448 HW2 Report

Mingfu Liang, Student ID:3146919, NetID:MLQ4767

February 2019

1. Problem 1

In this problem, we are going to use Monte-Carlo integration to evaluate the integrals below and compare the results with the corresponding analytical results I_{exact}. For the Monte-Carlo integration of each of the integrals use M sample points X_j and determine the Monte-Carlo estimate $\hat{I}$ of the integral as well as an estimate for the standard error of the mean, σ_I. We also need to absolute value of the error of the average of $\hat{I}$ across the N_{trials}, such as $\epsilon_{<\hat{I}>}$ and ϵ_m. As well as the corresponding standard deviation $std(\hat{I})$ and the average $< \hat{\sigma}_I >$ of $\hat{\sigma}_I$ across the N_{trials} as a function of $M = 4^p$ with p up to $p = 9$. The reason we use *loglog* plot for plotting these statistics since by log, we can see the linear relationship between the number of the sample, M, with different statistics much more clear.

1.1 Problem 1(a)

In Problem 1(a), we need to calculate the integral of $\int_0^\infty \cos(x)e^{-x}dx$ and we set $p(x) = e^(-x)$ and $g(x) = cos(x)$. The exact solution $I_{exact} = 1/2$. First I plot all the statistics which I need to measure, which are $\hat{I}$, $< \hat{\sigma}_I >$, $\epsilon_{<\hat{I}>}$, $std(\hat{I})$ and ϵ_m in Figure (1). Then Let's see how the $\hat{I}$, $std(\hat{I})$ and $< \hat{\sigma}_I >$ scale with M. Look at the Figure (1) we can see that, $\hat{I}$ almost not change when the M is larger. From Figure (2) we can see that actually the $\hat{I}$ did change but the amount is extremely small that even can be ignored, and the $< \hat{I} >$ is almost the same as the $I_{exact} = 1/2$. When the M grows larger, the $< \hat{I} >$ tend to be stable at 0.5, which is the same as I_{exact}.

Now let's look at the $< \hat{\sigma}_I >$. From Figure (3) we can see that there is linear relationship between M and $< \hat{\sigma}_I >$ and by the *ployfit* from *MATLAB*, I find that the relationship between M and $< \hat{\sigma}_I >$ is

$$log_{10}(< \hat{\sigma}_I >) = -0.4982 * log_{10}(M) - 0.2352$$

For $std(\hat{I})$ we can get the same conclusion as $< \hat{\sigma}_I >$ from Figure (4) and by the *ployfit* from *MATLAB*, I find that the relationship between M and $std(\hat{I})$ is

$$log_{10}(std(\hat{I})) = -0.4913 * log_{10}(M) - 0.2633$$

Be careful that the parameter may change when you run the code again, but this equation give you some approximation relationship about how these statistics scale with M.

To sum up, this is a successful Monte-Carlo Integration attempt and by increasing M, we can get a more accurate integration as we want.

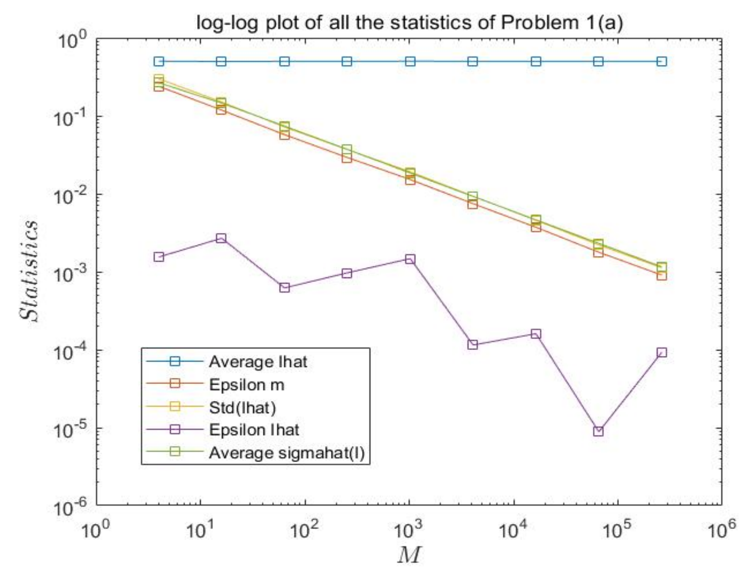

Figure 1: log-log plot of all the measured statistics of Problem 1(a)

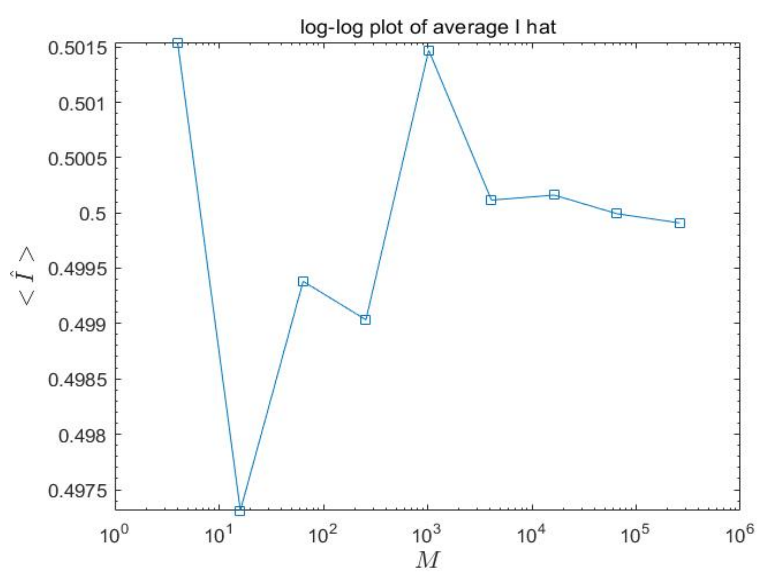

Figure 2: log-log plot of average $\hat{I}$, $<\hat{I}>$

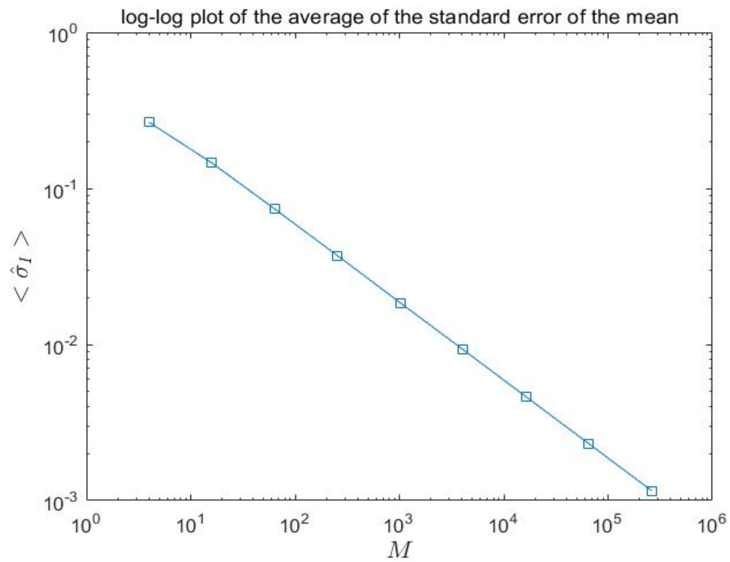

Figure 3: log-log plot of the average of the standard error of the mean, $< \hat{\sigma_I} >$

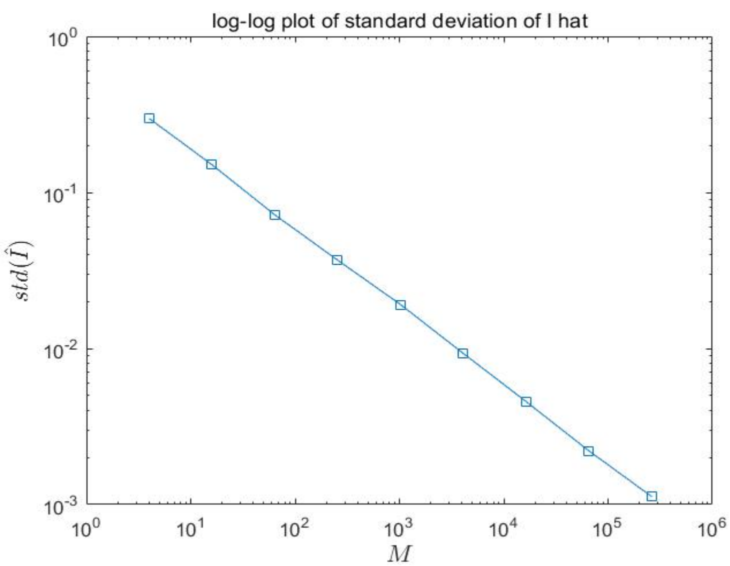

Figure 4: log-log plot of the standard deviation of $\hat{I}$, $std(\hat{I})$

1.2 Problem 1(b)

In this Problem 1(b), we may change the α for $\alpha = 1/4$ or $\alpha = 3/4$ with $g(x) = x^{-\alpha}$ and $p(x) = e^{-x}$ and consider about the integral of

$$\int_0^\infty x^{-\alpha} e^{-x} dx$$

1.2.1 Problem 1(b) when $\alpha = 1/4$

First let's look at the case when $\alpha = 1/4$. Figure (5) shows all the statistics when $\alpha = 1/4$, whose pattern are very similar to the problem 1(a). This is a successful Monte-Carlo Integration attempt and by increasing M, we can get a more accurate integration as we want.

