Pricing Asian Options using Monte Carlo Method considering the control variate approach

MA693 Final Project

Hao Ji

151541200

July 30, 2022

Abstract

In this Project, we explore some methods to simulate the options price of the arithmetic average Asian call and put options with a fixed strike, where the methods are the Monte Carlo method in combination with the Control Variate method. Furthermore, we generate options prices under the given circumstance in MATLAB. In addition, we compare our results with the put-call parity method's results. At the end, we conlude our investigated method is in compliance with put-call parity formula and also reduced the variance of options payoff values using control variate method.

Contents

Introduction

Financial derivatives refer to a financial contract based on an underlying financial instrument whose value depends on one or more underlying assets such as commodities, bonds, stocks, and currencies.

Options are a type of derivative financial instrument, which gives the holder or owner the right to buy or sell a certain quantity of a certain commodity at the exercise price, on or before the expiry date. Call options and Put options give the right of their holder or owner to buy shares at a specific price or sell shares at a specific price.

The holder of American-style options can propose to execute the contract on or before the expiration date, while European-style options can only be exercised on a given expiration date.

Geometric Brownian motion is used to imitate stock prices in the Black-Scholes pricing model. For certain options pricing problems that are too complicated to obtain analytical solutions to, the Monte Carlo Method can be an effective method to solve them.

An Asian options is a kind of average price options, where the payoff is a function of the average of the underlying asset price within a certain period of time. The role of Asian options is to avoid unexpected sudden price changes, such as artificially hyping stock prices, etc. It is characterized by lower volatility and lower prices than their analogous options, such as European-style options and American-style options. The type of averaging Asian options can be divided into two categories, which are arithmetic average Asian options and geometric Asian options.

Methods

2.1 The Monte Carlo method

The Monte Carlo method, based on two fundamental laws: the Central Limit Theorem and the Law of Large Numbers,can be used to estimate a mathematical expectation by generating random variables. Many methods are available to generate random variables, such as the LCG(Linear congruential generator) method, inverse CDF(cumulative distribution function) method, etc. The main characteristic of the Monte Carlo method is that approximated results can be calculated on random sampling. With the increase of sampling size, the probability that the obtained results be the correct results gradually increases, but until the real result is obtained, it is impossible to know whether the current result is correct or not.

2.1.1 The Black-Scholes model

The Black-Scholes model is a mathematical method used to modify financial markets, pricing options and derivatives.

The following are some basic well-known pricing formulas for standard European-style (Vanilla) call and put options:

$$Call\ Options\ price$$

$$C(S_0, K, \sigma, r, T) = S_0 N(d_+) - Ke^{-rT} N(d_-)$$

$$Put\ Options\ price$$

$$P(S_0, K, \sigma, r, T) = Ke^{-rT} N(-d_-) - S_0 N(-d_+)$$

$$where$$

$$d_+ = \frac{ln(S_0/K)+(r+\sigma^2/2)T}{\sigma\sqrt{T}},\ d_- = \frac{ln(S_0/K)+(r-\sigma^2/2)T}{\sigma\sqrt{T}}$$

$N\ is\ the\ cumulative\ standard\ normal\ distribution\ function.$

S_0: $initial\ price\ of\ the\ asset$

k: $strike\ price$

σ: $volatility\ parameter$

r: $interest\ rate$

T: $time\ to\ maturity$; $[0, T]$.

2.1.2 The geometric Brownian motion model

The geometric Brownian motion (GBM) is an exponentiated Brownian Motion. To put it simply, it is the continuous-time Stochastic Process, where the logarithm of the variable follows Brownian motion.

For any t, we have

$$S(t) = S_0 e^{\mu t + \sigma W(t)},\ t \geq 0\ ,\ S_0 > 0$$

$$can\ be\ written\ explicitly\ as\ a\ stock\ price\ evolution\ equation:$$

$$S(t) = S_0 e^{(\mu - \frac{\sigma^2}{2})t + (\sigma\sqrt{t})\varepsilon}, t \geq 0, r \geq 0, \sigma > 0$$

$$\mu: expected\ average\ annual\ rate,$$

$$t: time$$

$$\sigma: asset\ price\ volatility$$

$W(t)$: *Standard Brownian motion considered under the risk neutral probability P*

ε: *random number that is a standard normal distribution in the interval* $(0, 1)$.

2.2 Arithmetic average

Given a sequence of stock price $\{S_j\}\}_{j \geq 1}$,we have

$$X_A = \frac{1}{m}\sum_{i=1}^{m} S_{t_i}\ , i = 1, ..., m.$$

where the asset price on [0,T].

2.3 Asian options

In general form of options, the payoff value at time T is as follows

$$V_{call} = E[(S(T) - K)^+]$$

$$V_{put} = E[(K - S(T))^+]$$

For the call options, at time T if the stock price S(T) be more than the strike price K, we will exercise our options and buy the stock at the price of K and we can sell that at the market price S(T), and receive the payoff of $S(T) - K$. Obviously, if the stock price S(T) is less than K, we will not exercise our call options and will receive nothing as payoff.

On the other hand, for the put options,if the S(T) be less than K, we will exercise put options and receive the payoff of $K - S(T)$.

Asian options is a special type of options where the payoff depends on the average price of the underlying asset over a certain period of time, unlike European or American options whose payoff depends on the current price of the asset. To calculate this options price, we equally divide the time horizon[0,T] into m blocks, then the time step can be denoted by h, where $h = \frac{T}{m}$. The respective payoff functions of a fixed strike (K) arithmetic Asian call and put are:

Call Options

$$f(\left\{S_{t_i}\right\}_{0\leq i\leq m}, r, \sigma, K, T) = max(X_A - K, 0) = (X_A - K)^+$$

Put Options

$$f(\left\{S_{t_i}\right\}_{0\le i\le m}, r, \sigma, K, T) = max(K - X_A, 0) = (K - X_A)^+$$

The options payoff of the fixed strike arithmetic average Asian call options and put options can be estimated by using the following:

Present value of Call options payoff

$$V_{call} = E[e^{-rT}(X_A - K)^+]$$

Present value of Put options payoff

$$V_{put} = E[e^{-rT}(K - X_A)^+]$$

2.4 Put-Call-Parity

Put-call parity for standard (vanilla) options is the relationship between a put, a call and the underlying futures price, where the puts and calls share the same strike K and expiry T,i.e:

$$Payoff_{call} = max(S_T - K, 0)$$
$$Payoff_{put} = max(K - S_T, 0)$$
$$Payoff_{call} - Payoff_{put} = S_T - K$$

