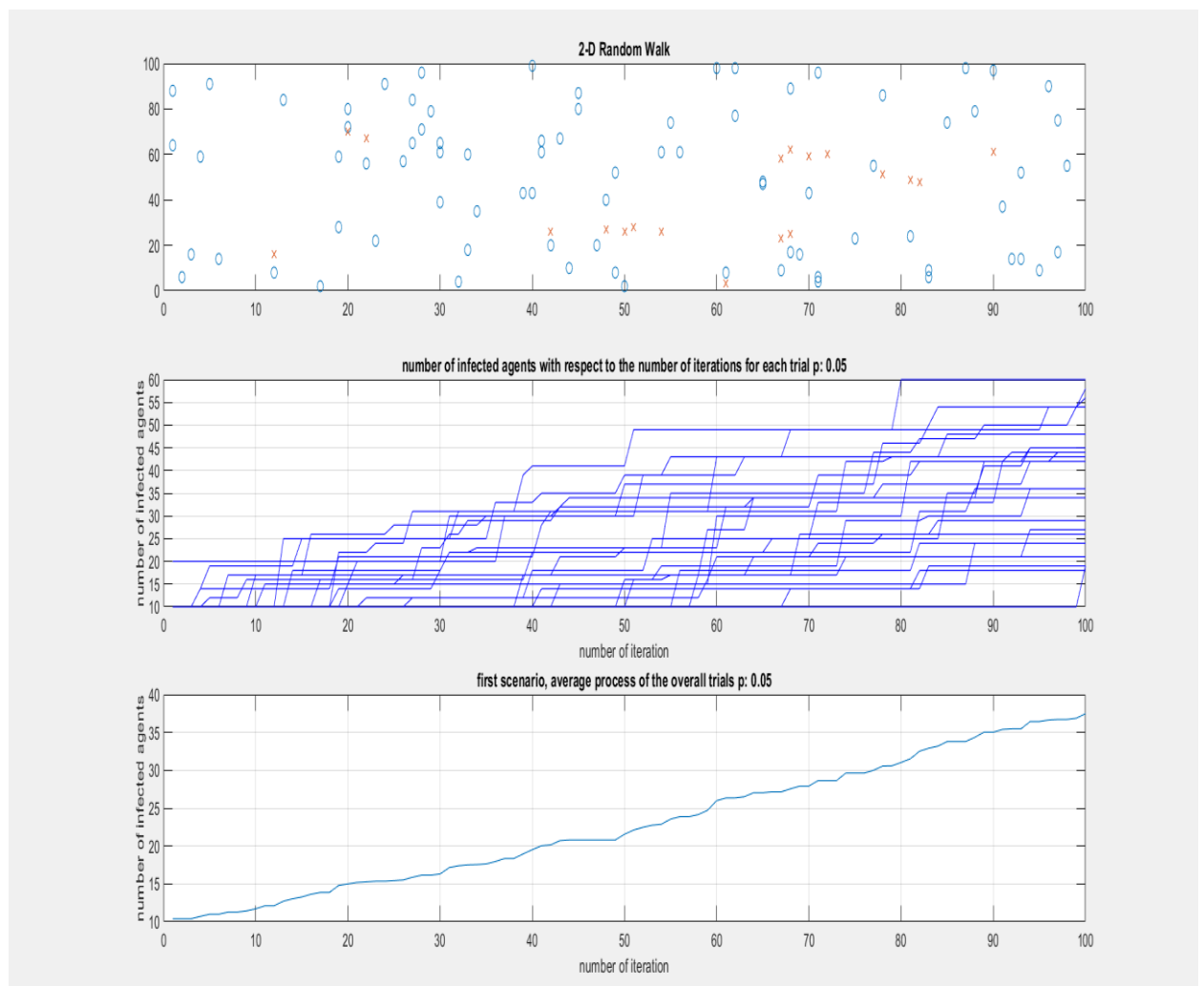


OSMAN KAAAN KURTÇA 040160090

## RANDOM SIGNALS AND NOISE – TERM PROJECT

### 1) Normal Case. First scenario

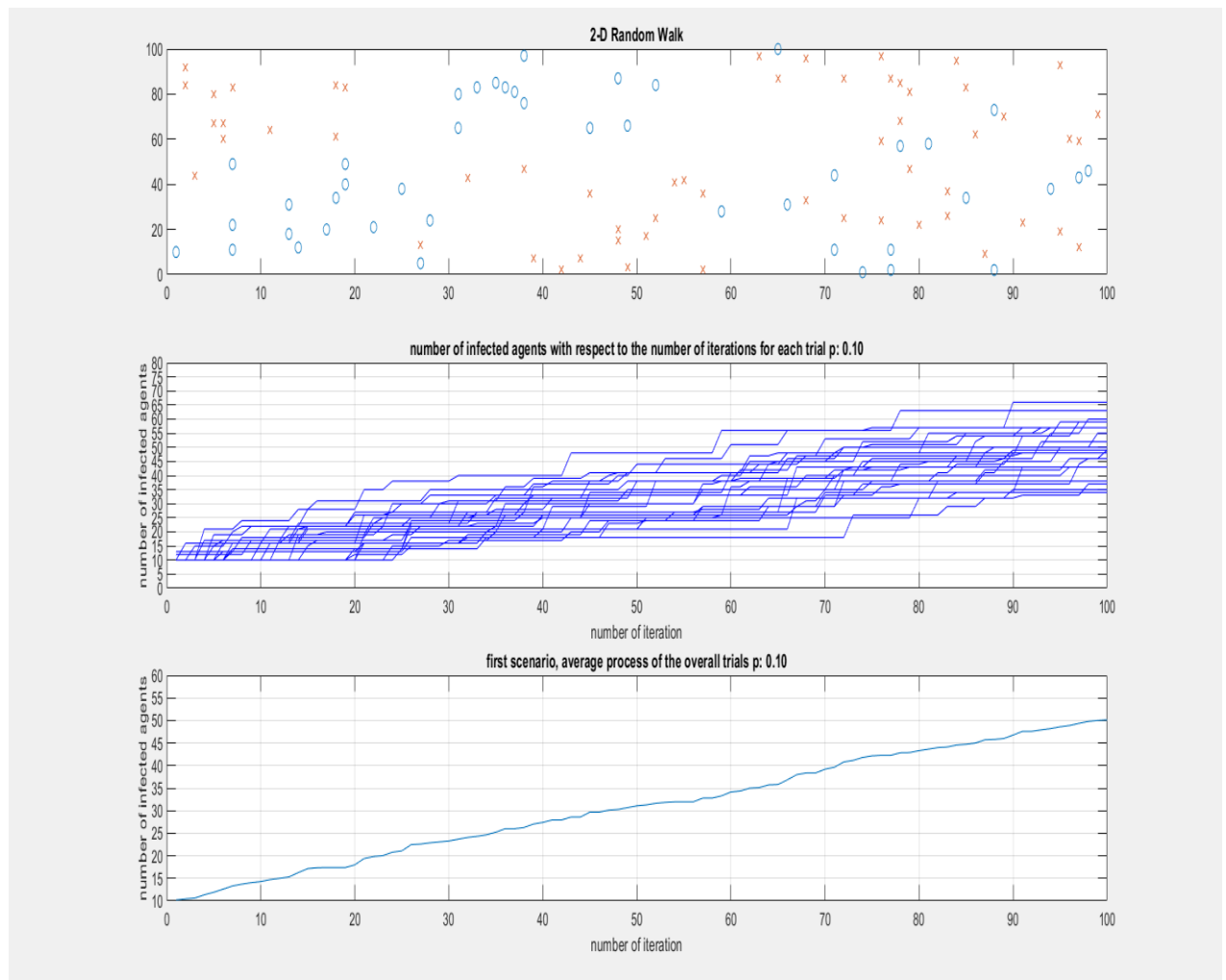
P=0.05



The second graph shows the change of the number of infected agents for 25 iterations.