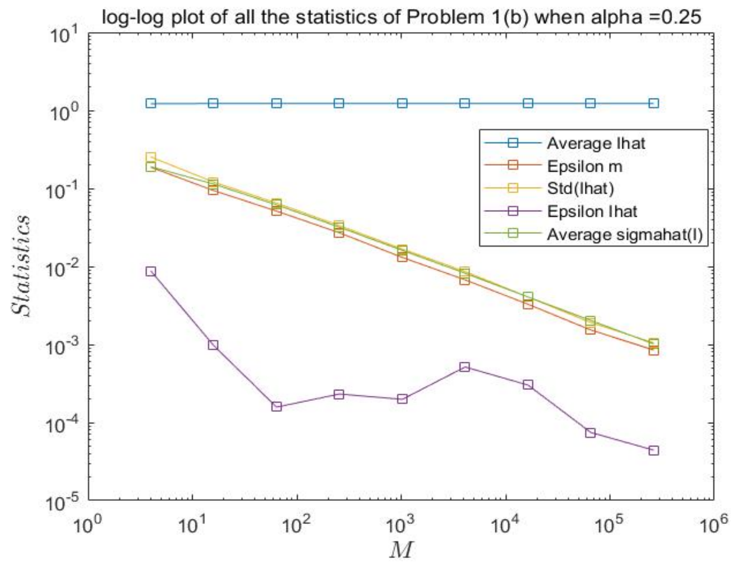

Figure 5: log-log plot of all the measured statistics of Problem 1(b) when $\alpha = 1/4$

To get the I_{exact}, we use $gamma(1-alpha)$ in $MATLAB$ and I get the approximation exact solution is 1.225416702465178. From Figure (6) we can see that when the M is bigger, the $<\hat{I}>$ tend to be stable at 1.2254, which is the same as I_{exact}.

Now let's look at the $<\hat{\sigma}_I>$. From Figure (7) we can see that there is linear relationship between M and $<\hat{\sigma}_I>$ and by the $ployfit$ from $MATLAB$, I find that the relationship between M and $<\hat{\sigma}_I>$ is

$$log_{10}(<\hat{\sigma}_I>) = -0.5022 * log_{10}(M) - 0.2685$$

For $std(\hat{I})$ we can get the same conclusion as $<\hat{\sigma}_I>$ from Figure (8) and by the $ployfit$ from $MATLAB$, I find that the relationship between M and $std(\hat{I})$ is

$$log_{10}(std(\hat{I})) = -0.4807 * log_{10}(M) - 0.2685$$

Be careful that the parameter may change when you run the code again, but this equation give you some approximation relationship about how these statistics scale with M.

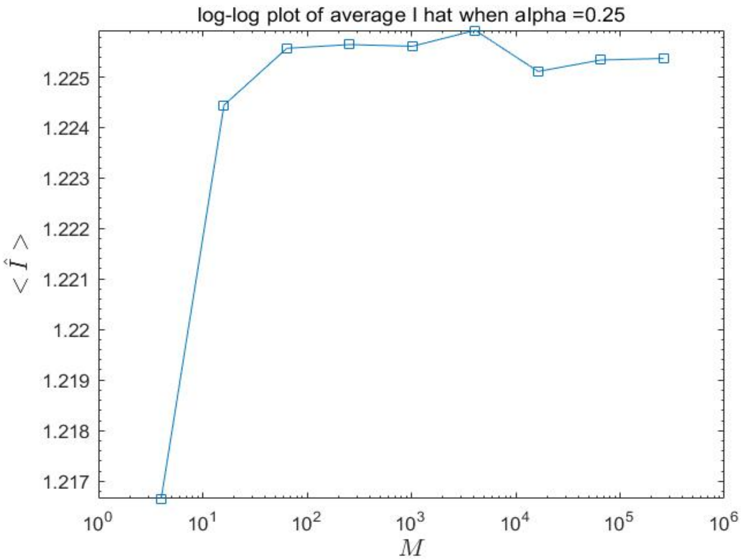

Figure 6: log-log plot of average $\hat{I}$, $< \hat{I} >$, when $\alpha = 1/4$

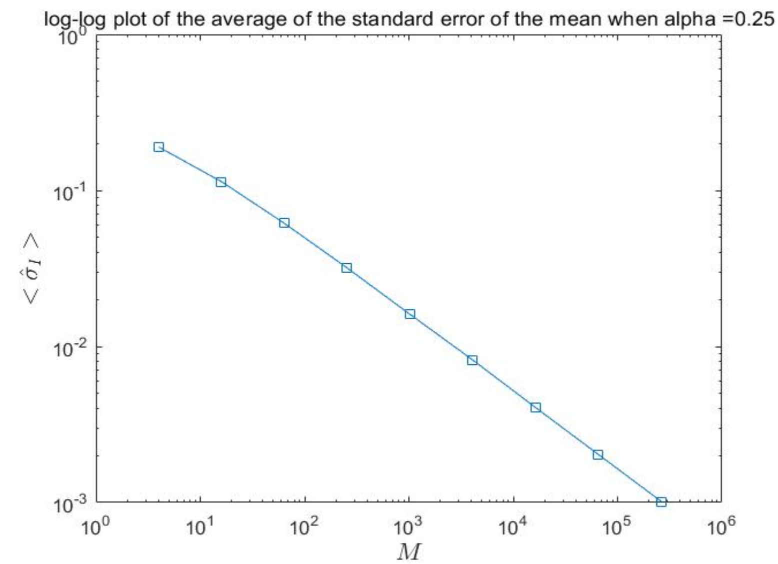

Figure 7: log-log plot of the average of the standard error of the mean, $< \hat{\sigma_I} >$, when $\alpha = 1/4$

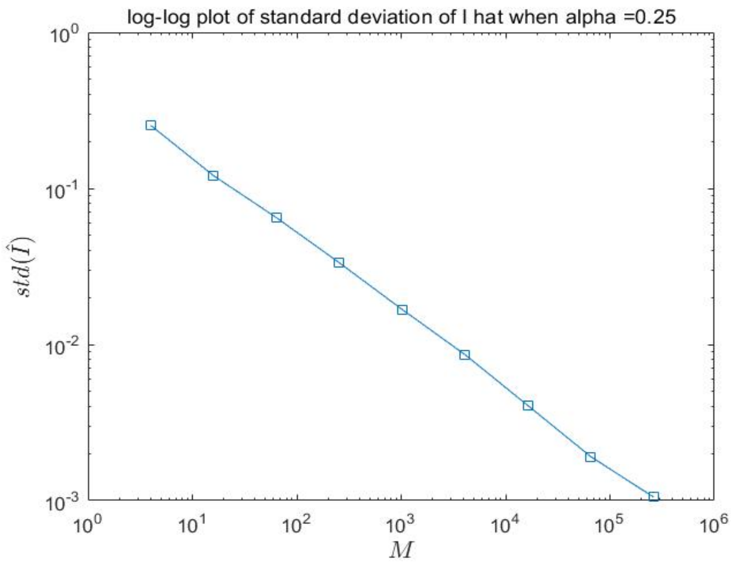

Figure 8: log-log plot of the standard deviation of $\hat{I}$, $std(\hat{I})$, when $\alpha = 1/4$

1.2.2 Problem 1(b) when $\alpha = 3/4$

Now let's look at $\alpha = 3/4$, Figure (9) shows all the statistics when $\alpha = 3/4$, and the pattern are very different from the $\alpha = 1/4$. To get insight into the results consider the analytical expression for $Var[g(x)]$ for general α. Firstly $Var(g(x)) = E((g(x) - E(g(x)))^2)$ and in this problem $g(x) = x^{-\alpha}$ and $E(g(x)) = \int_0^\infty x^{-\alpha}e^{-x}dx = gamma(1-\alpha)$. Then by some algebra we can get

$$Var(g(x)) = gamma(1-2\alpha) - gamma^2(1-\alpha)$$

When $\alpha = 1/4$, the $Var(g(x)) = gamma(1/2) - gamma^2(3/4)$ and it is finite; When $\alpha = 3/4$, the $Var(g(x)) = gamma(-1/2) - gamma^2(1/4)$ and since $gamma(-1/2)$ is not defined, therefore we can not use this formula to calculate the $Var(g(x))$ and we need to check the integral $E(g(x)) = \int_0^\infty x^{-\alpha}e^{-x}dx = gamma(1-\alpha)$ directly, and obviously when $\alpha = 3/4$ is infinite. Therefore we do not see the linear relationship exist in Figure (11) and Figure (12) for $< \hat{\sigma}_I >$ and $std(\hat{I})$ like those when $\alpha = 1/4$. When M scale, $< \hat{\sigma}_I >$ and $std(\hat{I})$ have some oscillation. This gives us a lesson that we should not use the $g(x) = x^{-\alpha}$ and $p(x) = e^{-x}$ when we want to integral from zero to infinite at $\alpha = 3/4$.