For Asian options payoff ,the put-call parity will be :

$$C - P = (X_A - K)^+ - (K - X_A)^+ = X_A - K$$

and then we apply Monte Carlo method :

$$\Rightarrow e^{-r(T-t)}E[X_A - K] = e^{-r(T-t)}E[X_A] - e^{-r(T-t)}K = S_t - Ke^{-r(T-t)}$$

based on above relation we can get:

$$C_t = E[e^{-r(T-t)}(K - X_A)^+] + S_t - Ke^{-r(T-t)}$$

or

$$P_t = E[e^{-r(T-t)}(X_A - K)^+] - S_t + Ke^{-r(T-t)}$$

$$C_t: call\ options\ price\ on\ time\ t,\ \ P_t: put\ options\ price\ on\ time\ t$$

$$r: risk - free\ return\ rate,\ T: expiry\ date$$

$$t: time,\ K: strike\ price$$

$$S_t: price\ of\ the\ asset\ at\ time\ t.$$

2.5 Control Variate Method

The Control Variate Method is one of the best methods for variance reduction purposes to reduce the variance, which we can use to improve the convergence of the Monte Carlo methods.

For the expected value of control variate Y_{call} and Y_{put} , we use the geometric average Asian call options and put options on the same asset with the same maturity. The options price at t=0 are given by:

$$Y_{call} = e^{-rT}max(G_A - K, 0)$$

$$Y_{put} = e^{-rT}max(K - G_A, 0)$$

where $G_A = (\prod_{i=1}^{m} S_{t_i})^{\frac{1}{m}}$.

The respective prices are given by the expectation of Y_{call} and Y_{put},

$$E[Y_{call}] = E[e^{-rT}(G_A - K)^+] = e^{r_0T_0 - rT}\{e^{-r_0T_0}[(S_0 e^{\frac{\sigma_0^2 T_0}{2}})e^{(r_0 - \frac{\sigma_0^2}{2})T_0 + \sigma_0\sqrt{T}Z} - K]^+\}$$

$$= e^{r_0T_0 - rT}C_{bsm}(S_0 e^{\frac{\sigma_0^2 T_0}{2}}, K, \sigma_0, r_0, T_0)$$

$$E[Y_{put}] = E[e^{-rT}max(K - G_A, 0)] = e^{r_0 T_0 - rT}\{e^{-r_0 T_0}[K - (S_0 e^{\frac{\sigma_0^2 T_0}{2}})e^{(r_0 - \frac{\sigma_0^2}{2})T_0 + \sigma_0\sqrt{T}Z}]^+\}$$

$$= e^{r_0 T_0 - rT} P_{bsm}(S_0 e^{\frac{\sigma_0^2 T_0}{2}}, K, \sigma_0, r_0, T_0)$$

$$where$$

$$T_0 = \frac{m+1}{2m}T, \sigma_0 = \frac{2m+1}{3m}\sigma^2, r_0 = r - \frac{1}{2}\sigma^2,$$

$$C_{bsm}(S_0, K, \sigma_0, r_0, T_0) = S_0 N(d_+) - Ke^{-rT} N(d\_),$$

$$P_{bsm}(S_0, K, \sigma_0, r_0, T_0) = Ke^{-rT} N(-d\_) - S_0 N(-d_+),$$

$$d_+ = \frac{ln(S_0/K)+(r+\sigma^2/2)T}{\sigma\sqrt{T}}, \; d\_ = \frac{ln(S_0/K)+(r-\sigma^2/2)T}{\sigma\sqrt{T}},$$

$$N \text{ is the standard normal cumulative distribution function.}$$

Then the general formula for Z_{call} and Z_{put} are

$$Z_{call} = e^{-rT}(X_A - K)^+ - c[Y_{call} - E[Y_{call}]]$$

$$Z_{put} = e^{-rT}(K - X_A)^+ - c[Y_{put} - E[Y_{put}]]$$

where Z_{call} is an unbiased estimator of the arithmetic Asian call options payoff V_{call}, Z_{put} is an unbiased estimator of the arithmetic Asian put options payoff V_{put}.

The optimal parameter c^* can be estimated by the following formula:

$$c^* = \frac{Cov(X,Y)}{Var(Y)}$$

Numerical result

In this section we are showing the results of pricing an Arithmetic average fixed strike Asian Call and Put Options under the GBM using Monte Carlo Method and the same model's results considering the Control Variate method (with initial parameters), It worth to mention that we ran our code by a computer with Core i7-8700k CPU and 32GB RAM.

$S = \$90\,, K = \$100,\ r = 0.05, \sigma = 0.3, t = 0, T = 2, m = 32,\ n = 10 * 2^{P},$

$P = 11, 12, 13, 14, 15, 16$

In the following Tables, first column from left hand side is the number of Monte Carlo loops, the second column is options payoff value, the third column is the call/put payoff value based on put-call parity method utilizing the results of our GBM method; it means we used call payoff values resulted from our GBM method to calculate put payoff values based on put-call parity method and same thing for put options , the fourth column is variance of options payoff values after applying Monte Carlo Method, the fifth column is standard error which is provided by the formula of $\sqrt{\frac{Variance}{n^{p}}}$, and the last column is run time in seconds.

Table 1: Asian Call Options

n	Call(GBM)	Put(GBM put-call)	Variance	stdError	time
20480	6.78300156	7.26674337	205.39086596	0.10014415	0.03969750
40960	6.73580893	7.21955074	202.41113853	0.07029707	0.04852320
81920	6.80523139	7.28897319	210.06381910	0.05063848	0.08810800
163840	6.80224520	7.285987	209.39237454	0.03574954	0.17905880
327680	6.75493104	7.23867285	207.79334155	0.02518204	0.35211130
655360	6.77734194	7.26108374	208.70090944	0.01784523	0.70260380

Table 1.1: Asian Put Options

n	Put(GBM)	Call(GBM put-call)	Variance	stdError	time
20480	11.42440736	10.94066556	167.4514212	0.09042311	0.0584101
40960	11.43620937	10.95246757	167.3578173	0.06392092	0.0817964
81920	11.48053319	10.99679139	168.6356958	0.04537115	0.1429081
163840	11.41468508	10.93094328	167.6890434	0.03199207	0.2829377
327680	11.46287262	10.97913082	168.0001749	0.02264279	0.5710489
655360	11.43199499	10.94825318	167.9746847	0.01600965	1.1398214

In the following Tables (2 and 2.1), second column from left hand side is the call options price based on GBM model, the third column is put options price based on put-call parity method using the second column's values and the fourth column is the call options based on the BSM as a benchmark for comparison.

