Math 623 **Semester :F03**
HW#4 Solution

Bear Spread option = Put (K1) – Put (K2) ,where K1=60 and K2=50
R=0.02, T=1, σ =0.2, S_0=100, no dividend

1a) $S(T) = S_0\, e^{((r-\frac{\sigma^2}{2})T+\frac{\sigma^2}{2}\sqrt{T}\xi)}$ under risk-neutral measure denoted by Q
where $\xi \sim N(0,1)$ and i.i.d.

1b) Bear spread option value = $P_{BS}(S_0$, K1=60, σ ,r ,T) – $P_{BS}(S_0$, K2=50, σ , r, T)
= 0.01879 – 0.00062541
= 0.01817

1c), d), e)
See code below

1f) As outlined in class, we need to know α such that

$$E^{\tilde{Q}}\left[S_0 e^{((r-\frac{\sigma^2}{2})T+\sigma\sqrt{T}(\eta-\alpha))} \right] = 55$$

where $\eta \sim N(0,1)$ and i.i.d. under $\tilde{Q}$-new measure
Taking expectation under new measure (treat α as deterministic), we get

$$\alpha = \frac{rT - \ln(\frac{55}{S_0})}{\sigma\sqrt{T}}$$ Substitute variables, we get α =3.08919

Steps
- Set α as above
- Generate N independent N(0,1)'s $\eta^{(1)}, \eta^{(2)}, \ldots, \eta^{(N)}$
- Compute S(T) under $\tilde{Q}$

$$S(T) = S_0 e^{(r-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}(\eta-\alpha)}$$

- Compute discounted option price along the paths

$$V^{(n)} = e^{-rT}[Max(K1-S(T),0) - Max(K2-S(T),0)]e^{\eta\alpha-\frac{1}{2}\alpha^2}$$

- Compute successive Monte Carlo estimates

$$\hat{V}_n = \frac{1}{n}\sum_{i=1}^{n} V^{(i)}$$

- Compute Standard Error, report N, $\hat{V}_N$ and plot Convergence diagrams.
See code below

1g), 1h)
See attachment

1i)

```
>> 1c) No. of Path=     1000000
        MC estimate=     0.018180
        Standard Error   0.000320
>> 1d) No. of Path=     1000000
        MC estimate=     0.018055
        Standard Error   0.000318
>> 1e) No. of Path=     1000000
        MC estimate=     0.018145
        Standard Error   0.0003201135
>> 1f) No. of Path=     1000000
        MC estimate=     0.018166
        Standard Error   0.0000214364
>> 1g) No. of Path=     1000000
        MC estimate=     0.018115
        Standard Error   0.0000214249
>> 1h) No. of Path=     1000000
        MC estimate=     0.018177
        Standard Error   0.0000214507
```

Code for 1c)

```
% Vanilla Monte Carlo Simulation
clear all
format long
r=0.02;
S0=100;
sig=0.2;
K1=60;
K2=50;
T=1;

N=1000000;
E=randn(1,N);

ST=S0*exp((r-0.5*sig^2)*T + sig*sqrt(T).*E);
optionprice= exp(-r*T)*(max(K1-ST,0)-max(K2-ST,0));
MC=cumsum(optionprice)./[1:N];
se=sqrt( sum ((optionprice-MC(N)).^2) / ((N-1)*N));

fprintf('1c) No. of Path=     %.0f\n',N)
fprintf('MC estimate=     %f\n',MC(N))
fprintf('Standard Error   %f',se)


plot(MC(1:50000))
```

```
title('1c) Convergence Diagram')
xlabel('No. of Path')
ylabel('Option price')
```

Code for 1d)

```
% Monte Carlo Simulation
% antithetic variable technique
clear all
format long
r=0.02;
S0=100;
sig=0.2;
K1=60;
K2=50;
T=1;

N=1000000;
E1=randn(1,N/2);
E=[E1,-E1];

ST=S0*exp((r-0.5*sig^2)*T + sig*sqrt(T).*E);
optionprice= exp(-r*T)*(max(K1-ST,0)-max(K2-ST,0));
MC=cumsum(optionprice)./[1:N];
se=sqrt( sum ((optionprice-MC(N)).^2) / ((N-1)*N));

fprintf('1d) No. of Path=     %.0f\n',N)
fprintf('MC estimate=     %f\n',MC(N))
fprintf('Standard Error   %f',se)


plot(MC(1:50000))
title('1d) Convergence Diagram')
xlabel('No. of Path')
ylabel('Option price')
```