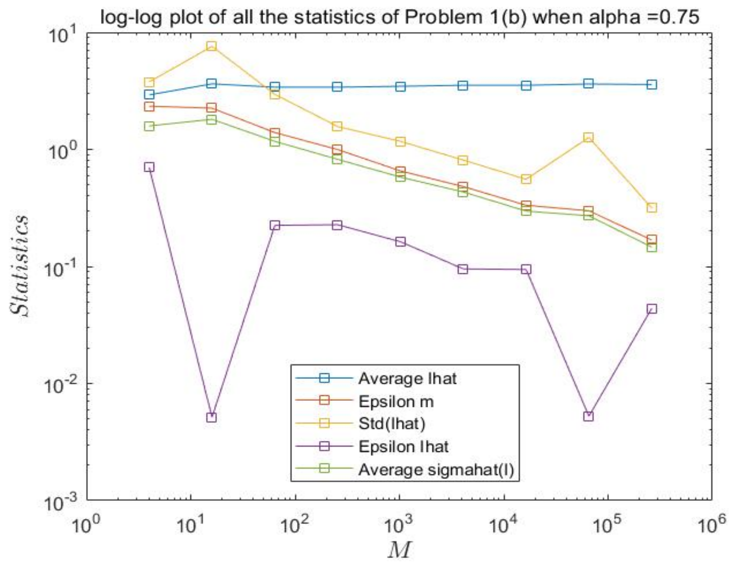

Figure 9: log-log plot of all the measured statistics of Problem 1(b) when $\alpha = 3/4$

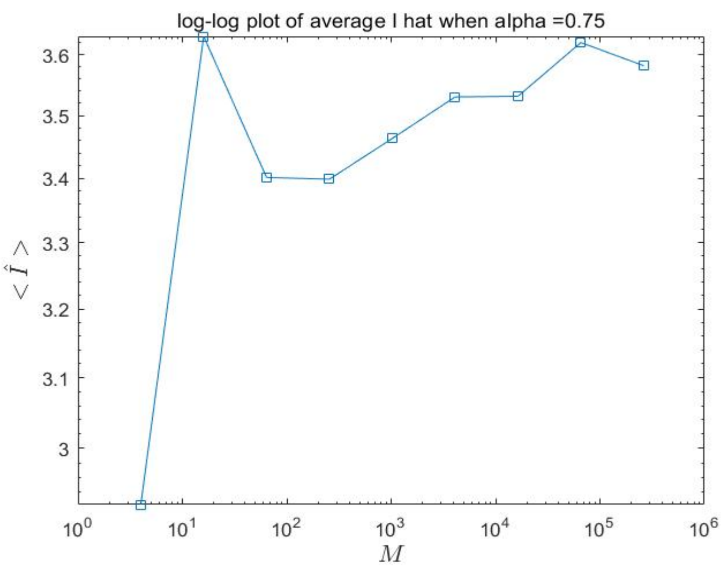

Figure 10: log-log plot of average $\hat{I}$, $< \hat{I} >$, when $\alpha = 3/4$

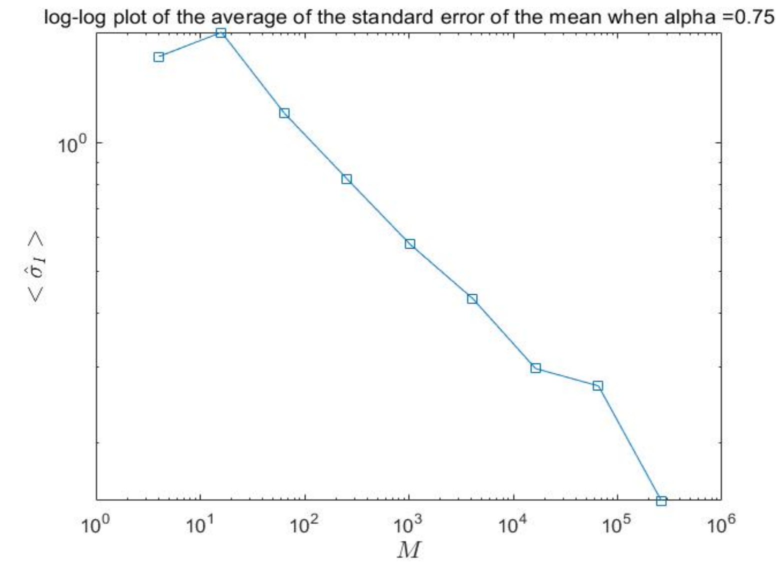

Figure 11: log-log plot of the average of the standard error of the mean, $< \hat{\sigma_I} >$, when $\alpha = 3/4$

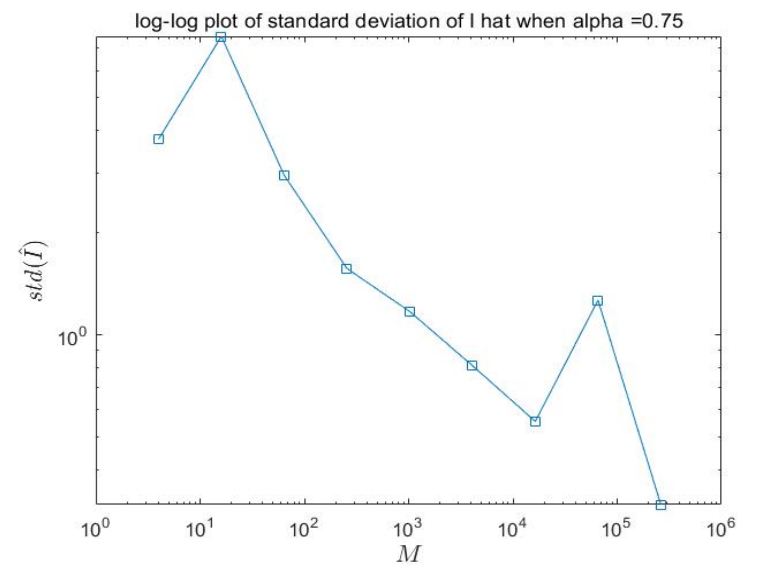

Figure 12: log-log plot of the standard deviation of $\hat{I}$, $std(\hat{I})$, when $\alpha = 3/4$

1.3 Problem 1(c)

Now let's use Monte-Carlo integration for the integral as the Problem 1(b) and change the integration interval from 0 to $x_{max} = 2$ and set $\alpha = 3/4$ and $g(x) = e^{-x}$ and $p(x) = x^{-\alpha}$. To calculate the exact integration of $\int_0^{x_{\max}} x^{-\alpha} e^{-x} dx$, using $MATLAB$ we get $I_{exact} = 3.562937573$. From Figure (13) we can see that all the statistics performs like the similar pattern discussed in Problem 1(a) and this is a successful Monte-Carlo Integration attempt and by increasing M, we can get a more accurate integration as we want.

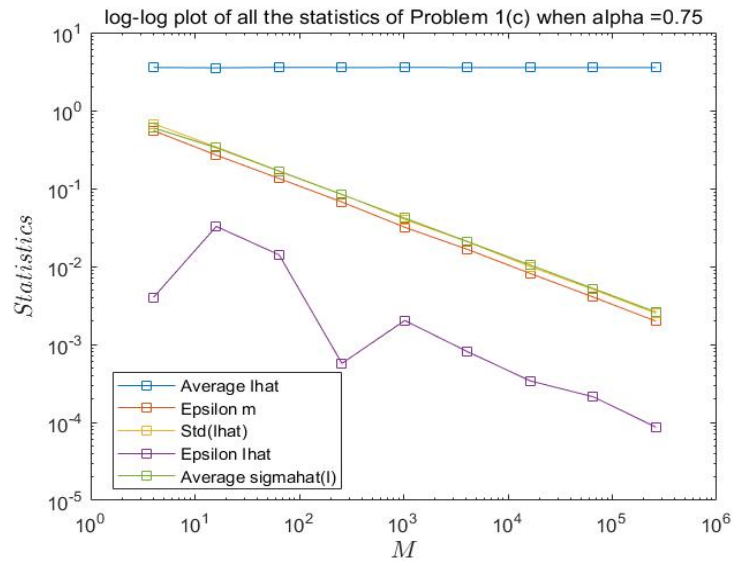

Figure 13: log-log plot of all the measured statistics of Problem 1(c)

From Figure (14) we can see that, with the increasing of M, we can see that the $< \hat{I} >$ will be stable at the I_{exact}, which is approximately 3.563.

