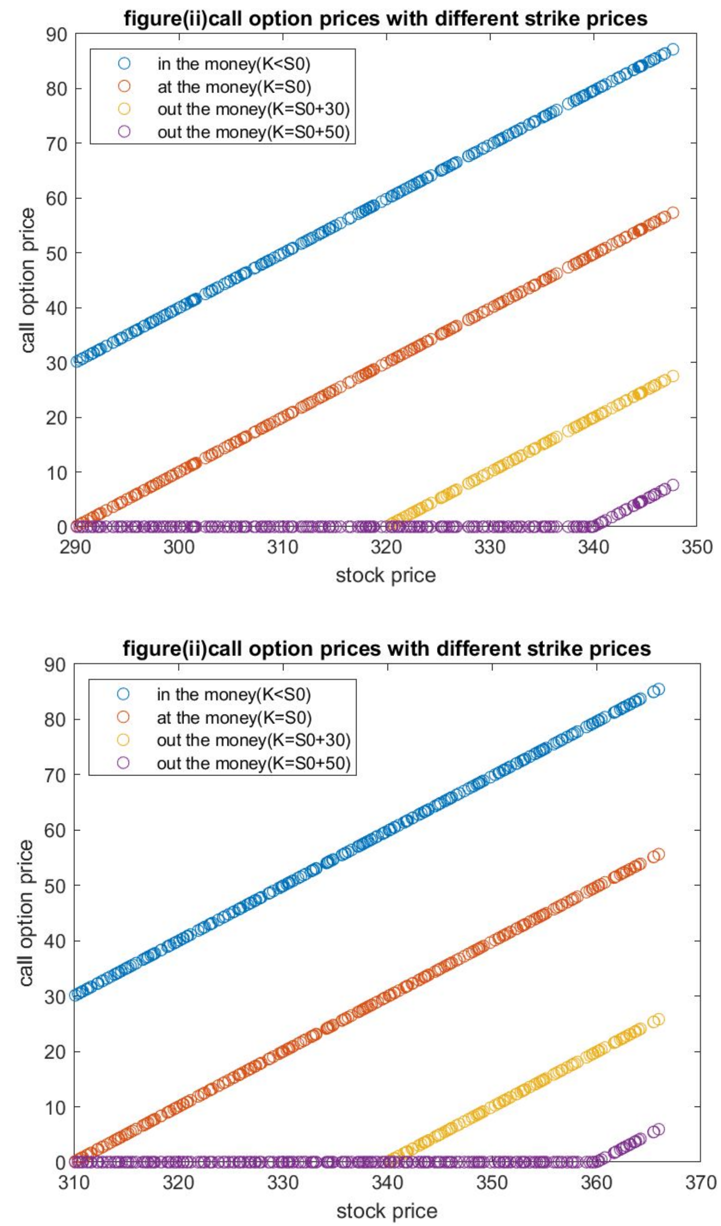
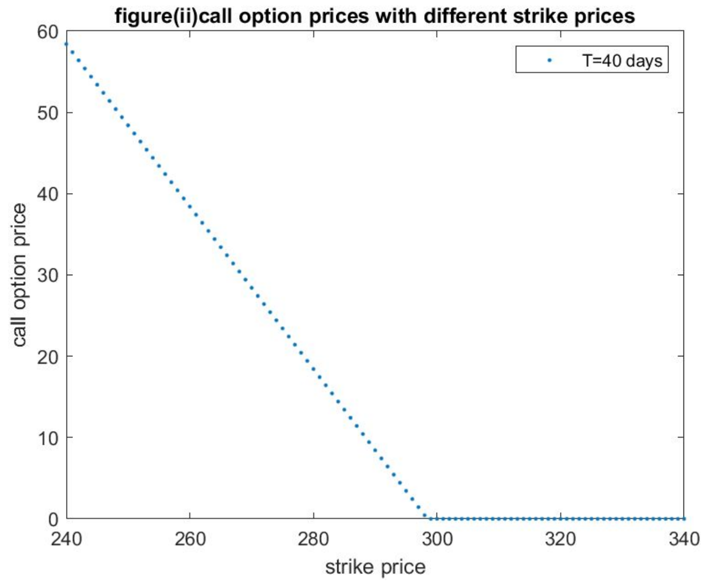


Figure 7



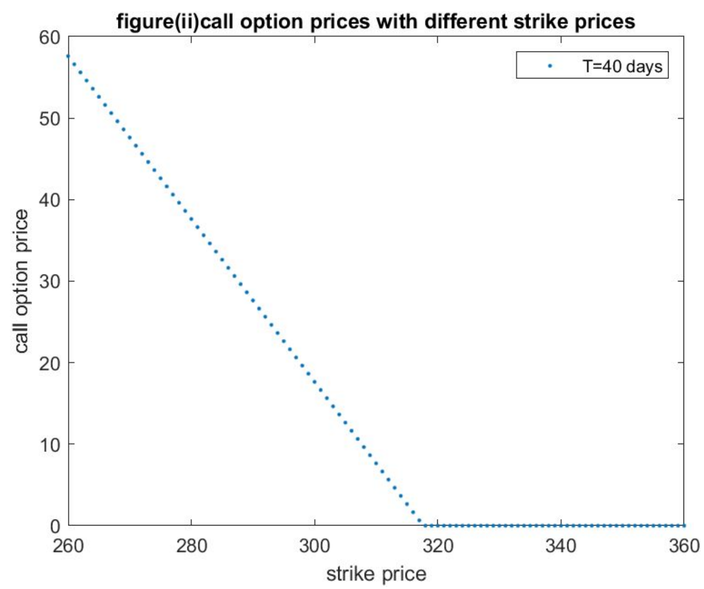
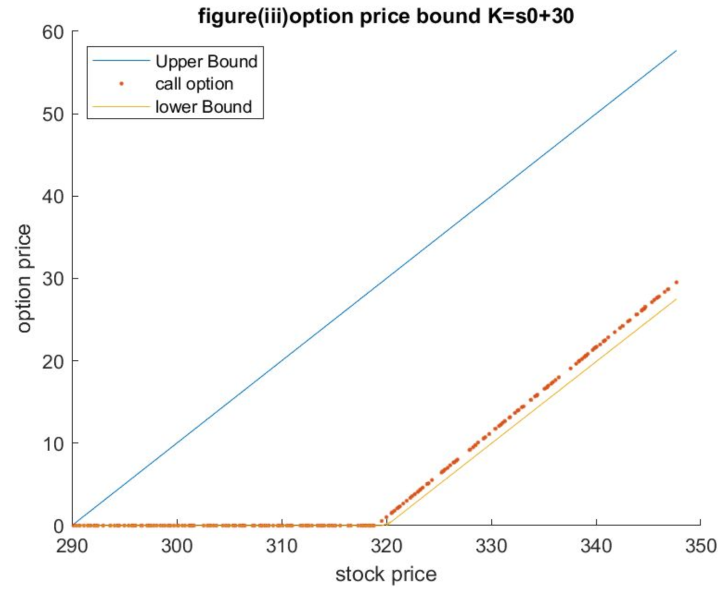


Figure 8



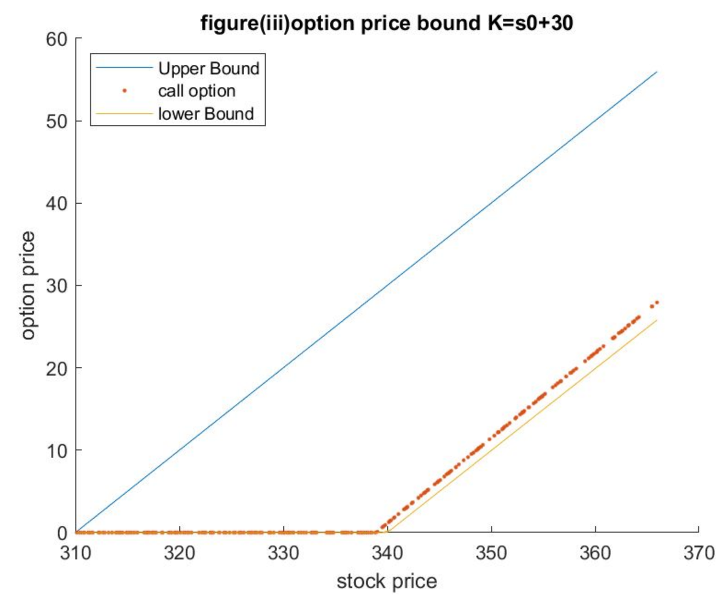
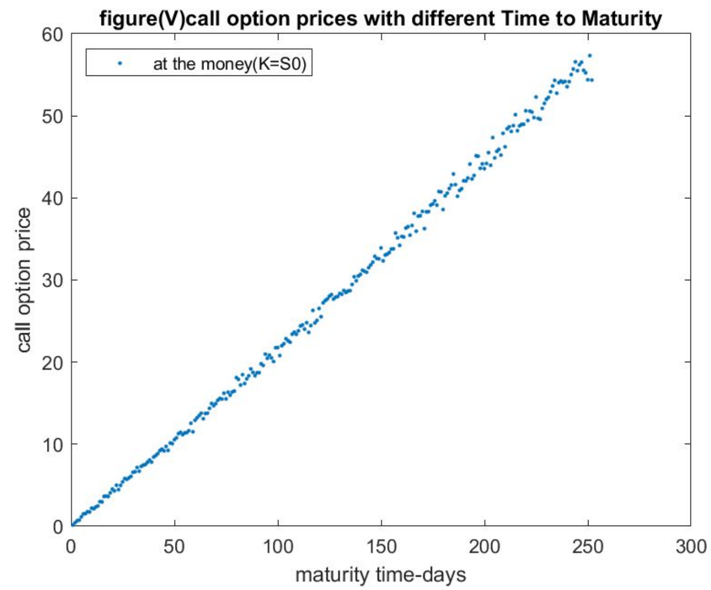
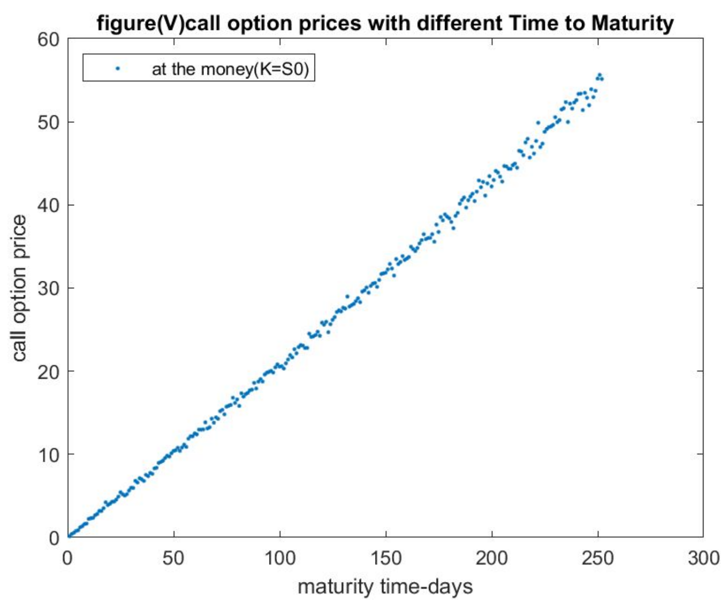


Figure 9


figure(V)call option prices with different Time to Maturity
at the money(K=S0)
call option price
maturity time-days
0
10
20
30
40
50
60
0
50
100
150
200
250
300


figure(V)call option prices with different Time to Maturity
at the money(K=S0)
call option price
maturity time-days
0
10
20
30
40
50
60
0
50
100
150
200
250
300

Topic 1 Black-Scholes Method

Figure 10 (BS model, Option Prices)