Code 1e)

```
% Monte Carlo Simulation
%Moment Matching
clear all
format long
r=0.02;
S0=100;
sig=0.2;
K1=60;
K2=50;
T=1;
```

```
N=1000000;
E1=randn(1,N);
mu=sum(E1)/N;
std=sqrt(sum((E1-mu).^2)/(N-1));
E=(E1-mu)/std;

ST=S0*exp((r-0.5*sig^2)*T + sig*sqrt(T).*E);
optionprice= exp(-r*T)*(max(K1-ST,0)-max(K2-ST,0));
MC=cumsum(optionprice)./[1:N];
se=sqrt( sum ((optionprice-MC(N)).^2) / ((N-1)*N));

fprintf('1e) No. of Path=     %.0f\n',N)
fprintf('MC estimate=     %f\n',MC(N))
fprintf('Standard Error   %.10f',se)


plot(MC(1:50000))
title('1e) Convergence Diagram')
xlabel('No. of Path')
ylabel('Option price')
```

Code 1f)

```
% Vanilla Monte Carlo Simulation
%important sampling technique
clear all
format long
r=0.02;
S0=100;
sig=0.2;
K1=60;
K2=50;
T=1;
alpha=0.1-5*log(55/100);

N=1000000;
n=randn(1,N);

ST=S0*exp((r-0.5*sig^2)*T + sig*sqrt(T).*(n-alpha));
optionprice= exp(-r*T)*(max(K1-ST,0)-max(K2-ST,0)).*exp(alpha*n-
0.5*alpha^2);
MC=cumsum(optionprice)./[1:N];
se=sqrt( sum ((optionprice-MC(N)).^2) / ((N-1)*N));

fprintf('1f) No. of Path=     %.0f\n',N)
fprintf('MC estimate=     %f\n',MC(N))
```

```
fprintf('Standard Error   %.10f',se)

plot(MC(1:50000))
title('1f) Convergence Diagram')
xlabel('No. of Path')
ylabel('Option price')
```

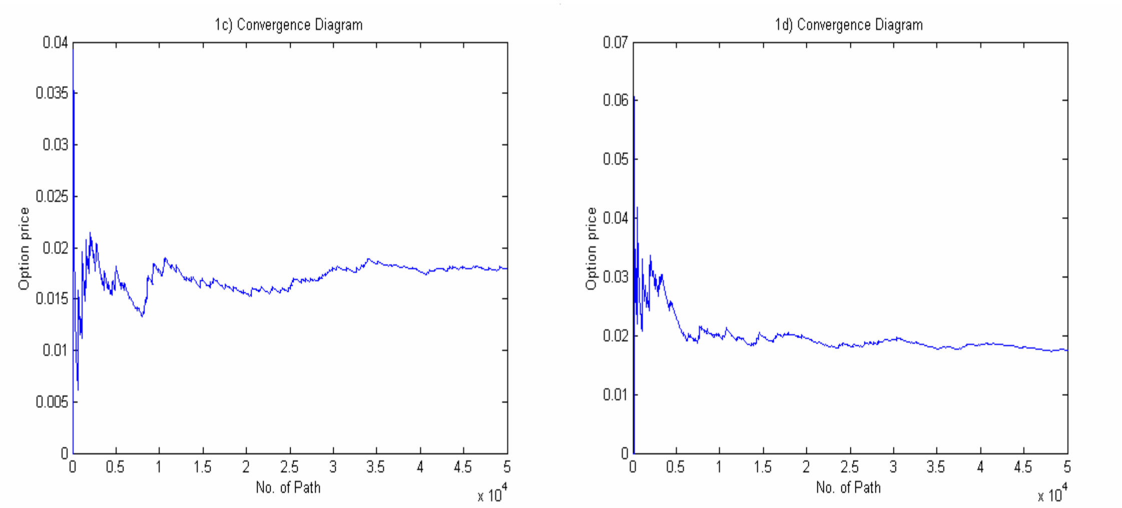
Code 1g)