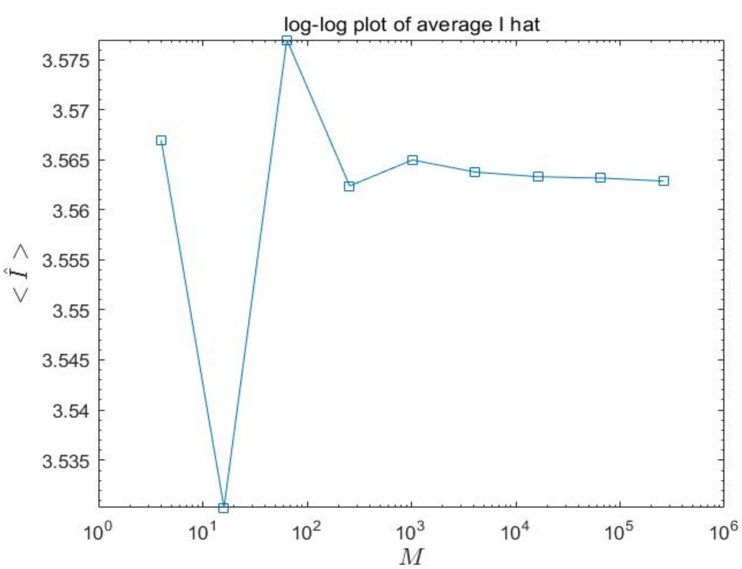

Figure 14: log-log plot of average $\hat{I}$, $< \hat{I} >$, when $x_{max} = 2$ and $\alpha = 3/4$

Now let's look at the $< \hat{\sigma}_I >$. From Figure (15) we can see that there is linear relationship between M and $< \hat{\sigma}_I >$ and by the $ployfit$ from $MATLAB$, I find that the relationship between M and $< \hat{\sigma}_I >$ is

$$log_{10}(< \hat{\sigma}_I >) = -0.4946 * log_{10}(M) + 0.1042$$

For $std(\hat{I})$ we can get the same conclusion as $< \hat{\sigma}_I >$ from Figure (16) and by the $ployfit$ from $MATLAB$, I find that the relationship between M and $std(\hat{I})$ is

$$log_{10}(std(\hat{I})) = -0.5061 * log_{10}(M) + 0.1384$$

Be careful that the parameter may change when you run the code again, but this equation give you some approximation relationship about how these statistics scale with M.

The results tell us that when we want to integrate $\int_0^{x_{\max}} x^{-\alpha} e^{-x} dx$, where x_{max} is a specific value, the choice of $g(x) = e^{-x}$ and $p(x) = x^{-\alpha}$ will be a good choice.

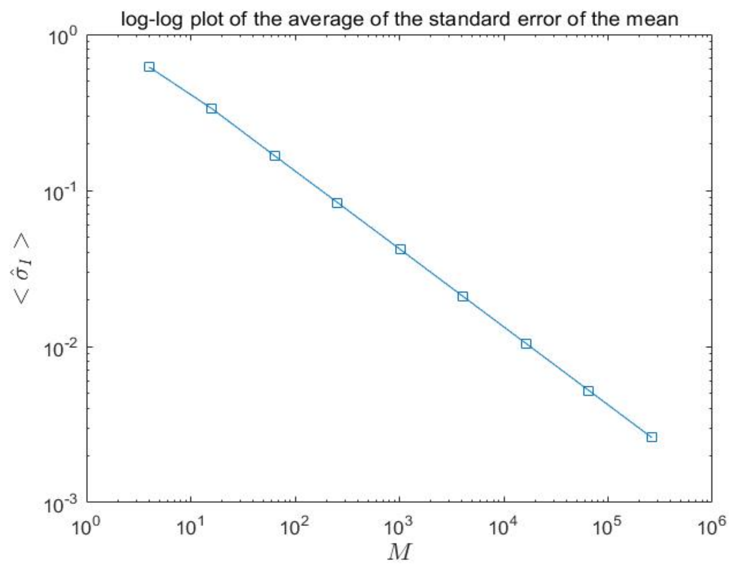

Figure 15: log-log plot of the average of the standard error of the mean, $< \hat{\sigma_I} >$, when $x_{max} = 2$ and $\alpha = 3/4$

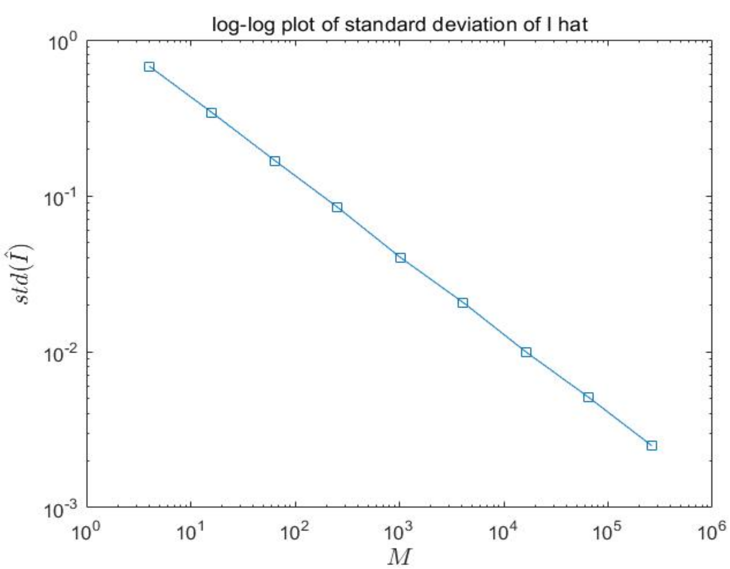

Figure 16: log-log plot of the standard deviation of $\hat{I}$, $std(\hat{I})$, when $x_{max} = 2$ and $\alpha = 3/4$

1.4 Problem 1(d)

Based on my results in parts 1b and 1c, when $\alpha = 3/4$, I may first compute the integral in part 1b as

$$\int_0^{x_{\max}} x^{-\alpha} e^{-x} dx,$$

where I set x_{max} to be a specific value and use $g(x) = e^{-x}$ and $p(x) = x^{-\alpha}$, which have been shown successful in part 1c. And then for the remaining part of the

$$\int_{x_{\max}}^{\infty} x^{-\alpha} e^{-x} dx,$$

I may use $g(x) = x^{-\alpha}$ and $p(x) = e^{-x}$ so that I can avoid the case that the $x^{-\alpha}$ to be infinite when $x = 0$.

2. Problem 2

For the Problem 2, we are going to do Phase Transition in the Ising Model. Be careful that when we changing the temperature, we should use the final state of the previous temperature as the initial condition since at hope if we choose a good t_{corr}, then we are expected to get an equilibrium by doing $N_{trials} * t_{torr}$ for previous temperature, and this state will a good initial condition for the next temperature.

2.1 Problem 2(a)

For $L = 25$,$t_{corr} = 200$, $N_{trials} = 400$, $H = 0$, $J = 1$, here I give the result of Engergy per spin $< U(\tilde{T}) >$ as shown in Figure (17), Specific Heat as shown in Figure (18), Magnetization per spin $m(\tilde{T})$ as shown in Figure (19), Susceptibility as shown in Figure (20), Faction of accepted spin flips as shown in Figure(21) and when $temperature = 3, 2.5, 2, 1.5$ the corresponding snapshots of the final spin configuration $\vec{\sigma}$. From the Figure (17) we can see that as temperature decrease, the energy per spin will decrease as well, which is the same as what we expected since when the temperature decrease, the system is more confined to energy minimal. From Figure (18) we can see that there is a peak in the middle of the temperature and it also the same as what we expected. From the Figure (19), we can conclude that when the temperature is decreasing, the magnetization per spin is increasing, this is also the same as what we want since we can visualize this result from Figure(25) that we can see the final state at temperature of 1.5, each lattice point are going to be the same states except only a few of them are different. From Figure (20) we can see that there is also a peak near $temperature = 2.4$ and the highest value of the susceptibility is 12 approximately. From Figure (21) we can see that the Fraction of accepted spin flips decrease from 0.5 to 0 nearly, which is the same as what we expect since at the beginning, is nearly 1/2 to flips a coins and it becomes harder to accept spin flips as the temperature decreasing since the lower the temperature the lower is the probability that a spin flip is accepted if it were to raise the energy: the system is more confined to energy minimal and less likely to climb over energy barriers. From Figure (22), (23), (24) and (25) we can see that as the temperature decrease, the state of the whole system are tending to the energy minimal.

