Math 623 : Hw # 3

1(a) $r = 0.01 = 1\%$

(b) $E^Q[S(t)] = S_0 e^{rT} \Rightarrow S_0 = 99.75$

(c) $E^Q[dS(t)/S(t) \mid S_t, Y_t] = 0.01dt$

$Var[dS_t / S_t \mid S_t, Y_t] = 0 + Y_t(\sqrt{(-0.7)^2 + 0.51})dt = Y_t dt$

(d) $E^Q[dY_t \mid S_t, Y_t] = 2(0.04 - Y_t)dt$

$Var[dY_t \mid S_t, Y_t] = (0.2 * \sqrt{Y_t})^2 dt = 0.04 Y_t dt$

(e) Consider equation $dS_t / S_t = 0.01dt + \sqrt{Y_t}(-0.7dW_t + \sqrt{0.51}dZ_t)$
since dW_t and dZ_t are independent so they can be rewrite as dN
where N~ NID(0,dt). Thus, this equation becomes
$dS_t / S_t = 0.01dt + \sqrt{Y_t} dN_t$, which is the typical GBM stock process.
That's why it is make sense to have Y_t as a square of volatility level at t.

(f) First, equation $dY_t = 2(0.04 - Y_t)dt + 0.2\sqrt{Y_t}dW_t$ is the mean reversion stochastic process, which process will be reverse to the level of Y_t =0.04 or volatility = 0.2. Second, this process start at Yt = 0.04, which means it is starting at the reversion level.

(g) $$\rho = \frac{Cov(dY_t, dS_t / S_t)}{\sqrt{Var(dY_t)Var(dS_t / S_t)}}$$

$$\begin{aligned} Cov(dY_t, dS_t / S_t) &= E[(dY_t - E[dY_t])(dS_t / S_t - E[S_t / S_t)] \\ &= E[(0.2\sqrt{Y_t}dW_t)(-0.7\sqrt{Y_t}dW_t + \sqrt{0.51 Y_t}dZ_t)] \\ &= -0.14 Y_t dt \end{aligned}$$

$$\rho = \frac{-0.14 Y_t dt}{\sqrt{0.04 Y_t dt}\sqrt{Y_t dt}} = -0.7$$

(h) Under recession, when stock price and return on stock fall down, stock volatility tends to be higher as confirmed by several empirical researches. That's why the relationship of stock return and volatility is negative sign.
This also can be explained by leverage issue. The drop in stock returns (negative return) lead to increase in leverage of firm, which make the stock riskier. Consequently, volatility increases.
Another explanation based on volatility feedback suggests that if volatility is priced, an anticipated increase in volatility raises the required return on equity, leading to an immediate stock price decline (negative return).

2 (a) $\Delta t = 0, \Delta t, 2\Delta t, ..., 30\Delta t = T$ where Δt =3/360, T=0.25

$$\begin{cases} S_{t+\Delta t} = S_t + 0.01 S_t \Delta t + \sqrt{Y_t} S_t (-0.7\sqrt{\Delta t}\xi_t + \sqrt{0.51\Delta t}\eta_t) \\ S_0 = 99.75 \end{cases}$$

where $\xi_t \sim NID(0,1)$, $\eta_t \sim NID(0,1)$ and independent of each other.

$$\begin{cases} Y_{t+\Delta t} = Y_t + 2(0.04 - Y_t)\Delta t + 0.2\sqrt{Y\Delta t_t}\xi_t \\ Y_0 = 0.04 \end{cases}$$

where $\xi_t \sim NID(0,1)$

(b), (c), (d), (e), (f). See code below

(e) 1. Generate MN independent N(0,1)'s ξ_t and MN independent N(0,1)'s of η_t

2. First path use $\xi_1{}^{(1)}{}_t$, $\xi_2{}^{(1)}{}_t$, $\xi_3{}^{(1)}{}_t$, …, $\xi_M{}^{(1)}{}_t$

$\eta_1{}^{(1)}{}_t$, $\eta_2{}^{(1)}{}_t$, $\eta_3{}^{(1)}{}_t$, …, $\eta_M{}^{(1)}{}_t$

Second path use -$\xi_1{}^{(1)}{}_t$, -$\xi_2{}^{(1)}{}_t$, -$\xi_3{}^{(1)}{}_t$, …, -$\xi_M{}^{(1)}{}_t$

-$\eta_1{}^{(1)}{}_t$, -$\eta_2{}^{(1)}{}_t$, -$\eta_3{}^{(1)}{}_t$, …, -$\eta_M{}^{(1)}{}_t$

And so on….

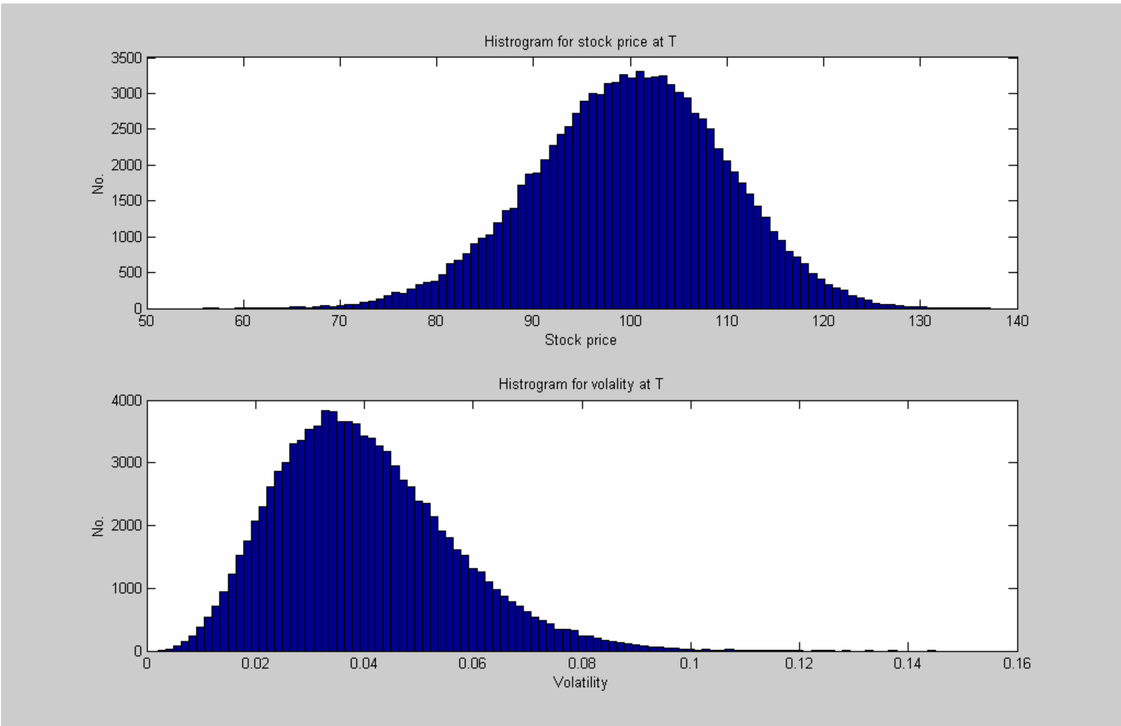
3. Use the 2N paths generated above as before.

