Homework #4

Jingyi Guo
Pittsburgh Campus

Problem 1: Conditional Monte Carlo

a) Standard Monte Carlo

Code:

```
n=10000;
S0=100;
K=100;
r=0.05;
T=1;
v0square=0.04;
N=50;
alpha=0.1;
phi=0.1; % or 1.0
delta=T/N;

S=S0*ones(n,1);
vsquare=v0square*ones(n,1);

for i=1:N
    z=randn(n,2);
    S=S+r*delta*S+sqrt(delta)*sqrt(vsquare).*S.*z(:,1);
    vsquare=vsquare+alpha*delta.*vsquare+phi*sqrt(delta).*vsqu
are.*z(:,2);
end

C_a=exp(-r*T).*max(S-K,0);
Cbar_a=mean(C_a)
Cstderr_a=std(C_a)/sqrt(n)
```

Result:

	$(\alpha, \psi) = (0.10, 0.10)$	$(\alpha, \psi) = (0.10, 1.0)$
Estimation	10.6588	10.3911
Std. Error	0.1480	0.1557

b) Conditional Monte Carlo

Code:

```
vsquare2=zeros(N+1,n);
vsquare2(1,:)=v0square*ones(1,n);
```

```
for i=1:N
    z=randn(1,n);
    vsquare2(i+1,:)=vsquare2(i,:)+alpha*delta.*vsquare2(i,:)+p
hi*sqrt(delta).*vsquare2(i,:).*z;
end

vol=zeros(1,n);

for i=2:(N+1)
    vol=vol+vsquare2(i,:);
end
vol=sqrt(vol./N);

% Black-Scholes formula:
dplus=1./(vol.*sqrt(T)).*(log(S0/K)+(r+1/2.*vol.^2)*T);
dminus=1./(vol.*sqrt(T)).*(log(S0/K)+(r-1/2.*vol.^2)*T);
C_b=S0*normcdf(dplus)-normcdf(dminus)*K*exp(-r*T);
Cbar_b=mean(C_b)
Cstderr_b=std(C_b)/sqrt(n)
```

Result:

	$(\alpha, \psi) = (0.10, 0.10)$	$(\alpha, \psi) = (0.10, 1.0)$
Estimation	10.6469	10.2933
Std. Error	0.0023	0.0225

The estimation in b) is less than that in a), and the standard error of b) is less than that of a).

Problem 2: Interest rates derivatives and CIR

a) Pricing zero coupon bond

Code:

```
alpha=0.2;
sig=0.1;
b=0.05;
r0=0.04;
n=1000;
N=50;
T=1;
delta=T/N;

r=zeros(n,N+1);
r(:,1)=r0*ones(n,1);
sumr=zeros(n,1);
```

```
for i=2:(N+1)
    d=4*alpha*b/(sig^2);
    lam=4*alpha*exp(-alpha*delta)/(sig^2*(1-exp(-
alpha*delta))).*r(:,i-1);
    chi=random('ncx2',d,lam);
    r(:,i)=(sig^2*(1-exp(-alpha*delta)))/(4*alpha).*chi;
    sumr=sumr+r(:,i);
end

ZCB=exp(-delta.*sumr);
ZCB_bar=mean(ZCB)
ZCB_stderr=std(ZCB)/sqrt(n)
```

Result:

ZCB_bar =

0.9599

ZCB_stderr =

3.3057e-04

The estimated price of zero coupon bond is 0.9599, the standard error is 3.3057e-04.

b) Pricing caplet

Code:

```
t=1;
del=1/12;
L=1;
R=0.05;

payoff=L*del*max(0,r(:,N+1)-R);
caplet=exp(-delta.*sumr).*payoff;
caplet_bar=mean(caplet)
caplet_stderr=std(caplet)/sqrt(n)
```

Result:

caplet_bar =

3.4216e-04

caplet_stderr =

2.4002e-05

The estimated price of caplet is 3.4216e-04, the standard error is 2.4002e-05.