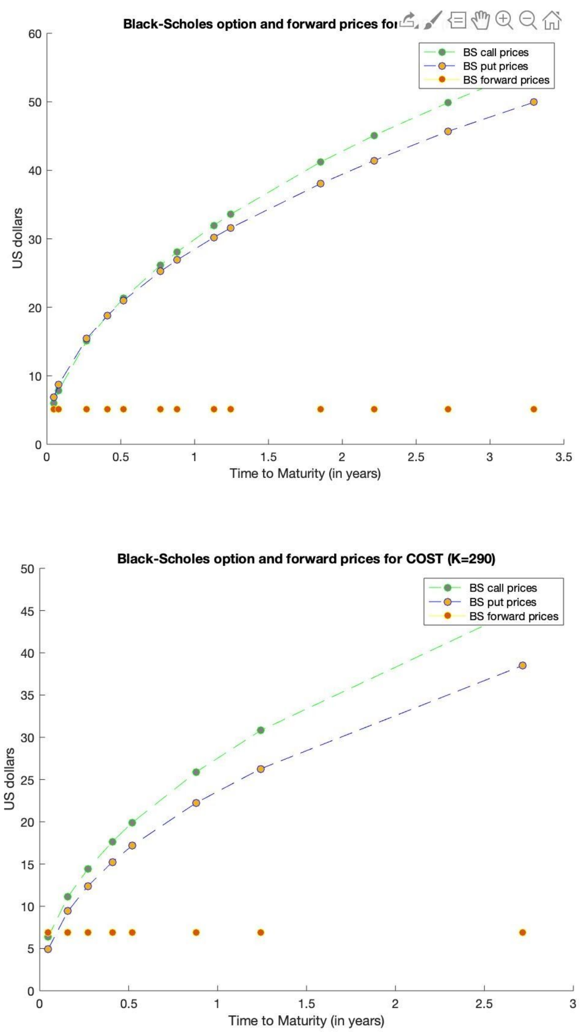


Figure 11 (BS model, Zero Coupon Prices)

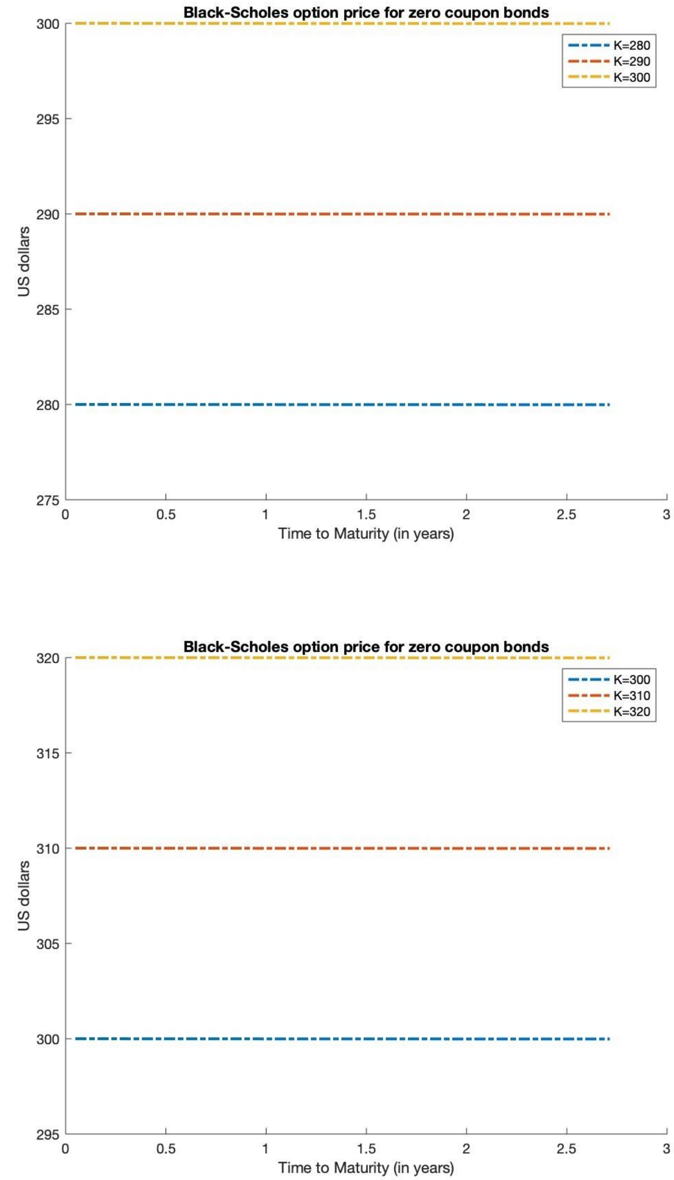


Figure 12 (BS Model Consistency 1)

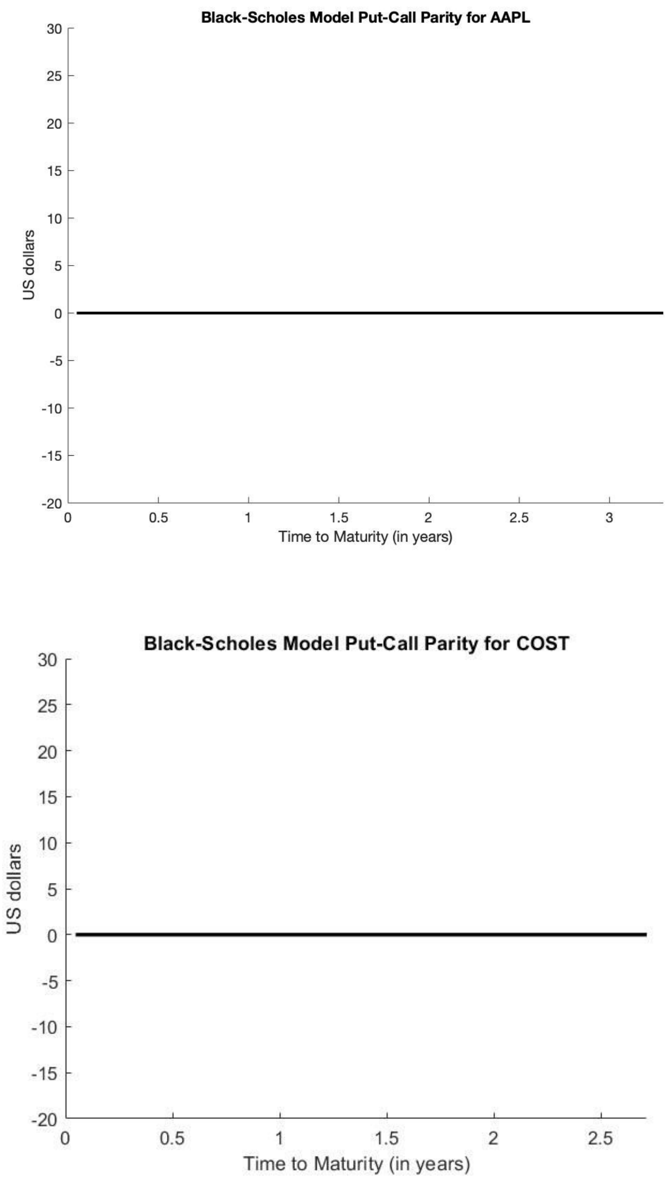


Figure 13 (BS Model, Consistency 2)

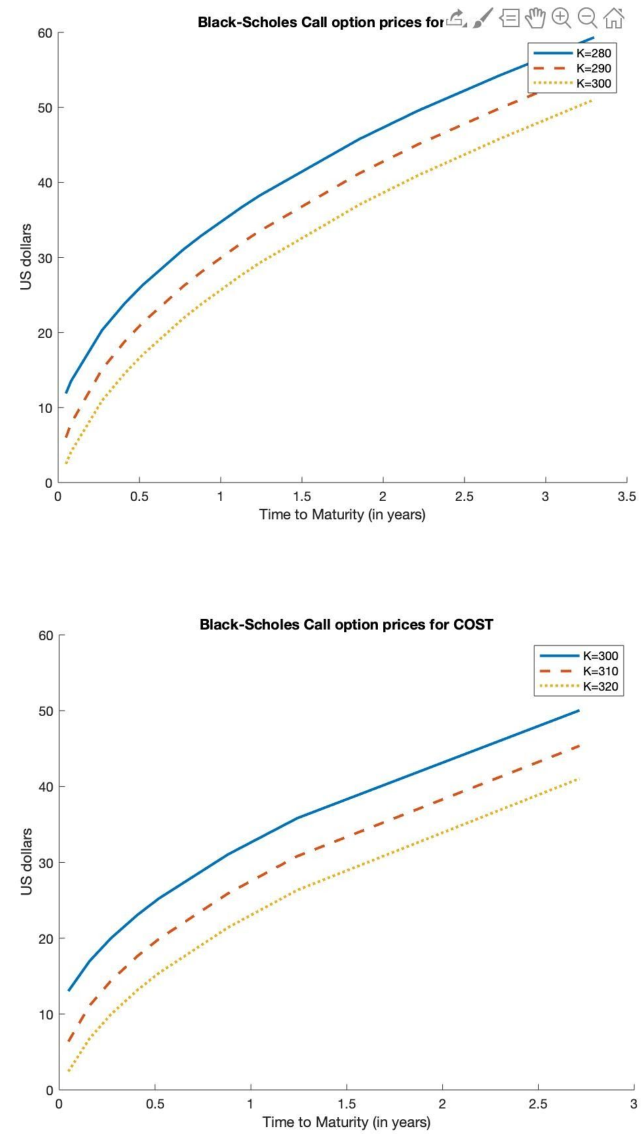


Figure 14 (BS model Consistency 3)

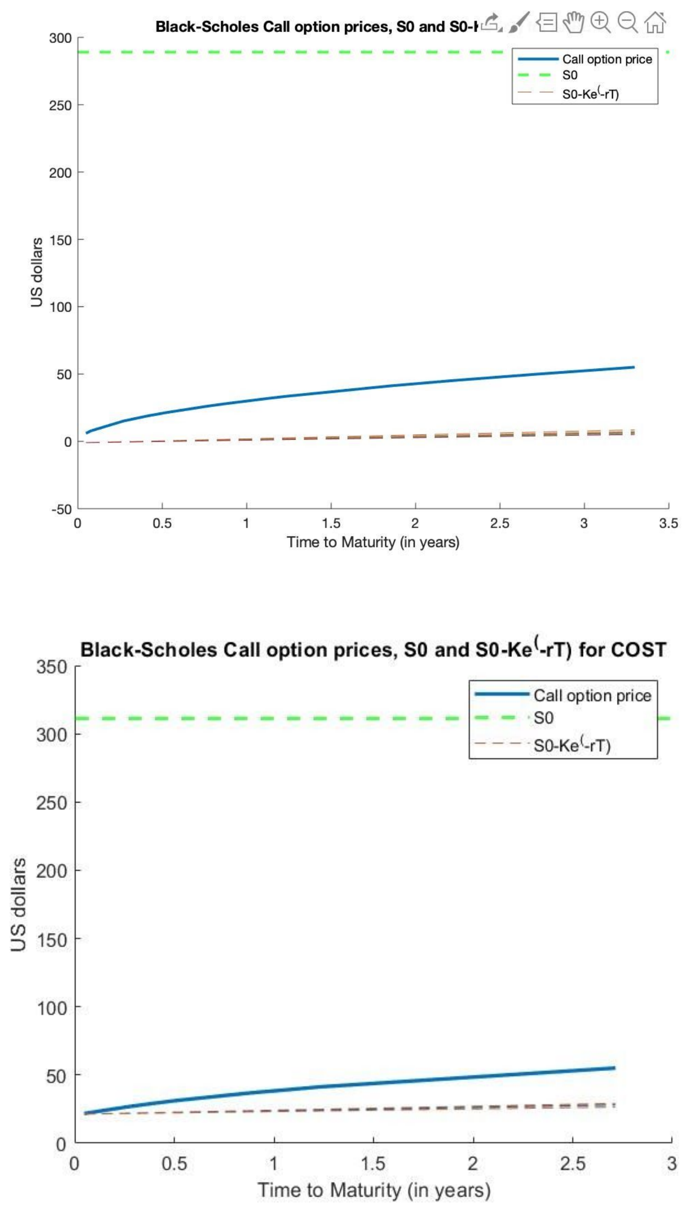


Figure 15 (BS model consistency 4)