This technique uses the symmetric of the normal distribution to reduce the variance and time saving.

<u>HW3 # 2b, 2c</u>

```
clear
format long

%set variable
dt=1/120;
T=0.25;
J=T/dt+1;

%N no. of path
N=100000;

%implement in Euler scheme

s=zeros(J,N);
y=zeros(J,N);

%set initial value

s(1,:)=99.75;
```

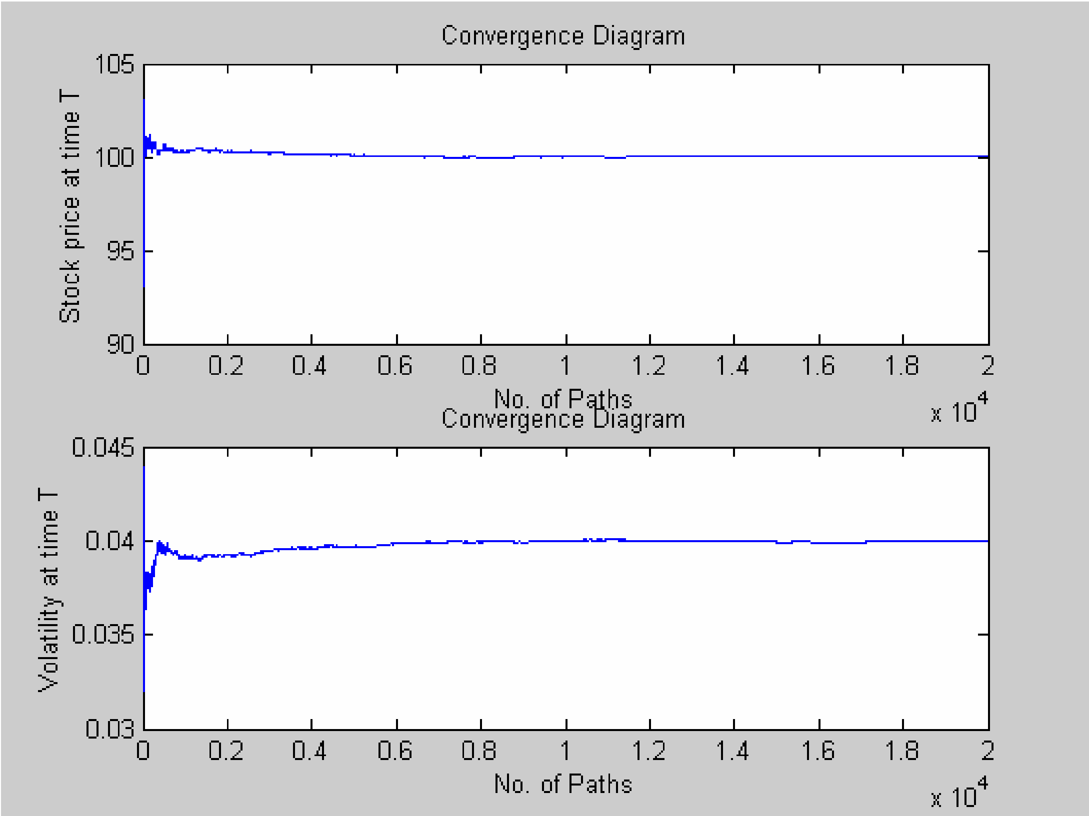
```
y(1,:)=0.04;

%generate N random variables of z and w
        z=randn(J-1,N);
        w=randn(J-1,N);
    for j=1:J-1

        y(j+1,:)= max(y(j,:),0) + 2*(0.04-max(y(j,:),0))*dt +
0.2*sqrt(max(y(j,:),0)*dt).*z(j,:);


        s(j+1,:)=max(s(j,:),0) + 0.01*max(s(j,:),0)*dt +
sqrt(max(y(j,:),0)).*max(s(j,:),0).*(-0.7*sqrt(dt).*z(j,:)...
                   +sqrt(0.51*dt).*w(j,:));
    end

stock_price_at_t_025=(s(J,:)');
mean(s(J,:))
volatility_at_t_025=(y(J,:)');

%x=50:10:150;
subplot(2,1,1)
hist(s(J,:)',10)
subplot(2,1,2)
hist(y(J,:)',10)
```



