Homework #6

Pittsburgh
Jingyi Guo

Problem 1: Longstaff and Schwartz method

Code:

```
S0=40;
K=40; % in this case will not exercise at time 0
sig=0.2;
r=0.06;
T=1; % or 2
N=50;
n=50000;
delta=T/N;

% First generate stock price paths
S=zeros(n,N);
San=zeros(n,N); %store antithetic values of S
S0v=ones(n,1)*S0;

for j=1:N
    z=randn(n,1);
    if j==1
        S(:,j)=S0v.*exp((r-
1/2*sig^2)*delta+sig*sqrt(delta).*z);
        San(:,j)=S0v.*exp((r-
1/2*sig^2)*delta+sig*sqrt(delta).*(-z));
    else
        S(:,j)=S(:,j-1).*exp((r-
1/2*sig^2)*delta+sig*sqrt(delta).*z);
        San(:,j)=San(:,j-1).*exp((r-
1/2*sig^2)*delta+sig*sqrt(delta).*(-z));
    end
end
S=[S;San];% first 50000 rows: original; second 50000 rows:
antithetic
P = max(K-S(:,50),0); % Payoff at final time

% Cash flow table
for i = (N-1):-1:1
    Si = S(:,i);
    itmP = find(K-Si>0); % ITM path label
    otmP = find(K-Si<=0);
    X = Si;
    Y = P*exp(-r*delta); % Discounted payoffs (a reduced-dim
column)
    Y(otmP) = 0;
    XMat =
[ones(size(X)),Laguerre(X,0),Laguerre(X,1),Laguerre(X,2)];
    XMat(otmP,:)=0;
    beta = XMat\Y;
    C = XMat*beta; % Value of continuation
    E = K-X; % Value of immediate exercise
    exP = find(C<E); % immediate exercise
    P(exP) = E(exP);
    rest = setdiff(1:n,exP);
    P(rest) = P(rest)*exp(-r*delta); % continue
end
P1=P(1:n)*exp(-r*delta);
Pa=P((n+1):2*n)*exp(-r*delta);
newP=(P1+Pa)/2;
P_bar = mean(newP)
P_stderr = std(newP)/sqrt(n)

function output = Laguerre(X,k) % k - order of laguerre, X -
input vector
    if k==0
        output = exp(-X/2);
    elseif k == 1
        output = exp(-X/2).*(1-X);
    elseif k == 2
        output = exp(-X/2).*(1-2*X+X.^2/2);
    end
end
```

Result:

	T=1	T=2
Estimate	2.2813	2.8563
Std. Err	0.0042	0.0055

Problem 2: GHS and Carpriotti Importance Sampling

(a) GHS

Code:

```
clear
S0=100;
r=0.05;
sig=0.20;
T=1;
K=120; % or 140, 160
n=10000;

% a
% B-S formula
dplus=1/(sig*sqrt(T))*(log(S0/K)+(r+1/2*sig^2)*T);
dminus=1/(sig*sqrt(T))*(log(S0/K)+(r-1/2*sig^2)*T);
BS_price = S0*normcdf(dplus)-normcdf(dminus)*K*exp(-r*T);

% standard MC
S_stdMC=S0*ones(n,1).*exp((r-
0.5*sig^2)*T+sig*sqrt(T).*randn(n,1));
C_stdMC=exp(-r*T).*max(S_stdMC-K,0);
C_stdMC_bar=mean(C_stdMC)
C_stdMC_stderr=std(C_stdMC)/sqrt(n)

% GHS
% solve for m:
syms x positive
S_T_m=S0*exp((r-1/2*sig^2)*T+sig*sqrt(T)*x);
eqn = x == S_T_m*sig*sqrt(T)/(S_T_m-K);
xsolve = solve(eqn) % xsolve = vpasolve(eqn,3)
m=double(xsolve)

% importance sampling
z=randn(n,1);
S_IS=S0*ones(n,1).*exp((r-0.5*sig^2)*T+sig*sqrt(T)*(z+m));
C_IS=exp(-r*T).*max(S_IS-K,0).*exp(-m*z-0.5*m^2);
C_IS_bar=mean(C_IS)
C_IS_stderr=std(C_IS)/sqrt(n)
```

Result:

Strike Price	BS Price	Std MC Est.	Std. Err	mhat	Imp. Samp. Est.	Std. Err
120	3.2475	3.3392	0.0875	1.4849	3.2706	0.0223

140	0.7850	0.7767	0.0410	2.0465	0.7773	0.0068
160	0.1590	0.1573	0.0186	2.6002	0.1577	0.0016

(b) Capriotti Importance Sampling

Code:

```
K=120; % or 140, 160
M=10000;
ind=randn(M,1);
dep=zeros(M,1);
mtrial=1;
bslink = @(m, X)((exp(-r*T)*max(S0*exp((r-
sig^2/2)*T+X*sig*sqrt(T))-K,0)).*sqrt(exp(-m*X+m^2/2)));
mhat=nlinfit(ind,dep,bslink,mtrial)
z=randn(n,1);
S_C=S0*ones(n,1).*exp((r-0.5*sig^2)*T+sig*sqrt(T)*(z+mhat));
C_C=exp(-r*T).*max(S_C-K,0).*exp(-mhat*z-0.5*mhat^2);
C_C_bar=mean(C_C);
C_C_stderr=std(C_C)/sqrt(n)
```

Result:

	K=120	K=140	K=160
mhat estimate	1.6505	2.2167	2.7377
C Std. Error	0.0216	0.0067	0.0016

(c) Capriotti Importance Sampling II

Code:

```
K=120; % or 140, 160
bslink = @(m, X)((exp(-r*T)*max(S0*exp((r-
sig^2/2)*T+X*sig*sqrt(T))-K,0)).*sqrt(exp(m(2))*exp(-
0.5*(X.^2-((X-m(1))/exp(m(2))).^2))));
ind = randn(M,1);
dep = zeros(M,1);
mtrial = [1,1];
mhat = nlinfit(ind,dep,bslink,mtrial);
m = mhat(1)
s = exp(mhat(2))

S_C2=S0*exp((r-0.5*sig^2)*T+sig*sqrt(T)*(s*z+m));
C_C2=exp(-r*T)*s*max(S_C2-K,0).*exp(-0.5*((s*z+m).^2-z.^2));
C_C2_bar=mean(C_C2);
C_C2_stderr=std(C_C2)/sqrt(n);
```

Result:

	K=120	K=140	K=160

mhat estimate	1.7441	2.3069	2.8080
shat estimate	0.6422	0.6061	0.3940