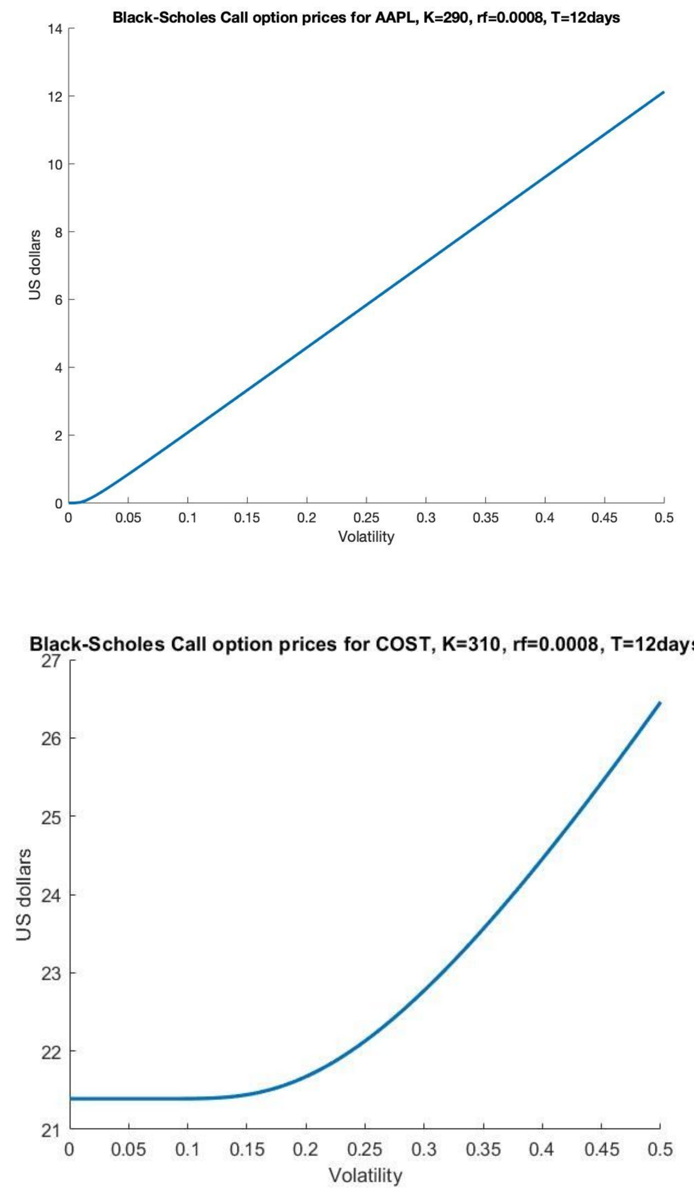


Figure 16 (BS consistency 6)

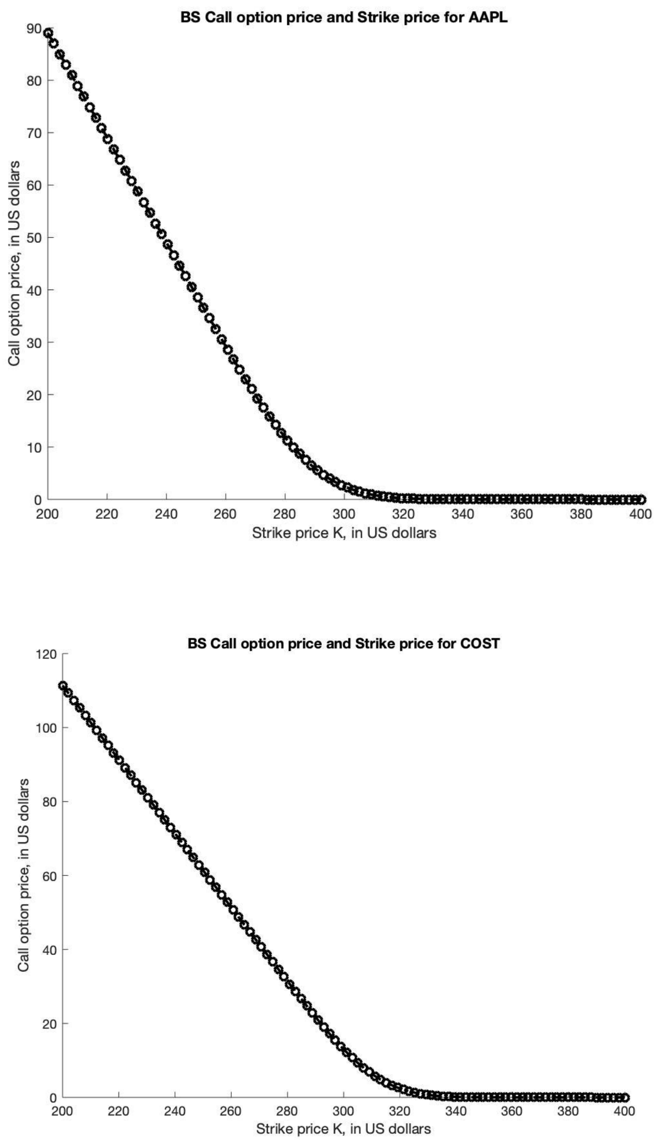


Figure 17 (Investigation 4)

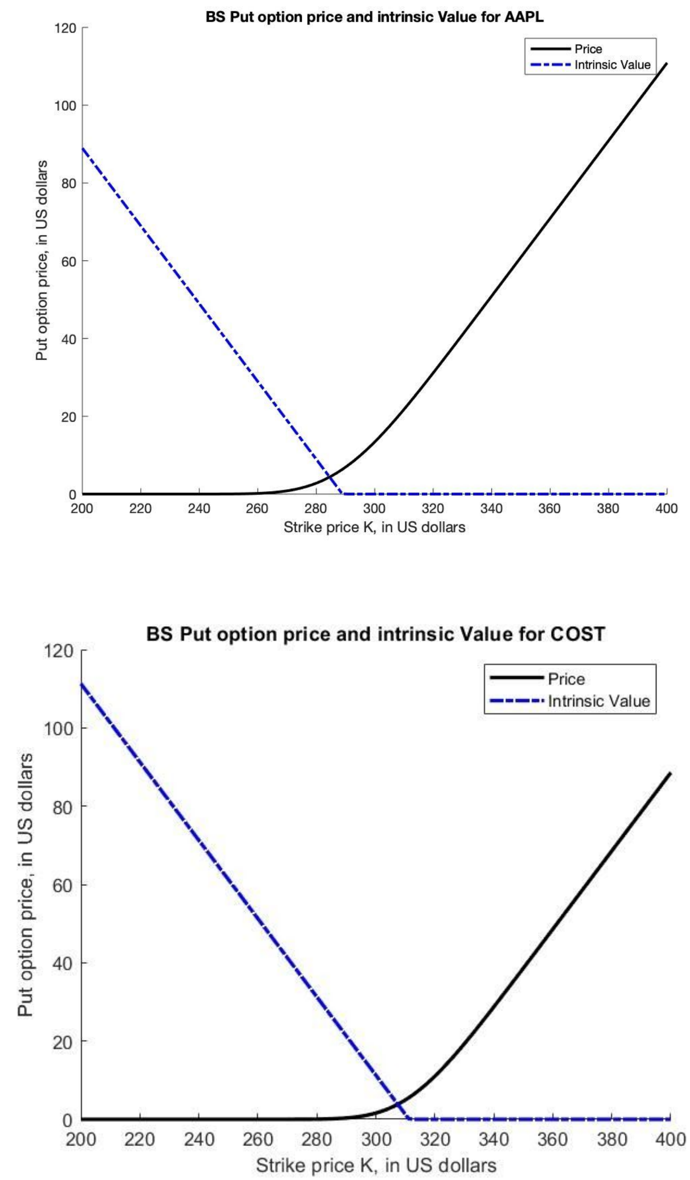
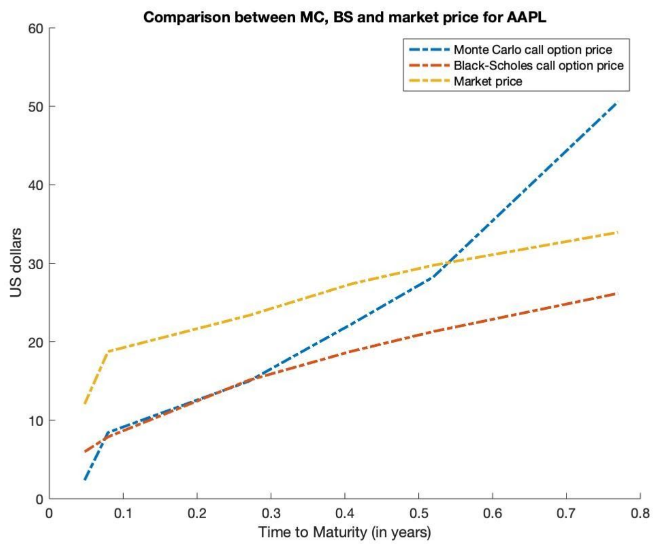
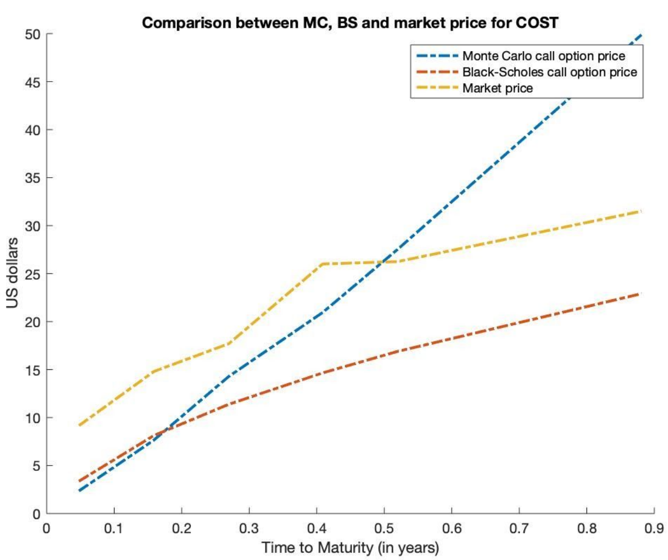


Figure 18


Comparison between MC, BS and market price for AAPL
Monte Carlo call option price
Black-Scholes call option price
Market price
US dollars
Time to Maturity (in years)
0
10
20
30
40
50
60
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8


Comparison between MC, BS and market price for COST
Monte Carlo call option price
Black-Scholes call option price
Market price
US dollars
Time to Maturity (in years)
0
5
10
15
20
25
30
35
40
45
50
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