HW3 # 2d

```
clear
format long

%set variable
dt=1/120;
T=0.25;
J=T/dt+1;

%N no. of path
N=100000;

%implement in Euler scheme

s=zeros(J,N);
y=zeros(J,N);

%set initial value

s(1,:)=99.75;
y(1,:)=0.04;

%serror=zeros(N,1);
%verror=zeros(N,1);
%generate N random variables of z and w
        z=randn(J-1,N);
        w=randn(J-1,N);
    for j=1:J-1

        y(j+1,:)= max(y(j,:),0) + 2*(0.04-max(y(j,:),0))*dt + 
0.2*sqrt(max(y(j,:),0)*dt).*z(j,:);


        s(j+1,:)=max(s(j,:),0) + 0.01*max(s(j,:),0)*dt + 
sqrt(max(y(j,:),0)).*max(s(j,:),0).*(-0.7*sqrt(dt).*z(j,:)...
                   +sqrt(0.51*dt).*w(j,:));
    end
sT=s(J,:);
s_square=cumsum(sT.^2)./[1:N];
mean_s=cumsum(sT)./[1:N];
s_se=sqrt(1./[1:N]).*sqrt(s_square - mean_s.^2);
vT=y(J,:);
mean_v=cumsum(vT)./[1:N];
v_square=cumsum(vT.^2)./[1:N];
v_se=sqrt(1./[1:N]).*sqrt(v_square - mean_v.^2);
```

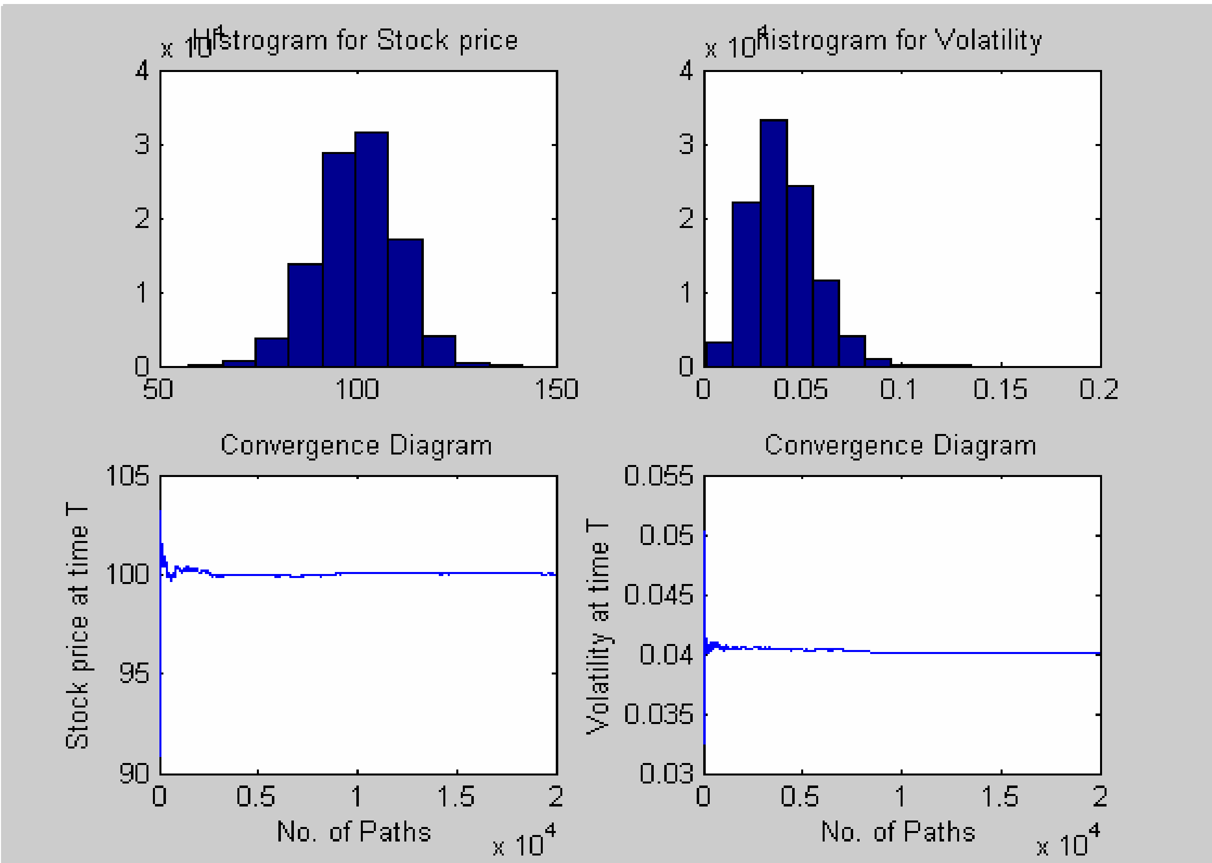
```
subplot(2,1,1)
plot(1:20000,mean_s(1:20000))
xlabel('No. of Paths')
ylabel('Stock price at time T')
Title('Convergence Diagram')

subplot(2,1,2)
plot(1:20000,mean_v(1:20000))
xlabel('No. of Paths')
ylabel('Volatility at time T')
Title('Convergence Diagram')
```



HW3 #2f

```
clear
format long

%set variable
dt=1/120;
T=0.25;
J=T/dt+1;

%N no. of path
N=100000;

%implement in Euler scheme

s=zeros(J,N);
y=zeros(J,N);

%set initial value

s(1,:)=99.75;
y(1,:)=0.04;

%generate N random variables of z and w
        z1=randn(J-1,N/2);
        z=[z1,-1*z1];
        w1=randn(J-1,N/2);
        w=[w1,-1*w1];

        for j=1:J-1

        y(j+1,:)= max(y(j,:),0) + 2*(0.04-max(y(j,:),0))*dt +
0.2*sqrt(max(y(j,:),0)*dt).*z(j,:);


        s(j+1,:)=max(s(j,:),0) + 0.01*max(s(j,:),0)*dt +
sqrt(max(y(j,:),0)).*max(s(j,:),0).*(-0.7*sqrt(dt).*z(j,:)...
                   +sqrt(0.51*dt).*w(j,:));
    end
sT=s(J,:);
s_square=cumsum(sT.^2)./[1:N];
mean_s=cumsum(sT)./[1:N];
s_se=sqrt(1./[1:N]).*sqrt(s_square - mean_s.^2);
vT=y(J,:);
mean_v=cumsum(vT)./[1:N];
v_square=cumsum(vT.^2)./[1:N];
```

```
v_se=sqrt(1./[1:N]).*sqrt(v_square - mean_v.^2);

subplot(2,2,1)
hist(s(J,:)',10);
title('Histrogram for Stock price')

subplot(2,2,2)
hist(y(J,:)',10);
title('histrogram for Volatility')

subplot(2,2,3)
plot(1:20000,mean_s(1:20000))
xlabel('No. of Paths')
ylabel('Stock price at time T')
Title('Convergence Diagram')

subplot(2,2,4)
plot(1:20000,mean_v(1:20000))
xlabel('No. of Paths')
ylabel('Volatility at time T')
Title('Convergence Diagram')
```



