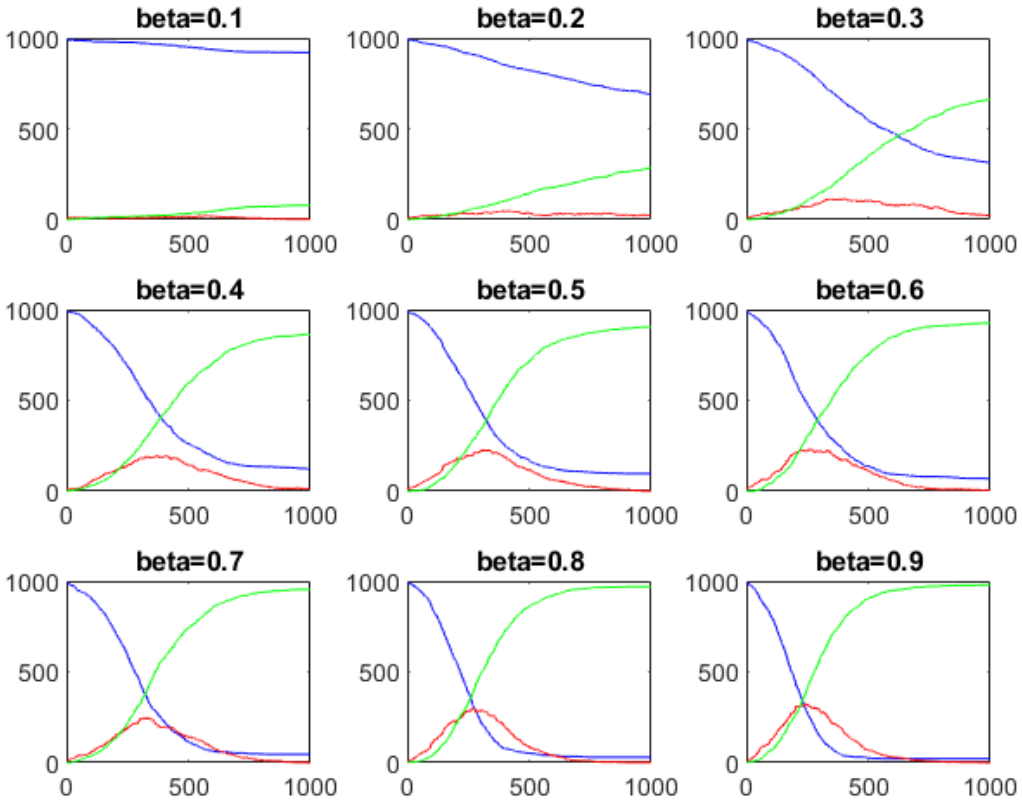
11.1

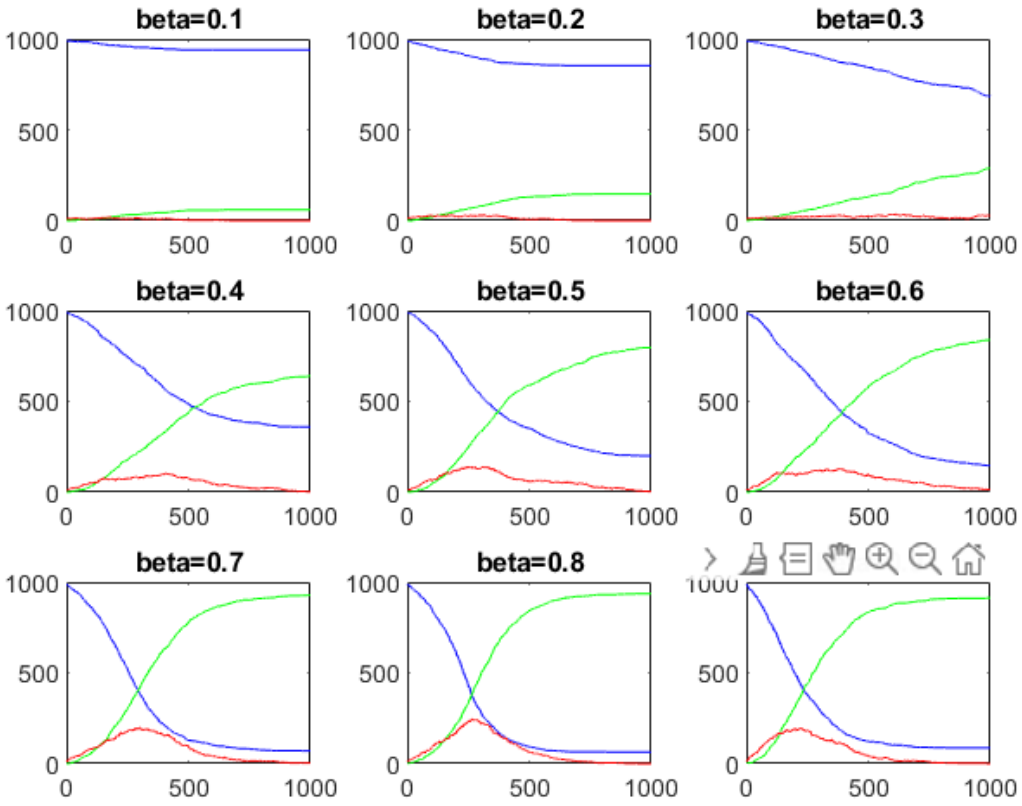
a)-c)

