Table of Contents

Part2a) Compare Least Square Method and Simple Monte Carlo

At the Money

```
for i=1:50 %running 50 times to compare 50 different results.
[Price_simpleMC_atthemoney(i),S]=AmericanPutMCsim2019...
    (157.005,157.005,0.013533,0.22013,10,5000,0.166667);
% I am comparing the same stock paths for two methods to make a fair
% comparison,  S=stockpath
[Price_least_squared_atthemoney(i)]=LeastSquareMethod...
    (157.005,157.005,0.013533,10,5000,0.166667,S);
end
figure(1)
plot(S) %This is the last Stock Path just to show it how it looks like
title('Stock Path')

Observed_Market_Value_atthemoney=5.87;

% Comparison by using Mean Square Error and Mean Absolute Error

% MAE and MSE at the money
 MAE_LeastSquared_atthemoney= abs(Price_least_squared_atthemoney...
     -Observed_Market_Value_atthemoney);
 MAE_SimpleMC_atthemoney= abs(Price_simpleMC_atthemoney-...
     Observed_Market_Value_atthemoney);
 figure(2)
 plot(MAE_LeastSquared_atthemoney)
 hold on
 plot(MAE_SimpleMC_atthemoney)
legend('MAE LeastSquared atthemone','MAE SimpleMC atthemoney')
 title('MAE Simple Method and Least Squared At The Money')

 %In The Money
Observed_Market_value_inthemoney=14.29;
for i=1:50 %running 50 times to compare 50 different results.
[Price_simpleMC_inthemoney(i),S]=AmericanPutMCsim2019...
    (157.005,170,0.013533,0.22013,10,5000,0.166667);
% I am comparing the same stock paths for two methods to make a fair
% comparison,  S=stockpath
[Price_least_squared_inthemoney(i)]=LeastSquareMethod...
    (157.005,170,0.013533,10,5000,0.166667,S);
end
```

```
MAE_LeastSquared_intthemoney= abs(Price_least_squared_inthemoney...
     -Observed_Market_value_inthemoney);
 MAE_SimpleMC_inthemoney= abs(Price_simpleMC_inthemoney-...
     Observed_Market_value_inthemoney);

figure(3)
plot(MAE_LeastSquared_intthemoney)
hold on
plot(MAE_SimpleMC_inthemoney)
legend('MAE LeastSquared intthemone','MAE SimpleMC atthemoney')
title('MAE Simple Method and Least Squared in The Money')

%Out Of Money
Observed_Market_value_outofmoney=0.63;
for i=1:50 %running 50 times to compare 50 different results.
[Price_simpleMC_outofmoney(i),S]=AmericanPutMCsim2019...
    (157.005,140,0.013533,0.22013,10,5000,0.166667);
% I am comparing the same stock paths for two methods to make a fair
% comparison,  S=stockpath
[Price_least_squared_outofmoney(i)]=LeastSquareMethod...
    (157.005,140,0.013533,10,5000,0.166667,S);
end
MAE_LeastSquared_outofmoney= abs(Price_least_squared_outofmoney...
     -Observed_Market_value_outofmoney);
MAE_SimpleMC_outofmoney= abs(Price_simpleMC_outofmoney-...
     Observed_Market_value_outofmoney);
figure(4)
plot(MAE_LeastSquared_outofmoney)
hold on
plot(MAE_SimpleMC_outofmoney)
legend('MAE LeastSquared outofmoney','MAE SimpleMC outofmoney')
title('MAE Simple Method and Least Squared out of Money')
```

