A4Q1:

```
function [c,p,stdc,stdp]=A4Q1(M, delt)
%generate the stable graph
randn('state', 100);

%define the initial value
sigma=0.20; %volatility
r=0.05; %risk-free rate
T=1.0; %total time T
K=100; %strike price
S0=100; %initial stock price
N=T/delt; %define the total number of N


%define the initial matrix:
S_old=zeros(M,1);
S_new=zeros(M,1);

S_old(1:M,1)=S0;

for i=1:N %time step loop
    S_new(:,1)=S_old(:,1)+S_old(:,1).*(r*delt+sigma*sqrt(delt)*randn(M,1));
    S_new(:,1)=max(0, S_new(:,1)); %consider the different cases for future
    %stock price
    S_old(:,1)=S_new(:,1);
end

%the payoff of the option:
payoffcall=max((S_new-K),0);
payoffput=max((K-S_new),0);

%get the value and std of the option
c=exp(-r*T)*(sum(payoffcall))/M;
p=exp(-r*T)*(sum(payoffput))/M;
stdc=std(exp(-r*T)*max((S_new-K),0));
stdp=std(exp(-r*T)*max((K-S_new),0));

end
```

```matlab
%define the simulations and timesteps:
M=[1000;2000;4000;8000;16000;32000;64000];
delt=[5/250, 2.5/250, 1/250];

%compute the range for call & put options:
%call option define the initial matrix
lowcal=ones(7,1);
upcal=ones(7,1);

%put:
lowput=ones(7,1);
upput=ones(7,1);

for i=1:7
    [c(i,3),p(i,3),stdc(i,3),stdp(i,3)]=A4Q1(M(i),delt(3));
    %the formula of the CI: MIU+/- quantile*std
    lowcal(i)=c(i,3)-1.96*stdc(i,3)/sqrt(M(i));
    upcal(i)=c(i,3)+1.96*stdc(i,3)/sqrt(M(i));
    lowput(i)=p(i,3)-1.96*stdp(i,3)/sqrt(M(i));
    upput(i)=p(i,3)+1.96*stdp(i,3)/sqrt(M(i));
end

%calculate the CI for call option:
CIcall=table(M,c(:,3),lowcal,upcal);
disp(CIcall)

%calculate the CI for put option:
CIput=table(M,p(:,3),lowput,upput);
disp(CIput)
```

Output for call option:

M	Var2	lowcal	upcal
1000	9.9954	9.0987	10.892
2000	10.568	9.895	11.241
4000	10.385	9.9281	10.841
8000	10.53	10.207	10.853
16000	10.511	10.284	10.738
32000	10.47	10.309	10.63
64000	10.492	10.377	10.606