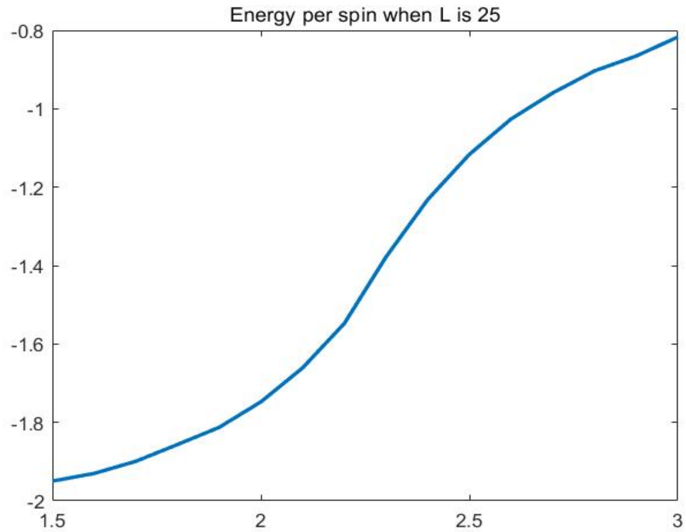

Figure 17: Energy per spin when $L = 25$

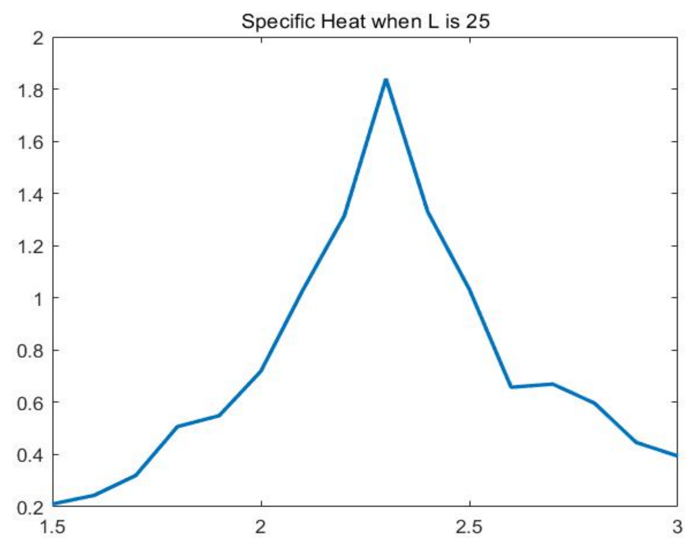

Figure 18: Specific Heat when $L = 25$

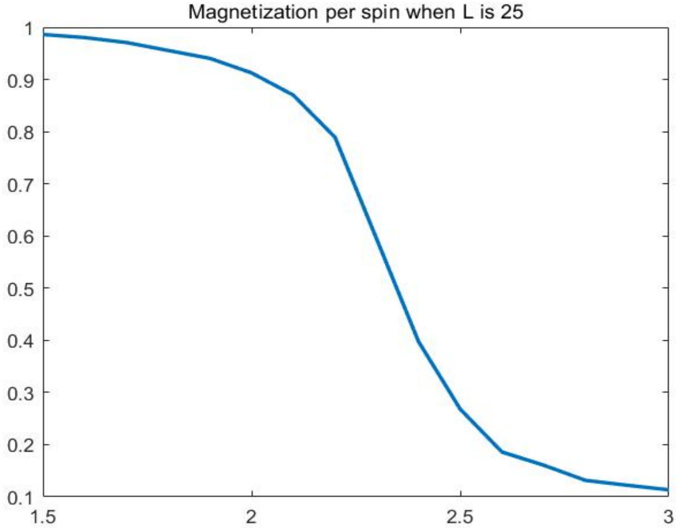

Figure 19: Magnetization per spin when $L = 25$

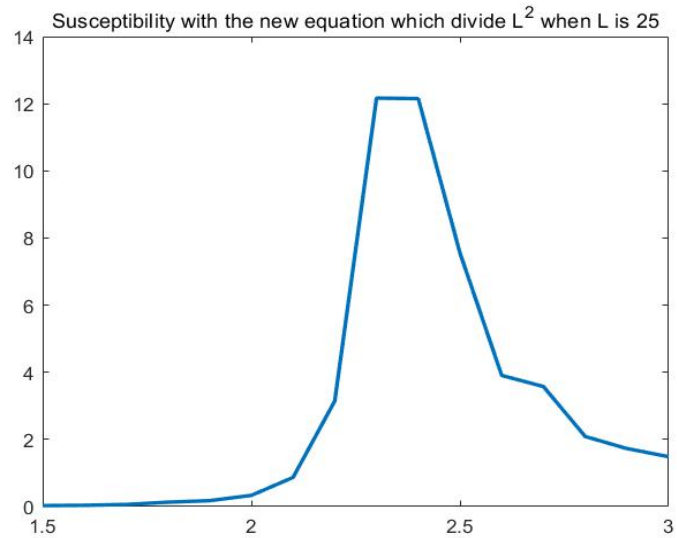

Figure 20: susceptibility when $L = 25$

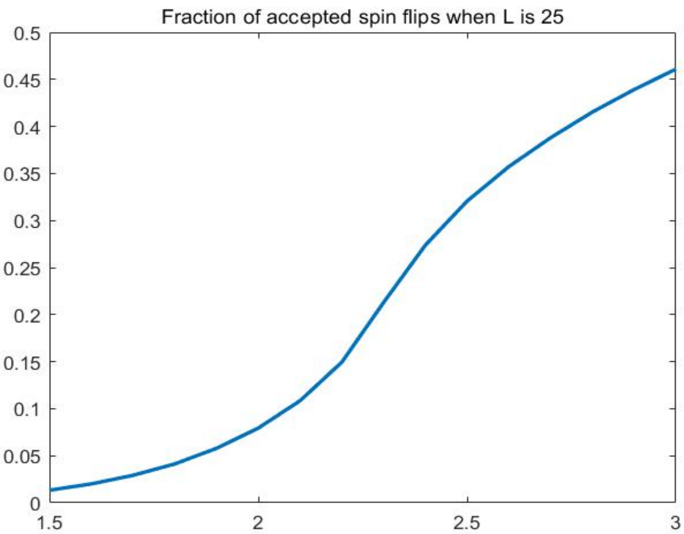

Figure 21: Fraction of accepted spin flips when $L = 25$

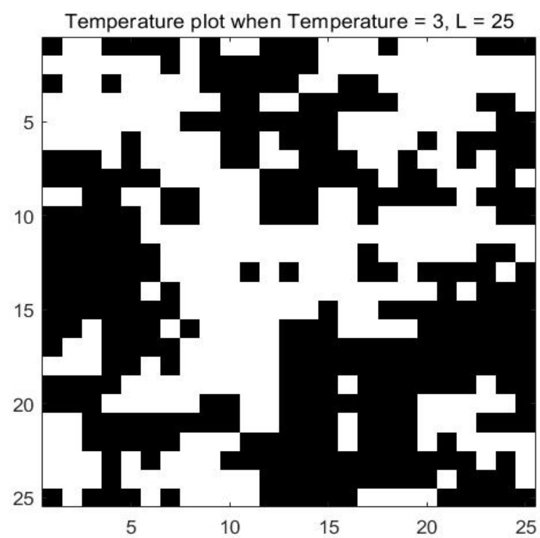

Figure 22: State snapshot when $Temperature = 3$, $L = 25$

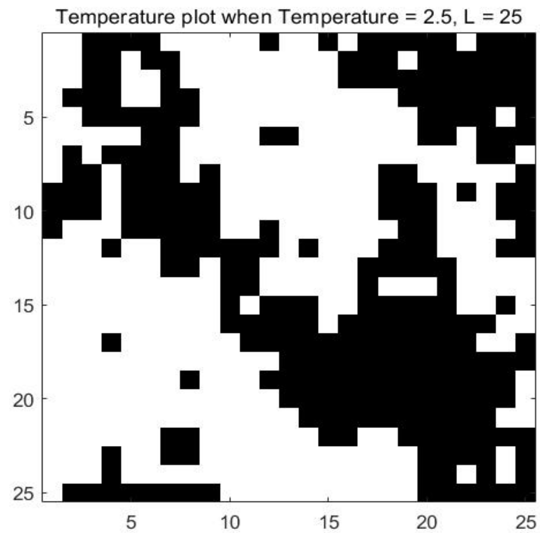

Figure 23: State snapshot when $Temperature = 2.5$, $L = 25$

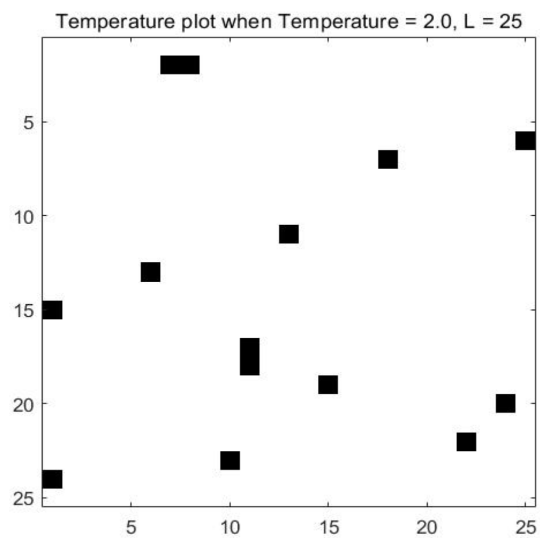

Figure 24: State snapshot when $Temperature = 2$, $L = 25$

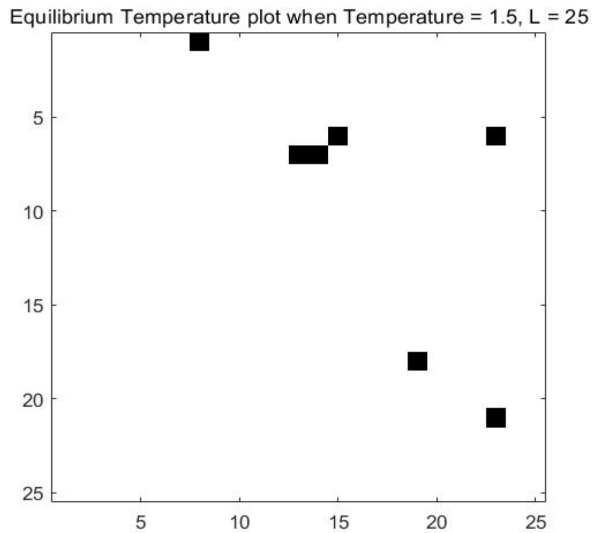

Figure 25: State snapshot when $Temperature = 1, 5$, $L = 25$

2.2 Problem 2(b)

My computations for $L = 25$ do show that there is a trend that for $\tilde{T} > \tilde{T}_c$, the magnetization decrease very fast and go to near zero, as shown in Figure (19), and we also see that there is a trend in Figure (18) that there will be diverges at $\tilde{T}_c$, where $\tilde{T}_c$ is approximately near 2.4. To get more convincing confirmation, I increase the $L = 125$ and get the results of Figure(26) and (27). From Figure (26) we can see that there is an obvious fast decrease when $\tilde{T}_c$ is near 2.4 and go to near zero, and from Figure (27) we see that the max susceptibility value become near to 200 when $L = 125$. We can imagine that when L goes to infinite, the results shown in L.Onsager's paper will happen that $m(\tilde{T}) = 0$ and the susceptibility diverges at $\tilde{T}_c$.