3(a) $C_{BS}(S_0=99.75, r=1\%, K=100, T=0.25, \sigma=0.2) = \$ 3.978$

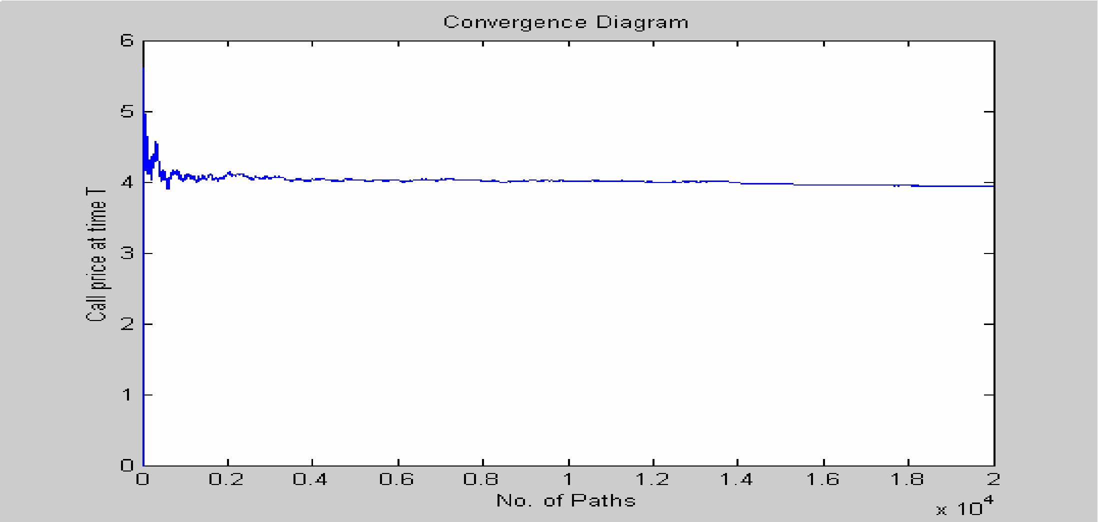
(b) Steps

1. Using the Euler scheme as in 2(a) and set of generated $S_{0.25}$ in 2(b) (suppose we use N paths)
2. At each $S_{T=0.25}$, calculate $C_{T=0.25} = (S_{T=0.25}-K)^+$
3. Discounted C_T to $C_{t=0}$ by equation $C_{t=0} = e^{-rT} C_{T=0.25}$
4. Calculate Monte Carlo estimate, $\overline{C} = \frac{1}{N}\sum C_{t=0}$
5. Standard Error = $\frac{1}{N}\sqrt{\frac{1}{N-1}\sum (C_{t=0}-\overline{C})}$

(c), (d). See code

```
HW3# 3b,3d
clear
format long

%set variable
dt=1/120;
T=0.25;
J=T/dt+1;
K=100;

%N no. of path
N=100000;

%implement in Euler scheme

s=zeros(J,N);
y=zeros(J,N);

%set initial value

s(1,:)=99.75;
y(1,:)=0.04;

%serror=zeros(N,1);
%verror=zeros(N,1);
%generate N random variables of z and w
      z=randn(J-1,N);

      w=randn(J-1,N);

   for j=1:J-1
```

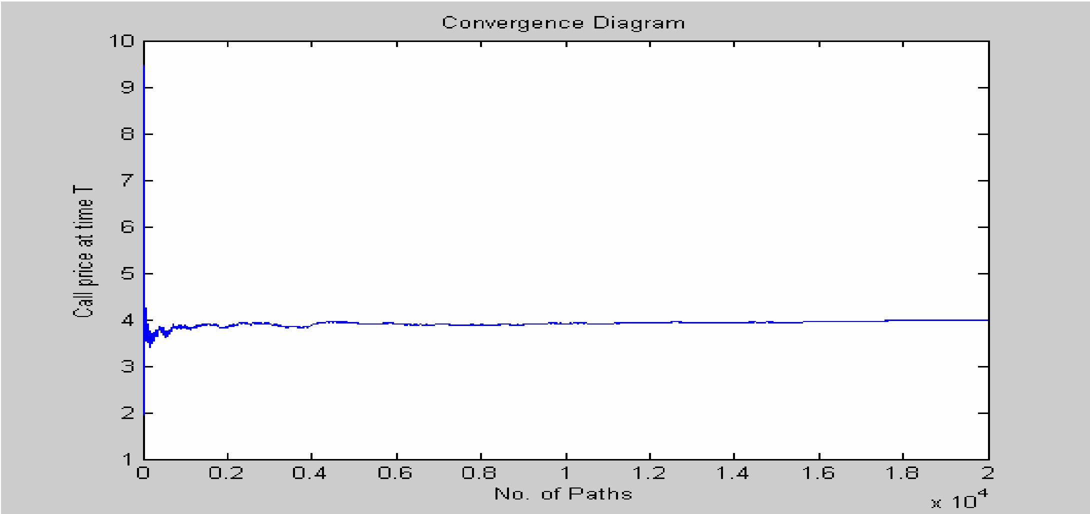
```
        y(j+1,:)= max(y(j,:),0) + 2*(0.04-max(y(j,:),0))*dt +
0.2*sqrt(max(y(j,:),0)*dt).*z(j,:);


        s(j+1,:)=max(s(j,:),0) + 0.01*max(s(j,:),0)*dt +
sqrt(max(y(j,:),0)).*max(s(j,:),0).*(-0.7*sqrt(dt).*z(j,:)...
                   +sqrt(0.51*dt).*w(j,:));
    end

sT=s(J,:);
call=max(sT-K,0);
mean_c=cumsum(call)./[1:N];
c_square=cumsum(call.^2)./[1:N];
c_se=sqrt(1./[1:N]).*sqrt(c_square - mean_c.^2);

plot(1:N,mean_c)
xlabel('No. of Paths')
ylabel('Call price at time T')
Title('Convergence Diagram')
```