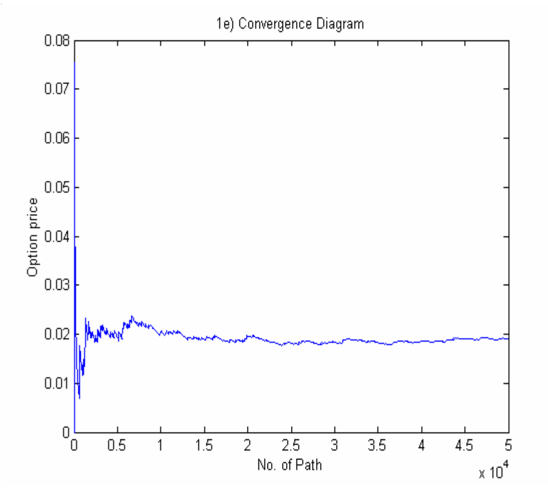
```
% Monte Carlo Simulation
% Important Sampling with antithetic variable technique
clear all
format long
r=0.02;
S0=100;
sig=0.2;
K1=60;
K2=50;
T=1;
alpha=0.1-5*log(55/100);

N=1000000;
E1=randn(1,N/2);
n=[E1,-E1];

ST=S0*exp((r-0.5*sig^2)*T + sig*sqrt(T).*(n-alpha));
optionprice= exp(-r*T)*(max(K1-ST,0)-max(K2-ST,0)).*exp(alpha*n-
0.5*alpha^2);
MC=cumsum(optionprice)./[1:N];
se=sqrt( sum ((optionprice-MC(N)).^2) / ((N-1)*N));

fprintf('1g) No. of Path=     %.0f\n',N)
fprintf('MC estimate=     %f\n',MC(N))
fprintf('Standard Error   %.10f',se)


plot(MC(1:50000))
title('1g) Convergence Diagram')
xlabel('No. of Path')
ylabel('Option price')
```

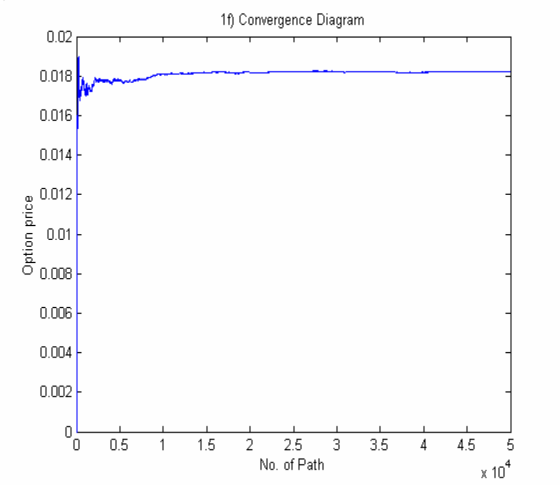
Code 1h)

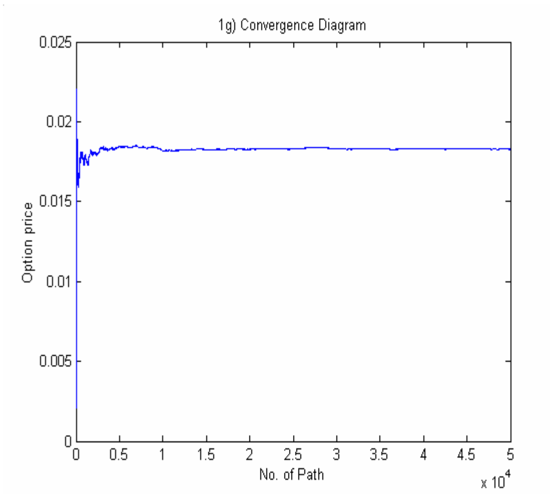
```
% Vanilla Monte Carlo Simulation
clear all
format long
r=0.02;
S0=100;
```

