Math 623 F03
HW#5 Solution

Question 1

1a) T= 0.5, r= 0.03, D=0.01, Sigma=0.2, S_0=75
European Strangle option = Euro Call option (K=120) + Euro Put option (K=80)
Apply BS formula; we obtain (For continuous paying dividend stock, basically, just replace S_0 with S_0e^{-DT})
Euro Strangle price = 6.73527724541701

2b) i ud=1
We have 4 equations as follow

Moment 0: $P_u + P_d = 1$

Moment 1: $P_u u + P_d d = e^{(r-D)\Delta t}$

Moment 2: $P_u u^2 + P_d d^2 = e^{(\sigma^2 + 2(r-D))\Delta t}$

$ud = 1$

Solve above equations, we obtain

$$\Rightarrow u = A + \sqrt{A^2 - 1}$$

$$d = A - \sqrt{A^2 - 1}, \text{ where } A = \frac{1}{2}(e^{-(r-D)\Delta t} + e^{((r-D)+\sigma^2)\Delta t})$$

$$P_u = \frac{e^{(r-D)\Delta t} - d}{u - d}$$

$$P_d = 1 - P_u$$

Plug in all variables with $\Delta t = 2^{\wedge}(-10)$, we get
u=1.00626971531204, d=0.99376934909534,
P_u=0.49999993896544, P_d=0.50000006103456.

ii P_u=P_d=1/2
We have 4 equations as follow

Moment 0: $P_u + P_d = 1$

Moment 1: $P_u u + P_d d = e^{(r-D)\Delta t}$

Moment 2: $P_u u^2 + P_d d^2 = e^{(\sigma^2 + 2(r-D))\Delta t}$

$$P_u = P_d = \frac{1}{2}$$

Solve above equations with $\Delta t = 2^{\wedge}(-10)$, we get

$$\Rightarrow u = e^{(r-D)\Delta t}(1 + \sqrt{e^{\sigma^2 \Delta t - 1}})$$

$$d = e^{(r-D)\Delta t}(1 - \sqrt{e^{\sigma^2 \Delta t - 1}})$$

$$P_u = P_d = \frac{1}{2}$$

Plug in all variables, we get
u=1.00627954144975, d=0.99377905315856
P_u=0.5, P_d=0.5

iii We also get 4 equations as follow

Moment 0: $P_u + P_d = 1$

Moment 1: $P_u u + P_d d = e^{(r-D)\Delta t}$

Moment 2: $P_u u^2 + P_d d^2 = e^{(\sigma^2 + 2(r-D))\Delta t}$

Moment 3: $P_u u^3 + P_d d^3 = e^{(3\sigma^2 + 3(r-D))\Delta t}$

$$\Rightarrow P_u = \frac{e^{(r-D)\Delta t} - d}{u - d}$$

$$P_d = \frac{u - e^{(r-D)\Delta t}}{u - d}$$

$$P_u u^2 + (1 - P_u) d^2 - e^{(\sigma^2 + 2(r-D))\Delta t} = 0$$

$$P_u u^3 + (1 - P_u) d^3 - e^{(3\sigma^2 + 3(r-D))\Delta t} = 0$$

Use function fsolve in Matlab to solve set of equations above with $\Delta t = 2 \wedge (-10)$, we obtain (See code below)
u=1.00633110436343, d=0.99382201218358
P_u=0.49544116935495, P_d=0.50455883064505

2c) See Code below
With $\Delta t = 2 \wedge (-10)$ we have, result as below
ud=1 : Option price = 6.87765727655044
Pu=Pd=0.5: Option price = 6.71913777606207
Matching 3rd moment: Option price = 6.88013164825830

Matching Moment 3

k (dt=2^(-k))	u	d	Pu	Pd	V
1	1.187692	0.894033	0.395073	0.604927	7.715675
2	1.122107	0.918321	0.425405	0.574595	6.999132
4	1.055248	0.954778	0.462551	0.537449	6.982708
8	1.012816	0.987809	0.490626	0.509374	6.884138
10	1.006331	0.993822	0.495441	0.504559	6.880132
11	1.004459	0.995619	0.496691	0.503309	6.877024