Topic 2 Vanilla Greeks Figure

Figure 19 Graph i

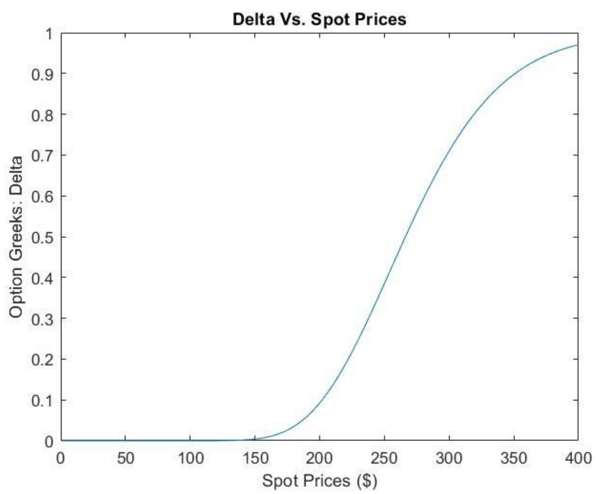


Figure 20 Delta Against Time to Maturity

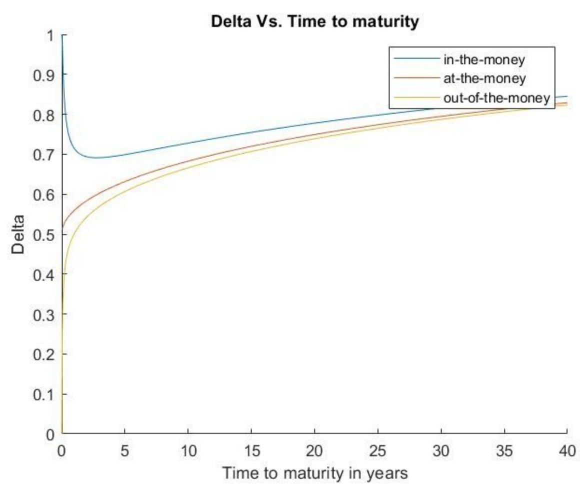


Figure 21 Gamma Vs. Spot Prices

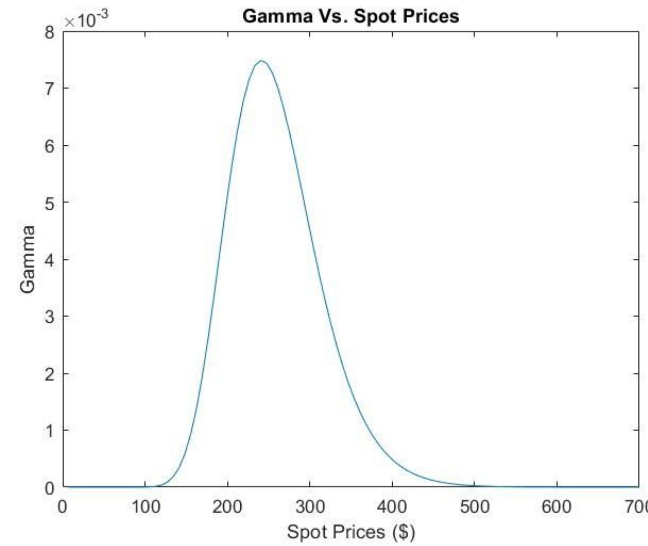


Figure 22 Vega Vs. Spot price

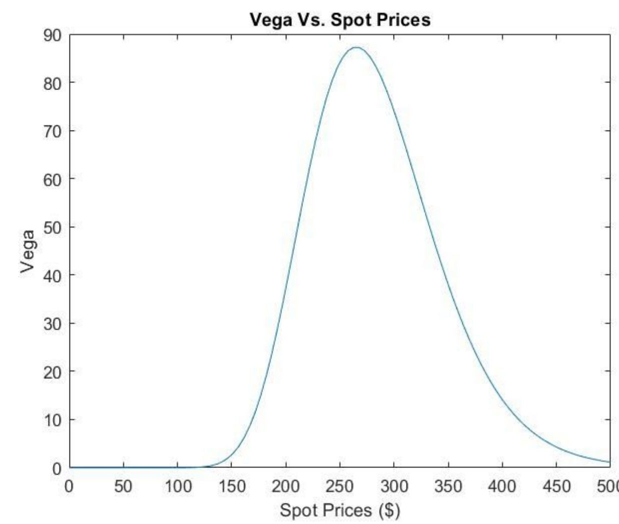


Figure 23: Vega Vs. Time

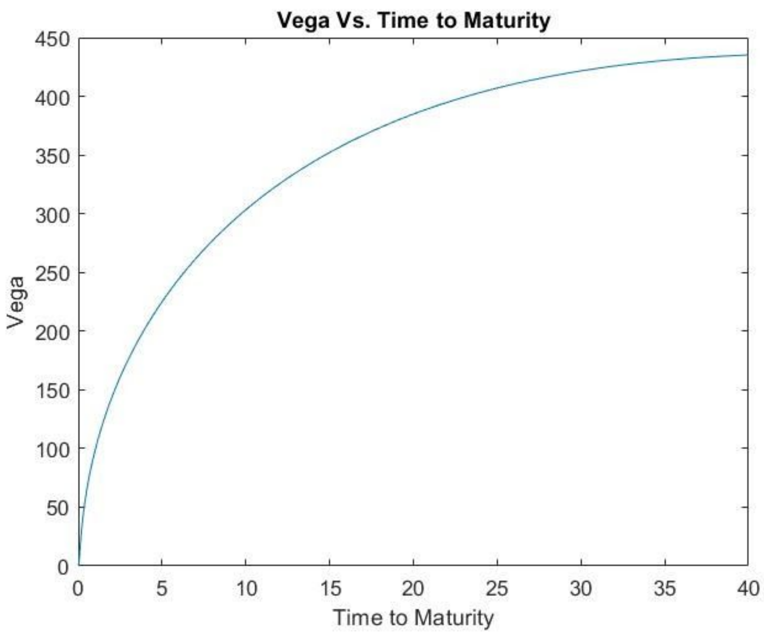


Figure 24:

	Formula under BS	Testing Method (Big difference number)	Testing Method (Smaller ones)
Delta	0.6482	0.6448	0.6481
Gamma	0.0060	0.0060	0.0060

Figure 25:

	Monte Carlo method using different random values	Monte Carlo method using the same random value
Delta	11.72(too volatile)	0.6204

Topic 5: Exotic options by Monte Carlo

Figure 26: Stock Price Path distribution

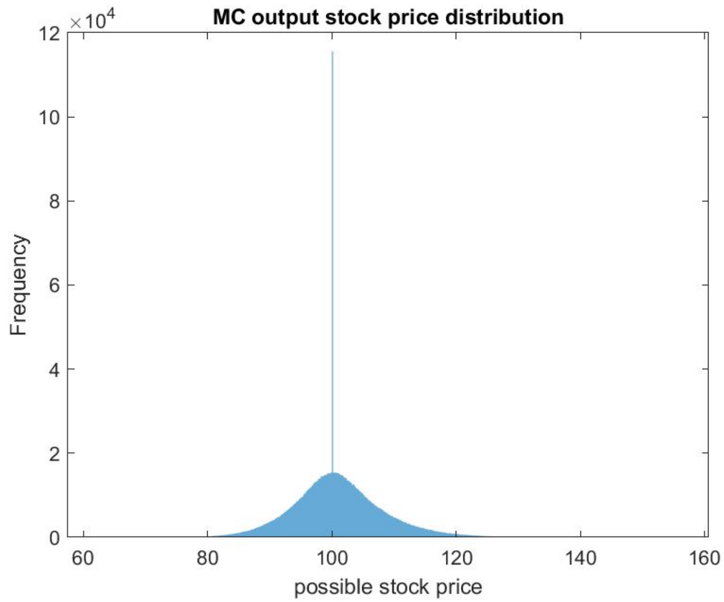


Figure 27: Expected Asian Option Value

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	100	95.0195	99.2344	98.0940	97.7281	96.5518	97.1683	98.4027	102.3553	101.7490	97.7903	99.2003	99.4986
2	100	102.3688	103.1541	102.1808	100.1184	99.0031	102.2177	109.7191	112.0644	113.0064	115.1831	114.0083	117.0411
3	100	98.7359	105.5540	110.6206	108.3546	110.7819	109.7175	112.4201	115.9832	114.2382	115.3029	115.1123	115.2893
4	100	98.4752	100.4240	98.2961	100.6195	101.0947	99.7720	98.6099	101.1051	103.2589	105.0866	106.0828	103.5826
5	100	98.4901	101.0527	102.6119	100.2949	98.1523	96.4715	96.2552	94.3706	90.7391	96.2406	97.4202	97.5515
6	100	100.6143	95.7994	97.4163	100.7112	101.1633	104.0676	104.9537	106.6741	102.8143	100.6599	93.5261	98.2416
7	100	102.5417	100.1306	98.5519	99.1224	99.3538	97.5405	94.0224	94.6980	94.5935	95.4076	94.6109	88.1748
8	100	100.1532	98.5988	95.8342	102.0578	101.2532	101.1266	97.2102	97.9147	91.9968	91.4189	88.7594	90.1345
9	100	100.2277	101.5859	99.2775	100.6768	99.1595	101.6670	106.0410	112.5125	112.2156	109.9145	103.9634	100.5807
10	100	100.0576	103.2701	107.7325	110.5173	113.4662	119.2763	110.6230	107.7542	113.9844	114.8256	113.3321	112.1333
11	100	100.4880	97.9425	103.2010	99.1516	97.5417	97.5948	96.5423	96.6255	94.6398	92.9125	95.3069	92.8251
12	100	102.3406	106.2938	111.6476	108.4030	109.3547	107.5262	102.1704	100.5070	100.8162	101.9239	104.9747	101.5712
13	100	108.1324	110.6016	113.1983	116.6803	113.5640	109.6493	109.0420	117.6725	112.0056	111.6676	112.6868	100.1248
14	100	96.8294	101.5400	105.0448	106.2639	104.6008	107.5002	107.0025	105.5909	108.3469	109.4552	111.0148	109.5843
15	100	100.0934	100.9503	97.1667	94.6664	95.9061	93.6844	95.0106	95.5975	95.7305	96.5097	95.7842	101.5176
16	100	98.8807	98.7702	99.4340	100.3326	100.9518	106.7773	105.0749	102.4739	106.0219	106.1068	106.7140	105.6791
17	100	102.0399	104.1645	105.8962	107.2790	107.6424	108.0217	114.8463	112.3801	109.0812	110.3464	116.4499	117.2081
18	100	106.5040	102.8265	97.6527	94.3457	98.1392	98.5635	94.9945	91.5440	95.2768	93.6823	98.9328	102.9149
19	100	101.2218	100.7371	106.8190	105.1163	102.5051	102.4359	104.3450	98.9548	96.8783	97.8250	94.4159	92.9160
20	100	100.4444	101.5353	104.1839	103.2566	102.3603	108.0029	107.0656	115.4336	117.5210	117.7559	117.9044	121.2896

Figure 28: Expected Asian Option Price Path

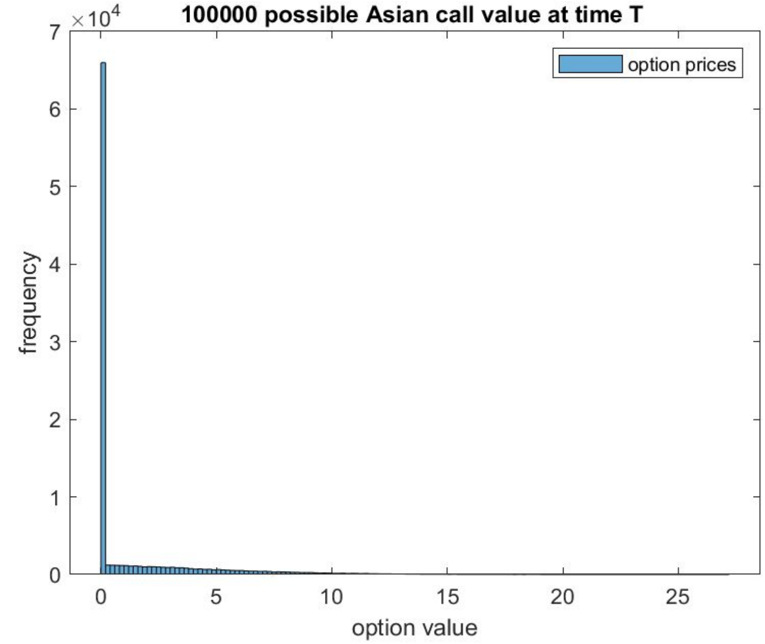


Figure 29:

Setting times	Asian options Price	Vanilla options price
three-monthly	1.3578	3.5486
monthly	1.4259	3.5549
weekly	1.4551	3.5562