Simple Monte Carlo Simulation American Option Pricing

Used Dr.K's function

```
function [Price_simpleMC,S]=AmericanPutMCsim2018(S_0,K,r,sigma,n,b,T)
dt=(T/b); % computes the lenght of a time stpe

SPaths=zeros(n,b+1); %preallocate the trajectories matrix, S.
S(:,1)=S_0*ones(n,1); %store initial value S0

% Generates trajectories
for i=2:b+1

 S(:,i)=S(:,i-1)+r*S(:,i-1)*(dt)+sigma*S(:,i-1)*sqrt(dt).*randn(n,1);
end

S4=max((K-S),0); %Stores the values of the option payoff
```

```
Price=zeros(n,b+1); % defines the index X(m) defined in Algorithm 3.

SS=zeros(n,1);

if (S_0==K)
    SS=ones(n,1);
else
    for o=1:n
        hh=find(S4(o,:)');
        if hh
            w=min(hh);
            SS(o)=w;
        else
            SS(o)=0;
        end
    end
end

for i=1:n
        if ((SS(i)>0)&(SS(i)<(b+1)))

        jj=SS(i);

        dd_1=(log(S(i,jj)/K)+(r+(sigma*sigma/2)*(T-dt*(jj-1))))/...
            (sigma*sqrt(T-dt*(jj-1)));
        dd_2=dd_1-sigma*sqrt(T-dt*(jj-1));


        BlackI=K*exp(-r*(T-dt*(jj-1)))*normcdf(-dd_2,0,1)-(S(i,jj))...
            *normcdf(-dd_1,0,1);

        if (S4(i,b+1)>0)
            Price(i,b+1)=exp(-r*dt*(jj-1))*BlackI;
        end

        % This loop computes the values needed on one given path

        for j=SS(i):(b)

            d_1=(log(S(i,j)/K)+(r+(sigma*sigma/2)*(T-dt*(j-1))))/...
                (sigma*sqrt(T-dt*(j-1)));
            d_2=d_1-sigma*sqrt(T-dt*(j-1));

            Black=K*exp(-r*(T-dt*(j-1)))*normcdf(-d_2,0,1)-
(S(i,j))*...
                normcdf(-d_1,0,1);

            Price(i,j)=exp(-r*(j-1)*dt)*(S4(i,j)-Black)+...
                exp(-r*dt*(jj-1))*BlackI;

        end

        % If the option is always out of the money its price equals 0
    elseif (SS(i)==0)
```

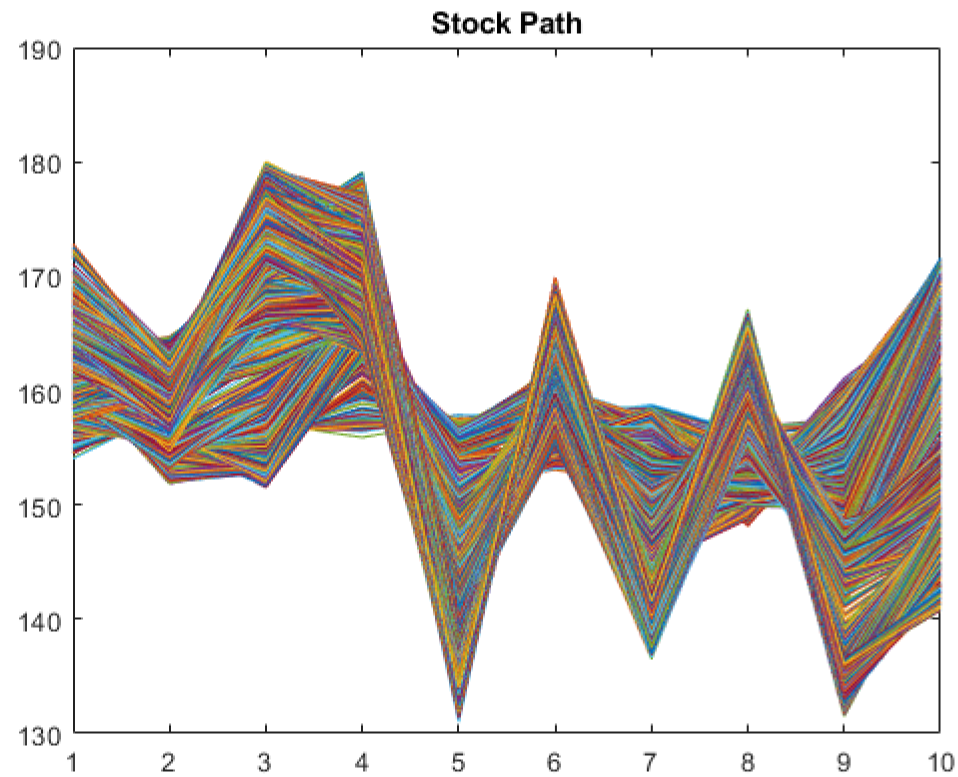
```
        Price(i,:)=0;

        %If the option enters for the first time in the money at t=T
    elseif(SS(i)==b+1)
        Price(i,b+1)=exp(-r*T)*S4(i,b+1);
    end
end
Price_simpleMC=mean(max(Price'));
end
```