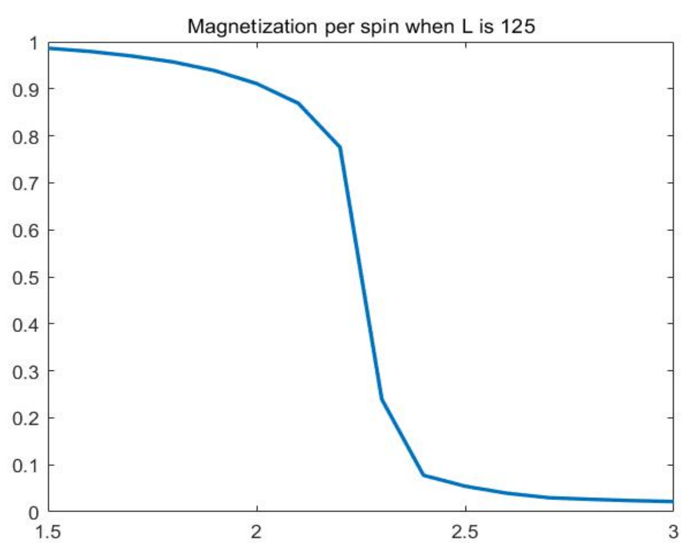

Figure 26: Magnetization per spin when $L = 125$

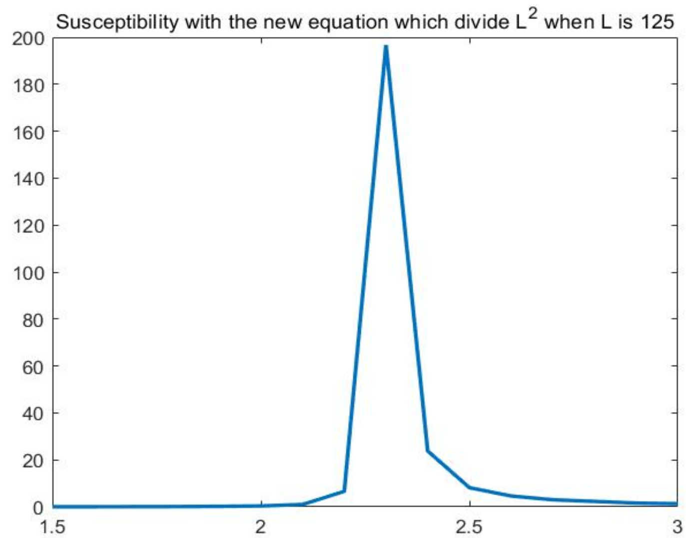

Figure 27: Susceptibility when $L = 125$

3. Part 1 MATLAB code

3.1 Problem 1(a)

```
%%%%%% author: Mingfu Liang
%%%%%% date: 02/27/2019
%%%%%% Problem 1(a)

%%%%%% initialize the parameters and the matrix used for storing the
%%%%%% statistics that we need to measure

N_trials=500;
p=9;
I_exact = 1/2;
I_hat_matrix = zeros(p,N_trials);
Sigma_hat_I = zeros(p,N_trials);
Sigma_absoluate_error = zeros(p,N_trials);
Sigma_std_error =  zeros(p,N_trials);
Std_var_I_hat = zeros(p,N_trials);

epsilon_I_hat_matrix = zeros(p,1);
epsilon_m_matrix = zeros(p,1);
average_I_hat=zeros(p,1);
Sigma_hat_I_mean = zeros(p,1);
Std_I_hat=zeros(p,1);
```

```
M_range=zeros(p,1);
abs_error_matrix=zeros(p,N_trials);

%%% To see how three quantities average of I_hat, average of Delta_hat and
%%% average of std(I_hat) scale with M, it is equal to see how they sacle
%%% with the p since M = 4^p with p up to 9

for i =1:p
    M = 4^i;
    M_range(i,1) = M;
    for trials =1:N_trials
     %%% use M=4^i sample points X_j for the Monte-Carlo integration of each of the integral
     %%% to estimate I_hat

     X_j = rand(1,M);
     X_hat = -log(1-X_j);
     g_X_j= cos(X_hat);

     I_hat = sum(g_X_j)/M;
     Abs_val_error = abs(I_hat-I_exact);
     abs_error=(g_X_j-I_hat).^2;
     abs_error_2=(g_X_j-I_exact).^2;
     std_error_mean =sqrt(sum(abs_error,2)/(M*(M-1))); % estimate the standard error of the mean

     abs_error_matrix(i,trials)=sqrt(sum(abs_error_2,2)/(M*M));
     I_hat_matrix(i,trials)=I_hat;
     Sigma_absoluate_error(i,trials)=Abs_val_error;
     Sigma_hat_I(i,trials)=std_error_mean;
    end

    %%% calculate average_{\hat{I}}, <\hat{\sigma{I}}>, std(\hat{I}),
    %%% \epsilon_{m} and \epsilon_{\hat{I}}
    average_I_hat(i,1) = mean(I_hat_matrix(i,:),2);
    Sigma_hat_I_mean(i,1)=mean(Sigma_hat_I(i,:),2);
    Std_I_hat(i,1)=std(I_hat_matrix(i,:));
    epsilon_I_hat_matrix(i,1)= abs(mean(I_hat_matrix(i,:),2)-I_exact);
    epsilon_m_matrix(i,1) = mean(Sigma_absoluate_error(i,:),2);

```

```
end

%%%%%% Evaluate the relationship between M and average_{\hat{I}}, <\hat{\sigma{I}}>, std(\hat{I})
p_I=polyfit(log10(M_range),log10(average_I_hat),1);
p_Std_I_hat=polyfit(log10(M_range),log10(Std_I_hat),1);
p_Sigma_hat_I_mean=polyfit(log10(M_range),log10(Sigma_hat_I_mean),1);

%%%%%% Visualize the result
figure;
loglog(M_range,average_I_hat,'-s');
hold on
loglog(M_range,epsilon_m_matrix,'-s');
hold on
loglog(M_range,Std_I_hat,'-s');
hold on
loglog(M_range,epsilon_I_hat_matrix,'-s');
hold on
loglog(M_range,Sigma_hat_I_mean,'-s');
legend('Average Ihat','Epsilon m','Std(Ihat)','Epsilon Ihat','Average sigmahat(I)')
xlabel('$M$','Interpreter','latex','FontSize',13)
ylabel('$Statistics$','Interpreter','latex','FontSize',13)
mytitle1 = ['log-log plot of all the statistics of Problem 1(a)'];
title(mytitle1);

figure;
loglog(M_range,average_I_hat,'-s');
mytitle4 = 'log-log plot of average I hat';
title(mytitle4);
xlabel('$M$','Interpreter','latex','FontSize',13)
ylabel('$<\hat{I}>$','Interpreter','latex','FontSize',13)

figure;
loglog(M_range,Sigma_hat_I_mean,'-s');
mytitle3 = 'log-log plot of the average of the standard error of the mean';
title(mytitle3);
xlabel('$M$','Interpreter','latex','FontSize',13)
ylabel('$<\hat{\sigma}_{I}>$','Interpreter','latex','FontSize',13)

```

```
figure;
loglog(M_range,Std_I_hat,'-s');
mytitle2 = 'log-log plot of standard deviation of I hat';
title(mytitle2);
xlabel('$M$','Interpreter','latex','FontSize',13)
ylabel('$std(\hat{I})$','Interpreter','latex','FontSize',13)
```