Problem 3: Replicating Broadie and Glasserman "Greeks" methodology

Code:

```
r=0.1;
K=100;
q=0.03;
sig=0.25;
T=0.2;
n=10000;
h=0.0001;
S0=90;
%European call with dividend

%% resimulation
% without control
z=randn(n,1);
S=S0*exp((r-q-1/2*sig^2)*T+sig*sqrt(T).*z);
C=exp(-r*T)*max(S-K,0);
Sh=(S0+h)*exp((r-q-1/2*sig^2)*T+sig*sqrt(T).*z);
Ch=exp(-r*T)*max(Sh-K,0);
delta=(Ch-C)./h;
delta_bar=mean(delta)
delta_stderr=std(delta)/sqrt(n)

Sv=S0*exp((r-q-1/2*(sig+h)^2)*T+(sig+h)*sqrt(T).*z);
Cv=exp(-r*T)*max(Sv-K,0);
vega=(Cv-C)./h;
delta_bar=mean(vega)
delta_stderr=std(vega)/sqrt(n)

% with control
Yd=delta;
X=S;
EX=exp((r-q)*T)*S0*ones(n,1);
deltac=Yd+(-corr(Yd,X)*std(Yd)/std(X))*(mean(X)-EX);
deltac_bar=mean(deltac)
deltac_stderr=std(deltac)/sqrt(n)*sqrt(1-(corr(Yd,X))^2)

Yv=vega;
vegac=Yv+(-corr(Yv,X)*std(Yv)/std(X))*(mean(X)-EX);
vegac_bar=mean(vegac)
vegac_stderr=std(vegac)/sqrt(n)*sqrt(1-(corr(Yv,X))^2)
```

```
%% pathwise
z=randn(n,1);
S=S0*exp((r-q-1/2*sig^2)*T+sig*sqrt(T).*z);

% without control
delta1=zeros(n,1);
vega1=zeros(n,1);
for i=1:n
    if (S(i)>=K)
        delta1(i)=exp(-r*T)*S(i)/S0;
        vega1(i)=exp(-r*T)*S(i)/sig*(log(S(i)/S0)-(r-
q+1/2*sig^2)*T);
    end
end

delta1_bar=mean(delta1)
delta1_stderr=std(delta1)/sqrt(n)

vega1_bar=mean(vega1)
vega1_stderr=std(vega1)/sqrt(n)

% with control
Y1d=delta1;
X=S;
EX=exp((r-q)*T)*S0*ones(n,1);
add=-corr(Y1d,X)*std(Y1d)/std(X);
delta1c=Y1d+add*(mean(X)-EX);
delta1c_bar=mean(delta1c)
delta1c_stderr=std(delta1c)/sqrt(n)*sqrt(1-(corr(Y1d,X))^2)

Y1v=vega1;
adv=-corr(Y1v,X)*std(Y1v)/std(X);
vega1c=Y1v+adv*(mean(X)-EX);
vega1c_bar=mean(vega1c)
vega1c_stderr=std(vega1c)/sqrt(n)*sqrt(1-(corr(Y1v,X))^2)

%% loglikelihood
% without control
z=randn(n,1);
S=S0*exp((r-q-1/2*sig^2)*T+sig*sqrt(T).*z);
delta2=exp(-r*T)*max(S-K,0)*1/(S0*sig^2*T).*(log(S./S0)-(r-q-
0.5*sig^2)*T);
```

```
delta2_bar=mean(delta2)
delta2_stderr=std(delta2)/sqrt(n)

d=(log(S./S0)-(r-q-sig^2/2)*T)/(sig*sqrt(T));
dd_over_dsig=(log(S0./S)+(r-q+sig^2)*T)/(sig^2*sqrt(T));
dlogg_over_dsig=-d.*dd_over_dsig-1/sig;
vega2=exp(-r*T).*max(S-K,0).*dlogg_over_dsig;
vega2_bar=mean(vega2)
vega2_stderr=std(vega2)/sqrt(n)

% with control
Y2d=delta2;
X=S;
EX=exp((r-q)*T)*S0*ones(n,1);
ad2=-corr(Y2d,X)*std(Y2d)/std(X);
delta2c=Y2d+ad2*(mean(X)-EX);
delta2c_bar=mean(delta2c)
delta2c_stderr=std(delta2c)/sqrt(n)*sqrt(1-(corr(Y2d,X))^2)

Y2v=vega2;
adv2=-corr(Y2v,X)*std(Y2v)/std(X);
vega2c=Y2v+adv*(mean(X)-EX);
vega2c_bar=mean(vega2c)
vega2c_stderr=std(vega2c)/sqrt(n)*sqrt(1-(corr(Y2v,X))^2)
```