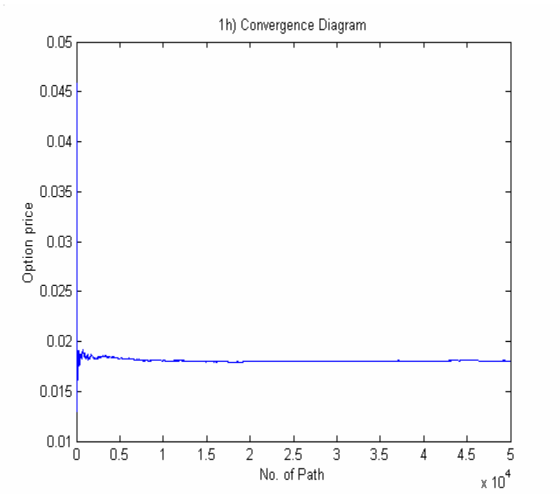
```
sig=0.2;
K1=60;
K2=50;
T=1;
alpha=0.1-5*log(55/100);

N=1000000;
E1=randn(1,N);
mu=sum(E1)/N;
std=sqrt(sum((E1-mu).^2)/(N-1));
n=(E1-mu)/std;

ST=S0*exp((r-0.5*sig^2)*T + sig*sqrt(T).*(n-alpha));
optionprice= exp(-r*T)*(max(K1-ST,0)-max(K2-ST,0)).*exp(alpha*n-
0.5*alpha^2);
MC=cumsum(optionprice)./[1:N];
se=sqrt( sum ((optionprice-MC(N)).^2) / ((N-1)*N));

fprintf('1h) No. of Path=     %.0f\n',N)
fprintf('MC estimate=     %f\n',MC(N))
fprintf('Standard Error   %.10f',se)


plot(MC(1:50000))
title('1h) Convergence Diagram')
xlabel('No. of Path')
ylabel('Option price')
```











2a) We can calculate Cholesky factorization by using Matlab function Chol(rho). From Matlab code, we get

$$A=\begin{pmatrix} 1 & 0 & 0 \\ -0.7 & 0.7141 & 0 \\ 0.1 & -0.8822 & 0.4602 \end{pmatrix}$$

and $\begin{bmatrix} W_{1,t} \\ . \\ . \\ W_{n,t} \end{bmatrix} = A \begin{bmatrix} Z_{1,t} \\ . \\ . \\ Z_{n,t} \end{bmatrix}$, where $Z_{1,t},\ldots,Z_{n,t}$ are independent BMs.

Then we can write stock process as

$$\frac{dS_{1,t}}{S_{1,t}} = rdt + (\sigma_1 a_{11})dZ_{1,t} + 0dZ_{2,t} + 0dZ_{3,t}$$

$$\frac{dS_{2,t}}{S_{2,t}} = rdt + (\sigma_2 a_{21})dZ_{1,t} + (\sigma_2 a_{22})dZ_{2,t} + 0dZ_{3,t}$$

$$\frac{dS_{3,t}}{S_{3,t}} = rdt + (\sigma_3 a_{31})dZ_{1,t} + (\sigma_3 a_{32})dZ_{2,t} + (\sigma_3 a_{33})dZ_{3,t}$$

This is as the form $\frac{dS_{i,t}}{S_{i,t}} = rdt + \sum_{j=1}^{3} \tilde{\sigma}_{ij} dZ_{j,t}, i = 1,2,3$

We get $\tilde{\sigma} = \begin{bmatrix} 0.2 & 0 & 0 \\ -0.1540 & 0.1571 & 0 \\ 0.0240 & -0.2117 & 0.1104 \end{bmatrix}$

2b) From 2a), we obtain terminal Stock price as

$$\begin{cases} S_{1,T} = S_{1,0} e^{(r-\frac{\tilde{\sigma}_{11}^2}{2})T + \tilde{\sigma}_{11}\sqrt{T}\xi_1} \\ S_{2,T} = S_{2,0} e^{(r-\frac{\tilde{\sigma}_{21}^2}{2}-\frac{\tilde{\sigma}_{22}^2}{2})T + \tilde{\sigma}_{21}\sqrt{T}\xi_1 + \tilde{\sigma}_{22}\sqrt{T}\xi_2} \\ S_{3,T} = S_{3,0} e^{(r-\frac{\tilde{\sigma}_{31}^2}{2}-\frac{\tilde{\sigma}_{32}^2}{2}-\frac{\tilde{\sigma}_{33}^2}{2})T + \tilde{\sigma}_{31}\sqrt{T}\xi_1 + \tilde{\sigma}_{32}\sqrt{T}\xi_2 + \tilde{\sigma}_{33}\sqrt{T}\xi_3} \end{cases}$$

$$\Rightarrow S_{i,T} = S_{i,0} e^{(r-\frac{1}{2}\sum_{j=1}^{3}\tilde{\sigma}_{ij}^2)T + \sqrt{T}\sum_{j=1}^{3}\tilde{\sigma}_{ij}\xi_j} \quad \text{i=1,2,3}$$