Code for 1b iii)

```
clear
format long

r=0.03;
sig=0.2;
D=0.01;
S0=75;
dt=2^(-10);
x0=[1.1;0.9];
options=optimset('Display','iter','TolFun',1e-10);
[x,fval]=fsolve(@moment3,x0,options);
u=x(1);
d=x(2);
Pu=(exp((r-D)*dt)-d)/(u-d);
 Pd=1-Pu;

function F = moment3(x)
r=0.03;
dt=2^(-10);
D=0.01;
vol=0.2;
F= [Pu*(x(1)^2)+Pd*(x(2)^2)-exp((2*(r-D)+vol^2)*dt)...
        ; Pu*(x(1)^3)+Pd*(x(2)^3)-exp((3*(r-D)+3*(vol^2))*dt)];
```

Code for c i)

```
%Implementation the pricing of a American Strangle option using binomial tree
% Under condition ud = 1

% Define variable
    clear all;
    format long;

    S0 = 75;
    r = 0.03;
    D=0.01;
    T = 0.5;
    dt = 2^(-10);
    vol = 0.2;

    A = 0.5*(exp(-(r-D)*dt) + exp(((r-D)+vol^2)*dt));
    d = A - sqrt(A^2 - 1);
    u = A + sqrt(A^2 - 1);
    Pu = (exp((r-D)*dt) - d)/(u-d);
    Pd = (u - exp((r-D)*dt))/(u-d);
```

```
    M = T/dt;

    S = zeros(M+1,M+1);
    V = zeros(M+1,M+1);

    S(1,1) = S0;

    for m = 2:M+1
        for n = 1:m
            S(n,m) = S0*u^(m-n)*d^(n-1);
        end
    end

    for n = 1:M+1
        if S(n,M+1) <= 80
            V(n,M+1) = 80-S(n,M+1);
        elseif (S(n,M+1) < 120) & (S(n,M+1) > 80)
            V(n,M+1) = 0;
        else
            V(n,M+1) = S(n,M+1)-120;
        end
    end

    for m = M:-1:1
        for n = M:-1:1
            % implement American property
            V(n,m) = max(exp(-r*dt)*(Pu*V(n,m+1) + Pd*V(n+1,m+1)),max(80-
S(n,m),0)+max(S(n,m)-120,0));
        end
    end

% display the result
disp ('The price of a American Strangle option using binomial tree with dt =
1/1000 and S(0)=75 is');
V(1,1)
disp ('the value of parameter');
u
d
Pu
Pd
disp('The Euro Strangle option price')
EuroStr=PBS(vol,80,S0,T,r)+CBS(vol,120,S0,T,r)
```

Code 1c ii)

```
% Math 623  Homework5  #1

% Implementation the pricing of a American Strangle option using binomial tree
% Under condition ud = 1

% Define variable


    clear all;

    format long;

    S0 = 75;
    r = 0.03;
    D=0.01;
    T = 0.5;
    dt = 2^(-10);
    vol = 0.2;

    d = (exp(r*dt))*(1 - sqrt((exp(dt*vol^2))-1));
    u = (exp(r*dt))*(1 + sqrt((exp(dt*vol^2))-1));
    Pu = 0.5;
    Pd = 0.5;
    M = T/dt;

    S = zeros(M+1,M+1);
    V = zeros(M+1,M+1);

    S(1,1) = S0;

    for m = 2:M+1
        for n = 1:m
            S(n,m) = S0*u^(m-n)*d^(n-1);
        end
    end

    for n = 1:M+1
        if S(n,M+1) <= 80
            V(n,M+1) = 80-S(n,M+1);
        elseif (S(n,M+1) < 120) & (S(n,M+1) > 80)
            V(n,M+1) = 0;
        else
            V(n,M+1) = S(n,M+1)-120;
        end
    end
```

```
   for m = M:-1:1
       for n = M:-1:1
           % implement American property
           V(n,m) = max(exp(-r*dt)*(Pu*V(n,m+1) + Pd*V(n+1,m+1)),max(80-
S(n,m),0)+max(S(n,m)-120,0));
       end
   end

% display the result
disp ('The price of a American Strangle option using binomial tree with dt =
1/1000 and S(0)=75 is');
V(1,1)
disp ('the value of parameter');
u
d
Pu
Pd
disp('The Euro Strangle option price')
EuroStr=PBS(vol,80,S0,T,r)+CBS(vol,120,S0,T,r)
```