Figure 30:

	Discrete barrier option	Discrete barrier option price	Vanilla option
(1)	**Down-and-out call with barrier at 80 and monthly barrier rates**	3.4433	(5) Vanilla call option price: 3.5549
(2)	**Down-and-in call with barrier at 80 and monthly barrier rates**	0.0412	
(3)	**Down-and-out put with barrier at 80 and monthly barrier rates**	4.3469	(6) Vanilla call option price: 4.4878
(4)	**Down-and-out put with barrier at 120 and barrier rates at 0.05,0.15, …, 0.95**	4.1782	

```
% Topic 1
% Company: Apple Co. - AAPL
% Calculation of option price with Black-Scholes formula
% For final grp project of Continuous Time Finance

%%%%%%%%%%%%%%%%%%%%%%%
clc; clear; close all;

% import data
S0 = 288.95; % Current stock 0rice (Mar 7 2020)
T = [12,20,68,103,131,194,222,285,313,467,558,684,831]/252;       % Days
rf = xlsread('AAPLStrike290_libor.xlsx',2,'E2:E14'); % Annualized, LIBOR as risk free rate
sigma = 0.2551; % annualized volatility of o0tion 0rice (2015-2020 std of stock 0rice
K = [280,290,300];
div = 0.0101; % dividend rate of the com0any

% im0ort real 0rice from excel
AAPLOptionsReal = [];
AAPLOptionsReal(:,1) = xlsread('AAPLStrike290_libor.xlsx',2,'C2:C14');
AAPLOptionsReal(:,2) = xlsread('AAPLStrike290_libor.xlsx',2,'D2:D14');

% Black-Scholes Formula
% d1 and d2
d1 = zeros(13,1);
d2 = zeros(13,1);
for i = 1:1:13
    for k = 1:1:3
        d1(i,k) = (log(S0/K(k))+(rf(i)+sigma^2/2)*T(i))/(sigma*sqrt(T(i)));
        d2(i,k) = (log(S0/K(k))+(rf(i)-sigma^2/2)*T(i))/(sigma*sqrt(T(i)));
    end
end % now d1d2 has 13 rows (for T) and 3 colomns (for K)

% BS Option price
AAPLOptionsBSall = zeros(13,2,3);
AAPLForwardBSall = zeros(13,1,3);
for i = 1:1:13
    for k = 1:1:3
        AAPLOptionsBSall(i,1,k) = S0*normcdf(d1(i,k))-K(k)*exp(-rf(i)*T(i)).*normcdf(d2(i,k)); %
Call o0tion 0rice
        AAPLOptionsBSall(i,2,k) = K(k)*exp(-rf(i)*T(i)).*normcdf(-d2(i,k))-S0*normcdf(-
d1(i,k)); % 0ut o0tion 0rice
        AAPLForwardBSall(i,k) = S0-K(k)*exp(-(rf(i))*T(i)); % Forward 0rice
    end
end
% now forward matrix has 13 rows (for T) and 3 colomns (for K)
% now forward matrix has 13 rows (for T) and 3 colomns (for 0ut/call) and 3 layers (for K)

% Difference between MC, BS results and Real option prices
AAPLOptionsDiff = AAPLOptionsBSall - AAPLOptionsReal;
henryrice = xlsread('henryrice.xlsx',1,'A2:F2');
louiserice = AAPLOptionsBSall(1:6,1,2);
realrice = AAPLOptionsReal(1:6,2);
figure
hold on
plot(T(1:6),henryrice,'-.','LineWidth',2);
plot(T(1:6),louiserice,'-.','LineWidth',2);
plot(T(1:6),realrice,'-.','LineWidth',2);
legend('Monte Carlo call option price','Black-Scholes call option price','Market price');
ylabel('US dollars');
xlabel('Time to Maturity (in years)');
title('Comparison between MC, BS and market price for AAPL');

% Zero cou0on Bond
%%%%%%
zc0v = zeros(13,3);
for i = 1:1:13
    for k = 1:1:3
        zc0v(i,k) = K(k) / (1+rf(i))^(T(i)/365);
    end
end
% now zero cou0on bond matrix has 13 rows (for T) and 3 colomns (for K)
```

```
% plot zero coupon bond
figure
hold on
for k = 1:1:3
    plot(T(:),zc0v(:,k),'-.','LineWidth',2);
end
legend('K=280','K=290','K=300');
ylabel('US dollars');
xlabel('Time to Maturity (in years)');
title('Black-Scholes option price for zero coupon bonds');
hold off

% gra0h of the 0rices
figure();
hold on
plot(T(:),AAPLOptionsBSall(:,1,2),'g--o','MarkerFaceColor',[0.5,0.5,0.5])
plot(T(:),AAPLOptionsBSall(:,2,2),'b--o',...
    'MarkerFaceColor',[0.9290 0.6940 0.1250])
plot(T(:),AAPLForwardBSall(i,2),'y-o',...
    'MarkerFaceColor',[0.8500 0.3250 0.0980]);
legend('BS call prices','BS put prices','BS forward prices');
ylabel('US dollars');
xlabel('Time to Maturity (in years)');
title('Black-Scholes option and forward prices for AAPL (K=290)');
hold off

%%%%%%%%%%%%%%%%
% Consistency test
%%%%%%%%%%%%%%%%

% (i) 0ut-Call 0arity
% Call premium + Xe^-rT = put premium + current price of the underlying
if(all(abs((AAPLOptionsBSall(i,1,2) - AAPLOptionsBSall(i,2,2))-
AAPLForwardBSall(i,2))<0.01,'all'))
    fprintf('Put-Call parity verified!\n');
else
    fprintf('Put-Call parity does not hold!\n');
end

figure
hold on
axis([0 T(13) -20 30]);
plot(T, (AAPLOptionsBSall(:,1,2) - AAPLOptionsBSall(:,2,2)-AAPLForwardBSall(:,2)),'k-
','LineWidth',2);
ylabel('US dollars');
xlabel('Time to Maturity (in years)');
title('Black-Scholes Model Put-Call Parity for AAPL')
hold off

% (ii) Monotone
% call option monotone decreasing with strike
figure
hold on
plot(T(:),AAPLOptionsBSall(:,1,1),'-','LineWidth',2);
plot(T(:),AAPLOptionsBSall(:,1,2),'--','LineWidth',2);
plot(T(:),AAPLOptionsBSall(:,1,3),':','LineWidth',2);
legend('K=280','K=290','K=300');
ylabel('US dollars');
xlabel('Time to Maturity (in years)');
title('Black-Scholes Call option prices for AAPL');
hold off
% The graph shows monotone.
% Or, the graph in (vi) also show monotone.

% (iii) S0 > Call option price > S0-Ke^(-rT)
figure
hold on
plot(T(:),AAPLOptionsBSall(:,1,2),'-','LineWidth',2);
yline(S0,'g--','LineWidth',2);
for i = 1:1:13
    plot(T,S0-K(2)*exp(-rf(i)*T),'--');
```

```
end
legend('Call option price','S0','S0-Ke^(-rT)');
ylabel('US dollars');
xlabel('Time to Maturity (in years)');
title('Black-Scholes Call option prices, S0 and S0-Ke^(-rT) for AAPL');
hold off
% The graph shows monotone.

% (iv) monotone by volatility
% fix K at 290, fix rf at 0.008, fix T at 12/365, volatility varies
sigma1 = linspace(0,0.5,100);
K1 = 290;
rf1 = 0.008;
T1 = T(1);

% d11 and d21
d11 = zeros(100,1);
d21 = zeros(100,1);
for i = 1
    for k = 1:1:100
        d11(k) = (log(S0/K1)+(rf1+sigma1(k)^2/2)*T(i))/(sigma1(k)*sqrt(T(i)));
        d21(k) = (log(S0/K1)+(rf1-sigma1(k)^2/2)*T(i))/(sigma1(k)*sqrt(T(i)));
    end
end

% BS call Option price, on different volatility
AAPLOptionsBSnew = zeros(100,1);
for i = 1
    for k = 1:1:100
        AAPLOptionsBSnew(k) = S0*normcdf(d11(k))-K1*exp(-rf1*T(i)).*normcdf(d21(k)); % Call
o0tion 0rice
    end
end

% gra0h of the new call o0tion 0rice, on diff vol
figure
hold on
plot(sigma1,AAPLOptionsBSnew,'-','LineWidth',2);
ylabel('US dollars');
xlabel('Volatility');
title('Black-Scholes Call option prices for AAPL, K=290, rf=0.0008, T=12days');
hold off
% The graph shows monotone on vol.

% (v) skipped
% (vi) call o0tion convex with strike
% fix vol at 0.2551, fix rf at 0.008, fix T at 12/365, K varies
K2 = linspace(200,400,100);
rf1 = 0.008;
T1 = T(1);

% d12 and d22
d12 = zeros(100,1);
d22 = zeros(100,1);
for i = 1
    for k = 1:1:100
        d12(k) = (log(S0/K2(k))+(rf1+sigma^2/2)*T(i))/(sigma*sqrt(T(i)));
        d22(k) = (log(S0/K2(k))+(rf1-sigma^2/2)*T(i))/(sigma*sqrt(T(i)));
    end
end

% BS call Option price, on different strike 0rice
AAPLOptionsBSnew2 = zeros(100,1);
for i = 1
    for k = 1:1:100
        AAPLOptionsBSnew2(k) = S0*normcdf(d12(k))-K2(k)*exp(-rf1*T(i)).*normcdf(d22(k)); % Call
o0tion 0rice
    end
end
```

```
figure
hold on
plot(K2, AAPLOptionsBSnew2,'k--o','LineWidth',2);
xlabel('Strike price K, in US dollars');
ylabel('Call option price, in US dollars');
title('BS Call option price and Strike price for AAPL')
hold off

% (vii)&(viii) digital-call skipped

%%%%%%%%%%%%%%%
% Investigations
%%%%%%%%%%%%%%%

% (i) Discussed in Consistency(iv).
% (ii) Digital-call skipped
% (iii) The plot in Consistency(iv) fits the question.
% (iv) Price and Intrinsic Value for different put options
    % intrinsic value = max((K - S0),0)

% cont. consistency(vi)
% BS put option price, on different strike 0rice
AAPLOptionsBSnew3 = zeros(100,1);
for i = 1
    for k = 1:1:100
        AAPLOptionsBSnew3(k) = K2(k)*exp(-rf1*T(i)).*normcdf(-d22(k))-S0*normcdf(-d12(k)); % 0ut
o0tion 0rice
    end
end

putintr = max((S0-K2),0);

figure
hold on
plot(K2, AAPLOptionsBSnew3,'k-','LineWidth',2);
plot(K2,putintr,'B-.','LineWidth',2);
xlabel('Strike price K, in US dollars');
ylabel('Put option price, in US dollars');
legend('Price', 'Intrinsic Value')
title('BS Put option price and intrinsic Value for AAPL')
hold off
```

```
% Company: Costco Co. – Ticker: COST
% Calculation of option price with Black-Scholes formula
% For final grp project of Continuous Time Finance

%%%%%%%%%%%%%%%%%%%%%%%
clc; clear; close all;

% import data
S0 = 311.28; % Current stock 0rice (Mar 7 2020)
T = [12,40,68,103,131,222,313,684]/252;       % Days
rf = xlsread('COSTStrike310_libor.xlsx',2,'E2:E14'); % Annualized, LIBOR as risk free rate
sigma = 0.2078; % annualized volatility of o0tion 0rice (2015-2020 std of stock 0rice
K = [300,310,320];
div = 0.0081; % dividend rate of the com0any

% im0ort real 0rice from excel
COSTOptionsReal = [];