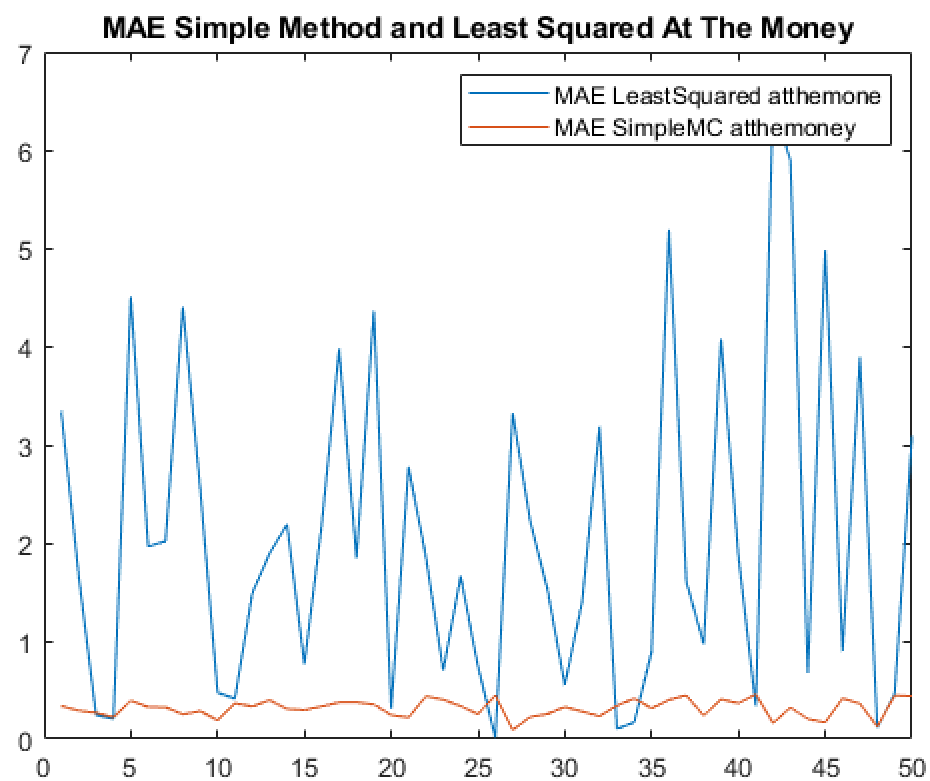
Least-Squares Method of Longstaff and Schwartz(2001) Function

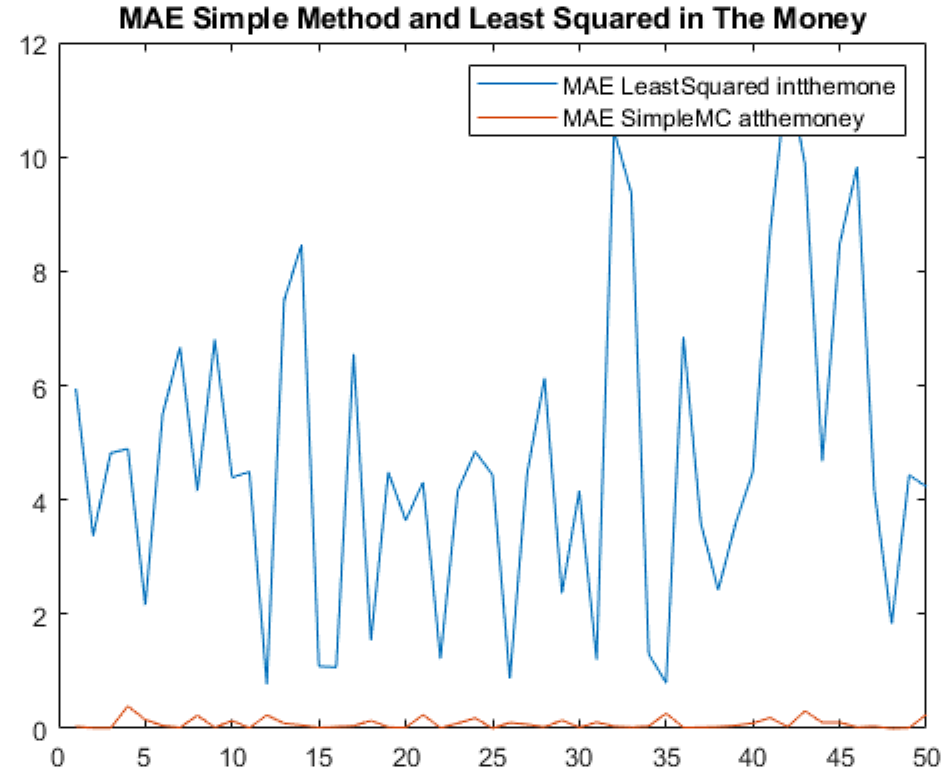
```
function
 [Price_least_squared]=LeastSquareMethod(SO,K,r,NRepl,NSteps,T,S)
dt = T/NSteps;
discountVet = exp(-r*dt*(1:NSteps)');
SPaths=S;
alpha = zeros(3,1) ; % regression parameters
CashFlows = max (0, K - SPaths ( : , NSteps) ) ;
ExerciseTime = NSteps*ones(NRepl,1);
for step = NSteps-1:-1:1
    InMoney = find(SPaths(: ,step) < K);
    XData = SPaths(InMoney,step);
    RegrMat = [ones(length(XData), 1), XData, XData.^2];
    YData = CashFlows(InMoney).*discountVet(ExerciseTime(InMoney)-
step);
    alpha = RegrMat \ YData;
    IntrinsicValue = K - XData;
    ContinuationValue = RegrMat * alpha;
    Index = find(IntrinsicValue>ContinuationValue) ;
    ExercisePaths = InMoney(Index);

    CashFlows(ExercisePaths) = IntrinsicValue(Index);
    ExerciseTime(ExercisePaths) = step;
end % for
Price_least_squared = max( K-SO, mean(CashFlows.*...
    discountVet(ExerciseTime)) );
end
```