Code for 1c iii)

```
% Math 623  Homework5  #1

% Implementation the pricing of a American Strangle option using binomial tree
% Under condition ud = 1

% Define variable
   clear all;
   format long;

   r = 0.03;
   T = 0.5;
   dt = 2^(-10);
   r=0.03;
   vol=0.2;
   D=0.01;
   S0=75;
   M=T/dt;

   x0=[1.1;0.9];
   options=optimset('Display','iter','TolFun',1e-10);
   [x,fval]=fsolve(@moment3,x0,options);
   u=x(1);
  d=x(2);
```

```
    Pu=(exp((r-D)*dt)-d)/(u-d);
    Pd=1-Pu;
    S = zeros(M+1,M+1);
    V = zeros(M+1,M+1);

    S(1,1) = S0;

    for m = 2:M+1
        for n = 1:m
            S(n,m) = S0*u^(m-n)*d^(n-1);
        end
    end

    for n = 1:M+1
        if S(n,M+1) <= 80
            V(n,M+1) = 80-S(n,M+1);
        elseif (S(n,M+1) < 120) & (S(n,M+1) > 80)
            V(n,M+1) = 0;
        else
            V(n,M+1) = S(n,M+1)-120;
        end
    end

    for m = M:-1:1
        for n = M:-1:1
            % implement American property
            V(n,m) = max(exp(-r*dt)*(Pu*V(n,m+1) + Pd*V(n+1,m+1)),max(80-
S(n,m),0)+max(S(n,m)-120,0));
        end
    end

 % display the result
disp ('The price of a American Strangle option using binomial tree with dt =
1/1000 and S(0)=75 is');
V(1,1)
disp ('the value of parameter');
u
d
Pu
Pd
disp('The Euro Strangle option price')
EuroStr=PBS(vol,80,S0,T,r)+CBS(vol,120,S0,T,r)
```

Question2

2a) $\rho = (AA)^T AA$ and $\tilde{\sigma} = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} * AA$

Calculate this by Matlab(See code below), we got

$$\tilde{\sigma} = \begin{bmatrix} 0.25 & 0 \\ 0.08 & 0.183 \end{bmatrix}$$

$$\Rightarrow \frac{dS_{1t}}{S_{1t}} = rdt + 0.25dZ_{1t}$$

$$\frac{dS_{2t}}{S_{2t}} = rdt + 0.08dZ_{1t} + 0.183dZ_{2t}$$

Where Z_{it} are the independent Brownian motion

2b) $\begin{bmatrix} X_{1t} \\ X_{2t} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \log S_{1t} \\ \log S_{2t} \end{bmatrix}$

Let define

$$\begin{bmatrix} Y_{1t} \\ Y_{2t} \end{bmatrix} = \begin{bmatrix} \log S_{1t} \\ \log S_{2t} \end{bmatrix}$$

Apply Ito's formula to Y, we obtain

$$\begin{bmatrix} dY_{1t} \\ dY_{2t} \end{bmatrix} = \begin{bmatrix} d\log S_{1t} \\ d\log S_{2t} \end{bmatrix} = \begin{bmatrix} (r - \frac{1}{2}\sum_{j=1}^{2} \tilde{\sigma}_{1j}^2)dt + \sum_{j=1}^{2} \tilde{\sigma}_{1j} dZ_{1t} \\ (r - \frac{1}{2}\sum_{j=1}^{2} \tilde{\sigma}_{2j}^2)dt + \sum_{j=1}^{2} \tilde{\sigma}_{2j} dZ_{2t} \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} dX_{1t} \\ dX_{2t} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} (r - \frac{1}{2}\sum_{j=1}^{2} \tilde{\sigma}_{1j}^2)dt + \sum_{j=1}^{2} \tilde{\sigma}_{1j} dZ_{1t} \\ (r - \frac{1}{2}\sum_{j=1}^{2} \tilde{\sigma}_{2j}^2)dt + \sum_{j=1}^{2} \tilde{\sigma}_{2j} dZ_{2t} \end{bmatrix}$$