Table 2: Asian Call Options with Control Variate method

n	Call(GBM)	Put(GBM put-call)	Call(BSM)	Variance	stdError	Time
20480	6.76677731	7.25051912	12.60180502	1.02658517	0.00707999	0.05930300
40960	6.75266006	7.23640187	10.48322036	0.93466440	0.00477692	0.08344180
81920	6.77186817	7.25560998	15.61588955	1.01517133	0.00352026	0.16662310
163840	6.76462571	7.24836751	9.91799824	1.00510657	0.00247683	0.33248540
327680	6.76647500	7.25021681	8.87124577	0.99796498	0.00174515	0.66344650
655360	6.76613264	7.24987444	32.26670210	1.00580874	0.00123885	1.33105180

Table 2.1: Asian Put Options with Control Variate method

n	Put(GBM)	Call(GBM put-call)	Put(BSM)	Variance	stdError	Time
20480	11.46724306	10.98350126	17.24457526	0.35112861	0.00414065	0.05923580
40960	11.46495473	10.98121292	19.26340029	0.35168506	0.00293020	0.08150650
81920	11.46962928	10.98588748	14.90961804	0.34481390	0.00205162	0.16245040
163840	11.46926956	10.98552776	19.87114782	0.34337211	0.00144768	0.32173380
327680	11.46784122	10.98409941	21.09129117	0.34867336	0.00103154	0.64346210
655360	11.46899612	10.98525432	7.60939541	0.34494731	0.00072550	1.28335570

We used a benchmark method’s result to compare with out GBM model. To do this, we considered standard BSM to reach call and put toptions prices using all parameters we have used in the GBM. In the BSM, we used Arithmetic average of stock price instead of S(t), beacuse our options are in Asian-style.

When we compare the results from the simulation, we can see that the reduction of the standard error is about half as the increase of the sample size n when it’s doubled up. Also, as we compare the results between [Table 1 and Table 2(Call Options)] and [Table 1.1 and Table 2.1(Put Options)], we can easily see that the options price values are very close to each other. Moreover, as we can see in the fourth column, the call/put options prices of BSM are different from GBM results. These results are fluctruating a lot while GBM’s values are consistent by varying n. Furthermore, the variance and standard error are decreasing, which has been reduced very well after applying the Control Variate Method. Therefore, the Control Variate Method is more efficient to improve the accuracy of derivatives pricing.

In the following Tables, first column from left hand side is the number of time steps M, the second column is call options payoff value, the third column is the variance of call options payoff values after applying Monte Carlo Method, the fourth column is standard error, the fifth column is put options payoff value, the sixth column is variance of call options payoff values after applying Monte Carlo Method and the last column is seventh column is standard error.

Table 3: Asian call and put options on different M with Control Variate method ,with initial parameters:

$S = \$90$, $K = \$100$, $r = 0.05, \sigma = 0.3, T = 2, m = 2, 4, 8, 16, 32, 64$, $n = 10 * 2^{16} = 655360.$

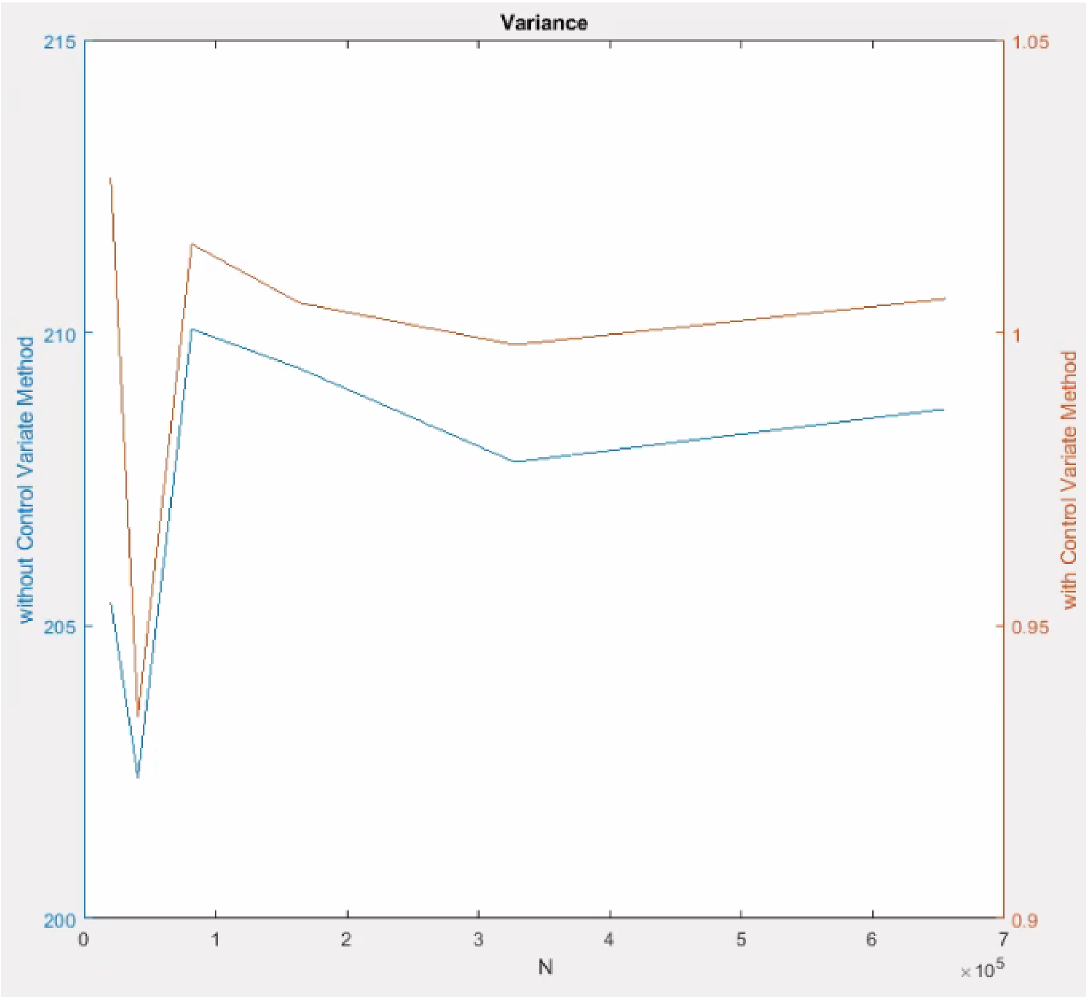
M	call option	variance	stderror	put option	variance	stderror
2	10.64255308	2.20452468	0.00183408	13.31980033	0.65358016	0.00099864
4	8.55056497	1.5511519	0.00153846	12.31307436	0.50650167	0.00087912
8	7.52374954	1.20862715	0.00135802	11.8246551	0.40853956	0.00078955
16	7.0154536	1.07375116	0.00128001	11.58671512	0.36527121	0.00074657
32	6.76663728	0.99974718	0.00123511	11.46858723	0.3484476	0.00072917
64	6.64055331	0.98626892	0.00122675	11.40937072	0.33881727	0.00071902