, where $\xi_1, \xi_2, \xi_3 \sim N(0,1)$, i.i.d and independent of each other

2c)

i) Denote $Y_T = (S_{1,T} S_{2,T} S_{3,T})^{1/3}$

$$Y_T = (S_{1,0} S_{2,0} S_{3,0})^{1/3} \exp\left\{ \begin{aligned} &(r - \frac{1}{6}(\tilde{\sigma}_{11}^2 + \tilde{\sigma}_{21}^2 + \tilde{\sigma}_{22}^2 + \tilde{\sigma}_{31}^2 + \tilde{\sigma}_{32}^2 + \tilde{\sigma}_{33}^2))T \\ &+ \frac{1}{3}\sqrt{T}\left((\tilde{\sigma}_{11} + \tilde{\sigma}_{21} + \tilde{\sigma}_{31})\xi_1 + (\tilde{\sigma}_{22} + \tilde{\sigma}_{23})\xi_2 + \tilde{\sigma}_{33}\xi_3\right) \end{aligned} \right\}$$

or

$$Y_T = (S_{1,0} S_{2,0} S_{3,0})^{1/3} \exp\left\{ (r - \frac{1}{6}\sum_{j=1}^{3}\sum_{i=1}^{3}\tilde{\sigma}_{ij}^2)T + \frac{1}{3}\sqrt{T}\left(\sum_{j=1}^{3}\sum_{i=1}^{3}\tilde{\sigma}_{ij}\xi_j\right) \right\}$$

ii) Denote $\xi \sim N(0,1)$ i.i.d. and independent of ξ_1, ξ_2, ξ_3

$$Y_T = (S_{1,0}S_{2,0}S_{3,0})^{1/3}\exp\left\{\begin{matrix}(r-\frac{1}{6}\sum_{j=1}^{3}\sum_{i=1}^{3}\tilde{\sigma}_{ij}^2)T \\ +\frac{1}{3}\sqrt{T}\left(\sqrt{(\tilde{\sigma}_{11}+\tilde{\sigma}_{21}+\tilde{\sigma}_{31})^2+(\tilde{\sigma}_{22}+\tilde{\sigma}_{23})^2+\tilde{\sigma}_{33}{}^2}\right)\xi\end{matrix}\right\}$$

or

$$Y_T = (S_{1,0}S_{2,0}S_{3,0})^{1/3}\exp\left\{(r-\frac{1}{6}\sum_{j=1}^{3}\sum_{i=1}^{3}\tilde{\sigma}_{ij}^2)T+\frac{1}{3}\sqrt{T}\sqrt{\sum_{i=1}^{3}\left(\sum_{j=1}^{3}\tilde{\sigma}_{ij}\right)^2}\xi\right\}$$

2d) Denote $\sigma' = \sqrt{\frac{1}{9}\sum_{i=1}^{3}\left(\sum_{j=1}^{3}\tilde{\sigma}_{ij}\right)^2}$ and $\Psi = \frac{1}{6}\sum_{j=1}^{3}\sum_{i=1}^{3}\tilde{\sigma}_{ij}^2$

Then, we can rewrite above equation as

$$Y_T = (S_{1,0}S_{2,0}S_{3,0})^{1/3}\exp\left((\frac{\sigma'^2}{2}-\Psi)T\right)\exp\left\{(r-\frac{1}{2}\sigma'^2)T+\sqrt{T}\sigma'\xi\right\}$$

If we denote $S_0' = (S_{1,0}S_{2,0}S_{3,0})^{1/3}\exp\left((\frac{\sigma'^2}{2}-\Psi)T\right)$, then we get

$$S_T' = Y_T = S_0'\exp\left\{(r-\frac{1}{2}\sigma'^2)T+\sqrt{T}\sigma'\xi\right\}$$

Thus, this process is like typical stock process under risk-neutral measure with S_0'=38.6609 and σ' = 0.0472 (See code for calculation)