If we consider about A, we notice that A = $\tilde{\sigma}^{-1}$ since

$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} \begin{bmatrix} X_{1t} \\ X_{2t} \end{bmatrix} = \begin{bmatrix} \log S_{1t} \\ \log S_{2t} \end{bmatrix}$, and we need to satisfy $\begin{bmatrix} dX_{1t} \\ dX_{2t} \end{bmatrix} = \begin{bmatrix} \mu_1 dt + dZ_{1t} \\ \mu_2 dt + dZ_{2t} \end{bmatrix}$

Then $A = \begin{bmatrix} 4 & 0 \\ -1.746 & 5.455 \end{bmatrix}$

And $\begin{bmatrix}\mu_1\\ \mu_2\end{bmatrix} = \begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\begin{bmatrix}(r-\frac{1}{2}\sum_{j=1}^{2}\tilde{\sigma}_{1j}^2)\\ (r-\frac{1}{2}\sum_{j=1}^{2}\tilde{\sigma}_{2j}^2)\end{bmatrix}$

We get from Matlab code that

$$\begin{bmatrix}\mu_1\\ \mu_2\end{bmatrix} = \begin{bmatrix}\text{-0.065}\\ 0.001091089\end{bmatrix}$$

2c) See code below (I constructed the tree with Pu=Pd=0.5)

<u>Steps</u>

- Build the product tree for (X_{1t},X_{2t}) at nodes (m,n_1,n_2)
  - $(m+1,n_1+1,n_2+1)$ with prob. Pu_1Pu_2,
  - $(m+1,n_1+1,n_2)$ with prob. Pu_1Pd_2,
  - $(m+1,n_1,n_2+1)$ with prob. Pd_1Pu_2,
  - $(m+1,n_1,n_2)$ with prob. Pd_1Pd_2,

  choosing parameters such that

  $$P_{ui}+P_{di}=1$$

  $$P_{ui}\log(u_i)+P_{di}\log(d_i)=\mu_i\Delta t$$

  $$P_{ui}(\log(u_i))^2+P_{di}(\log(d_i))^2=(\mu_i\Delta t)^2+\Delta t$$

  $$P_{ui}=P_{di}=\frac{1}{2}$$

  where i=1,2

  and

  $$X_{1,n1,n2}^m = X_{1,0}+n_1\log(u_1)+(m-n_1)\log(d_1)$$

  $$X_{2,n1,n2}^m = X_{2,0}+n_2\log(u_2)+(m-n_2)\log(d_2)$$

2d) Following from 2c), at each point (M,n_1,n_2), we know X_1,X_2 and we obtain S_1, S_2 from following equation

$$\begin{bmatrix}\log S_{1t}\\ \log S_{2t}\end{bmatrix} = \tilde{\sigma}^{-1}\begin{bmatrix}X_{1t}\\ X_{2t}\end{bmatrix}$$

Then we compute V^M, V^{M-1},…,V^0 successively;

$$V^M = \left[K-(S_{1,n1,n2}^M+S_{2,n1,n2}^M)/2\right]^+$$

and

$$V_{n1,n2}^{m-1} = e^{-r\Delta t}\left[P_{u1}P_{u2}V_{n1+1,n2+1}^m + P_{u1}P_{d2}V_{n1+1,n2}^m + P_{d1}P_{u2}V_{n1,n2+1}^m + P_{d1}P_{d2}V_{n1,n2}^m\right]$$