3.2 Problem 1(b)

```
%%%%%% author: Mingfu Liang
%%%%%% date: 02/27/2019
%%%%%% Problem 1(b)

%%%%%% initialize the parameters and the matrix used for storing the
%%%%%% statistics that we need to measure

N_trials=500;
p=9;
alpha = 3/4;
I_exact = gamma(1-alpha);

I_hat_matrix = zeros(p,N_trials);
Sigma_hat_I = zeros(p,N_trials);
Sigma_absoluate_error = zeros(p,N_trials);
Sigma_std_error =  zeros(p,N_trials);
Std_var_I_hat = zeros(p,N_trials);

epsilon_I_hat_matrix = zeros(p,1);
epsilon_m_matrix = zeros(p,1);
average_I_hat=zeros(p,1);
Sigma_hat_I_mean = zeros(p,1);
Std_I_hat=zeros(p,1);
M_range=zeros(p,1);
abs_error_matrix=zeros(p,N_trials);

%%% To see how three quantities average of I_hat, average of Delta_hat and
%%% average of std(I_hat) scale with M, it is equal to see how they sacle
%%% with the p since M = 4^p with p up to 9

for i =1:p
    M = 4^i;
```

```
        M_range(i,1) = M;
        for trials =1:N_trials
         %%% use M=4^i sample points X_j for the Monte-Carlo integration of each of the integral
         %%% to estimate I_hat
         X_j = rand(1,M);
         X_hat = -log(1-X_j);
         g_X_j= X_hat.^(-alpha);

         I_hat = sum(g_X_j)/M;
         Abs_val_error = abs(I_hat-I_exact);
         abs_error=(g_X_j-I_hat).^2;
         abs_error_2=(g_X_j-I_exact).^2;
         std_error_mean =sqrt(sum(abs_error,2)/(M*(M-1))); % estimate the standard error of the mean

         abs_error_matrix(i,trials)=sqrt(sum(abs_error_2,2)/(M*M));
         I_hat_matrix(i,trials)=I_hat;
         Sigma_absoluate_error(i,trials)=Abs_val_error;
         Sigma_hat_I(i,trials)=std_error_mean;
        end

        %%% calculate average_{\hat{I}}, <\hat{\sigma{I}}>, std(\hat{I}),
        %%% \epsilon_{m} and \epsilon_{\hat{I}}

        average_I_hat(i,1) = mean(I_hat_matrix(i,:),2);
        Sigma_hat_I_mean(i,1)=mean(Sigma_hat_I(i,:),2);
        Std_I_hat(i,1)=std(I_hat_matrix(i,:));
        epsilon_I_hat_matrix(i,1)= abs(mean(I_hat_matrix(i,:),2)-I_exact);
        epsilon_m_matrix(i,1) = mean(Sigma_absoluate_error(i,:),2);

end

%%%%%% Evaluate the relationship between M and average_{\hat{I}}, <\hat{\sigma{I}}>, std(\hat{I})
p_I=polyfit(log10(M_range),log10(average_I_hat),1);
p_Std_I_hat=polyfit(log10(M_range),log10(Std_I_hat),1);
p_Sigma_hat_I_mean=polyfit(log10(M_range),log10(Sigma_hat_I_mean),1);

%%%%%% Visualize the result
```

```
figure;
loglog(M_range,average_I_hat,'-s');
hold on
loglog(M_range,epsilon_m_matrix,'-s');
hold on
loglog(M_range,Std_I_hat,'-s');
hold on
loglog(M_range,epsilon_I_hat_matrix,'-s');
hold on
loglog(M_range,Sigma_hat_I_mean,'-s');
legend('Average Ihat','Epsilon m','Std(Ihat)','Epsilon Ihat','Average
    sigmahat(I)')
xlabel('$M$','Interpreter','latex','FontSize',13)
ylabel('$Statistics$','Interpreter','latex','FontSize',13)
mytitle1 = ['log-log plot of all the statistics of Problem 1(b) when alpha =
    ', num2str(alpha)];
title(mytitle1);

figure;
loglog(M_range,average_I_hat,'-s');
mytitle4 = ['log-log plot of average I hat when alpha =', num2str(alpha)];
title(mytitle4);
xlabel('$M$','Interpreter','latex','FontSize',13)
ylabel('$<\hat{I}>$','Interpreter','latex','FontSize',13)

figure;
loglog(M_range,Sigma_hat_I_mean,'-s');
mytitle3 = ['log-log plot of the average of the standard error of the mean
    when alpha =', num2str(alpha)];
title(mytitle3);
xlabel('$M$','Interpreter','latex','FontSize',13)
ylabel('$<\hat{\sigma}_{I}>$','Interpreter','latex','FontSize',13)

figure;
loglog(M_range,Std_I_hat,'-s');
mytitle2 = ['log-log plot of standard deviation of I hat when alpha =',
    num2str(alpha)];
title(mytitle2);
xlabel('$M$','Interpreter','latex','FontSize',13)
ylabel('$std(\hat{I})$','Interpreter','latex','FontSize',13)
```

3.3 Problem 1(c)

```
%%%%%% author: Mingfu Liang
%%%%%% date: 02/27/2019
%%%%%% Problem 1(c)

%%%%%% initialize the parameters and the matrix used for storing the
%%%%%% statistics that we need to measure

N_trials=500;
p=9;
alpha = 3/4;
x_max = 2;
I_exact = 3.562937573;

I_hat_matrix = zeros(p,N_trials);
Sigma_hat_I = zeros(p,N_trials);
Sigma_absoluate_error = zeros(p,N_trials);
Sigma_std_error =  zeros(p,N_trials);
Std_var_I_hat = zeros(p,N_trials);

epsilon_I_hat_matrix = zeros(p,1);
epsilon_m_matrix = zeros(p,1);
average_I_hat=zeros(p,1);
Sigma_hat_I_mean = zeros(p,1);
Std_I_hat=zeros(p,1);
M_range=zeros(p,1);

%%% To see how three quantities average of I_hat, average of Delta_hat and
%%% average of std(I_hat) scale with M, it is equal to see how they sacle
%%% with the p since M = 4^p with p up to 9

for i =1:p
    M = 4^i;
    M_range(i,1) = M;
    for trials =1:N_trials
     %%% use M=4^i sample points X_j for the Monte-Carlo integration of each
         of the integral
     %%% to estimate I_hat

```

```
        X_j = rand(1,M);
        Integral_alpha = 4*(x_max^(1-alpha));
        X_hat = (Integral_alpha*(1-alpha)*X_j).^(4);
        g_X_j= Integral_alpha*exp(-X_hat);

        I_hat = sum(g_X_j)/M;
        Abs_val_error = abs(I_hat-I_exact);
        abs_error=(g_X_j-I_hat).^2;
        std_error_mean =sqrt(sum(abs_error,2)/(M*(M-1))); % estimate the standard error of the mean

        I_hat_matrix(i,trials)=I_hat;
        Sigma_absoluate_error(i,trials)=Abs_val_error;
        Sigma_hat_I(i,trials)=std_error_mean;
      end

    %%% calculate average_{\hat{I}}, <\hat{\sigma{I}}>, std(\hat{I}),
    %%% \epsilon_{m} and \epsilon_{\hat{I}}

    average_I_hat(i,1) = mean(I_hat_matrix(i,:),2);
    Sigma_hat_I_mean(i,1)=mean(Sigma_hat_I(i,:),2);
    Std_I_hat(i,1)=std(I_hat_matrix(i,:));
    epsilon_I_hat_matrix(i,1)= abs(mean(I_hat_matrix(i,:),2)-I_exact);
    epsilon_m_matrix(i,1) = mean(Sigma_absoluate_error(i,:),2);

end

%%%%%% Evaluate the relationship between M and average_{\hat{I}}, <\hat{\sigma{I}}>, std(\hat{I})
p_I=polyfit(log10(M_range),log10(average_I_hat),1);
p_Std_I_hat=polyfit(log10(M_range),log10(Std_I_hat),1);
p_Sigma_hat_I_mean=polyfit(log10(M_range),log10(Sigma_hat_I_mean),1);

%%%%%% Visualize the result
figure;
loglog(M_range,average_I_hat,'-s');
hold on
loglog(M_range,epsilon_m_matrix,'-s');
hold on
loglog(M_range,Std_I_hat,'-s');
```

```
hold on
loglog(M_range,epsilon_I_hat_matrix,'-s');
hold on
loglog(M_range,Sigma_hat_I_mean,'-s');
legend('Average Ihat','Epsilon m','Std(Ihat)','Epsilon Ihat','Average
    sigmahat(I)')
xlabel('$M$','Interpreter','latex','FontSize',13)
ylabel('$Statistics$','Interpreter','latex','FontSize',13)
mytitle1 = ['log-log plot of all the statistics of Problem 1(c) when alpha =
    ',num2str(alpha)];
title(mytitle1);

figure;
loglog(M_range,average_I_hat,'-s');
mytitle4 = 'log-log plot of average I hat';
title(mytitle4);
xlabel('$M$','Interpreter','latex','FontSize',13)
ylabel('$<\hat{I}>$','Interpreter','latex','FontSize',13)

figure;
loglog(M_range,Sigma_hat_I_mean,'-s');
mytitle3 = 'log-log plot of the average of the standard error of the mean';
title(mytitle3);
xlabel('$M$','Interpreter','latex','FontSize',13)
ylabel('$<\hat{\sigma}_{I}>$','Interpreter','latex','FontSize',13)

figure;
loglog(M_range,Std_I_hat,'-s');
mytitle2 = 'log-log plot of standard deviation of I hat';
title(mytitle2);
xlabel('$M$','Interpreter','latex','FontSize',13)
ylabel('$std(\hat{I})$','Interpreter','latex','FontSize',13)
```