HW3#3c,3d

```
clear
format long

%set variable
dt=1/120;
T=0.25;
J=T/dt+1;
K=100;

%N no. of path
N=100000;

%implement in Euler scheme

s=zeros(J,N);
y=zeros(J,N);

%set initial value

s(1,:)=99.75;
y(1,:)=0.04;

%serror=zeros(N,1);
%verror=zeros(N,1);
%generate N random variables of z and w
        z1=randn(J-1,N/2);
        z=[z1,-1*z1];

        w1=randn(J-1,N/2);
        w=[w1,-1*w1];
    for j=1:J-1

        y(j+1,:)= max(y(j,:),0) + 2*(0.04-max(y(j,:),0))*dt +
0.2*sqrt(max(y(j,:),0)*dt).*z(j,:);


        s(j+1,:)=max(s(j,:),0) + 0.01*max(s(j,:),0)*dt +
sqrt(max(y(j,:),0)).*max(s(j,:),0).*(-0.7*sqrt(dt).*z(j,:)...
                   +sqrt(0.51*dt).*w(j,:));
    end

sT=s(J,:);
call=max(sT-K,0);
mean_c=cumsum(call)./[1:N];
c_square=cumsum(call.^2)./[1:N];
```

```
c_se=sqrt(1./[1:N]).*sqrt(c_square - mean_c.^2);

plot(1:N,mean_c)
xlabel('No. of Paths')
ylabel('Call price at time T')
Title('Convergence Diagram')


implied_vol=blsimpv(S0,K,r,T,mean_c(N),1000)
```



4 (a) $$\frac{\partial C}{\partial t}+\frac{1}{2}YS^2\frac{\partial^2 C}{\partial S^2}+\rho\beta YS\frac{\partial C}{\partial S\partial Y}+\frac{1}{2}\beta^2 Y\frac{\partial^2 C}{\partial Y^2}+rS\frac{\partial C}{\partial S}+K(\theta-Y)\frac{\partial C}{\partial Y}-rC=0$$

where K=2, θ=0.04, β=0.2, ρ=-0.7, r=0.01

Note: here K, θ,β are parameters from (Heston) stochastic volatility model

Terminal Condition

C(S,Y,T)= $(S\text{-Strike})^+$, Strike =100

(b) Boundary Condition

S=0

C(S=0,Y,t) ≈ 0

since it is likely that the stock will be in the money if s=0

S=Smax=400

C(S=400,Y,t)= $400-100e^{-r(T-t)}$

Y=0 use implicit boundary condition

$$\frac{\partial C}{\partial t}+rS\frac{\partial C}{\partial S}+K\theta\frac{\partial C}{\partial Y}-rC=0$$

Y=Ymax=0.2 use implicit boundary condition: full PDE

$$\frac{\partial C}{\partial t}+\frac{1}{2}Y_{\max}S^2\frac{\partial^2 C}{\partial S^2}+\rho\beta Y_{\max}S\frac{\partial C}{\partial S\partial Y}+\frac{1}{2}\beta^2 Y_{\max}\frac{\partial^2 C}{\partial Y^2}+rS\frac{\partial C}{\partial S}+K(\theta-Y_{\max})\frac{\partial C}{\partial Y}-rC=0$$

(c) Grid: $t=0,\triangle t,2\triangle t,\ldots,M\triangle t=T=0.25$

$S=0,\triangle S,2\triangle S,\ldots,N\triangle S=S_{\max}=400$

$Y=0,\triangle Y,2\triangle Y,\ldots,J\triangle Y=Y_{\max}=0.2$

Approximation $C(n\triangle S,\, j\triangle Y, m\triangle t)\approx C_{n,j}^m$

<u>Explicit Scheme</u>

$$\frac{C_{n,j}^m-C_{n,j}^{m-1}}{\triangle t}+\frac{1}{2}(j\triangle Y)(n\triangle S)^2\frac{C_{n+1,j}^m-2C_{n,j}^m-C_{n-1,j}^m}{2\triangle S}$$

$$+\rho\beta(j\triangle Y)(n\triangle S)\frac{C_{n+1,j+1}^m-C_{n+1,j-1}^m-C_{n-1,j+1}^m+C_{n-1,j-1}^m}{4\triangle S\triangle Y}$$

$$+\frac{1}{2}\beta^2(j\triangle Y)\frac{C_{n,j+1}^m-2C_{n,j}^m+C_{n,j-1}^m}{(\triangle Y)^2}+r(n\triangle S)\frac{C_{n+1,j}^m-C_{n-1,j}^m}{2\triangle S}+K(\theta-j\triangle Y)\frac{C_{n,j+1}^m-C_{n,j-1}^m}{2\triangle Y}$$

$$-rC_{n,j}^m=0$$

$$\Rightarrow\; C_{n,j}^{m-1}=p_1C_{n+1,j-1}^m+p_2C_{n,j-1}^m+p_3C_{n-1,j-1}^m+p_4C_{n+1,j}^m+p_5C_{n,j}^m+p_6C_{n-1,j}^m$$

$$p_7 C^m_{n+1,j+1} + p_8 C^m_{n,j+1} + p_9 C^m_{n-1,j+1}$$

where $p_1 = \frac{-\rho\beta j \triangle t}{4}$, $p_2 = \frac{\beta^2 j \triangle t - K\theta \triangle t + Kj \triangle Y \triangle t}{2\triangle Y}$, $p_3 = -p_1$

$$p_4 = \frac{j \triangle Y n^2 \triangle t + \triangle t r n}{2},\ p_5 = \frac{-j(\triangle Y n)^2 \triangle t - \beta^2 j \triangle t - r \triangle Y \triangle t + \triangle Y}{\triangle Y}$$

$$p_6 = \frac{j \triangle Y n^2 \triangle t - \triangle t r n}{2},\ p_7 = p_3,\ p_8 = \frac{\beta^2 j \triangle t + K\theta \triangle t - Kj \triangle Y \triangle t}{2\triangle t},\ p_9 = p_1$$