As we can see in the Table 3, as the number of time steps M (number of subintervals) increases, the variance of the payoff values decreases and the amounts are convergent.

Chart 4: Variance with initial parameters:

$S = \$90, K = \$100, r = 0.05, \sigma = 0.3, T = 2, m = 32, n = 10 * 2^{P}$,

$P = 11, 12, 13, 14, 15, 16$



In Chart 4,the blue line is the variance based on Monte Carlo Method and the orange line is the variance based on Monte Carlo Method considering control variate method,which shows the variance of the simulation with the control variate method and without the control variate method on a different scale, in which the two lines are the same in variation trend and up-and-down similarly when the number of Monte Carlo loops N is increasing.

Chart 5: Call options price based on different expiry times (T) with initial parameters:

$S = \$90\ , K = \$100,\ r = 0.05, \sigma = 0.3, T = \{1, 9, 15\}, m = 32,$

$n = 10 * 2^{16} = 1655360.$

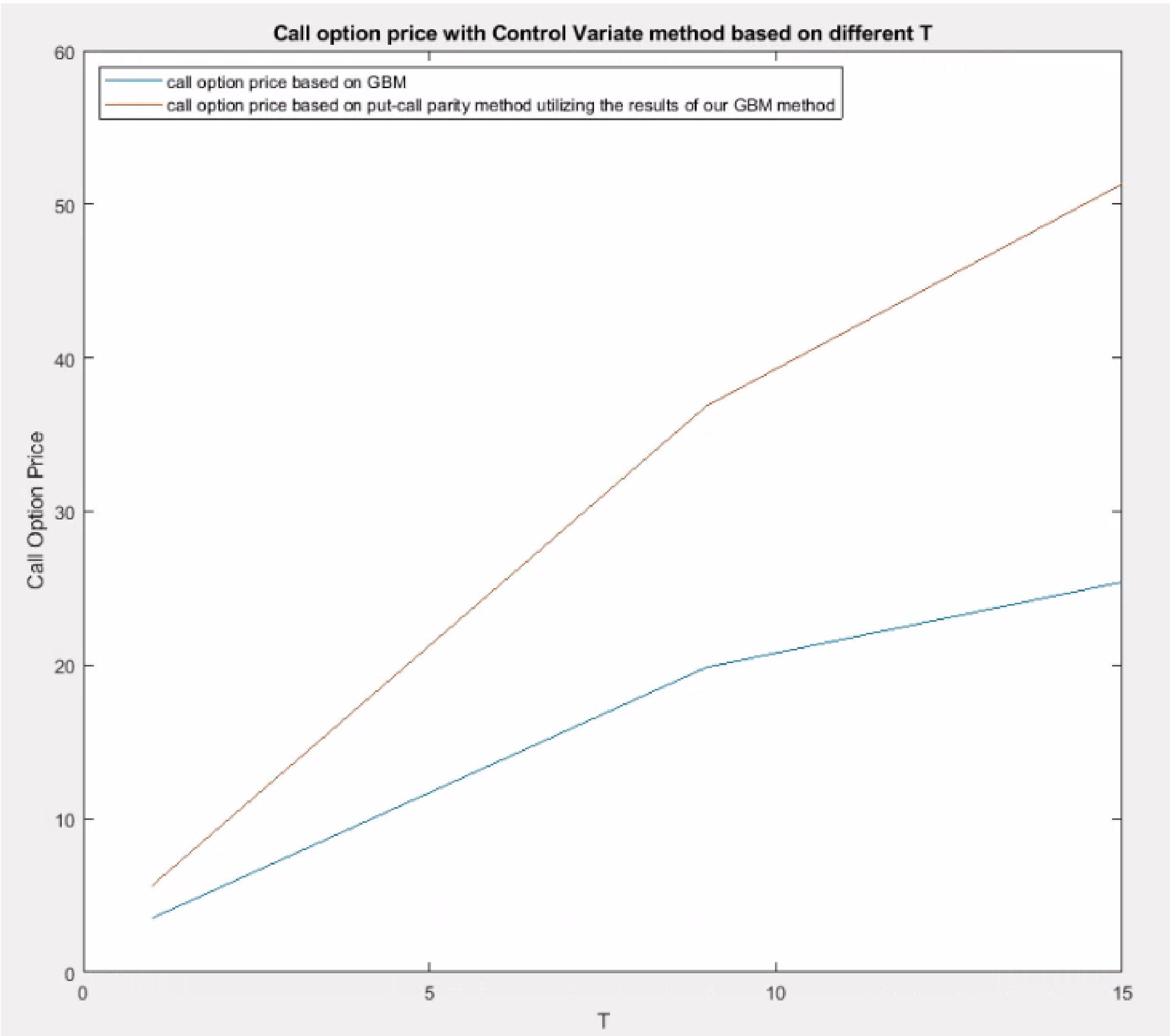
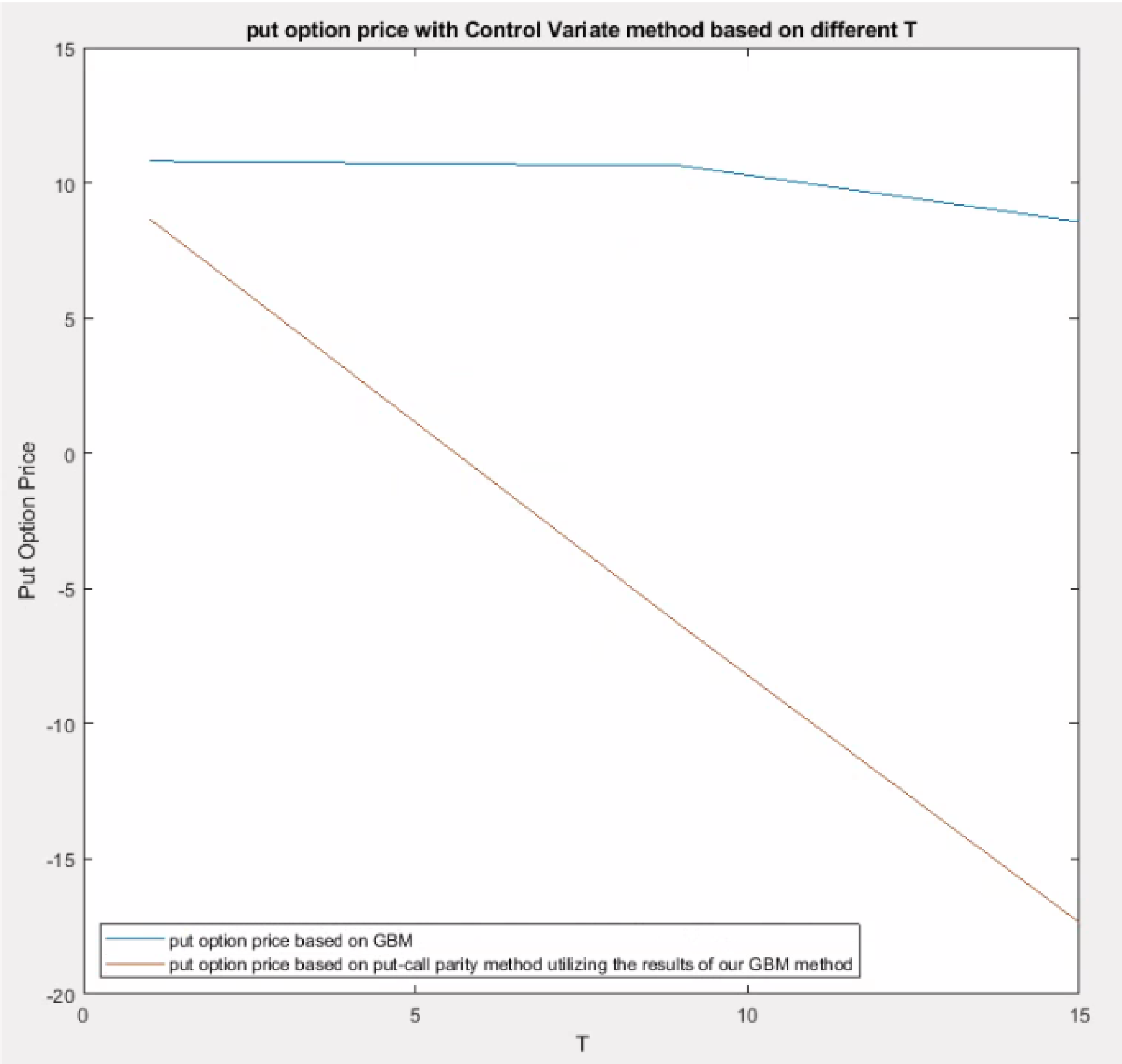