3.4 Problem 2

```
%%%%%% author: Mingfu Liang
%%%%%% date: 02/27/2019
%%%%%% Problem 2 Phase Transition in the Ising Model
tic

%%%%%%%%% initialize the parameter %%%%%%%%%%%%%%%%%
```

```
H=0; % H is non-dimensionalized version of B
J=1; % J is non-dimensionalized coupling strength
L=25; % Consider a two-dimensional system of spin 1/2 magnetic dipoles
    arrayed on a square L*L
L_2 = L^2; % The size of the one Monte-Carlo step, L_2 is the number of
    spins in the system
t_corr = 200; % fisrt establish equilibrium by taking t_corr Monte-Carlo
    steps as intial configuration
N_trials = 400; % take N_trials * t_corr Monte-Carlo steps
T_high = 3; % The highest value in the relevant range of the temperature
T_low =1.5; % The lowest value in the relevant range of the temperature
delta_T = -0.1; % The change of temperature in each steps
T_range =zeros(16,1);% Initialize the range of temperature


% Generate the time range which should be [1.5,3.0]
h=1;
for T = T_high:delta_T:T_low
    T_range(h,1)=T;
    h=h+1;
end

posit = 1: L; % define the index variables
up_shift = circshift(posit,1); % shift the variables up one unit
down_shift = circshift(posit,-1); % shift the variables down one unit
counter =0; % count the total iteration, using for debug
T_init = T_high; % To calculate the initialization configuration

%%%%%%%% Initialize the M_{Sigma}, U_{Sigma} and the flip counter matrix
M_Sigma= zeros(16,400);
U_Sigma_matrix=zeros(16,400);
U_Sigma = zeros(16,1);
U_Sigma_Square =zeros(16,400);
flip_tol = zeros(16,1);


%%%%%%%%% Metropolis Algorithm using the Ising model %%%%%%

%%%%%%%%% Then take N_trials*t_corr Monte-Carlo steps and measure
%%%%%%%%% the expectation values listed below by sampling the corresponding
```

```
    quantities every
%%%%%%%%%% t_corr Monte-Carlo steps.
temp_count = 1;
for T = T_high:delta_T:T_low
    %%%%%%%%%% first establish equilibrium by taking t_corr Monte-Carlo
    %%%%%%%%%% steps using an initial configuration

            if exist('Sigma_mat')==0
                Sigma_mat = randsrc(L); % generate square matrix L*L and each lattice point (i,j) the dipole is only 1 or -1
            end
            for k = 1:1:t_corr
                    [i_ind,j_ind]=ind2sub([L,L],randperm(L^2)); % in each Monte-Carlo step generate a random permutation of all spin indices
                for select_ind = 1: L_2 % perform one Monte-Carto step with L*L attempted spin flips
                i_select = i_ind(select_ind);
                j_select = j_ind(select_ind);
                Sigma_select_old = Sigma_mat(i_select, j_select);
                Sigma_select_new = -1*Sigma_mat(i_select, j_select);% propose the candidate spin flip
                Delta_U = 2*H*Sigma_select_old + 2*J*Sigma_select_old*(Sigma_mat(up_shift(i_select), j_select)+Sigma_mat(down_shift(i_select), j_select)+Sigma_mat(i_select, up_shift(j_select))+Sigma_mat(i_select, down_shift(j_select)));
                accept_prob = min(exp(-Delta_U/T),1);
                U = rand;
                        if U < accept_prob
                            Sigma_mat(i_select,j_select)=Sigma_select_new;
                        end
                end
            end

        % initialize the accepted flip counter
        flip_count =0;

        for N=1:N_trials

```

```
    for t = 1:t_corr
        [i_ind,j_ind]=ind2sub([L,L],randperm(L^2)); % in each Monte-Carlo step generate a random permutation of all spin indices
        for select_ind = 1: L_2 % perform one Monte-Carto step with L*L attempted spin flips
                i_select = i_ind(select_ind);
                j_select = j_ind(select_ind);
                % define the Sigma_{select_old} for energy
                % difference calculation
                Sigma_select_old = Sigma_mat(i_select,j_select);
                % propose the candidate spin flip
                Sigma_select_new = -1*Sigma_mat(i_select,j_select);
                % Compute the energy difference Delta_{U} using (5.3)
                Delta_U = 2*H*Sigma_select_old+2*J*Sigma_select_old*(Sigma_mat(up_shift(i_select),j_select)+Sigma_mat(down_shift(i_select),j_select)+Sigma_mat(i_select,up_shift(j_select))+Sigma_mat(i_select,down_shift(j_select)));
                accept_prob = min(exp(-Delta_U/T),1);
                U = rand;
                if U < accept_prob
                    Sigma_mat(i_select,j_select)=Sigma_select_new;
                    flip_count = flip_count+1;
                end
        end
        counter = counter+1;
    end

    % Calculate M_Sigma and U_Sigma
    for select_ind = 1: L_2
    i_select = i_ind(select_ind);
    j_select = j_ind(select_ind);
    M_Sigma(temp_count,N)=M_Sigma(temp_count,N)+ Sigma_mat(i_select,j_select);
    U_Sigma(temp_count)=U_Sigma(temp_count)-J*Sigma_mat(i_select,j_select)*(Sigma_mat(up_shift(i_select),j_select)+Sigma_mat(down_shift(i_select),j_select)+Sigma_mat(i_select,up_shift(j_select))+Sigma_mat(i_select,down_shift(j_select)))/2;
```

```
                U_Sigma_Square(temp_count,N)=U_Sigma_Square(temp_count,N)+(-J*
                    Sigma_mat(i_select,j_select)*(Sigma_mat(up_shift(i_select),
                    j_select)+Sigma_mat(down_shift(i_select),j_select)+Sigma_mat(
                    i_select,up_shift(j_select))+Sigma_mat(i_select,down_shift(
                    j_select)))/2);
                %U_Sigma_matrix(temp_count,N)=-J*Sigma_mat(i_select,j_select)*(
                    Sigma_mat(up_shift(i_select),j_select)+Sigma_mat(down_shift(
                    i_select),j_select)+Sigma_mat(i_select,up_shift(j_select))+
                    Sigma_mat(i_select,down_shift(j_select)))/2;
            end

        end
        flip_tol(temp_count,1)= flip_count;
        temp_count=temp_count+1;

        %%%%%%%%% Snapchat
        if T == 2
            figure;
            imagesc(Sigma_mat)
            colormap(gray(2))
            axis ij
            axis square
            mytitle5=['Temperature plot when Temperature = 2.0, L = ',
                num2str(L)];
            title(mytitle5);
        end
        if T == 2.5
            figure;
            imagesc(Sigma_mat)
            colormap(gray(2))
            axis ij
            axis square
            mytitle6=['Temperature plot when Temperature = 2.5, L = ',
                num2str(L)];
            title(mytitle6)
        end
        if T == 3
            figure;
            imagesc(Sigma_mat)
            colormap(gray(2))
```

```
            axis ij
            axis square
            mytitle11=['Temperature plot when Temperature = 3, L = ',num2str(L)];
            title(mytitle11)
        end
end

%%%%%%%% Equilibrium Temperature plot when Temperature = 1.5 %%%%%%%
figure;
imagesc(Sigma_mat)
colormap(gray(2))
axis ij
axis square
mytitle8=['Equilibrium Temperature plot when Temperature = 1.5, L = ', num2str(L)];
title(mytitle8)

%%%%%%%% measure the expectation values listed below by sampling the
%%%%%%%% corresponding quantities

%%% Fraction of accepted spin flips, here flip_tol() is the total number of
%%% spin flips of different temperature, which is 16*1 vector, then divide
%%% by the total attempt of spin flips
Filp_Faction_Accp = flip_tol()/(L_2*t_corr*N_trials);

%%% Define the M(Sigma), which is 16*400 matrix, each row has 400 sample
%%% obtained from specified temperature
%M_Sigma_vec = M_Sigma/L_2;

%%% Define U(T_tiuda) from samples
U_T_tiuda = U_Sigma/(L_2*N_trials);

%%% Define the Magnetization of whole state from samples
M_T_tiuda =sum(abs(M_Sigma),2)/N_trials;

%%% Define the Magnetization per spin from samples
m_T_tiuda = sum(abs(M_Sigma),2)/(N_trials*L_2);

%%% Define the <U(\tiuda(T))^2> from samples
```

```
U_T_tiuda_square= sum(U_Sigma_Square.^2,2)/(N_trials);

%%% Define the <M(\Sigma)^2> from samples
M_T_tiuda_square = sum(M_Sigma.^2,2)/N_trials;

%%% Define Sepcific Heat from samples
SepcificHeat = (U_T_tiuda_square - ((U_Sigma/N_trials).^2))./(L_2*T_range.^2);

%%% Define Susceptibility without divide L^2 from samples
Susceptibility = (M_T_tiuda_square-M_T_tiuda.^2)./T_range;

%%% Define the correct Susceptibility from samples
Susceptibility_L_2 = (M_T_tiuda_square-M_T_tiuda.^2)./(T_range*L_2);

%%% Visualize the statistics
figure;
plot(T_range,Susceptibility_L_2,'linewidth',2)
mytitle14 =['Susceptibility with the new equation which divide L^2 when L is ',num2str(L)];
title(mytitle14);
figure;
plot(T_range,U_T_tiuda,'linewidth',2)
mytitle1 =['Energy per spin when L is ',num2str(L)];
title(mytitle1);
figure;
plot(T_range,SepcificHeat,'linewidth',2)
mytitle2 =['Specific Heat when L is ',num2str(L)];
title(mytitle2);
figure;
plot(T_range,Susceptibility,'linewidth',2)
mytitle3 =['Susceptibility of whole state when L is ',num2str(L)];
title(mytitle3);
figure;
plot(T_range,m_T_tiuda,'linewidth',2)
mytitle4 =['Magnetization per spin when L is ',num2str(L)];
title(mytitle4);
figure;
plot(T_range,Filp_Faction_Accp,'linewidth',2)
mytitle10 =['Fraction of accepted spin flips when L is ',num2str(L)];
```

```
title(mytitle10);
toc
```