<u>Boundary Condition</u>

Y=0

$$\frac{C^m_{n,0} - C^{m-1}_{n,0}}{\triangle t} + r(n\triangle S)\frac{C^m_{n+1,0} - C^m_{n-1,0}}{2\triangle S} + K\theta \frac{3C^m_{n,0} - 4C^m_{n,1} + C^m_{n,2}}{2\triangle Y} - rC^m_{n,0} = 0$$

$$\Rightarrow C^{m-1}_{n,0} = a_1 C^m_{n+1,0} + a_2 C^m_{n,0} + a_3 C^m_{n-1,0} + a_4 C^m_{n,1} + a_5 C^m_{n,2}$$

where

$$a_1 = \frac{rn\triangle t}{2},\ a_2 = \frac{2\triangle Y - 3K\theta \triangle t - 2r\triangle t \triangle Y}{2\triangle Y},\ a_3 = -\frac{rn\triangle t}{2},\ a_4 = \frac{2K\theta \triangle t}{\triangle Y},\ a_5 = -\frac{K\theta \triangle t}{2\triangle Y}$$

Y=Ymax=0.2

$$\frac{C^m_{n,J} - C^{m-1}_{n,J}}{\triangle t} + \frac{1}{2}(J\triangle Y)(n\triangle S)^2 \frac{C^m_{n+1,J} - 2C^m_{n,J} - C^m_{n-1,J}}{2\triangle S}$$

$$+\rho\beta(J\triangle Y)(n\triangle S)\frac{3(C^m_{n+1,J} - C^m_{n-1,J}) - 4(C^m_{n+1,J-1} - C^m_{n-1,J-1}) + (C^m_{n+1,J2} - C^m_{n-1,J-2})}{4\triangle S \triangle Y}$$

$$+\frac{1}{2}\beta^2 (J\triangle Y)\frac{2C^m_{n,J} - 5C^m_{n,J-1} + 4C^m_{n,J-2} - C^m_{n,J-3}}{(\triangle Y)^2} + r(n\triangle S)\frac{C^m_{n+1,J} - C^m_{n-1,J}}{2\triangle S}$$

$$+K(\theta - J\triangle Y)\frac{3C^m_{n,J} - 4C^m_{n,J-1} + C^m_{n,J-2}}{2\triangle Y} - rC^m_{n,J} = 0$$

$$\Rightarrow C^{m-1}_{n,J} = b_1 C^m_{n+1,J} + b_2 C^m_{n,J} + b_3 C^m_{n-1,J} + b_4 C^m_{n+1,J-1} + b_5 C^m_{n,J-1} + b_6 C^m_{n-1,J-1}$$

$$b_7 C^m_{n+1,J-2} + b_8 C^m_{n,J-2} + b_9 C^m_{n-1,J-2} + b_{10} C^m_{n,J-3}$$

where

$$b_1 = \frac{2\triangle t J \triangle Y n^2 + 3\rho\beta J n \triangle t + 2rn\triangle t}{4}$$

$$b_2 = \frac{2\triangle Y - 2\triangle t J(\triangle Y n)^2 + 2\beta^2 J\triangle t + 3K(\theta - J\triangle Y)\triangle t - 2r\triangle t \triangle Y}{2\triangle Y},\ b_3 = \frac{2\triangle t J \triangle Y n^2 - 3\rho\beta J n \triangle t - 2rn\triangle t}{4}$$

$$b_4 = -\rho\beta J n \triangle t,\ b_5 = \frac{-5\beta^2 J\triangle t - 4K(\theta - J\triangle Y)\triangle t}{2\triangle Y},\ b_6 = \rho\beta J n\triangle t,\ b_7 = \frac{\rho\beta J n \triangle t}{4}$$

$$b_8 = \frac{4\beta^2 J\triangle t - K(\theta - J\triangle Y)\triangle t}{2\triangle Y},\ b_9 = -b_7,\ b_{10} = \frac{-\beta^2 J\triangle t}{2\triangle Y}$$

Terminal Condition

$$C_{n,j}^{M} = (n\triangle S - Strike)^{+}$$

Procedure

- Compute C^M from Terminal Condition
- Successively compute C^{M-1}, C^{M-2},…,C^0
- Compute C^{m-1} from C^m explicitly