Code 2a)-d)

```
rho=[1.0 -0.7 0.1;-0.7 1.0 -0.7;0.1 -0.7 1.0];
A=chol(rho)'
sigma=[0.2 0.22 0.24];
S0=[42 40 35];
T=0.25
sigma_hat=diag(sigma)*A

new_sigma=sqrt(sum(sum(sigma_hat).^2))/3
new_S = (S0(1)*S0(2)*S0(3))^(1/3)*exp((-
sum(sum(sigma_hat.^2))/6+0.5*new_sigma^2)*T)
```

3a) This option’s payoff = C(K=40)-C(K=60) where underlying asset is X(t)

- Generate 3N path ~N(0,1)’s of $\xi_1^{(1)},..,\xi_1^{(N)}$, $\xi_2^{(1)},..,\xi_2^{(N)}$ and $\xi_3^{(1)},..,\xi_3^{(N)}$

- Use equations $S_{1,T} = S_{1,0}e^{(r-\frac{\tilde{\sigma}_{11}^2}{2})T+\tilde{\sigma}_{11}\sqrt{T}\xi_1}$

$$S_{2,T} = S_{2,0}e^{(r-\frac{\tilde{\sigma}_{21}^2}{2}-\frac{\tilde{\sigma}_{22}^2}{2})T+\tilde{\sigma}_{21}\sqrt{T}\xi_1+\tilde{\sigma}_{22}\sqrt{T}\xi_2}$$

$$S_{3,T} = S_{3,0}e^{(r-\frac{\tilde{\sigma}_{31}^2}{2}-\frac{\tilde{\sigma}_{32}^2}{2}-\frac{\tilde{\sigma}_{33}^2}{2})T+\tilde{\sigma}_{31}\sqrt{T}\xi_1+\tilde{\sigma}_{32}\sqrt{T}\xi_2+\tilde{\sigma}_{33}\sqrt{T}\xi_3}$$

We get $S^{(n)}{}_{i,T}$ for i=1,2.3 where $1 \le n \le N$

- Compute corresponding $X^{(n)}_T$ and then we get $V^{(n)}$
- Compute Monte Carlo Estimate $\hat{V}_N = \frac{1}{N}\sum_{i=1}^{N} V^{(i)}$

See code below

3b) - Generate 3N path ~N(0,1)'s of $\xi_1^{(1)},..,\xi_1^{(N)}$, $\xi_2^{(1)},..,\xi_2^{(N)}$ and $\xi_3^{(1)},..,\xi_3^{(N)}$
- Use path as follow

$$\xi_1^{(1)},-\xi_1^{(1)},\xi_1^{(2)},-\xi_1^{(2)},...,\xi_1^{(N)},\xi_1^{(N)}$$

$$\xi_2^{(1)},-\xi_2^{(1)},\xi_2^{(2)},-\xi_2^{(2)},...,\xi_2^{(N)},\xi_2^{(N)}$$

$$\xi_2^{(1)},-\xi_2^{(1)},\xi_2^{(2)},-\xi_2^{(2)},...,\xi_2^{(N)},\xi_2^{(N)}$$

- Compute $X^{(n)}_T$ and $V^{(n)}$ as in a) using 2N of each stock
- Compute MC estimates as in a)

3c) This bull spread is consisted of long call option with strike price = 40 and short with strike price = 60. Thus, option price is

Option value = C_{BS}(K1=40)-C_{BS}(K2=60)

Where S_0=38.6609, σ =0.0472 , T=1/4 and r= 2% as in Q2 d)

Plug in Black-Scholes formula of call, we get

Option price = 0.048845

See code for calculation

3d) From property of stock price, $S_{1,T}, S_{2,T}, S_{3,T} \ge 0$. Consider geometric mean and arithmetic mean of more than zero variables, arithmetic mean always more than or equal geometric mean. $\frac{1}{3}(S_{1,T}S_{2,T}S_{3,T}) \ge (S_{1,T}S_{2,T}S_{3,T})^{1/3}$

Then, $E^Q[\Phi(X_T)] > E^Q[\Phi(Y_T)]$ and C<V.