Chart 5.1: Put options price based on different expiry times (T) with initial parameters:

$S = \$90, K = \$100, r = 0.05, \sigma = 0.3, T = \{1, 9, 15\}, m = 32,$

$n = 10 * 2^{16} = 1655360.$



In the charts (Chart 5 & Chart 5.1), as the expiry time T increases from 1 to 9, we can see that the call options price based on GBM and the call options price based on put-call parity method utilizing the results of our

GBM method are both increasing. On the other hand, the put options price based on GBM and the put options price based on the put-call parity method utilizing the results of our GBM method are both decreasing.

Chart 6: Call options price based on different strikes (K) with initial parameters:

$S = \$90\ , K = \{\$100, \$110, \$120\},\ r = 0.05, \sigma = 0.3, T = 2, m = 32,$

$n = 10 * 2^{16} = 1655360.$

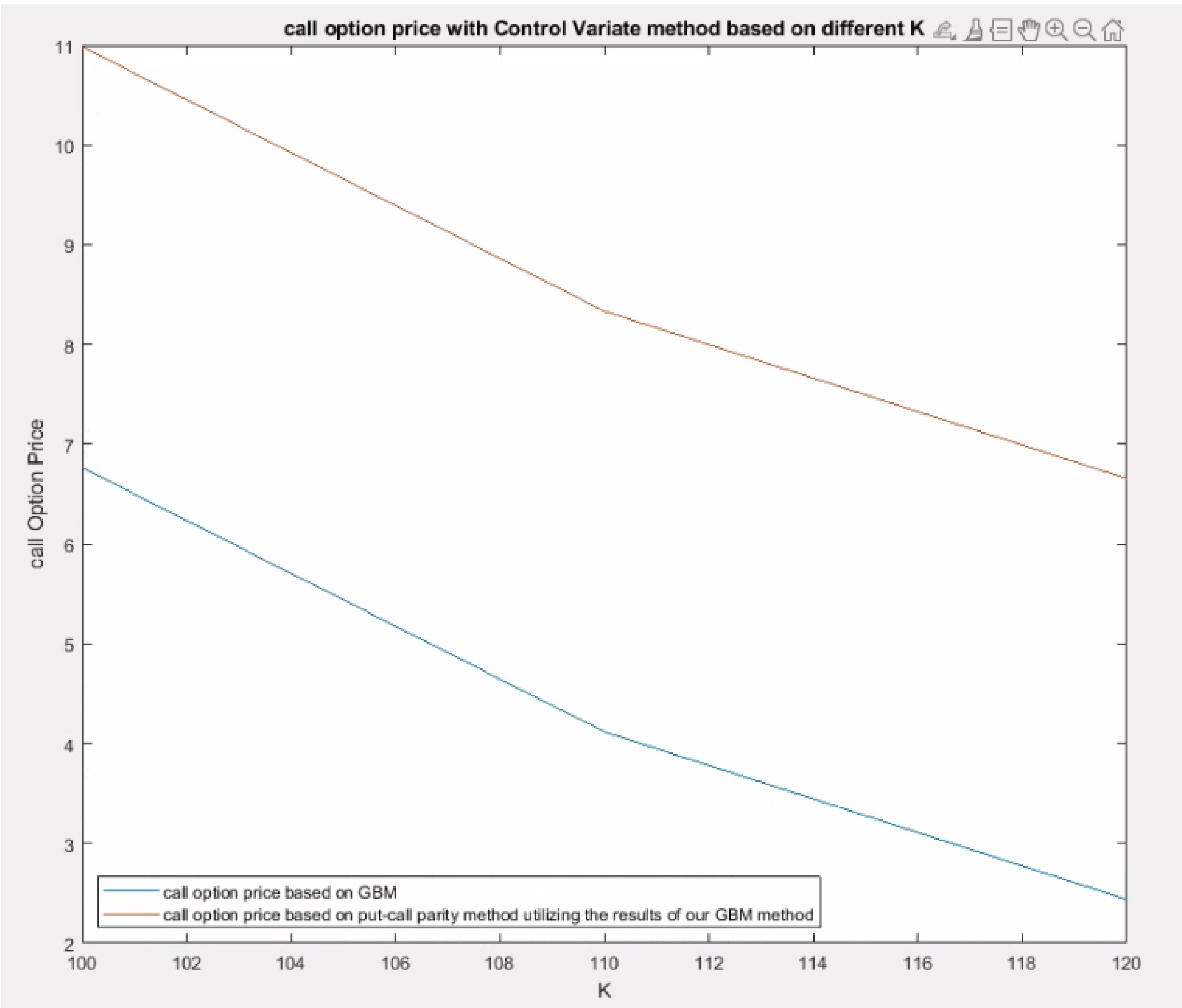
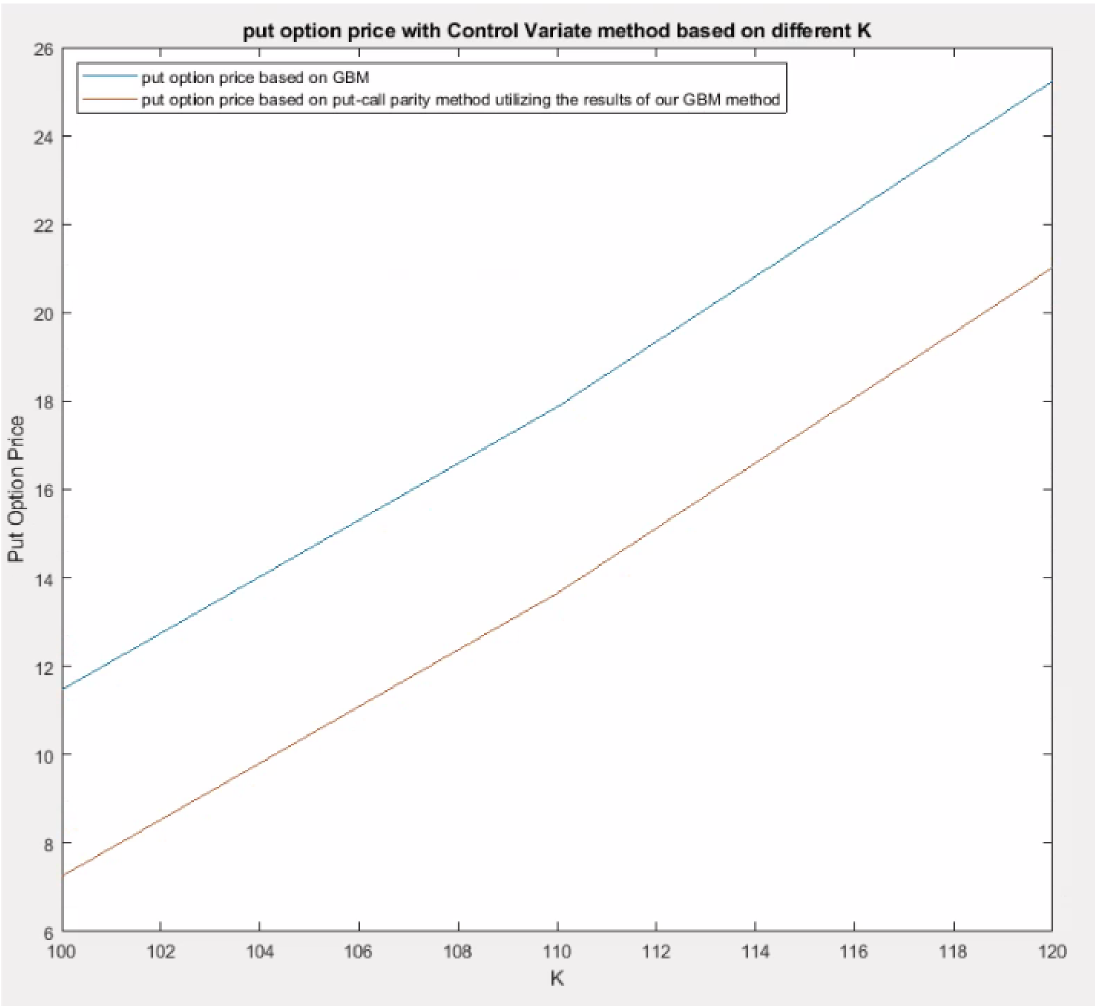


Chart 6.1: Put options price based on different strikes (K) with initial parameters:

$S = \$90\ , K = \{\$100, \$110, \$120\},\ r = 0.05, \sigma = 0.3, T = 2, m = 32,$

$n = 10 * 2^{16} = 1655360.$



In the charts (Chart 6 & Chart 6.1), the strike price K is increasing from \$100 to \$120, we can see that the call options price based on GBM and the call options price based on put-call parity method utilizing the results of our GBM method are both decreasing. On the other hand, the put options price based on GBM and the put options price based on the put-call parity method utilizing the results of our GBM method are both increasing.

Therefore, from the above charts, we can easily see that whether K or T is changing, they are all the same in variation trends between options price based on GBM and options price based on put-call parity method utilizing the results of our GBM method. Also, Based on the plots, the GBM results are compliant with put-call parity results.

Conclusions

We have employed GBM path simulations to compute the prices of Arithmetic average Asian call options and put options, and also with a fixed strike we used the Monte Carlo method. And by using the control variate method, which is one of the variance reduction techniques, we can conclude that the performance of the control variate method to reduce the variance is efficient. We also compared the GBM,s results with BSM as a benchmark. In addition, as the total number of simulations grows, the variance of prices decreases and we converge to a more accurate result, but we have to spend more time to run the code of the implemented method.