The third graph shows the average change for the 25 iterations.

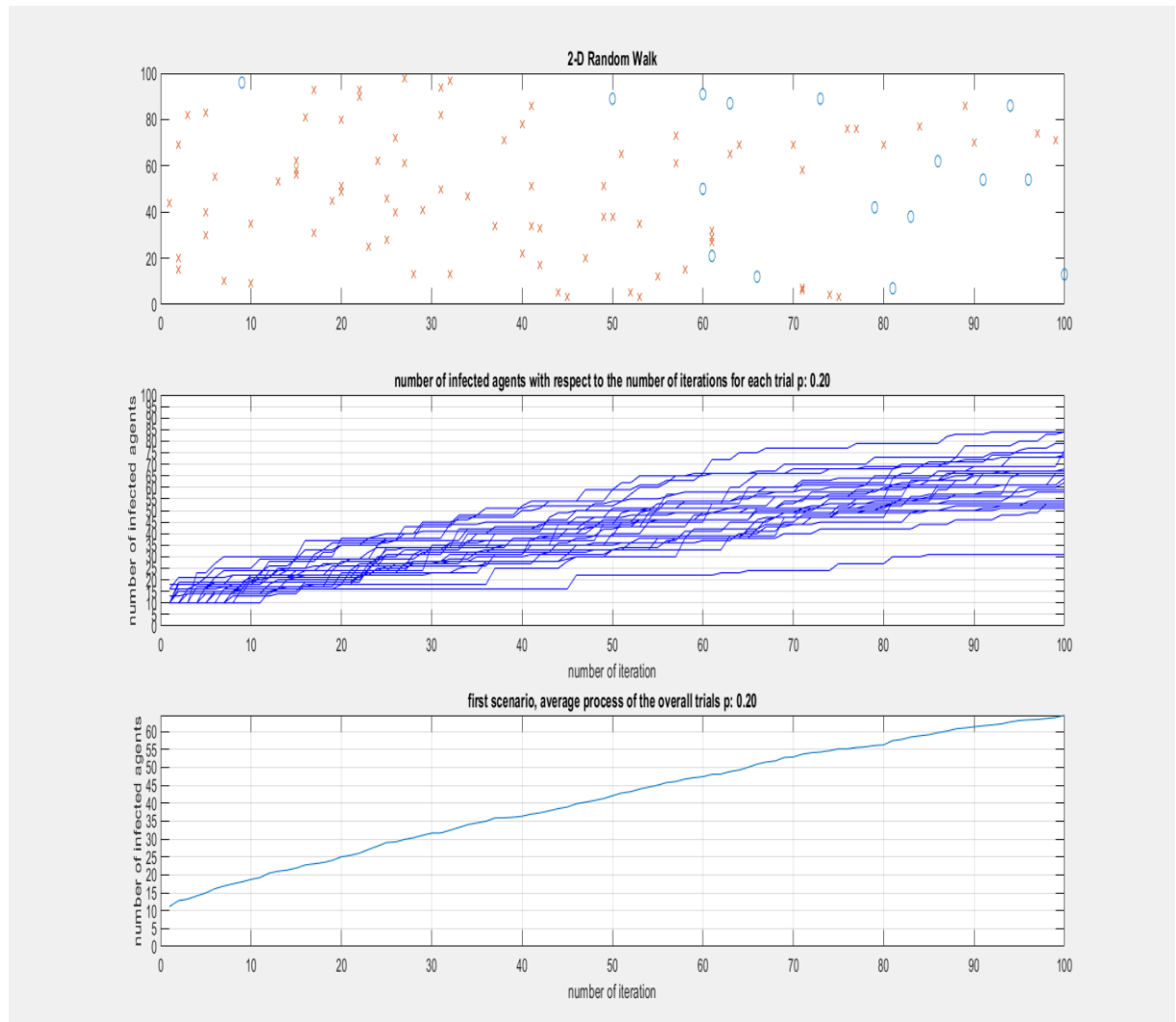
P=0.1



The second graph shows the change of the number of infected agents for 25 iterations.

The third graph shows the average change for the 25 iterations.