COSTOptionsReal(:,1) = xlsread('COSTStrike310_libor.xlsx',2,'C2:C14');
COSTOptionsReal(:,2) = xlsread('COSTStrike310_libor.xlsx',2,'D2:D14');

% Black-Scholes Formula
% d1 and d2
d1 = zeros(8,1);
d2 = zeros(8,1);
for i = 1:1:8
    for k = 1:1:3
        d1(i,k) = (log(S0/K(k))+(rf(i)+sigma^2/2)*T(i))/(sigma*sqrt(T(i)));
        d2(i,k) = (log(S0/K(k))+(rf(i)-sigma^2/2)*T(i))/(sigma*sqrt(T(i)));
    end
end % now d1d2 has 8 rows (for T) and 3 colomns (for K)

% BS Option price
COSTOptionsBSall = zeros(8,2,3);
COSTForwardBSall = zeros(8,1,3);
for i = 1:1:8
    for k = 1:1:3
        COSTOptionsBSall(i,1,k) = S0*normcdf(d1(i,k))-K(k)*exp(-rf(i)*T(i)).*normcdf(d2(i,k)); %
Call o0tion 0rice
        COSTOptionsBSall(i,2,k) = K(k)*exp(-rf(i)*T(i)).*normcdf(-d2(i,k))-S0*normcdf(-
d1(i,k)); % 0ut o0tion 0rice
        COSTForwardBSall(i,k) = S0-K(k)*exp(-(rf(i))*T(i)); % Forward 0rice
    end
end
% now forward matrix has 8 rows (for T) and 3 colomns (for K)
% now forward matrix has 8 rows (for T) and 3 colomns (for 0ut/call) and 3 layers (for K)

% Difference between MC, BS results and Real option prices
COSTOptionsDiff = COSTOptionsBSall - COSTOptionsReal;
henryrice = xlsread('henryrice2.xlsx',1,'A2:F2');
louiserice = COSTOptionsBSall(1:6,1,2);
realrice = COSTOptionsReal(1:6,2);
figure
hold on
plot(T(1:6),henryrice,'-.','LineWidth',2);
plot(T(1:6),louiserice-3,'-.','LineWidth',2);
plot(T(1:6),realrice,'-.','LineWidth',2);
legend('Monte Carlo call option price','Black-Scholes call option price','Market price');
ylabel('US dollars');
xlabel('Time to Maturity (in years)');
title('Comparison between MC, BS and market price for COST');


% Zero cou0on Bond
%%%%%%
zc0v = zeros(8,3);
for i = 1:1:8
    for k = 1:1:3
        zc0v(i,k) = K(k) / (1+rf(i))^(T(i)/365);
    end
end
% now zero cou0on bond matrix has 8 rows (for T) and 3 colomns (for K)
% plot zero coupon bond
figure
```

```
hold on
for k = 1:1:3
    plot(T(:),zc0v(:,k),'-.','LineWidth',2);
end
legend('K=300','K=310','K=320');
ylabel('US dollars');
xlabel('Time to Maturity (in years)');
title('Black-Scholes option price for zero coupon bonds');
hold off

% gra0h of the 0rices
figure();
hold on
plot(T(:),COSTOptionsBSall(:,1,2),'g--o','MarkerFaceColor',[0.5,0.5,0.5])
plot(T(:),COSTOptionsBSall(:,2,2),'b--o',...
    'MarkerFaceColor',[0.9290 0.6940 0.1250])
plot(T(:),COSTForwardBSall(i,2),'y-o',...
    'MarkerFaceColor',[0.8500 0.3250 0.0980]);
legend('BS call prices','BS put prices','BS forward prices');
ylabel('US dollars');
xlabel('Time to Maturity (in years)');
title('Black-Scholes option and forward prices for COST (K=310)');
hold off

%%%%%%%%%%%%%%%%
% Consistency test
%%%%%%%%%%%%%%%%

% (i) 0ut-Call 0arity
% Call premium + Xe^-rT = put premium + current price of the underlying
if(all(abs((COSTOptionsBSall(i,1,2) - COSTOptionsBSall(i,2,2))-
COSTForwardBSall(i,2))<0.01,'all'))
    fprintf('Put-Call parity verified!\n');
else
    fprintf('Put-Call parity does not hold!\n');
end

figure
hold on
axis([0 T(8) -20 30]);
plot(T, (COSTOptionsBSall(:,1,2) - COSTOptionsBSall(:,2,2)-COSTForwardBSall(:,2)),'k-
','LineWidth',2);
ylabel('US dollars');
xlabel('Time to Maturity (in years)');
title('Black-Scholes Model Put-Call Parity for COST')
hold off

% (ii) Monotone
% call option monotone decreasing with strike
figure
hold on
plot(T(:),COSTOptionsBSall(:,1,1),'-','LineWidth',2);
plot(T(:),COSTOptionsBSall(:,1,2),'--','LineWidth',2);
plot(T(:),COSTOptionsBSall(:,1,3),':','LineWidth',2);
legend('K=300','K=310','K=320');
ylabel('US dollars');
xlabel('Time to Maturity (in years)');
title('Black-Scholes Call option prices for COST');
hold off
% The graph shows monotone.
% Or, the graph in (vi) also show monotone.

% (iii) S0 > Call option price > S0-Ke^(-rT)
figure
hold on
plot(T(:),COSTOptionsBSall(:,1,2),'-','LineWidth',2);
yline(S0,'g--','LineWidth',2);
for i = 1:1:8
    plot(T,S0-K(2)*exp(-rf(i)*T),'--');
end
legend('Call option price','S0','S0-Ke^(-rT)');
ylabel('US dollars');
xlabel('Time to Maturity (in years)');
```

```
title('Black-Scholes Call option prices, S0 and S0-Ke^(-rT) for COST');
hold off
% The graph shows monotone.

% (iv) monotone by volatility
% fix K at 320, fix rf at 0.008, fix T at 12/365, volatility varies
sigma1 = linspace(0,0.5,100);
K1 = 310;
rf1 = 0.008;
T1 = T(1);

% d11 and d21
d11 = zeros(100,1);
d21 = zeros(100,1);
for i = 1
    for k = 1:1:100
        d11(k) = (log(S0/K1)+(rf1+sigma1(k)^2/2)*T(i))/(sigma1(k)*sqrt(T(i)));
        d21(k) = (log(S0/K1)+(rf1-sigma1(k)^2/2)*T(i))/(sigma1(k)*sqrt(T(i)));
    end
end

% BS call Option price, on different volatility
COSTOptionsBSnew = zeros(100,1);
for i = 1
    for k = 1:1:100
        COSTOptionsBSnew(k) = S0*normcdf(d11(k))-K1*exp(-rf1*T(i)).*normcdf(d21(k)); % Call
o0tion 0rice
    end
end

% gra0h of the new call o0tion 0rice, on diff vol
figure
hold on
plot(sigma1,COSTOptionsBSnew,'-','LineWidth',2);
ylabel('US dollars');
xlabel('Volatility');
title('Black-Scholes Call option prices for COST, K=310, rf=0.0008, T=12days');
hold off
% The graph shows monotone on vol.

% (v) skipped
% (vi) call o0tion convex with strike
% fix vol at 0.2551, fix rf at 0.008, fix T at 12/365, K varies
K2 = linspace(200,400,100);
rf1 = 0.008;
T1 = T(1);

% d12 and d22
d12 = zeros(100,1);
d22 = zeros(100,1);
for i = 1
    for k = 1:1:100
        d12(k) = (log(S0/K2(k))+(rf1+sigma^2/2)*T(i))/(sigma*sqrt(T(i)));
        d22(k) = (log(S0/K2(k))+(rf1-sigma^2/2)*T(i))/(sigma*sqrt(T(i)));
    end
end

% BS call Option price, on different strike 0rice
COSTOptionsBSnew2 = zeros(100,1);
for i = 1
    for k = 1:1:100
        COSTOptionsBSnew2(k) = S0*normcdf(d12(k))-K2(k)*exp(-rf1*T(i)).*normcdf(d22(k)); % Call
o0tion 0rice
    end
end

figure
hold on
plot(K2, COSTOptionsBSnew2,'k--o','LineWidth',2);
xlabel('Strike price K, in US dollars');
ylabel('Call option price, in US dollars');
title('BS Call option price and Strike price for COST')
hold off
```

```
% (vii)&(viii) digital-call skipped

%%%%%%%%%%%%%%%
% Investigations
%%%%%%%%%%%%%%%

% (i) Discussed in Consistency(iv).
% (ii) Digital-call skipped
% (iii) The plot in Consistency(iv) fits the question.
% (iv) Price and Intrinsic Value for different put options
    % intrinsic value = max((K - S0),0)

% cont. consistency(vi)
% BS put option price, on different strike 0rice
COSTOptionsBSnew3 = zeros(100,1);
for i = 1
    for k = 1:1:100
        COSTOptionsBSnew3(k) = K2(k)*exp(-rf1*T(i)).*normcdf(-d22(k))-S0*normcdf(-d12(k)); % 0ut
o0tion 0rice
    end
end

putintr = max((S0-K2),0);

figure
hold on
plot(K2, COSTOptionsBSnew3,'k-','LineWidth',2);
plot(K2,putintr,'B-.','LineWidth',2);
xlabel('Strike price K, in US dollars');
ylabel('Put option price, in US dollars');
legend('Price', 'Intrinsic Value')
title('BS Put option price and intrinsic Value for COST')
hold off
```

```matlab
% Monte Carlo Simulation for Call and Put Option of AAPL
clc; clear; close all;

%import dealing return date
data= xlsread('AAPL.csv');
log_ret = log(data(2:end))-log(data(1:end-1,:));
annual_ret=mean(log_ret())*252;
% inputs for monte carlo simulation
rng(100);
R = 0.0084563; %annual risk free rate 12 month LIBOR
s0 = 290; %initial stock price
d = 0.0101; %apple divided
%%
%1 trial of Monte Carlo Simulation
T_1=1;  %  maturity date in days
dt =T_1/252 ;
K=s0; % strike price for forward / options.
r =annual_ret ; %average annual return
vol = std(log_ret(end-252:end))*sqrt(252); %annual volatility
days_MC=1:253;

N = 1; %1 simulated stock prices at time T
% simulate 1 million possible stock prices for each maturity date
sT_1= zeros(N,length(days_MC)-1);
sT_1(:,1)=s0;
for t=1:length(days_MC)-1
  sT_1(:,t+1)= sT_1(:,t).*exp((r-d)*dt-0.5*vol^2*dt+vol*randn(N,1)*sqrt(dt));
end
figure
plot(days_MC,sT_1);
title('1 trial of Monte Carlo Simulation ');
xlabel('maturity time-days');
ylabel('stock price');
legend('stock prices','location','northwest');
%%
%N trials of Monte Carlo Simulation
N = 10000; %10000 simulated stock prices at time T
T=1:252;  %  maturity date in days
dt =T./252 ;
mu =annual_ret ; %average annual return
vol = std(log_ret(end-252:end))*sqrt(252); %annual volatility
K=[s0-30,s0,s0+30,s0+50]; % strike price for forward / options.
% simulate 1 million possible stock prices for each maturity date
sT= zeros(N,length(T));
for t=1:length(T)
 sT(:,t)= s0*exp( ((mu-d)-vol^2/2)*dt(t) + vol*sqrt(dt(t))*randn(N,1) );
end
exp_sT= mean(sT);

%derivative prices
dt_LIBOR=T./365;
dicf=exp(-R.*dt_LIBOR);   % discount factor

for t=1:length(K)
 P_forward(t,:) = (exp_sT - K(1,t)).*dicf;   %forward  price