3e) - Generate 3N path ~N(0,1)'s of $\xi_1^{(1)},..,\xi_1^{(N)}$, $\xi_2^{(1)},..,\xi_2^{(N)}$ and $\xi_3^{(1)},..,\xi_3^{(N)}$
- Compute $V^{(n)}$ as in a) and for $C^{(n)}$ use the equations in 2c) i) using the same path
- Compute exact value of control variate or C
- Compute C_N and V_N as before
- MC estimate of V = V_N+C-C_N

3f) See code below

3g)
```
>> 3a) No. of path: 1000000
       MC estimate : 0.103520
       Standard Error : 0.000327
>> 3b) No. of path: 1000000
       MC estimate(Antithetic variables) : 0.103544
        Standard Error : 0.000327
>> 3c) BS price of bull spread on Geometric avg.: 0.048845
>> 3e) No. of path: 1000000
        MC estimate(Control Variate) : 0.103723
        Standard Error : 0.000174
>> 3f) No. of path: 1000000
        MC estimate(Control Variate+Antithetic Variables) : 0.103871
        Standard Error : 0.000174
```

Code 3a)

```
clear
format long
r=0.02;
T=1/4;
K1=40;
K2=60;

rho=[1.0 -0.7 0.1;-0.7 1.0 -0.7;0.1 -0.7 1.0];
A=chol(rho)';
sigma=[0.2 0.22 0.24];
S0=[42 40 35]';

sigma_hat=diag(sigma)*A;
new_sigma=sqrt(sum(sum(sigma_hat).^2))/3;

N=1000000;
z=randn(3,N);

b=sum((sigma_hat)'.^2);

ST(1,:)=S0(1)*exp((r-0.5*b(1))*T + sqrt(T)*sigma_hat(1,:)*z);
ST(2,:)=S0(2)*exp((r-0.5*b(2))*T + sqrt(T)*sigma_hat(2,:)*z);
ST(3,:)=S0(3)*exp((r-0.5*b(3))*T + sqrt(T)*sigma_hat(3,:)*z);
```

```
XT=sum(ST)/3;

V=exp(-r*T)*max(XT-K1,0)-max(XT-K2,0);
MCV=(cumsum(V)./[1:N]);
se=sqrt(sum((V-MCV(N)).^2)/(N*(N-1)));
fprintf('3a) No. of path: %.0f\n' , N);
fprintf(' MC estimate : %f\n' , MCV(N));
fprintf(' Standard Error : %f\n' , se);
```

Code 3b)

```
clear
format long
r=0.02;
T=0.25;
K1=40;
K2=60;

rho=[1.0 -0.7 0.1;-0.7 1.0 -0.7;0.1 -0.7 1.0];
A=chol(rho)';
sigma=[0.2 0.22 0.24];
S0=[42 40 35]';

sigma_hat=diag(sigma)*A;

new_sigma=sqrt(sum(sum(sigma_hat).^2))/3;

N=1000000;

%Antithetic varible
z1=randn(3,N/2);
z=[z1,-z1];

b=sum(sigma_hat'.^2);

ST(1,:)=S0(1)*exp((r-0.5*b(1))*T + sqrt(T)*sigma_hat(1,:)*z);
ST(2,:)=S0(2)*exp((r-0.5*b(2))*T + sqrt(T)*sigma_hat(2,:)*z);
ST(3,:)=S0(3)*exp((r-0.5*b(3))*T + sqrt(T)*sigma_hat(3,:)*z);


XT=mean(ST);

V=exp(-r*T)*max(XT-K1,0)-max(XT-K2,0);
MCV=(cumsum(V)./[1:N]);
se=sqrt(sum((V-MCV(N)).^2)/(N*(N-1)));
fprintf('3b) No. of path: %.0f\n' , N);
fprintf(' MC estimate(Antithetic variables) : %f\n' , MCV(N));
```

```
fprintf(' Standard Error : %f\n' , se);
```

Code 3c)

```
rho=[1.0 -0.7 0.1;-0.7 1.0 -0.7;0.1 -0.7 1.0];
A=chol(rho)';
sigma=[0.2 0.22 0.24];
S0=[42 40 35];
T=0.25;
sigma_hat=diag(sigma)*A;

new_sigma=sqrt(sum(sum(sigma_hat).^2))/3;
new_S = (S0(1)*S0(2)*S0(3))^(1/3)*exp((-
sum(sum(sigma_hat.^2))/6+0.5*new_sigma^2)*T);
exact=CBS(new_sigma,K1,new_S,T,r)-CBS(new_sigma,K2,new_S,T,r);

fprintf('3c) BS price of bull spread on Geometric avg.: %f\n',exact)
```