See code below

I did two way, with condition ud=1 and Pu=Pd=0.5

With ud=1

Option value = 6.09406916514486

With Pu=Pd=0.5

Option value = 6.13462313233680

For Rho, I run code twice (the one with condition ud=1) with different interest rate, first with r=15.5% and 14.5%. Then I use central difference approximation as follow.

$$\rho \approx \frac{V(r=15.5\%)-V(r=14.5\%)}{2*0.5\%}$$

$$\rho \approx \frac{6.08585121688618-6.10229231308614}{2*0.5\%} = -16.44109619996037$$

Code for 2c

```
% Construct 2-dimentional tree
clear
format long
% Variable
% r - interest rate
% Pu1 - risk neutral probility for X1 going up
% Pd1 - risk neutral probility for X1 going down
% Pu2 - risk neutral probility for X2 going up
% Pd2 - risk neutral probility for X1 going down
% m1 - drift term of X1 process
% m2 - drift term of X2 process
% T - Product tree of X1,X2 at time M
deltat= 1/288;
r=0.015;
K=60;
T=4/12;
M=T/deltat+1;
s0=[52 ;56];
rho = [1 0.4;0.4 1];
AA=chol(rho)';
sigma=[0.25,0.2]';
sighat = diag(sigma)*AA;
A=sighat^(-1);
m1=A(1,1)*(r-0.5*sighat(1,1)^2)+A(1,2)*(r-0.5*sighat(2,1)^2-
0.5*sighat(2,2)^2);
m2=A(2,1)*(r-0.5*sighat(1,1)^2)+A(2,2)*(r-0.5*sighat(2,1)^2-
0.5*sighat(2,2)^2);
Pu1=0.5;
Pd1=0.5;
Pu2=0.5;
Pd2=0.5;
```

```
logu=zeros(2,1);
logd=zeros(2,1);
x0=[log(s0(1,1))*A(1,1);log(s0(1,1))*A(2,1)+log(s0(2,1))*A(2,2)];
logu(1,1) = m1*deltat + sqrt(deltat);
logd(1,1) = m1*deltat - sqrt(deltat);
logu(2,1) = m2*deltat + sqrt(deltat);
logd(2,1) = m2*deltat - sqrt(deltat);
X2=zeros(M,M,M);
X1= zeros(M,M,M);

for i=1:M
    for j=1:i
        for k=1:i
            X1(i,j,k)=x0(1)+(i-j)*logu(1,1)+(j-1)*logd(1,1);
            X2(i,j,k)=x0(2)+(i-k)*logu(2,1)+(k-1)*logd(2,1);
        end
    end
end
```

Code for 2d with condition Pu=Pd=0.5

```
% Valuation of Basket option with two correlated stocks
clear
format long
% Variable
% r - interest rate
% Pu1 - risk neutral probility for X1 going up
% Pd1 - risk neutral probility for X1 going down
% Pu2 - risk neutral probility for X2 going up
% Pd2 - risk neutral probility for X1 going down
% m1 - drift term of X1 process
% m2 - drift term of X2 process
% T - Product tree of X1,X2 at time M

deltat= 1/288;
r=0.015;
K=60;
T=4/12;
M=T/deltat+1;
s0=[52 ;56];
rho = [1 0.4;0.4 1];
AA=chol(rho)';
sigma=[0.25,0.2]';
sighat = diag(sigma)*AA;
A=sighat^(-1);
```

```
m1=A(1,1)*(r-0.5*sighat(1,1)^2)+A(1,2)*(r-0.5*sighat(2,1)^2-
0.5*sighat(2,2)^2);
m2=A(2,1)*(r-0.5*sighat(1,1)^2)+A(2,2)*(r-0.5*sighat(2,1)^2-
0.5*sighat(2,2)^2);
Pu1=0.5;
Pd1=0.5;
Pu2=0.5;
Pd2=0.5;
logu=zeros(2,1);
logd=zeros(2,1);
x0=[log(s0(1,1))*A(1,1);log(s0(1,1))*A(2,1)+log(s0(2,1))*A(2,2)];

logu(1,1) = m1*deltat + sqrt(deltat);
logd(1,1) = m1*deltat - sqrt(deltat);
logu(2,1) = m2*deltat + sqrt(deltat);
logd(2,1) = m2*deltat - sqrt(deltat);
V=zeros(M,M,M);
X1= zeros(M,M);
X2= zeros(M,M);

for j=1:M
   for i=1:j
   X1(i,j)=x0(1)+(i-1)*logu(1)+(j-i)*logd(1);
   X2(i,j)=x0(2)+(i-1)*logu(2)+(j-i)*logd(2);
   end
end

S1=exp(sighat(1,1)*X1(:,M)+sighat(1,2)*X2(:,M));
S2=exp(sighat(2,1)*X1(:,M)+sighat(2,2)*X2(:,M));
for i=1:M
   for j=1:M
      V(i,j,M)=max(K-(S1(i)+S2(j))/2,0);
   end
end

for k=M:-1:2
   for i=1:(k-1)
      for j=1:(k-1)
         V(i,j,k-1)=exp(-
r*deltat)*(Pu1*Pu2*V(i+1,j+1,k)+Pd1*Pu2*V(i,j+1,k)+Pu1*Pd2*V(i+1,j,k)+
Pd1*Pd2*V(i,j,k));
      end
   end
end
V(1,1,1)
```