Result:

	Delta Est.	Delta Std. Err	Vega Est.	Vega Std. Err
Resimulation	0.2145	0.0045	11.5976	0.2697
Resimulation with control	0.2179	0.0031	11.8078	0.1772
Pathwise	0.2247	0.0046	12.2323	0.2773
Pathwise with control	0.2263	0.0031	12.3347	0.1787
Likelihood	0.2136	0.0075	10.5798	0.5827
Likelihood with control	0.2142	0.0059	10.6044	0.5171

Problem 4: Greeks of Digital Options

a) In Problem3, Assignment 3, we have deduced the pricing formula for digital option as follows:

$$P(S_0, K, r, \sigma, T) = e^{-rT} N(\frac{1}{\sigma\sqrt{T}}(log\left(\frac{S_0}{K}\right) + (r - \frac{1}{2}\sigma^2)T))$$

So delta is:

$$Delta = \frac{\partial}{\partial S_0} P(S_0, K, r, \sigma, T) = e^{-rT} N'\left(\frac{1}{\sigma\sqrt{T}}\left(log\left(\frac{S_0}{K}\right) + \left(r - \frac{1}{2}\sigma^2\right)T\right)\right) \cdot \frac{1}{\sigma\sqrt{T}} \cdot \frac{1}{S_0}$$

$$= e^{-rT} \cdot pdf(d_-) \cdot \frac{1}{\sigma\sqrt{T}} \cdot \frac{1}{S_0}$$

where $d_- = \frac{1}{\sigma\sqrt{T}}\left(log\left(\frac{S_0}{K}\right) + \left(r - \frac{1}{2}\sigma^2\right)T\right)$

Code:

```
S0=95;
K=100;
r=0.05;
sig=0.2;
T=1;
n=10000;
h=0.0001;
d_minus=1/(sig*sqrt(T))*(log(S0/K)+(r-1/2*sig^2)*T);
delta_a=exp(-r*T)*normpdf(d_minus)*1/(sig*sqrt(T))*1/S0
```

Result:

```
delta_a =

    0.0199
```

So the delta of digital option is 0.0199.

b) Resimulation method

Code:

```
z=randn(n,1);
S=S0*exp((r-1/2*sig^2)*T+sig*sqrt(T).*z);
C=exp(-r*T)*max(S-K,0)./abs(S-K); % pays 1 or 0
Sh=(S0+h)*exp((r-1/2*sig^2)*T+sig*sqrt(T).*z);
Ch=exp(-r*T)*max(Sh-K,0)./abs(Sh-K);
delta=(Ch-C)./h;
delta_bar=mean(delta)
delta_stderr=std(delta)/sqrt(n)
```

Result:

```
delta_b_bar =
```

5.9952e-15

delta_b_stderr =

5.2333e-15

So the estimation of delta is 5.9952e-15 and its standard error is 5.2333e-15.

c) Likelihood method

Formula is: $Delta$: $e^{-rT} I_{\{S_T \geq K\}} \frac{1}{S_0 \sigma^2 T} \times (ln\left(\frac{S_T}{S_0}\right) - (r - \frac{1}{2}\sigma^2)T)$

Code:

```
z=randn(n,1);
S=S0*exp((r-1/2*sig^2)*T+sig*sqrt(T).*z);
delta_c=exp(-r*T)*max(S-K,0)./abs(S-
K)*1/(S0*sig^2*T).*(log(S./S0)-(r-0.5*sig^2)*T);
delta_c_bar=mean(delta_c)
delta_c_stderr=std(delta_c)/sqrt(n)
```

Result:

delta_c_bar =

0.0200

delta_c_stderr =

2.9544e-04

So the estimation of delta is 0.02 and its standard error is 2.9544e-04.