References

[1] Giuseppe Campolieti and Roman N. Makarov.(2014). Financial Mathematics: A Comprehensive Treatment.March)

[2] Paul Glasserman.(2004).Monte Carlo Methods in Financial Engineering.

Appendix

Appendix [Asian put options with Monte Carlo method]

```
rng(10);
clc;
clear;
asian_mc_put(2.0,32)

function asian_mc_put(T,m)

K=100;
r=0.05;
s0=90; sig=0.3;
h=T/m;

fprintf(' n          put-option      Put-Call parity(call)      Variance
stdError              time       \n')
r0=r-sig^2/2;
for i=1:m
    t1(i)=i*h;
end

for p=1:6
    n(p)=10*2^(p+10);
end
for p=1:6
    tic;
    temp1=0.0;
    temp2=0.0;
    for k=1:n(p)

        s1(1)=s0;
        for i=1:m
              rni=randn;
              s1(i+1)=s1(i)*exp(r0*h+sig*sqrt(h)*rni);
              s(i)=s1(i+1);

        end
        temp1=temp1+max(K-sum(s)/m,0);
        temp2=temp2+max(K-sum(s)/m,0)*max(K-sum(s)/m,0);
    end
    v=(exp(-r*T))*temp1/n(p);
    var=(exp(-r*T))*((n(p)*temp2-temp1^2)/((n(p)-1)*n(p)));
    stddev=sqrt(var);
    stderr=stddev/sqrt(n(p));
    time=toc;
     %put-call parity to calculate call option based on put option
     C=v+s0-(K/(exp(r*T)));
     fprintf('%6d  %15.8f   %15.8f    %15.8f    %15.8f     %15.8f
\n',n(p),v,C,var,stderr,time);
end
end
```

Appendix [Asian call options with Monte Carlo method]

```
rng(10);
clc;
clear;
asian_mc_call(2.0,32)

function asian_mc_call(T,m)
K=100;
r=0.05;
s0=90; sig=0.3;
h=T/m;
fprintf(' n          call-option     Put-Call parity(put)     Variance
stdError              time       \n')
r0=r-sig^2/2;
for i=1:m
    t1(i)=i*h;
end
for p=1:6
    n(p)=10*2^(p+10);
end
for p=1:6
    tic;
    temp1=0;
    temp2=0;
    for k=1:n(p)
        s1(1)=s0;
        for i=1:m
              rni=randn;
              s1(i+1)=s1(i)*exp(r0*h+sig*sqrt(h)*rni);
              s(i)=s1(i+1);
        end
        temp1=temp1+max(sum(s)/m-K,0);
        temp2=temp2+max(sum(s)/m-K,0)*max(sum(s)/m-K,0);
    end
    v=(exp(-r*T))*temp1/n(p);
    var=(exp(-r*T))*((n(p)*temp2-temp1^2)/((n(p)-1)*n(p)));
    stddev=sqrt(var);
    stderr=stddev/sqrt(n(p));
    time=toc;
    %put-call parity to calculate put option based on call option
P=v-s0+(K/(exp(r*T)));

     fprintf('%6d  %15.8f   %15.8f    %15.8f    %15.8f     %15.8f
\n',n(p),v,P,var,stderr,time);
end
end
```

Appendix [Asian put options with Monte Carlo method considering control variate]

```
rng(10);
clc;
clear;
asian_mc_put_cv(2.0,32)
function asian_mc_put_cv(T,m)
K=100;
r=0.05;
s0=90; sig=0.3;
h=T/m;
fprintf(' n           put(GBM)            call(GBM)           put(BSM)
Variance            stdError             time      \n')
% calculate expectation_Y under the BSM model
r0=r-sig^2/2;
sig0=sig*sqrt((2*m+1)/(3*m));
T0=(m+1)/(2*m)*T;
expectation_Y=exp(r0*T0-r*T)*asian_put_bsm(s0*exp(0.5*sig0*sig0*T0),K,sig0
,r0,T0);
for p=1:6
 n(p)=10*2^(p+10);
end
s(1)=s0;s2(1)=s0;
for p=1:6
   tic;
   temp1=0;
   temp2=0;
   temp3=0;
   temp4=0;
   temp5=0;
   for j=1:n(p)
     %GeometricalBrownianMotion
       for i=2:m+1
            rni=randn;

           s2(i)=s2(i-1)*exp((r-sig^2/2)*h+sig*sqrt(h)*rni); %asset price
under BSM model
           s1(i-1)=s2(i);
           s3(i-1)=log(s2(i));
       end
       ave_a1=sum(s1)/m;
       ave_g1=exp(sum(s3))^(1/m);
       geo_payoff=max(K-ave_g1,0);
       arith_payoff=max(K-ave_a1,0) ;
       temp1=temp1+arith_payoff;
       temp2=temp2+arith_payoff^2;
       temp3=temp3+geo_payoff;
       temp4=temp4+geo_payoff^2;
       temp5=temp5+geo_payoff*arith_payoff;
   end
S=sum(s1)/m;
d1=(log(S/K)+(r+0.5*sig*sig)*T)/(sig*sqrt(T));
d2=(log(S/K)+(r-0.5*sig*sig)*T)/(sig*sqrt(T));
v2=K*exp(-r*T)*normcdf(-d2)-S*normcdf(-d1);
```

```
    lamda_st=(n(p)*temp5-temp1*temp3)/(n(p)*temp4-temp3*temp3);
    v=(exp(-r*T))*((temp1-lamda_st*temp3)/(n(p)))+lamda_st*expectation_Y;

temp6=n(p)*temp2-temp1*temp1+lamda_st*lamda_st*(n(p)*temp4-temp3*temp3);
    temp7=2*lamda_st*(n(p)*temp5-temp1*temp3);
    varx=(exp(-r*T))*(temp6-temp7)/((n(p)-1)*n(p));
    stderr=sqrt(varx)/sqrt(n(p));
    time=toc;
    %put-call parity to calculate call option based on put option
     C=v+s0-(K/(exp(r*T)));
     fprintf('%6d  %15.8f   %15.8f   %15.8f    %15.8f    %15.8f     %15.8f
\n',n(p),v,C,v2,varx,stderr,time);
end
end

function z=asian_put_bsm(S0,K,sig,r,t)
d1=(log(S0/K)+(r+sig^2/2)*t)/(sig*sqrt(t));
d2=(log(S0/K)+(r-sig^2/2)*t)/(sig*sqrt(t));
z=K*exp(-r*t)*normcdf(-d2)-S0*normcdf(-d1);
end
```