Code 3e)

```
clear
format long
T=1;
r=0.02;
T=0.25;
K1=40;
K2=60;

rho=[1.0 -0.7 0.1;-0.7 1.0 -0.7;0.1 -0.7 1.0];
A=chol(rho)';
sigma=[0.2 0.22 0.24];
S0=[42 40 35]';

sigma_hat=diag(sigma)*A;

new_sigma=sqrt(sum(sum(sigma_hat).^2))/3;

N=1000000;
z=randn(3,N);

b=sum(sigma_hat'.^2);

%Arithmetic mean
ST(1,:)=S0(1)*exp((r-0.5*b(1))*T + sqrt(T)*sigma_hat(1,:)*z);
ST(2,:)=S0(2)*exp((r-0.5*b(2))*T + sqrt(T)*sigma_hat(2,:)*z);
ST(3,:)=S0(3)*exp((r-0.5*b(3))*T + sqrt(T)*sigma_hat(3,:)*z);
XT=mean(ST);
V1=exp(-r*T)*(max(XT-K1,0)-max(XT-K2,0));
```

```
%Geomatric mean
YT(1,:)= geomean(S0)*exp((r-
sum(sum(sigma_hat.^2))/6)*T+1/3*sqrt(T)*(sum(sigma_hat)*z));
V2=exp(-r*T)*max(YT-K1,0)-max(YT-K2,0);

%Exact price for option on Yt
new_sigma=sqrt(sum(sum(sigma_hat).^2))/3;
new_S = (S0(1)*S0(2)*S0(3))^(1/3)*exp((-
sum(sum(sigma_hat.^2))/6+0.5*new_sigma^2)*T);
exact=CBS(new_sigma,K1,new_S,T,r)-CBS(new_sigma,K2,new_S,T,r);

%MC estimate
V=V1+(exact-V2);
MCV=cumsum(V)./[1:N];
se=sqrt(sum((V-MCV(N)).^2)/(N*(N-1)));
fprintf('3e) No. of path: %.0f\n' , N);
fprintf(' MC estimate(Control Variate) : %f\n' , MCV(N));
fprintf(' Standard Error : %f\n' , se);
```

Code 3f)

```
clear
format long
T=1;
r=0.02;
T=0.25;
K1=40;
K2=60;

rho=[1.0 -0.7 0.1;-0.7 1.0 -0.7;0.1 -0.7 1.0];
A=chol(rho)';
sigma=[0.2 0.22 0.24];
S0=[42 40 35]';

sigma_hat=diag(sigma)*A;

new_sigma=sqrt(sum(sum(sigma_hat).^2))/3;

N=1000000;
%Antithetic Variable
z1=randn(3,N/2);
z=[z1,-z1];

b=sum(sigma_hat'.^2);

%Arithmetic mean
ST(1,:)=S0(1)*exp((r-0.5*b(1))*T + sqrt(T)*sigma_hat(1,:)*z);
```

```
ST(2,:)=S0(2)*exp((r-0.5*b(2))*T + sqrt(T)*sigma_hat(2,:)*z);
ST(3,:)=S0(3)*exp((r-0.5*b(3))*T + sqrt(T)*sigma_hat(3,:)*z);
XT=mean(ST);
V1=exp(-r*T)*(max(XT-K1,0)-max(XT-K2,0));

%Geomatric mean
YT(1,:)= geomean(S0)*exp((r-
sum(sum(sigma_hat.^2))/6)*T+1/3*sqrt(T)*(sum(sigma_hat)*z));
V2=exp(-r*T)*max(YT-K1,0)-max(YT-K2,0);

%Exact price for option on Yt
new_sigma=sqrt(sum(sum(sigma_hat).^2))/3;
new_S = (S0(1)*S0(2)*S0(3))^(1/3)*exp((-
sum(sum(sigma_hat.^2))/6+0.5*new_sigma^2)*T);
exact=CBS(new_sigma,K1,new_S,T,r)-CBS(new_sigma,K2,new_S,T,r);

%MC estimate
V=V1+(exact-V2);
MCV=cumsum(V)./[1:N];
se=sqrt(sum((V-MCV(N)).^2)/(N*(N-1)));

fprintf('3f) No. of path: %.0f\n' , N);
fprintf(' MC estimate(Control Variate+Antithetic Variables) : %f\n' ,
MCV(N));
fprintf(' Standard Error : %f\n' , se);
```