```
% number of simulations
M = [1000, 2000, 4000, 8000, 16000, 32000, 64000];
% number of timesteps
delta = [5/250, 2.5/250, 1/250];

% Loop through each timestep and simulation to calculate the option value
for i = 1:3
    for j = 1:7
        [Call_Value(j,i),Put_Value(j,i)]= MonteCarlo(M(j),delta(i));

    end
end

% Compute Black-Scholes price of the option
[blsCall, blsPut] = blsprice(100,100,0.05,1.0,0.20);



% plot for delta = 5/250
subplot(3,2,1)
plot(M,blsCall*ones(size(M)));
title("European Call: plot for delta_t = 5/250")
hold on;
plot(M,Call_Value(:,1));
hold off;

subplot(3,2,2)
plot(M,blsPut*ones(size(M)));
title('European Put: plot for delta_t = 5/250')
hold on;
plot(M,Put_Value(:,1));
hold off;

% plot for delta = 2.5/250
subplot(3,2,3)
plot(M,blsCall*ones(size(M)));
title('European Call: plot for delta_t = 2.5/250')
hold on;
plot(M,Call_Value(:,2));
hold off;

subplot(3,2,4)
plot(M,blsPut*ones(size(M)));
title('European Put: plot for delta_t = 2.5/250')
hold on;
plot(M,Put_Value(:,2));
hold off;

% plot for delta = 1/250
subplot(3,2,5)
```

```
plot(M,blsCall*ones(size(M)));
title('European Call: plot for delta_t = 1/250')
hold on;
plot(M,Call_Value(:,3));
hold off;

subplot(3,2,6)
plot(M,blsPut*ones(size(M)));
title('European Put: plot for delta_t = 1/250')
hold on;
plot(M,Put_Value(:,3));
hold off;
```