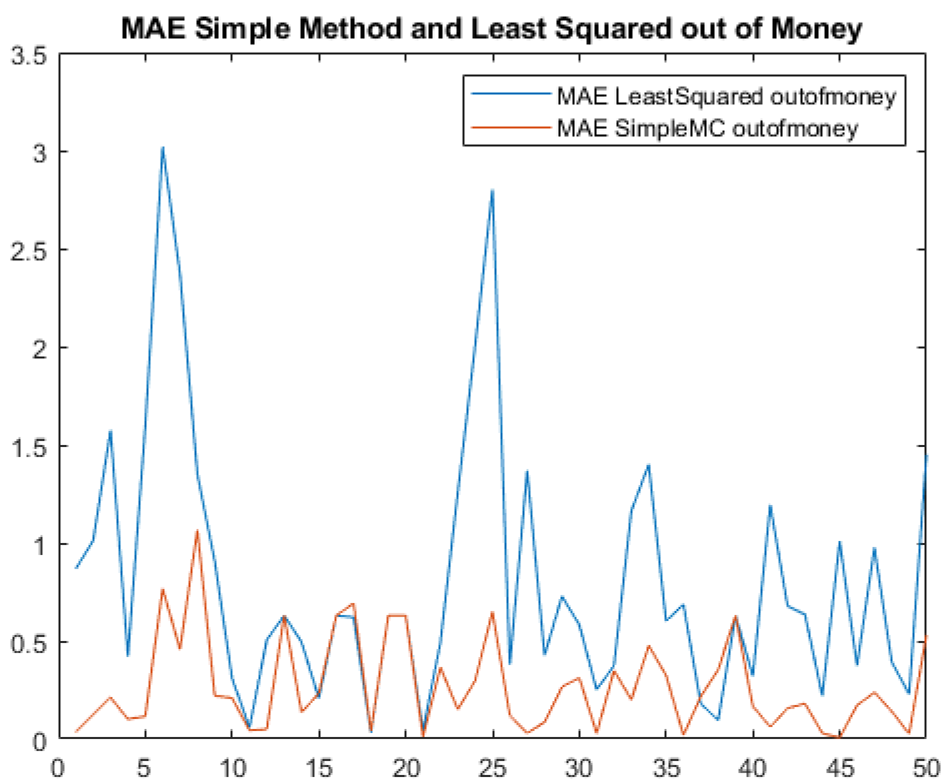
```
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
```

```
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
Warning: Rank deficient, rank = 1, tol =  1.730881e-10.
```


Stock Path
190
180
170
160
150
140
130
1
2
3
4
5
6
7
8
9
10


MAE Simple Method and Least Squared At The Money
MAE LeastSquared atthemone
MAE SimpleMC atthemoney
7
6
5
4
3
2
1
0
0
5
10
15
20
25
30
35
40
45
50


MAE Simple Method and Least Squared in The Money
MAE LeastSquared intthemone
MAE SimpleMC atthemoney
0
2
4
6
8
10
12
0
5
10
15
20
25
30
35
40
45
50


MAE Simple Method and Least Squared out of Money
MAE LeastSquared outofmoney
MAE SimpleMC outofmoney
0
0.5
1
1.5
2
2.5
3
3.5
0
5
10
15
20
25
30
35
40
45
50

Published with MATLAB® R2018b