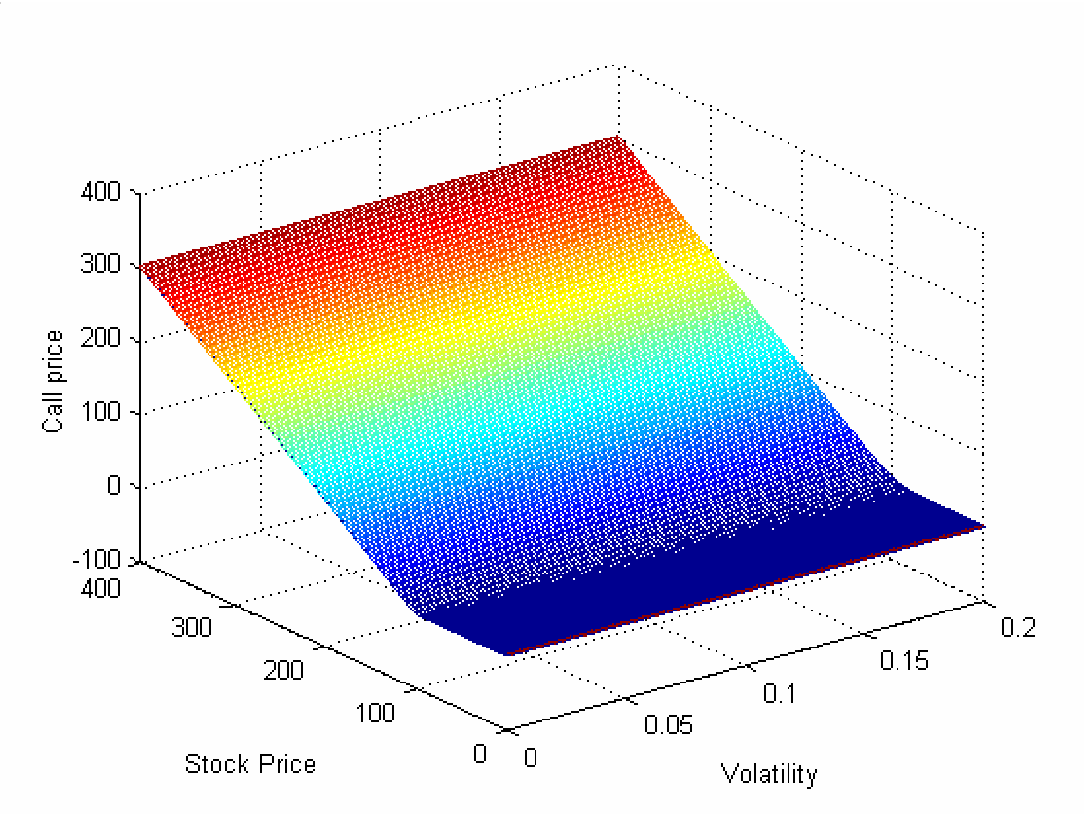
HW3#4c-g

```
format long
clear

% variables
Smax=400;
ymax=0.2;
T=0.25;
r=0.01;
theta=0.04;
k=2;
B=0.2;
strike=100;
rho=-0.7;


%pick deltas arbitary while deltay and deltat has to be picked carefully
ds= 5;
dy= 0.001;%round((ds*B^2*ymax)/(Smax*(ymax-r)+k*ymax-
ds*k*theta)*10000)/10000;
dt= 0.0001;%((dy*ds)^2)/(ymax*(dy*Smax)^2+(B*ds)^2*ymax+r*(dy*ds)^2);

%Grid
J=ymax/dy;
N=Smax/ds;
M=T/dt;

c=zeros(N+1,J+1,M+1);
s=[0:ds:Smax];
ss=s'*ones(1,J+1);

%Terminal Condition
c(:,:,M+1)= max(ss-strike,0);

%Boundary condition
%at s=0
c(1,:,:)=0;
```

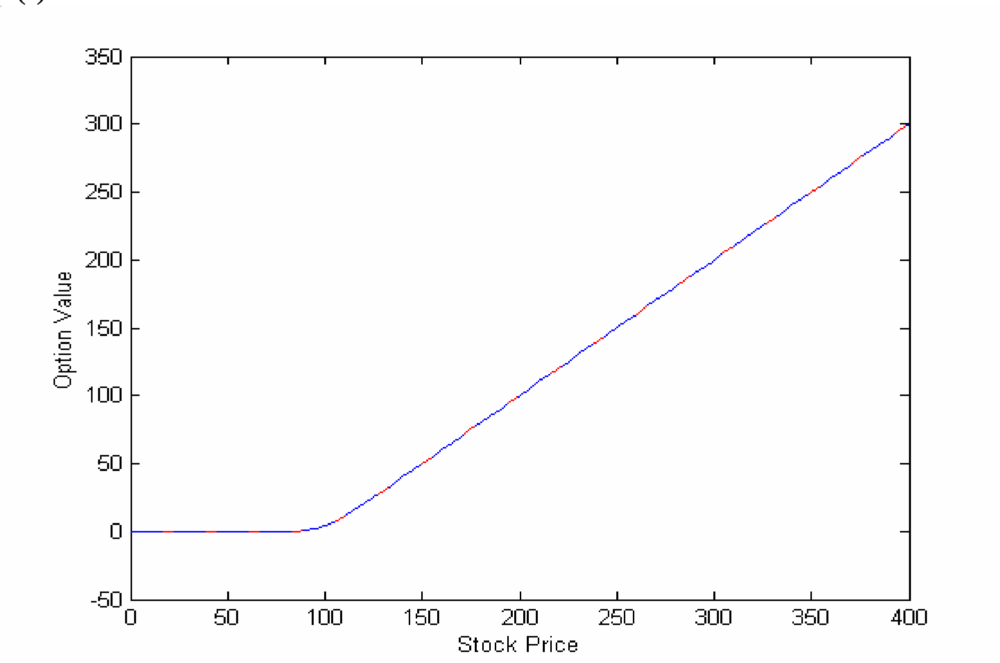
```
%at s=400
t=[0:dt:T];
c(N+1,:,:)=ones(J+1,1)*(Smax-strike.*exp(-r.*(T-t)));

nvec=[0:N]';
nmat=[0:N]'*ones(1,J+1);
jmat=ones(1,N+1)'*[0:J];

%coefficients for implicit boundary y=0
a1=r*nvec*dt/2;
a2=(2*dy-3*k*theta*dt-2*r*dt*dy)/(2*dy)*ones(N+1,1);
a3=-a1;
a4=(2*k*theta*dt)/(dy)*ones(N+1,1);
a5=(-k*theta*dt)/(2*dy)*ones(N+1,1);

%coefficients for implicit boundary y=ymax
b1=(2*dt*J*dy*nvec.^2+3*rho*B*J*nvec*dt+2*r*nvec*dt)/4;
b2=(2*dy-2*dt*J*(nvec*dy).^2+2*B^2*J*dt+3*k*(theta-J*dy)*dt-
2*r*dt*dy)/(2*dy);
b3=(2*dt*J*dy*nvec.^2-3*rho*B*J*nvec*dt-2*r*nvec*dt)/4;
b4=-rho*B*J*nvec*dt;
b5=(-5*B^2*J*dt-4*k*(theta-J*dy)*dt)/(2*dy)*ones(N+1,1);
b6=-b4;
b7=b6/4;
b8=(4*B^2*J*dt+k*(theta-J*dy)*dt)/(2*dy)*ones(N+1,1);
b9=-b7;
b10=(-B^2*J*dt)/(2*dy)*ones(N+1,1);

%coefficients for explicit method
p1=-rho*B*jmat.*nmat*dt/4;
p2=((B^2)*jmat.*dt-k*theta*dt+k*jmat.*dy*dt)/(2*dy);
p3=rho*B*jmat.*nmat*dt/4;
p4=(jmat.*dy.*(nmat.^2)*dt+dt*r*nmat)/2;
p5=(-jmat.*((dy*nmat).^2)*dt-(B^2)*jmat*dt-r*dy*dt+dy)/dy;
p6=(jmat.*(nmat.^2)*dy*dt-dt*r*nmat)/2;
p7=p3;
p8=((B^2)*jmat*dt+k*theta*dt-k*jmat*dy*dt)/(2*dy);
p9=p1;


%calculate backward in time with implicit boundary condition
for m=M:-1:1

  %explicit method
   c(2:N,2:J,m)=p1(2:N,2:J).*c(3:N+1,1:J-1,m+1) + p2(2:N,2:J).*c(2:N,1:J-1,m+1)
+ p3(2:N,2:J).*c(1:N-1,1:J-1,m+1) + ...
```

```
              p4(2:N,2:J).*c(3:N+1,2:J,m+1)   + p5(2:N,2:J).*c(2:N,2:J,m+1)   +
p6(2:N,2:J).*c(1:N-1,2:J,m+1) + ...
              p7(2:N,2:J).*c(3:N+1,3:J+1,m+1) + p8(2:N,2:J).*c(2:N,3:J+1,m+1) +
p9(2:N,2:J).*c(1:N-1,3:J+1,m+1);

   %implicit boundary for y=0
    c(2:N,1,m)=a1(2:N).*c(3:N+1,1,m+1) + a2(2:N).*c(2:N,1,m+1) +
a3(2:N).*c(1:N-1,1,m+1) + ...
             a4(2:N).*c(2:N,2,m+1) + a5(2:N).*c(2:N,3,m+1);

   %implicit boundary for y=ymax
    c(2:N,J+1,m)=b1(2:N).*c(3:N+1,J+1,m+1) + b2(2:N).*c(2:N,J+1,m+1) +
b3(2:N).*c(1:N-1,J+1,m+1) +...
              b4(2:N).*c(3:N+1,J,m+1)   + b5(2:N).*c(2:N,J,m+1)   +
b6(2:N).*c(1:N-1,J,m+1) +...
              b7(2:N).*c(3:N+1,J-1,m+1) + b8(2:N).*c(2:N,J-1,m+1) +
b9(2:N).*c(1:N-1,J-1,m+1) +...
              b10(2:N).*c(2:N,J-2,m+1);
end

%call_price=c(100/ds+1,0.04/dy+1,1)
%4e
%mesh([0:dy:ymax]',[0:ds:Smax]',c(:,:,1))

%4f
%plot([0:ds:Smax]',c(:,0.04/dy+1,1),'r')
%hold
%plot([0:ds:Smax]',CBS(0.2,strike,[0:ds:Smax]',T,r),'--b')

4(g)
for i=1:N+1
implied_vol(i)=blsimpv(ds*(i-1),strike,r,T,c(i,0.04/dy+1,1),100);
end
plot([0:ds:Smax]',implied_vol)
```