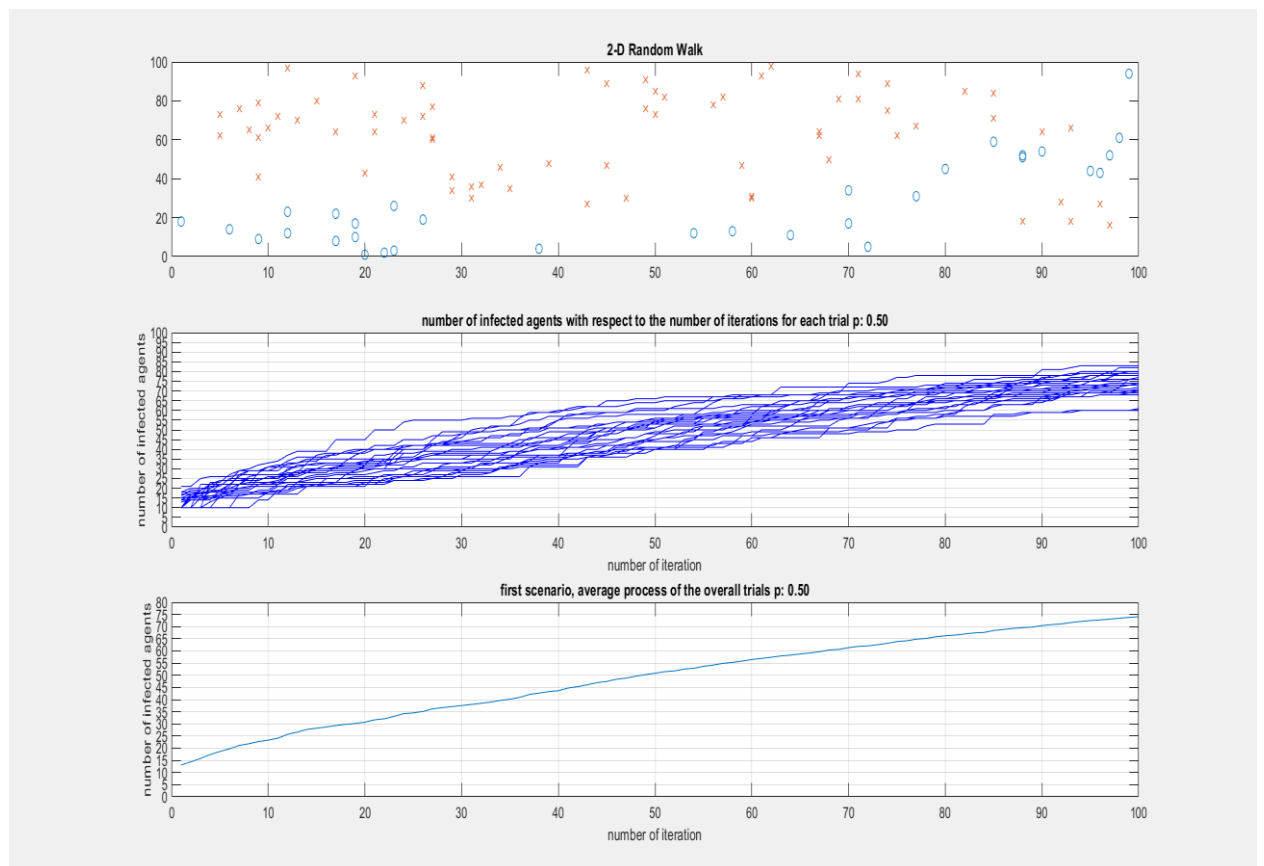
P=0.2



The second graph shows the change of the number of infected agents for 25 iterations.

The third graph shows the average change for the 25 iterations.

P=0.5

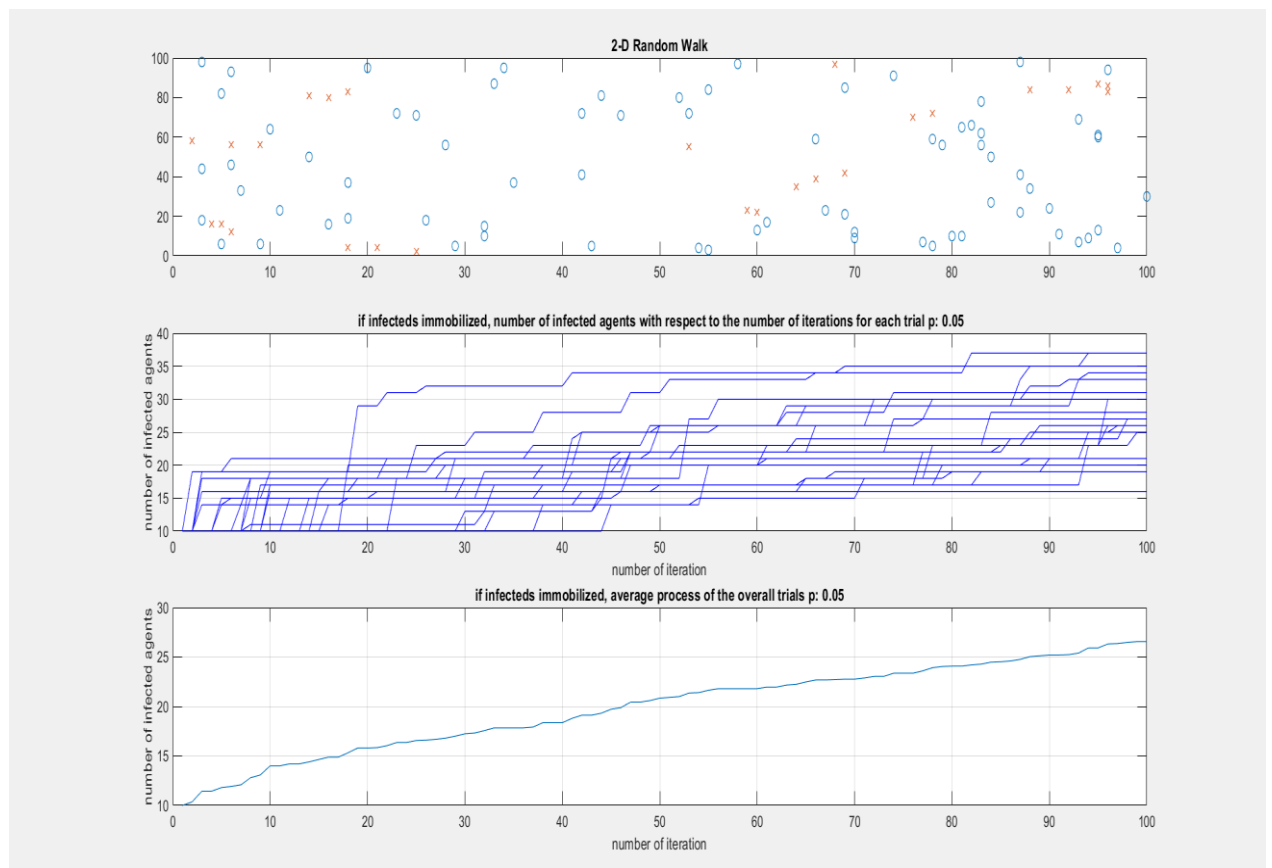


The second graph shows the change of the number of infected agents for 25 iterations.

The third graph shows the average change for the 25 iterations.