Code for 2c with condition ud=1

```
% Valuation of Basket option with two correlated stocks
clear
format long
% Variable
% r - interest rate
% Pu1 - risk neutral probility for X1 going up
% Pd1 - risk neutral probility for X1 going down
% Pu2 - risk neutral probility for X2 going up
% Pd2 - risk neutral probility for X1 going down
% m1 - drift term of X1 process
% m2 - drift term of X2 process
% T - Product tree of X1,X2 at time M
deltat= 1/288;
r=0.015-0.0005
K=60;
T=4/12;
M=T/deltat+1;
s0=[52 ;56];
rho = [1 0.4;0.4 1];
AA=chol(rho)';
sigma=[0.25,0.2]';
sighat = diag(sigma)*AA;
A=sighat^(-1);
m1=A(1,1)*(r-0.5*sighat(1,1)^2)+A(1,2)*(r-0.5*sighat(2,1)^2-
0.5*sighat(2,2)^2);
m2=A(2,1)*(r-0.5*sighat(1,1)^2)+A(2,2)*(r-0.5*sighat(2,1)^2-
0.5*sighat(2,2)^2);
logu=zeros(2,1);
logd=zeros(2,1);
x0=[log(s0(1,1))*A(1,1);log(s0(1,1))*A(2,1)+log(s0(2,1))*A(2,2)];

logu(1,1) = sqrt((m1*deltat)^2 + deltat);
logd(1,1) = -logu(1,1);
logu(2,1) = sqrt((m2*deltat)^2 + deltat);
logd(2,1) = -logu(2,1);
Pu1=0.5*(1+(m1*deltat)/logu(1));
Pd1=1-Pu1;
Pu2=0.5*(1+(m2*deltat)/logu(2));
Pd2=1-Pu2;
V=zeros(M,M,M);
X1= zeros(M,M);
X2= zeros(M,M);

for j=1:M
   for i=1:j
```

```
    X1(i,j)=x0(1)+(i-1)*logu(1)+(j-i)*logd(1);
    X2(i,j)=x0(2)+(i-1)*logu(2)+(j-i)*logd(2);
    end
end

S1=exp(sighat(1,1)*X1(:,M)+sighat(1,2)*X2(:,M));
S2=exp(sighat(2,1)*X1(:,M)+sighat(2,2)*X2(:,M));
for i=1:M
    for j=1:M
        V(i,j,M)=max(K-(S1(i)+S2(j))/2,0);
    end
end

for k=M:-1:2
    for i=1:(k-1)
        for j=1:(k-1)
            V(i,j,k-1)=exp(-
r*deltat)*(Pu1*Pu2*V(i+1,j+1,k)+Pd1*Pu2*V(i,j+1,k)+Pu1*Pd2*V(i+1,j,k)+
Pd1*Pd2*V(i,j,k));
        end
    end
end
V(1,1,1)
```