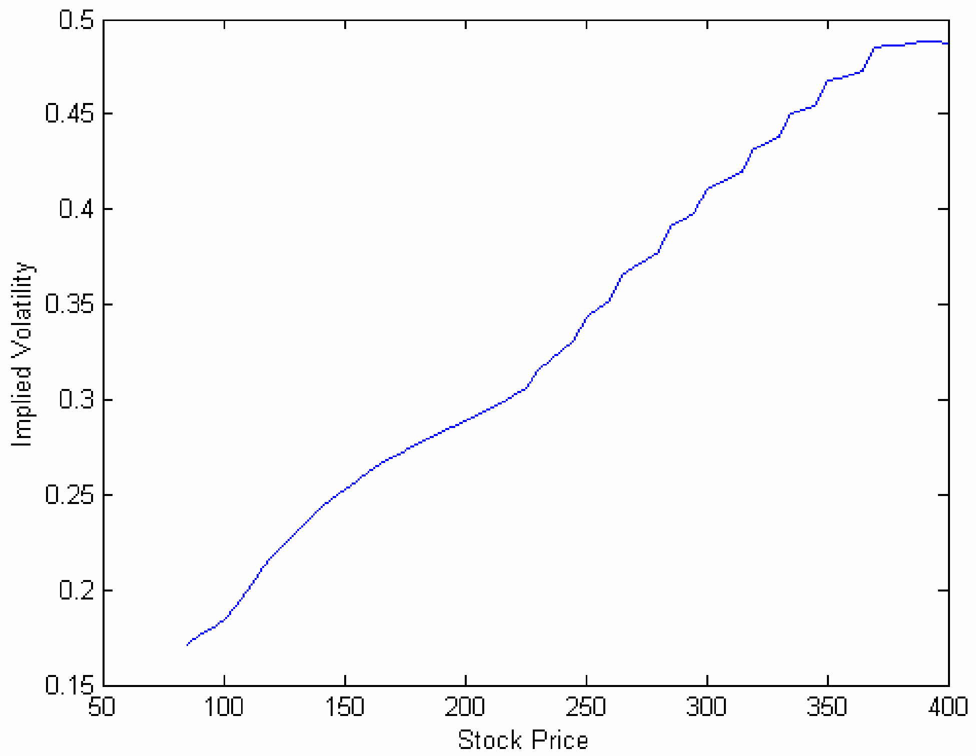
4(e)


400
300
200
100
0
-100
Call price
400
300
200
100
0
Stock Price
0
0.05
0.1
0.15
0.2
Volatility

4(f)


350
300
250
200
150
100
50
0
-50
Option Value
0
50
100
150
200
250
300
350
400
Stock Price

4(g)



0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
Implied Volatility
50
100
150
200
250
300
350
400
Stock Price