## 2) When infecteds are immobilized. Second scenario

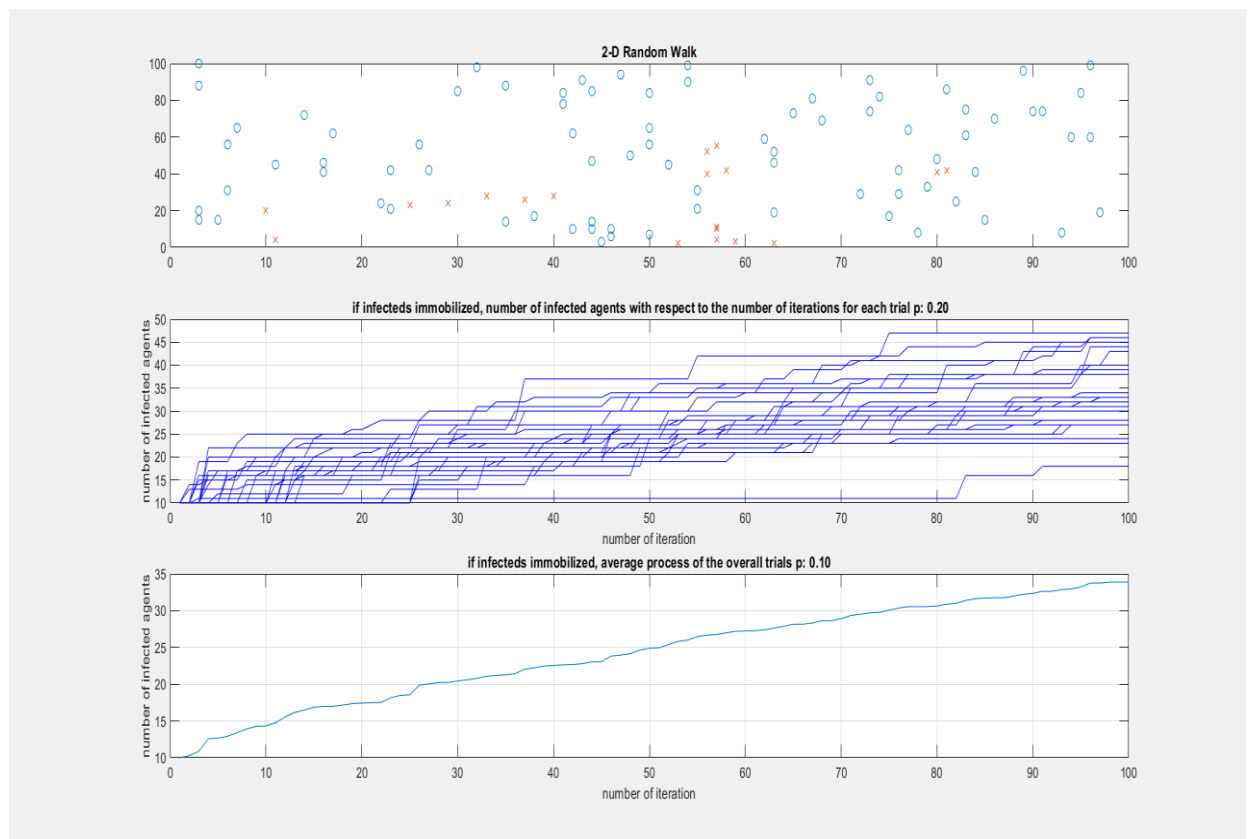
P=0.05



The second graph shows the change of the number of infected agents for 25 iterations.

The third graph shows the average change for the 25 iterations.

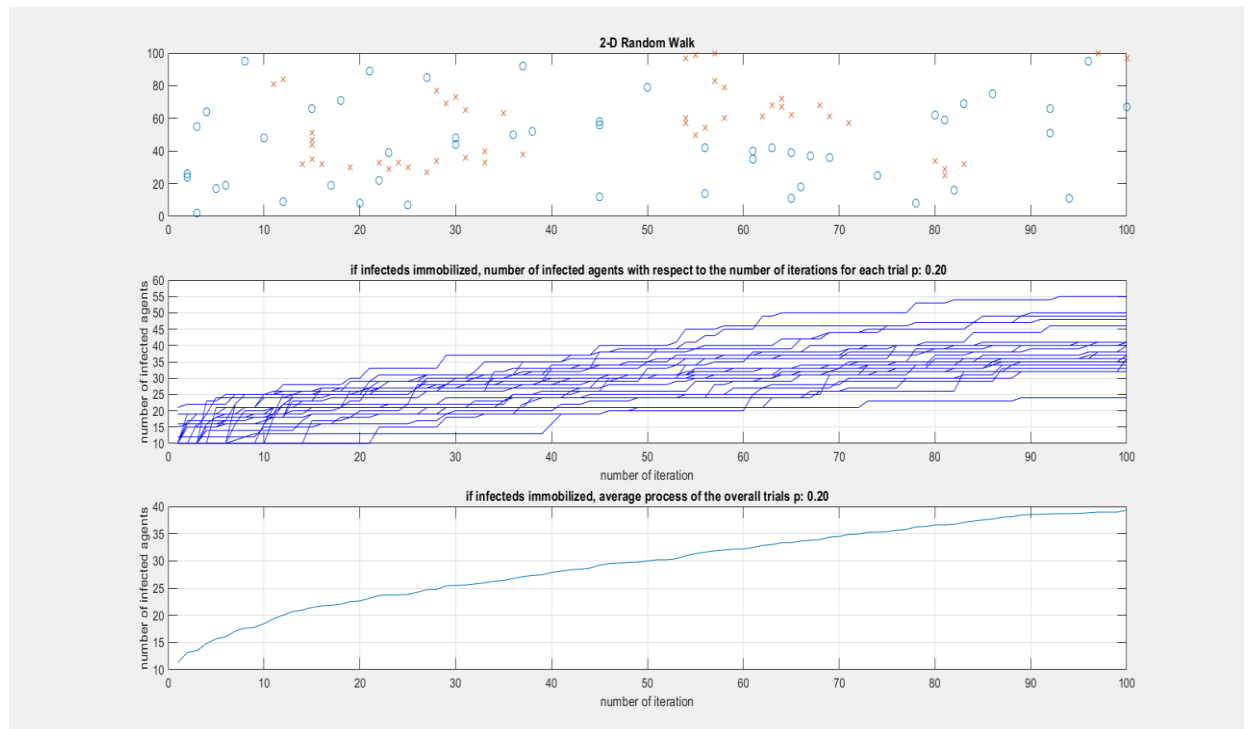
P=0.1



The second graph shows the change of the number of infected agents for 25 iterations.

The third graph shows the average change for the 25 iterations.