Appendix [Asian call options with Monte Carlo method considering control variate]

```
rng(10);
clc;
clear;
asian_mc_call_cv(2.0,32)
function asian_mc_call_cv(T,m)
K=100;
r=0.05;
s0=90; sig=0.3;
h=T/m;
fprintf(' n          call(GBM)            put(GBM)          call(BSM)
Variance            stdError               time      \n')
% calculate expectation_Y under the BSM model
r0=r-sig^2/2;
sig0=sig*sqrt((2*m+1)/(3*m));
T0=(m+1)/(2*m)*T;
expectation_Y=exp(r0*T0-r*T)*asian_call_bsm(s0*exp(sig0*sig0*T0/2),K,sig0,
r0,T0);
s(1)=s0;s2(1)=s0;
for p=1:6
 n(p)=10*2^(p+10);
end
for p=1:6
   tic;
   temp1=0;
   temp2=0;
   temp3=0;
   temp4=0;
   temp5=0;
   for j=1:n(p)
       %GeometricalBrownianMotion
       for i=2:m+1
            rni=randn;
           s2(i)=s2(i-1)*exp((r-sig^2/2)*h+sig*sqrt(h)*rni); %asset price
under BSM model
           s1(i-1)=s2(i);
           s3(i-1)=log(s2(i));
       end
    ave_a1=sum(s1)/m;
       ave_g1=exp(sum(s3))^(1/m);
       geo_payoff=max(ave_g1-K,0);
       arith_payoff=max(ave_a1-K,0);
       temp1=temp1+arith_payoff;
       temp2=temp2+arith_payoff^2;
       temp3=temp3+geo_payoff;
       temp4=temp4+geo_payoff^2;
       temp5=temp5+geo_payoff*arith_payoff;
   end
   S=sum(s1)/m;
   d1=(log(S/K)+(r+sig^2/2)*T)/(sig*sqrt(T));
d2=(log(S/K)+(r-sig^2/2)*T)/(sig*sqrt(T));
v2=S*normcdf(d1)-K*exp(-r*T)*normcdf(d2);
```

```
lamda_st=(n(p)*temp5-temp1*temp3)/(n(p)*temp4-temp3*temp3);
   v=(exp(-r*T))*((temp1-lamda_st*temp3)/(n(p)))+lamda_st*expectation_Y;

temp6=n(p)*temp2-temp1*temp1+lamda_st*lamda_st*(n(p)*temp4-temp3*temp3);
   temp7=2*lamda_st*(n(p)*temp5-temp1*temp3);
   varx=(exp(-r*T))*(temp6-temp7)/((n(p)-1)*n(p));
    stderr=sqrt(varx)/sqrt(n(p));
   time=toc;
   %put-call parity to calculate put option based on call option
  P=v-s0+(K/(exp(r*T)));
   fprintf('%6d  %15.8f   %15.8f   %15.8f    %15.8f    %15.8f     %15.8f
\n',n(p),v,P,v2,varx,stderr,time);

end
end
 function z=asian_call_bsm(S0,K,sig,r,t)
d1=(log(S0/K)+(r+sig^2/2)*t)/(sig*sqrt(t));
d2=(log(S0/K)+(r-sig^2/2)*t)/(sig*sqrt(t));
z=S0*normcdf(d1)-K*exp(-r*t)*normcdf(d2);
end
```

Appendix [Variance with different N]

```
rng(10);
clc;
clear;

C=[205.390866
202.4111385
210.0638191
209.3923745
207.7933416
208.7009094
];
t=[20480
40960
81920
163840
327680
655360
];
C_mat=[1.02658517
0.9346644
1.01517133
1.00510657
0.99796498
1.00580874
];
t_1=[0.04279360  0.08290810 0.16425430 0.32439950 0.64401220
1.30088260];

figure

%hold on;
[hAx,hLine1,hLine2] =plotyy(t,C,t,C_mat);
title('Variance')
xlabel('N')

ylabel(hAx(1),'without Control Variate Method') % left y-axis
ylabel(hAx(2),'with Control Variate Method') % right y-axis
```

Appendix [Asian call options with different T considering control variate method]

```
rng(10);
clc;
clear;

C=[3.53660387
19.85982136
25.40676371
];
t=[1
9
15
];
C_mat=[5.68048233
36.87093901
51.32411235
];


figure

plot(t,C);
title('Call option price with Control Variate method based on
different T')
xlabel('T')
hold on;
plot(t,C_mat);
het1=legend('call option price based on GBM','call option price based
on put-call parity method utilizing the results of our GBM method')
set(het1,'Location','NorthWest')
ylabel('Call Option Price')
```

Appendix [Asian put options with different T considering control variate method]

```
rng(10);
clc;
clear;

C=[10.80342478
10.63375417
8.56076762
];
t=[1
9
15
];
C_mat=[8.65954632
-6.37736348
-17.35658101
];


figure

plot(t,C);
title('put option price with Control Variate method based on
different T')
xlabel('T')
hold on;
plot(t,C_mat);
het1=legend('put option price based on GBM','put option price based
on put-call parity method utilizing the results of our GBM method')
set(het1,'Location','SouthWest')
ylabel('Put Option Price')
```

Appendix [Asian call options with different K considering control variate method]

```
rng(10);
clc;
clear;

C=[6.76613264
4.11658853
2.43784157
];
t=[100
110
120
];
C_mat=[10.98525432
8.33440709
6.65722362
];


figure

plot(t,C);
title('call option price with Control Variate method based on
different K')
xlabel('K')
hold on;
plot(t,C_mat);
het1=legend('call option price based on GBM','call option price based
on put-call parity method utilizing the results of our GBM method')
set(het1,'Location','SouthWest')
ylabel('call Option Price')
```

Appendix [Asian put options with different K considering control variate method]

```
rng(10);
clc;
clear;

C=[11.46899612
17.86652308
25.23771379
];
t=[100
110
120
];
C_mat=[7.24987444
13.64870451
21.01833174
];

figure

plot(t,C);
title('put option price with Control Variate method based on
different K')
xlabel('K')
hold on;
plot(t,C_mat);
het1=legend('put option price based on GBM','put option price based
on put-call parity method utilizing the results of our GBM method')
set(het1,'Location','NorthWest')
ylabel('Put Option Price')
```