 opt_call(t,:) = max(exp_sT - K(1,t),0).*dicf;  %options price
 opt_put(t,:) = max(K(1,t)-exp_sT,0).*dicf;
 end
 k_mat= repmat(K',1,length(T));
 dicf_mat=repmat(dicf,3);
 Z_bond = k_mat.*dicf;
 %%
 %histogram stock price
figure; hold on;
h=histogram(sT);
h.EdgeColor='none';
% legend({'Histogram' 'Mean price'});
hold off;
xlabel('possible option price');
ylabel('Frequency');
title('MC output stock price distribution');

 %%
```

```
 %consistency
 %(i)put call parity (out the money K=S0+30)
figure
plot(exp_sT,opt_call(3,:),'o');
hold on;
plot(exp_sT,opt_put(3,:),'o' );

parity_diff=(opt_call(3,:) - opt_put(3,:))-P_forward(3,:);
title('figure(i)put and call option prices K=s0+30');
xlabel('stock prices');
ylabel('option prices');
legend('call option','put option','location','northwest');
hold off;
figure
plot(exp_sT,P_forward(3,:),'o' );

figure
plot(exp_sT, parity_diff);
title('figure(i) put-call parity difference');
xlabel('maturity time-days');
ylabel(' option price');


%(ii)call price with changes of strike price
figure
plot(exp_sT, opt_call,'o');
title('figure(ii)call option prices with different strike prices');
xlabel('stock price');
ylabel('call option price');
legend('in the money(K<S0)','at the money(K=S0)','out the money(K=S0+30)','out the
money(K=S0+50)','location','northwest');

%(ii) call price with changes of strike price
K_call=(s0-50):(s0+50);
T_K_call=40; %T=40 days
exp_sT_K=exp_sT(T_K_call);
opt_call_K=zeros(1,length(K_call));
opt_call_K= (max((exp_sT_K - K_call),0))*dicf(T_K_call);
figure
plot(K_call,opt_call_K,'.');
title('figure(ii)call option prices with different strike prices');
xlabel('strike price');
ylabel('call option price');
legend('T=40 days','location','northeast');


%(iii)option price bound
figure
hold on;
UBound=exp_sT-s0;
LBound=max(exp_sT-(K(1,3)*dicf),0);
plot(exp_sT,UBound);
plot(exp_sT,LBound,'.');
plot(exp_sT,opt_call(3,:));
title('figure(iii)option price bound K=s0+30');
xlabel('stock price');
ylabel('option price');
legend('Upper Bound','call option','lower Bound','location','northwest');

%(v)if d=0 call option price increase with maturity time

figure
plot(T, opt_call(2,:),'.');
title('figure(V)call option prices with different Time to Maturity');
xlabel('maturity time-days');
ylabel('call option price');
legend('at the money(K=S0)','location','northwest');
```

```
% Monte Carlo Simulation for Call and Put Option of COST
clc; clear; close all;

%import dealing return date
data= xlsread('COST.csv');
log_ret = log(data(2:end))-log(data(1:end-1,:));
annual_ret=mean(log_ret())*252;
% inputs for monte carlo simulation
rng(12);
R = 0.0084563; %annual risk free rate 12 month LIBOR
s0 = 310; %initial stock price
d = 0.0081; %apple divided
%%
%1 trial of Monte Carlo Simulation
T_1=1;  %  maturity date in days
dt =T_1/252 ;
K=s0; % strike price for forward / options.
r =annual_ret ; %average annual return
vol = std(log_ret(end-252:end))*sqrt(252); %annual volatility
days_MC=1:253;

N = 1; %1 simulated stock prices at time T
% simulate 1 million possible stock prices for each maturity date
sT_1= zeros(N,length(days_MC)-1);
sT_1(:,1)=s0;
for t=1:length(days_MC)-1
  sT_1(:,t+1)= sT_1(:,t).*exp((r-d)*dt-0.5*vol^2*dt+vol*randn(N,1)*sqrt(dt));
end
figure
plot(days_MC,sT_1);
title('1 trial of Monte Carlo Simulation ');
xlabel('maturity time-days');
ylabel('stock price');
legend('stock prices','location','northwest');
%%
%N trials of Monte Carlo Simulation
N = 10000; %10000 simulated stock prices at time T
T=1:252;  %  maturity date in days
dt =T./252 ;
mu =annual_ret ; %average annual return
vol = std(log_ret(end-252:end))*sqrt(252); %annual volatility
K=[s0-30,s0,s0+30,s0+50]; % strike price for forward / options.
% simulate 1 million possible stock prices for each maturity date
sT= zeros(N,length(T));
for t=1:length(T)
 sT(:,t)= s0*exp( (mu-vol^2/2)*dt(t) + vol*sqrt(dt(t))*randn(N,1) );
end
exp_sT= mean(sT);

%derivative prices
dt_LIBOR=T./365;
dicf=exp(-R.*dt_LIBOR);   % discount factor

for t=1:length(K)
 P_forward(t,:) = (exp_sT - K(1,t)).*dicf;   %forward  price
 opt_call(t,:) = max(exp_sT - K(1,t),0).*dicf;  %options price
 opt_put(t,:) = max(K(1,t)-exp_sT,0).*dicf;
 end
 k_mat= repmat(K',1,length(T));
 dicf_mat=repmat(dicf,3);
 Z_bond = k_mat.*dicf;

 %histogram stock price
figure; hold on;
h=histogram(sT);
h.EdgeColor='none';
hold off;
xlabel('possible option price');
ylabel('Frequency');
title('MC output stock price distribution');

 %%
 %consistency
```

```
 %(i)put call parity (out the money K=S0+30)
figure
plot(exp_sT,opt_call(3,:),'o');
hold on;
plot(exp_sT,opt_put(3,:),'o' );

parity_diff=(opt_call(3,:) - opt_put(3,:))-P_forward(3,:);
title('figure(i)put and call option prices K=s0+30');
xlabel('stock prices');
ylabel('option prices');
legend('call option','put option','location','northwest');
hold off;
figure
plot(exp_sT,P_forward(3,:),'o' );

figure
plot(exp_sT, parity_diff);
title('figure(i) put-call parity difference');
xlabel('maturity time-days');
ylabel(' option price');


%(ii)call price with changes of strike price
figure
plot(exp_sT, opt_call,'o');
title('figure(ii)call option prices with different strike prices');
xlabel('stock price');
ylabel('call option price');
legend('in the money(K<S0)','at the money(K=S0)','out the money(K=S0+30)','out the
money(K=S0+50)','location','northwest');

%(ii) call price with changes of strike price
K_call=(s0-50):(s0+50);
T_K_call=40; %T=40 days
exp_sT_K=exp_sT(T_K_call);
opt_call_K=zeros(1,length(K_call));
opt_call_K= (max((exp_sT_K - K_call),0))*dicf(T_K_call);
figure
plot(K_call,opt_call_K,'.');
title('figure(ii)call option prices with different strike prices');
xlabel('strike price');
ylabel('call option price');
legend('T=40 days','location','northeast');


%(iii)option price bound
figure
hold on;
UBound=exp_sT-s0;
LBound=max(exp_sT-(K(1,3)*dicf),0);
plot(exp_sT,UBound);
plot(exp_sT,LBound,'.');
plot(exp_sT,opt_call(3,:));
title('figure(iii)option price bound K=s0+30');
xlabel('stock price');
ylabel('option price');
legend('Upper Bound','call option','lower Bound','location','northwest');

%(v)if d=0 call option price increase with maturity time

figure
plot(T, opt_call(2,:),'.');
title('figure(V)call option prices with different Time to Maturity');
xlabel('maturity time-days');
ylabel('call option price');
legend('at the money(K=S0)','location','northwest');
```

```matlab
%%%%%%%%%%%%%%%%%%%
% Topic 5
% The Greeks
%%%%%%%%%%%%%%%%%%%
clc;clear;
% d1 = (log(S0/K) + (r+sigma^2/2)*T)/(sigma*T^(1/2));
% d2 = d1 - (sigma*T^(1/2));
% N_d1 = normcdf(d1);
% N_m_d1 = normcdf(-d1);
% N_d2 = normcdf(d2);
% N_m_d2 = normcdf(-d2);
% C = S0 * N_d1 - K*exp(-r*T) * N_d2;
% P = K*exp(-r*T) * N_m_d2 - S0 * N_m_d1;
sigma = 0.2551;
rf = 0.087;
S0 = 289.03;
%delta of call option
S_APPL_Greeks = linspace(0,400,100);
Delta_Changing_Spot = zeros(length(S_APPL_Greeks),1);
K_Greeks = 290;
T_Greeks = 180/252;

%% vanilla greek
d1 = (log(S0/K_Greeks) + (rf+sigma^2/2)*T_Greeks)/(sigma*T_Greeks^(1/2));
d2 = d1 - (sigma*T_Greeks^(1/2));
N_d1 = normcdf(d1);
N_m_d1 = normcdf(-d1);
N_d2 = normcdf(d2);
N_m_d2 = normcdf(-d2);
N_deri_d1 = 1/(2 * pi)^0.5 * exp(-d1 ^ 2/2);

Delta_call_v = N_d1;
Gamma_call_v = N_deri_d1/(S0 * sigma * sqrt(T_Greeks));
Vega_call_v = S0 * N_deri_d1 * sqrt(T_Greeks);
Theta_call_v = -((S0*N_deri_d1*sigma)/(2*sqrt(T_Greeks)))-(rf*K_Greeks*exp(-rf*T_Greeks)*N_d2);
Rho_call_v = K_Greeks * T_Greeks * exp(-rf * T_Greeks)* N_d2;
fprintf('the value of Greeks are:
\ndelta: %6.4f\ngamma: %6.4f\nvega: %6.4f\ntheta: %6.4f\nrho: %6.4f\n',Delta_call_v,Gamma_call_v,
Vega_call_v,Theta_call_v,Rho_call_v)

%% vanilla greek Blacksholes testing
n = 1000;
rv = randn(n,1) ;
delta_MC = zeros(n,1);
gamma_MC = zeros(n,1);

for i = 1:n
    delta_MC = (BlackScholes(K_Greeks,T_Greeks,S0+rv(i),sigma,rf) -
BlackScholes(K_Greeks,T_Greeks,S0,sigma,rf))./rv(i);
    gamma_MC = (BlackScholes(K_Greeks,T_Greeks,S0+rv(i),sigma,rf) - 2 *
BlackScholes(K_Greeks,T_Greeks,S0,sigma,rf) + BlackScholes(K_Greeks,T_Greeks,S0 -
rv(i),sigma,rf))/(rv(i)^2);

end
fprintf('The BlackScholes testing result of Greeks
are:\ndelta: %6.4f\ngamma: %6.4f\n',mean(delta_MC),mean(gamma_MC));

% Note: Delta = N(d1) for European Call

for  i = 1:length(S_APPL_Greeks)
     d1_sp = (log(S_APPL_Greeks(i)/K_Greeks)...
         + (rf+((sigma)^2)/2)*T_Greeks)/(sigma*sqrt(T_Greeks));
     Delta_Changing_Spot(i) = normcdf(d1_sp);
end


figure;
plot(S_APPL_Greeks,Delta_Changing_Spot);
title('Delta Vs. Spot Prices');
xlabel('Spot Prices ($)');
ylabel({'Option Greeks: Delta'});
```

```
T_APPL_Greeks = linspace(0,40,100);
K_Greeks_2 = [260 S0 300];