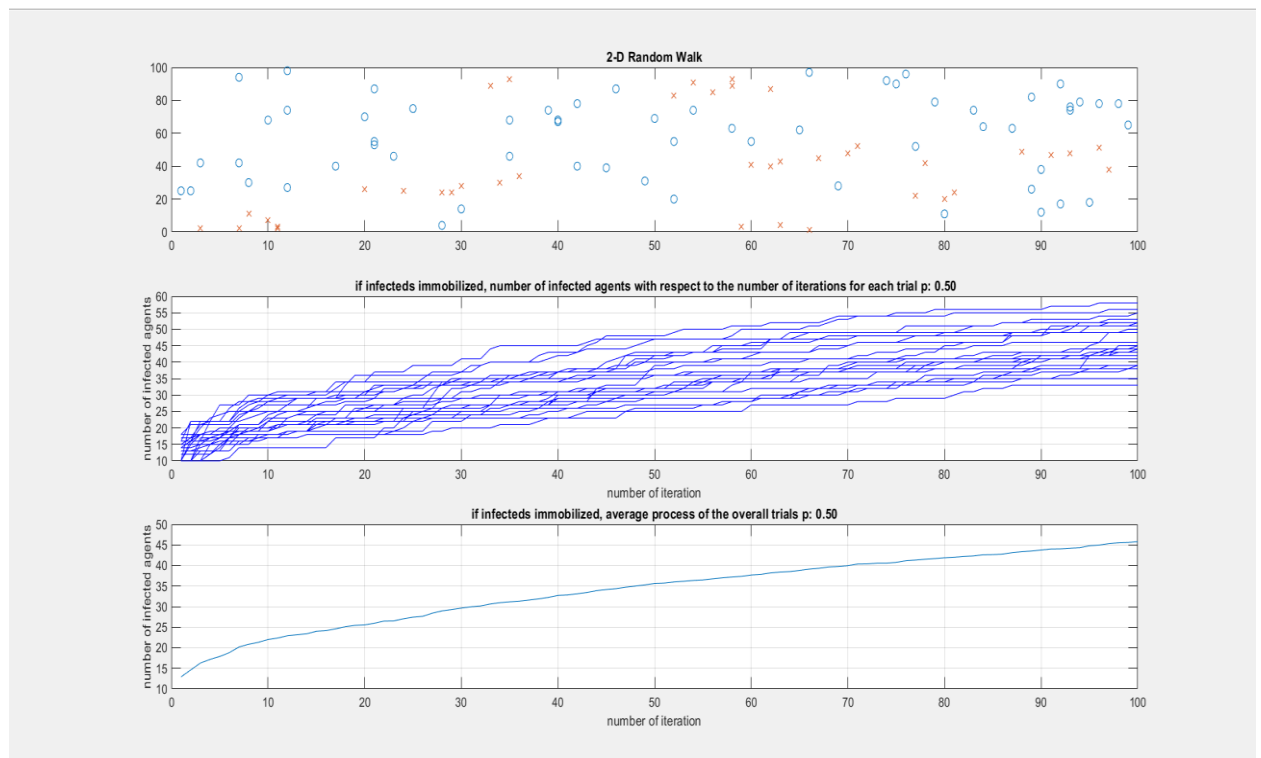
P=0.2



The second graph shows the change of the number of infected agents for 25 iterations.

The third graph shows the average change for the 25 iterations.

P=0.5



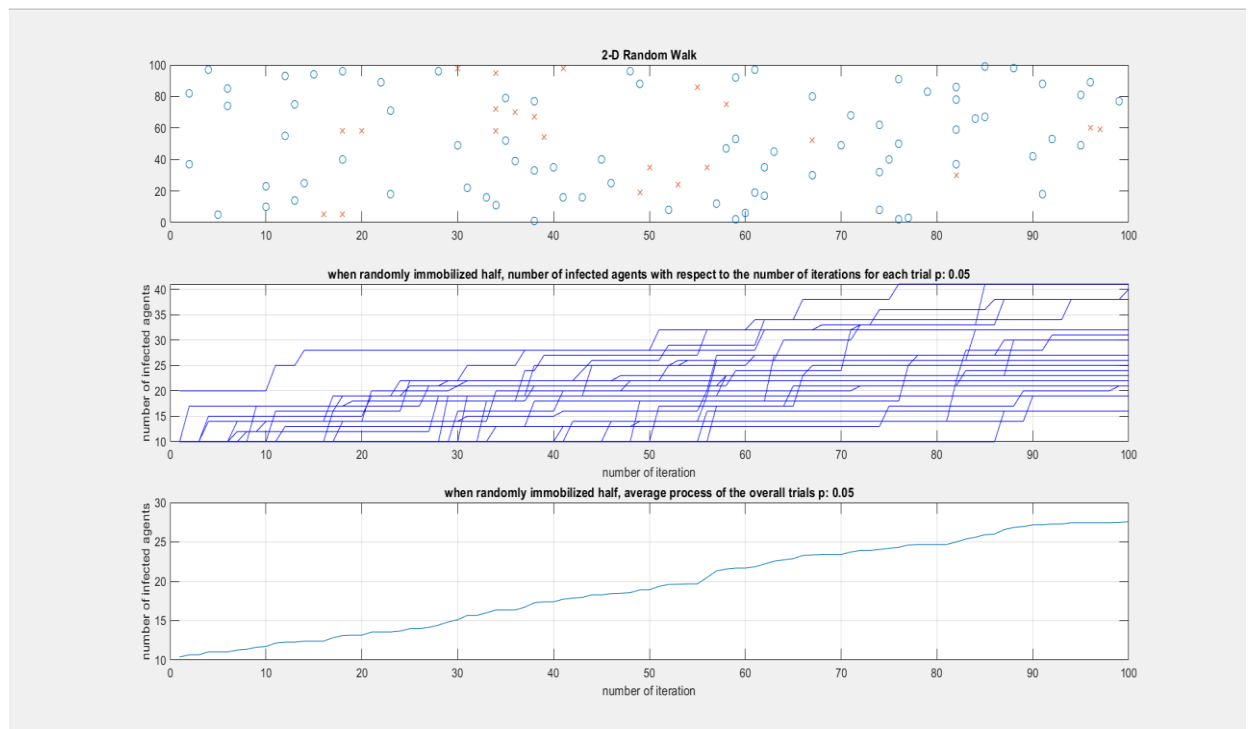
The second graph shows the change of the number of infected agents for 25 iterations.

The third graph shows the average change for the 25 iterations.