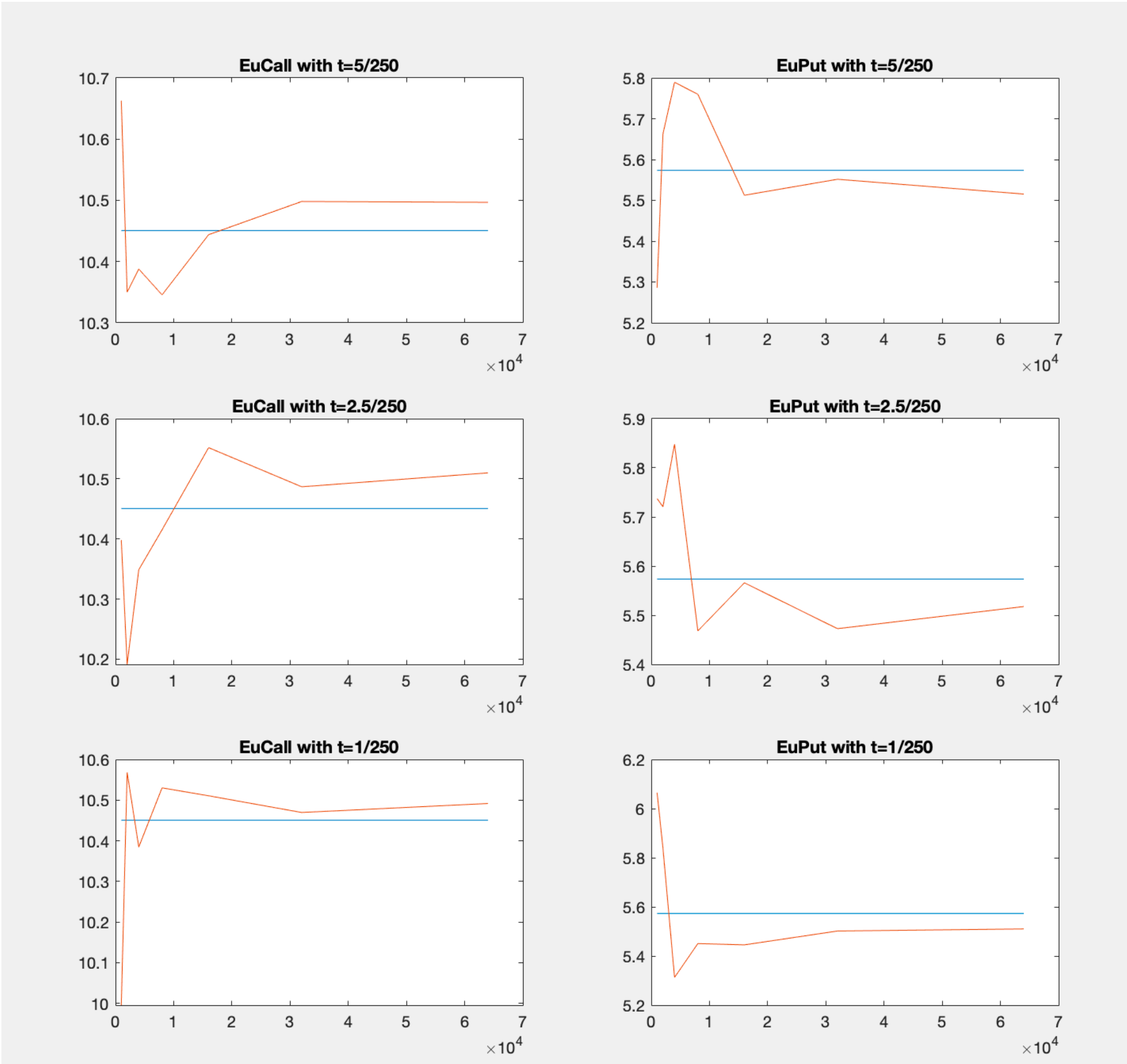
Output for put option:

M	Var2	lowput	upput
1000	6.0649	5.5033	6.6266
2000	5.8398	5.4572	6.2224
4000	5.3131	5.0515	5.5746
8000	5.4507	5.2659	5.6355
16000	5.4455	5.3134	5.5776
32000	5.5018	5.4075	5.5962
64000	5.5104	5.4436	5.5771

The tables tell us: when the number of simulations(M) increase, the mean of the option value will approach to the exact mean value.

```matlab
%define the simulations and timesteps:
M=[1000,2000,4000,8000,16000,32000,64000];
delt=[5/250, 2.5/250, 1/250];

for i=1:length(delt)
    for j=1:length(M)
        [c(j,i),p(j,i)]=A4Q1(M(j),delt(i));
    end
end

[blsc,blsp]=blsprice(100,100,0.05,1.00,0.2);
%---------------------------------------------
%draw the graph for delta=5/250
%call option:
subplot(3,2,1)
plot(M,blsc*ones(size(M)));
title('EuCall with t=5/250');
hold on;
plot(M,c(:,1));
hold off;
%put option:
subplot(3,2,2)
plot(M,blsp*ones(size(M)));
title('EuPut with t=5/250');
hold on;
plot(M,p(:,1));
hold off;

%--------------------------------------------
%draw the graph for delta=2.5/250
%call option:
subplot(3,2,3)
plot(M,blsc*ones(size(M)));
title('EuCall with t=2.5/250');
hold on;
plot(M,c(:,2));
hold off;
%put option:
subplot(3,2,4)
plot(M,blsp*ones(size(M)));
title('EuPut with t=2.5/250');
hold on;
plot(M,p(:,2));
hold off;

%----------------------------------------
%draw the graph for delta=1/250
%call option:
subplot(3,2,5)
plot(M,blsc*ones(size(M)));
title('EuCall with t=1/250');
hold on;
plot(M,c(:,3));
hold off;
%put option:
subplot(3,2,6)
plot(M,blsp*ones(size(M)));
title('EuPut with t=1/250');
hold on;
plot(M,p(:,3));
hold off;
```

The graph output:



By observing from the graph, we find as the total number of simulations increases, the option price tends to approach to the exact value of option calculated by BS equation, which means as the total simulation increases, the error of the option prices will become smaller. In this case, we need to try more times of simulations to reduce the error when we use Monte Carlo method.