Call_Price_Greeks = zeros(length(T_APPL_Greeks),length(K_Greeks_2));
Delta_Changing_T = zeros(length(T_APPL_Greeks)-1,length(K_Greeks_2));
% Note:Delta =N(d1)
rf = 0.0058;
for i = 1:length(K_Greeks_2)
    for j = 1:length(T_APPL_Greeks)
        d1_ap = (log(S0/K_Greeks_2(i))...
            + (rf+((sigma)^2)/2)*T_APPL_Greeks(j))/...
         (sigma * sqrt(T_APPL_Greeks(j)));
        Delta_Changing_T(j,i) = normcdf(d1_ap);
    end
end
% Graphs (ii)
figure;
hold on
plot(T_APPL_Greeks,Delta_Changing_T);
title('Delta Vs. Time to maturity');
xlabel('Time to maturity in years');
ylabel({'Delta'});
legend('in-the-money','at-the-money','out-of-the-money');
hold off


S_APPL_Greeks = linspace(0,700,100);
Gamma_Changing_Spot = zeros(length(S_APPL_Greeks),1);
for i = 1:length(S_APPL_Greeks)
    d1_ga = (log(S_APPL_Greeks(i)/K_Greeks)...
            + (rf+((sigma)^2)/2)*T_Greeks)/(sigma*sqrt(T_Greeks));
        N_deri_d1 = 1/(2 * pi)^0.5 * exp(-d1_ga ^ 2/2);
        Gamma_Changing_Spot(i) = N_deri_d1/...
            (S_APPL_Greeks(i) * sigma * sqrt(T_Greeks));
end
% Graphs (iii)
figure;
plot(S_APPL_Greeks,Gamma_Changing_Spot);
title('Gamma Vs. Spot Prices');
xlabel('Spot Prices ($)');
ylabel({'Gamma'});


Vol_Vega = linspace(0.001,3,100);
Vega_Changing_Vol = zeros(length(Vol_Vega),1);
for i = 1:length(Vol_Vega)
    d1_vol = (log(S0/K_Greeks)...
           + (rf+((Vol_Vega(i))^2)/2)*T_Greeks)/...
        (Vol_Vega(i) * sqrt(T_Greeks));
         N_deri_d1 = 1/(2 * pi)^0.5 * exp(-d1_vol ^ 2/2);
         Vega_Changing_Vol(i) = S0*N_deri_d1*...
         sqrt(T_Greeks);
end

% Graphs (iv)
figure;
plot(Vol_Vega,Vega_Changing_Vol);
title('Vega Vs. Volatilities');
xlabel('Volatilities');
ylabel({'Vega'});




S_APPL_Greeks = linspace(0,500,100);
Vega_Changing_Spot = zeros(length(S_APPL_Greeks),1);

for i = 1:length(S_APPL_Greeks)
    d1_veg = (log(S_APPL_Greeks(i)/K_Greeks)...
            +(rf+((sigma)^2)/2)*T_Greeks)/(sigma*sqrt(T_Greeks));
            N_deri_d1 = 1/(2 * pi)^0.5 * exp(-d1_veg ^ 2/2);
            Vega_Changing_Spot(i) = S_APPL_Greeks(i) * N_deri_d1 * sqrt(T_Greeks);
```

```
end
% Graphs (v)
figure;
plot(S_APPL_Greeks,Vega_Changing_Spot);
title('Vega Vs. Spot Prices');
xlabel('Spot Prices ($)');
ylabel({'Vega'});




T_APPL_Greeks = linspace(0,40,1000);
Vega_Changing_T = zeros(length(T_APPL_Greeks),1);
for i = 1:length(T_APPL_Greeks)
        d1_vega = (log(S0/K_Greeks)...
              + (rf+((sigma)^2)/2)*T_APPL_Greeks(i))/...
        (sigma * sqrt(T_APPL_Greeks(i)));
        N_deri_d1 = 1/(2 * pi)^0.5 * exp(-d1_vega ^ 2/2);
        Vega_Changing_T(i) = S0* N_deri_d1 *...
                    sqrt(T_APPL_Greeks(i));
end
% Graphs (vi)
figure;
plot(T_APPL_Greeks,Vega_Changing_T);
title('Vega Vs. Time to Maturity');
xlabel('Time to Maturity');
ylabel({'Vega'});



%N_deri_d1 = 1/(2 * pi)^0.5 * exp(-d1 ^ 2/2);
%Gamma = N_deri_d1 /(S0 * sigma * T^0.5);
%Theta = -S0 * N_deri_d1 * sigma/ (2 * T) - r * K * exp(-r*T) * N_d2;
%Vega = S0 * N_deri_d1 * T^0.5;
%Rho = K * T * exp(-r * T) * N_d2;


%%% Monte Carlo Greek (Same random number)
drift = (rf-sigma^2/2)*T_Greeks;
diffusion = sigma*sqrt(T_Greeks);

Opt_p1 = zeros(n,1);
Opt_p2 = zeros(n,1);

rad = randn(n,1);

ST1 = S0*exp( drift + diffusion * rad );
call_payoff1 = get_call_payoff(ST1,K_Greeks);
discounted_payoff1 = call_payoff1 * exp(-rf*T_Greeks);


Opt_p1 = mean(discounted_payoff1);

ST2 = (S0+0.1)*exp(drift + diffusion * rad);
call_payoff2 = get_call_payoff(ST2,K_Greeks);
discounted_payoff2 = call_payoff2 * exp(-rf*T_Greeks);

Opt_p2 = mean(discounted_payoff2);

delta_MC = (Opt_p2 - Opt_p1)/0.1;
fprintf('Monte carlo simulation value of Greeks are: \ndelta: %6.4f\n',delta_MC)



function call_payoff = get_call_payoff(ST, K)

call_payoff = max(ST - K,0);
end

function price = BlackScholes(K,T,S0,sigma,r)

%BlackScholes: Determines the price of an option of type Option from the
```

```
%   inputs using the Black-Scholes-Merton formula

    d1 = (log(S0/K)+(r+1/2*sigma^2)*T)/(sigma*sqrt(T));

    d2 = d1 - sigma*sqrt(T);

    price = (S0 * normcdf(d1) - K*exp(-r*T)*normcdf(d2));

end

function
[BSM_call,delta_call,gamma_call,vega_call,theta_call,rho_call]=BSM_price(S0,K,r,d,sigma,T)

d1=(log(S0/K)+(r-d+0.5*sigma^2)*(T))/(sigma*sqrt(T));
d2=d1-sigma*sqrt(T);

% risk-neutral probabilities
Nd1=normcdf(d1);
Nd2=normcdf(d2);

% call price
BSM_call=S0*exp(-d*T)*Nd1-K*exp(-r*T)*Nd2;

% delta
delta_call=exp(-d*T)*Nd1;

derivatives=1/sqrt(2*pi)*exp(-0.5*d1^2);

%theta
theta_call=-((S0*derivatives*sigma*exp(-d*T))/(2*sqrt(T)))-((-d)*S0*exp(-d*T)*Nd1)-(r*K*exp(-
r*T)*Nd2);

%gamma
gamma_call=(derivatives*exp(-d*T))/(S0*sigma*sqrt(T));
%vega
vega_call=S0*sqrt(T)*derivatives*exp(-d*T);
%rho
rho_call=K*T*exp(-r*T)*Nd2;

end
```

```
% Topic 5
% Monte Carlo Simulation for Exotic Options of COST
clc; clear; close all;

% Pricing Asian options
s0 = 100;
vol = 0.1;
r = 0.05;
d = 0.03;
k = 103;
N = 1e5;
T = 1; % 1YEAR
%%
%1.PRICING ASIAN OPTIONS
%1.1 Asian call option with monthly setting
dt_M =1/12 ;
sT_M= zeros(N,12);
sT_M(:,1)=s0;
for t=1:12
 sT_M(:,t+1)= sT_M(:,t).*exp((r-d)*dt_M-0.5*vol^2*dt_M+vol*randn(N,1)*sqrt(dt_M));
end
exp_P= mean(sT_M);
% histogram
figure;
h=histogram(sT_M);
h.EdgeColor='none';
xlabel('possible stock price');
ylabel('Frequency');
title('MC output stock price distribution');

% price of Asian call options monthly
C_Asian_p= max(mean(sT_M,2)-k,0);
discf=exp(-(r-d)*T);
C_Asian_M=mean(C_Asian_p)*discf;
% price of Vanilla  options monthly
exp_sT_M=max((sT_M(:,end)-k),0);
C_M=mean(exp_sT_M)*discf;
figure;
histogram(C_Asian_p);
title('100000 possible Asian call value at time T ');
xlabel('option value');
ylabel('frequency');
legend('option prices','location','northeast');

%1.2 Asian call option with three-monthly(quaterly) setting
dt_Q =1/4;
sT_Q= zeros(N,4);
sT_Q(:,1)=s0;
for t=1:4
 sT_Q(:,t+1)= sT_Q(:,t).*exp((r-d)*dt_Q-0.5*vol^2*dt_Q+vol*randn(N,1)*sqrt(dt_Q));
end


% price of Asian call options quaterly
C_Asian_Q= max(mean(sT_Q,2)-k,0);

C_Asian_Q=mean(C_Asian_Q)*discf;
% price of Vanilla  options   quaterly
exp_sT_Q=max((sT_Q(:,end)-k),0);
C_Q=mean(exp_sT_Q)*discf;

%1.3 Asian call option with weekly setting

dt_W =1/52;
sT_W= zeros(N,52);
sT_W(:,1)=s0;
for t=1:52
 sT_W(:,t+1)= sT_W(:,t).*exp((r-d)*dt_W-0.5*vol^2*dt_W+vol*randn(N,1)*sqrt(dt_W));
end
% price of Asian call options weekly
C_Asian_W= max(mean(sT_W,2)-k,0);
C_Asian_W=mean(C_Asian_W)*discf;
% price of Vanilla  options weekly
```

```
exp_sT_W=max((sT_W(:,end)-k),0);
C_W=mean(exp_sT_W)*discf;
%% Answer question1
display('1.ANSWER FOR PRICING ASIAN OPTIONS');
fprintf('Asian Quaterly %8.4f  Vanilla Quaterly %8.4f\n ',C_Asian_Q,C_Q);
fprintf('Asian Monthly  %8.4f  Vanilla Monthly %8.4f\n ',C_Asian_M,C_M);
fprintf('Asian Weekly   %8.4f  Vanilla Weekly %8.4f\n ',C_Asian_W,C_W);
display('both option prices are increasing with the decrease of maturity time');
display('price of Asian call option is lower than Vanilla call option ');
%%
%2.pricing discrete barrier option
%2.1 down and out monthly call option with barrier at 80
H_DO=80;
M_logic =min((sT_M>=H_DO),[],2);
C_DO= mean(M_logic*C_M)*discf;
%2.2 down and in monthly call option with barrier at 80
H_DI=80;
M_logic2 =max((sT_M<=H_DI),[],2);
C_DI= mean(M_logic2*C_M)*discf;
%2.3 down and out monthly put option with barrier at 80
exp_sT_M_P=max((k-sT_M(:,end)),0);
P_M=mean(exp_sT_M_P)*discf;
M_logic3 =min((sT_M>=H_DO),[],2);
P_DO= mean(M_logic3*P_M)*discf;
%2.4 down and out monthly put option with barrier at 120 dates at
%0.5,0.15...0.95
H_DO2=120;
t=0.05;
dt_4=0.1;
s0_4=s0.*exp((r-d)*t-0.5*vol^2*t+vol*randn(N,1)*sqrt(t));
sT_4=zeros(N,11);
sT_4(:,1) = s0;
sT_4(:,2) = s0_4;
for i=2:10
    sT_4(:,i+1)=sT_4(:,i).*exp((r-d)*dt_4-0.5*vol^2*dt_4+vol*randn(N,1)*sqrt(dt_4));
end
Put_option=max(k-sT_4(:,end),0);
M_logic4 =1-max((sT_M>=H_DO2),[],2);
P_DO_4=mean(M_logic4.*Put_option)*discf;
%% Answer questions 2
%
display('2.ANSWER FOR pricing discrete barrier option');
fprintf('down and out monthly call option with barrier at 80 %8.4f;Vanilla option %8.4f\n
',C_DO,C_M);
fprintf('down and in monthly call option with barrier at 80 %8.4f;Vanilla option %8.4f\n
',C_DI,C_M);
fprintf('down and out monthly put option with barrier at 80 %8.4f;Vanilla option %8.4f\n
',P_DO,P_M);
fprintf('down and out monthly put option with barrier at 120 %8.4f;Vanilla option %8.4f\n
',P_DO_4,P_M);
DIFF=(C_DO+C_DI)-C_M;
fprintf('difference between down and out plus down and in minus Vanilla call is%8.4f ,which is
close to 0\n ',DIFF);
display(' this holds formula  C = Cdi + Cdo valid');
```