3) When randomly half of the agents are immobilized. Third scenario

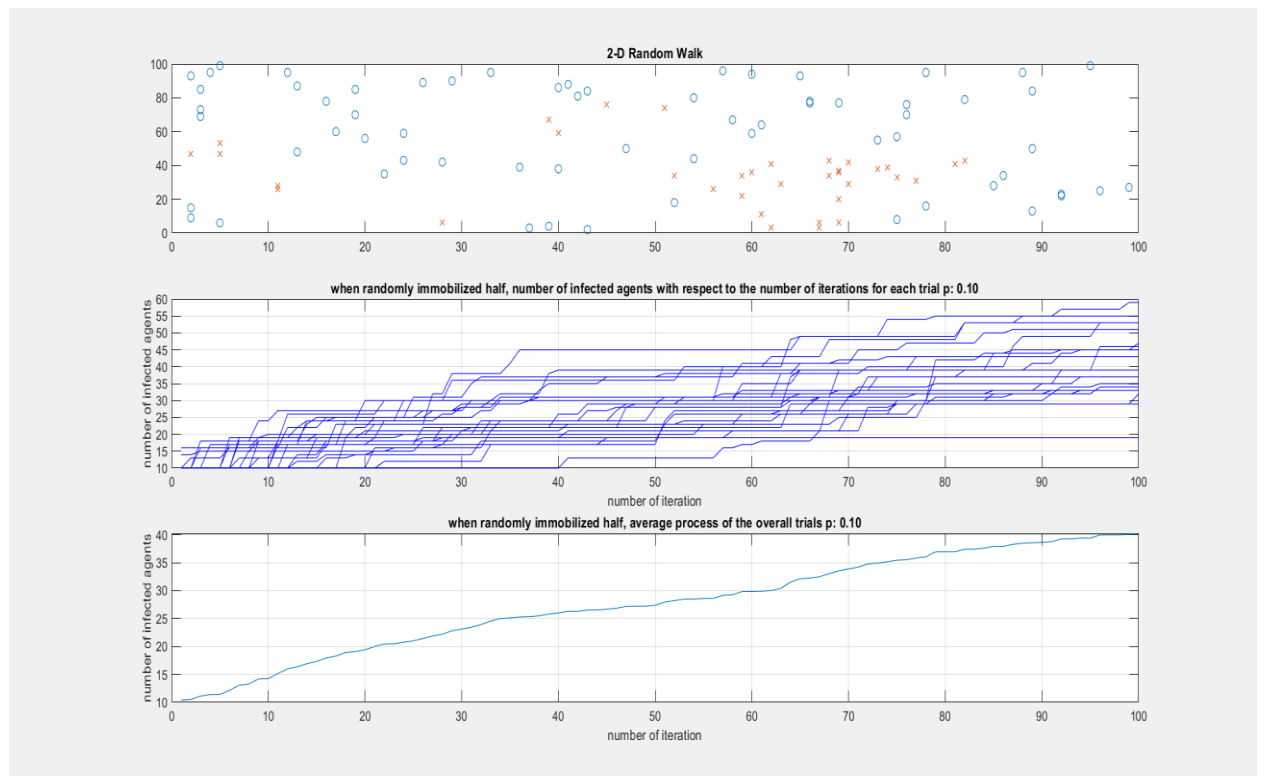
P=0.05



The second graph shows the change of the number of infected agents for 25 iterations.

The third graph shows the average change for the 25 iterations.

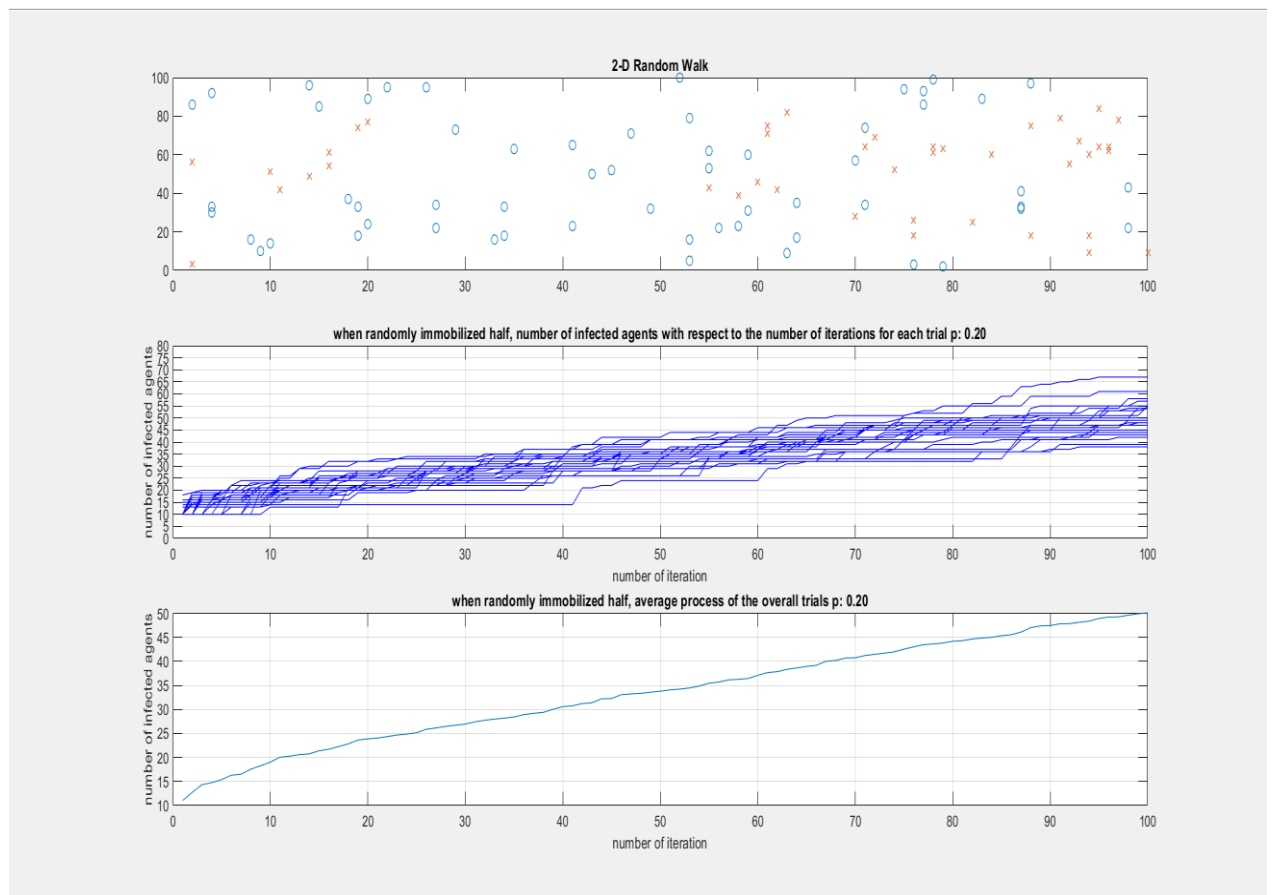
P=0.1



The second graph shows the change of the number of infected agents for 25 iterations.

The third graph shows the average change for the 25 iterations.

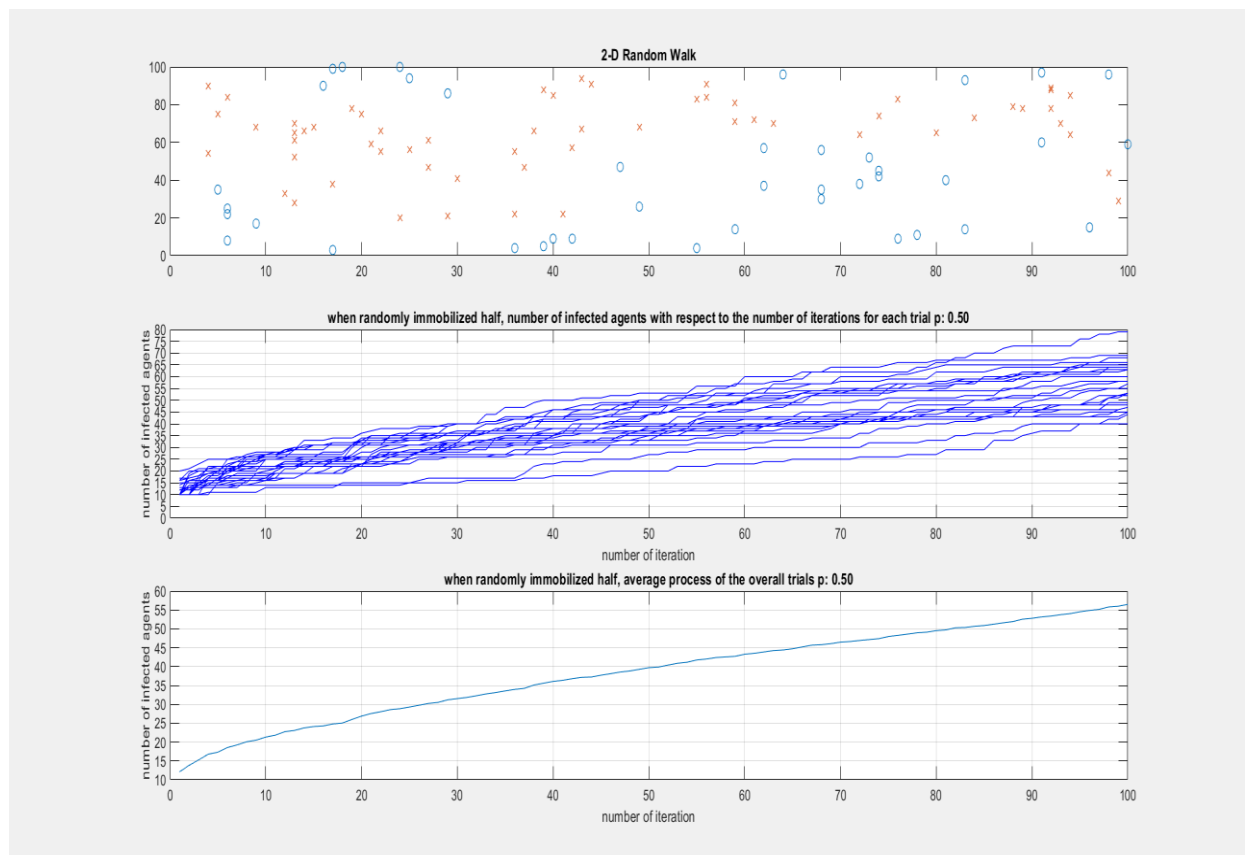
P=0.2



The second graph shows the change of the number of infected agents for 25 iterations.

The third graph shows the average change for the 25 iterations.

P=0.5



The second graph shows the change of the number of infected agents for 25 iterations.

The third graph shows the average change for the 25 iterations.

4-Compare the results that are obtained for these three scenarios in terms of the average processes.

If we look average process of all 3 scenarios for same probabilities, We can say that the worst case is first scenario. After 100 random walk steps, we reach the most infecteds at first scenario because all agents are moved per iteration. On the other hand; if we randomly choose 50 agents and we immobilize them, we observe that number of infected agents are less than first scenario. Since the number of moving agents decreased by half according to the initial situation, the virus infection rate also decreased. Finally, In second scenario, if we immobilize the infected agents, we reach the best case. Since infecteds are immobilized, the probabilities of healthy agents violating the social distance rule are reduced. Therefore; final infecteds number is least for this scenario.

5- Regarding the obtained average results, determine whether these are stationary processes or not.

When we look at the average processes, we see that processes' mean and variance parameters change over time. So that, these are not stationary processes.

6-For all scenarios and all  $p$  probabilities, We plot the scattering of both infected and healthy agents at first subplot.

# PROJECT CODES

```
clc;clear all;close all;

% normal_simulation(0.05)
% pause(1);
% normal_simulation(0.1)
% pause(1);
% normal_simulation(0.2)
% pause(1);
% normal_simulation(0.5)
% pause(1);
% immobilized_infecteds_simulation(0.05)
% pause(1);
% immobilized_infecteds_simulation(0.1)
% pause(1);
% immobilized_infecteds_simulation(0.2)
% pause(1);
% immobilized_infecteds_simulation(0.5)
% pause(1);
% randomly_half_simulation(0.05)
% pause(1);
% randomly_half_simulation(0.1)
% pause(1);
% randomly_half_simulation(0.2)
% pause(1);
randomly_half_simulation(0.5)
pause(1);

function normal_simulation(p)
figure;
iternumb=25;
totalnumofinf=zeros(1,100);
for i=1:iternumb
    % declare num. of agents
    numAgent = 100;
    area = 100;
    %initial positions
    position = randi([0,area],numAgent,2);
    %-> declare the positions of the initially infected agents
    %-> declare the positions of the initially healthy agents
    index = randsample(1:length(position), 10);
    infecteds = position(index,:);
    Temp=position;
    Temp(index,:)=[];
    healthies=Temp;
    %step size
    stepsize = 1.0;
    numofinf=zeros(1,100);

    for ii = 1:100
        % generate difference
        d_position_inf = stepsize.*(2*randi([0,1],length(infecteds),2)-1);
        d_position_hlt = stepsize.*(2*randi([0,1],length(healthies),2)-1);

        % update
        infecteds = infecteds + d_position_inf;
        healthies = healthies + d_position_hlt;
        %->update the infecteds
        %->update the healties

        %apply the boundry conditions
        %->B.C. for the infecteds
        %->B.C. for the healties
        infecteds(infecteds > area) = area-2*stepsize;
        infecteds(infecteds <= 0) = 2*stepsize;
        healthies(healthies > area) = area-2*stepsize;
        healthies(healthies <= 0) = 2*stepsize;
        %-----
        %->check healty agents if they are violating the social distance rule.
        ind=[];
        for j=1:length(healthies)
            for k=1:length(infecteds)

                if sqrt((healthies(j,1)-infecteds(k,1)).^2 + (healthies(j,2)-infecteds(k,2)).^2) < 5
                    ind=[ind; j];
                    break;
                end
            end
        end
        %->if they are, they get infected with "p" probabily.
        prob=rand;
        if prob < p
            temp2 = healthies(ind,:);
            infecteds=[infecteds;temp2];
            healthies(ind,:)=[];
        end
        %-----
        %->update the positions of infected and healthy agents
        %->record the num of infected.
        numofinf(ii)=length(infecteds);

        subplot(311)
        plot(healthies(:,1),healthies(:,2),'o',infecteds(:,1),infecteds(:,2),'x');
        title('2-D Random Walk')
        subplot(312)
```

```

        plot(1:ii,numofinf(1:ii),'-b')
        hold on; grid on;
        yticks([0:5:100])
        str = sprintf('number of infected agents with respect to the number of iterations for each trial p: %.2f', p)
        title(str)
        xlabel('number of iteration');ylabel('number of infected agents');
        drawnow;
        pause(0.1);
    end
    totalnumofinf=totalnumofinf+numofinf;
end
averagenumofinf=zeros(1,100);
averagenumofinf=totalnumofinf./iternumb;
subplot(313)
plot(1:100,averagenumofinf)
grid on;
yticks([0:5:100])
str = sprintf('first scenario, average process of the overall trials p: %.2f', p);
title(str)
xlabel('number of iteration');ylabel('number of infected agents');
end

function immobilized_infecteds_simulation(p)
figure;
iternumb=25;
totalnumofinf=zeros(1,100);
for i=1:iternumb
    % declare num. of agents
    numAgent = 100;
    area = 100;
    %initial positions
    position = randi([0,area],numAgent,2);
    %-> declare the positions of the initially infected agents
    %-> declare the positions of the initially healthy agents
    index = randsample(1:length(position), 10);
    infecteds = position(index,:);
    Temp=position;
    Temp(index,:)=[];
    healthies=Temp;
    %step size
    stepsize = 1.0;
    numofinf=zeros(1,100);

    for ii = 1:100
        % generate difference
        d_position_inf = stepsize.*(2*randi([0,1],length(infecteds),2)-1);
        d_position_hlt = stepsize.*(2*randi([0,1],length(healthies),2)-1);

        % update
        %infecteds = infecteds + d_position_inf;
        healthies = healthies + d_position_hlt;
        %->update the infecteds
        %->update the healthies

        %apply the boundary conditions
        %->B.C. for the infecteds
        %->B.C. for the healthies
        infecteds(infecteds > area) = area-2*stepsize;
        infecteds(infecteds <= 0) = 2*stepsize;
        healthies(healthies > area) = area-2*stepsize;
        healthies(healthies <= 0) = 2*stepsize;
        %-----
        %->check healthy agents if they are violating the social distance rule.
        ind=[];
        for j=1:length(healthies)
            for k=1:length(infecteds)

                if sqrt((healthies(j,1)-infecteds(k,1)).^2 + (healthies(j,2)-infecteds(k,2)).^2) < 5
                    ind=[ind; j];
                    break;
                end
            end
        end
        %->if they are, they get infected with "p" probabily.
        prob=rand;
        if prob < p
            temp2 = healthies(ind,:);
            infecteds=[infecteds;temp2];
            healthies(ind,:)=[];
        end
        %-----
        %->update the positions of infected and healthy agents
        %->record the num of infected.
        numofinf(ii)=length(infecteds);

        subplot(311)
        plot(healthies(:,1),healthies(:,2),'o',infecteds(:,1),infecteds(:,2),'x');
        title('2-D Random Walk')
        subplot(312)
        plot(1:ii,numofinf(1:ii),'-b')
        hold on; grid on;
        yticks([0:5:100])
        str = sprintf(' if infecteds immobilized, number of infected agents with respect to the number of iterations for each
trial p: %.2f', p)
        title(str)
        xlabel('number of iteration');ylabel('number of infected agents');
        drawnow;
        pause(0.1);
    end
    totalnumofinf=totalnumofinf+numofinf;
end
averagenumofinf=zeros(1,100);

```

```

averagenumofinf=totalnumofinf./iternumb;
subplot(313)
plot(1:100,averagenumofinf)
grid on;
yticks([0:5:100])
str = sprintf('if infecteds immobilized, average process of the overall trials p: %.2f', p);
title(str)
xlabel('number of iteration');ylabel('number of infected agents');
end

function randomly_half_simulation(p)
figure;
iternumb=25;
totalnumofinf=zeros(1,100);
for i=1:iternumb
    % declare num. of agents
    numAgent = 100;
    area = 100;
    %initial positions
    position = randi([0,area],numAgent,2);
    %-> declare the positions of the initially infected agents
    %-> declare the positions of the initially healthy agents
    index = randsample(1:length(position), 10);
    infecteds = position(index,:);
    Temp=position;
    Temp(index,:)=[];
    healthies=Temp;
    %step size
    stepsize = 1.0;
    numofinf=zeros(1,100);

    for ii = 1:100
        % generate difference
        % d_position_inf = stepsize.*(2*randi([0,1],round(length(infecteds)./2),2)-1);
        % d_position_hlt = stepsize.*(2*randi([0,1],round(length(infecteds)./2),2)-1);
        % inf_index = randsample(1:length(infecteds), round(length(infecteds)./2));
        % hlt_index = randsample(1:length(healthies), round(length(infecteds)./2));
        % update
        % infecteds(inf_index,:) = infecteds(inf_index,:) + d_position_inf;
        % healthies(hlt_index,:) = healthies(hlt_index,:) + d_position_hlt;
        %->update the infecteds
        %->update the healties

        allagents=[healthies;infecteds];
        indekss=randsample(1:length(allagents),50);
        indd=[];
        inddd=[];
        for iii=1:50
            if indekss(iii)<= (100-length(infecteds))
                indd=[indd; indekss(iii)];
            else
                inddd=[inddd; indekss(iii)];
            end
        end

        d_position_inf = stepsize.*(2*randi([0,1],length(inddd),2)-1);
        d_position_hlt = stepsize.*(2*randi([0,1],length(indd),2)-1);

        infecteds((inddd-length(healthies)),:) = infecteds((inddd-length(healthies)),:) + d_position_inf;
        healthies(indd,:) = healthies(indd,:) + d_position_hlt;

        %apply the boundry conditions
        %->B.C. for the infecteds
        %->B.C. for the healties
        infecteds(infecteds > area) = area-2*stepsize;
        infecteds(infecteds <= 0) = 2*stepsize;
        healthies(healthies > area) = area-2*stepsize;
        healthies(healthies <= 0) = 2*stepsize;
        %-----
        %->check healty agents if they are violating the social distance rule.
        ind=[];
        for j=1:length(healthies)
            for k=1:length(infecteds)
                if sqrt((healthies(j,1)-infecteds(k,1)).^2 + (healthies(j,2)-infecteds(k,2)).^2) < 5
                    ind=[ind; j];
                    break;
                end
            end
        end
        %-->if they are, they get infected with "p" probabiltly.
        prob=rand;
        if prob < p
            temp2 = healthies(ind,:);
            infecteds=[infecteds;temp2];
            healthies(ind,:)=[];
        end
        %-----
        %->update the positions of infected and healthy agents
        %->record the num of infected.
        numofinf(ii)=length(infecteds);

        subplot(311)
        plot(healthies(:,1),healthies(:,2),'o',infecteds(:,1),infecteds(:,2),'x');
        title('2-D Random Walk')
        subplot(312)
        plot(1:ii,numofinf(1:ii),'-b')
        hold on; grid on;
    end
end

```



```

        yticks([0:5:100])
        str = sprintf('when randomly immobilized half, number of infected agents with respect to the number of iterations for
each trial p: %.2f', p)
        title(str)
        xlabel('number of iteration');ylabel('number of infected agents');
        drawnow;
        pause(0.1);
    end
    totalnumofinf=totalnumofinf+numofinf;
end
averagenumofinf=zeros(1,100);
averagenumofinf=totalnumofinf./iternumb;
subplot(313)
plot(1:100,averagenumofinf)
yticks([0:5:100])
grid on;
str = sprintf('when randomly immobilized half, average process of the overall trials p: %.2f', p);
title(str)
xlabel('number of iteration');ylabel('number of infected agents');
end

```