This is for gamma=0.01

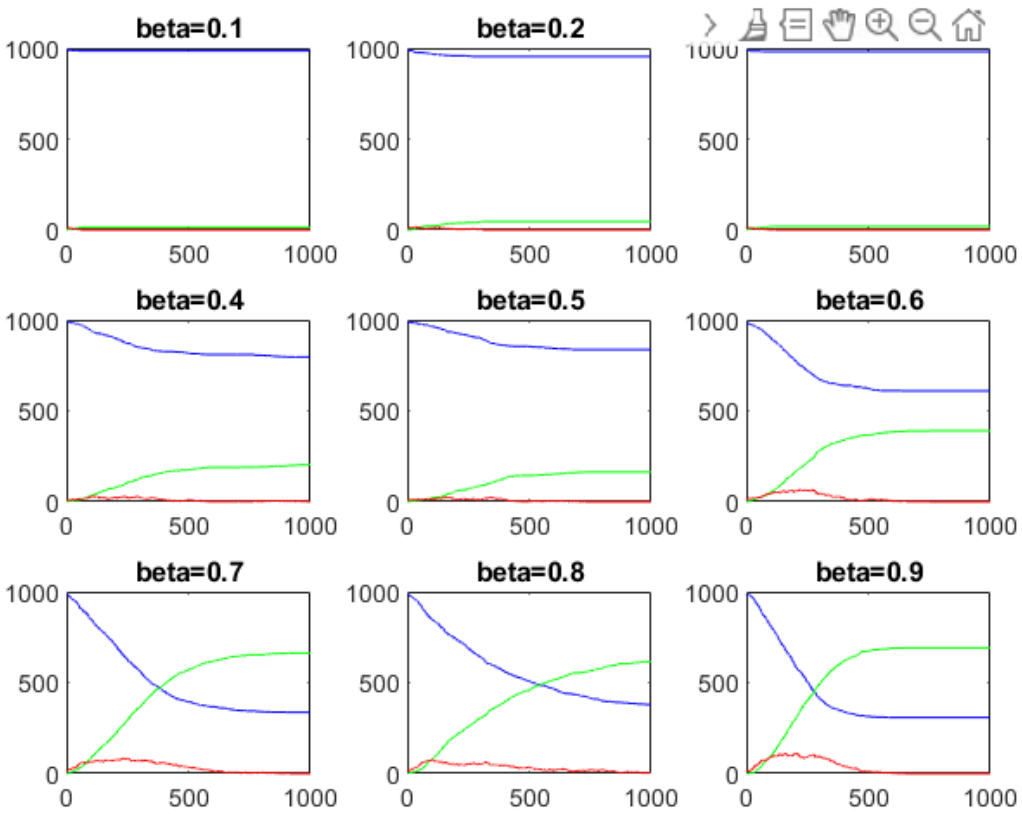


d)

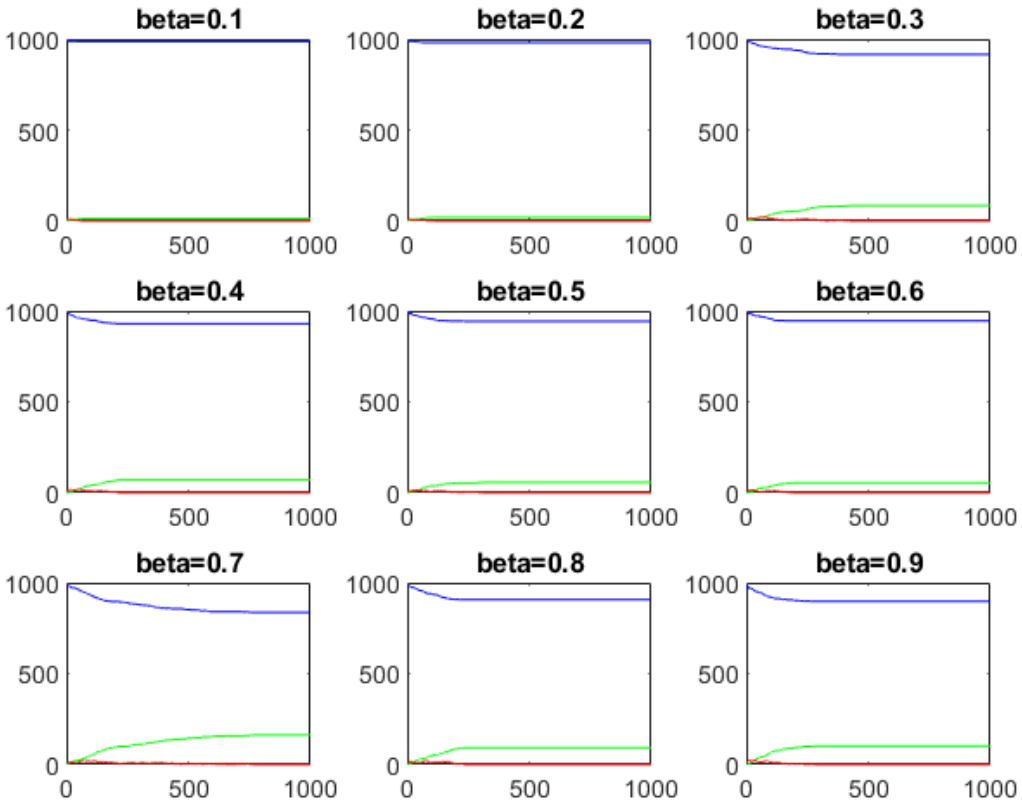
gamma=0.015


beta=0.1
beta=0.2
beta=0.3
beta=0.4
beta=0.5
beta=0.6
beta=0.7
beta=0.8
1000
500
0
0
500
1000

Gamma=0.02:

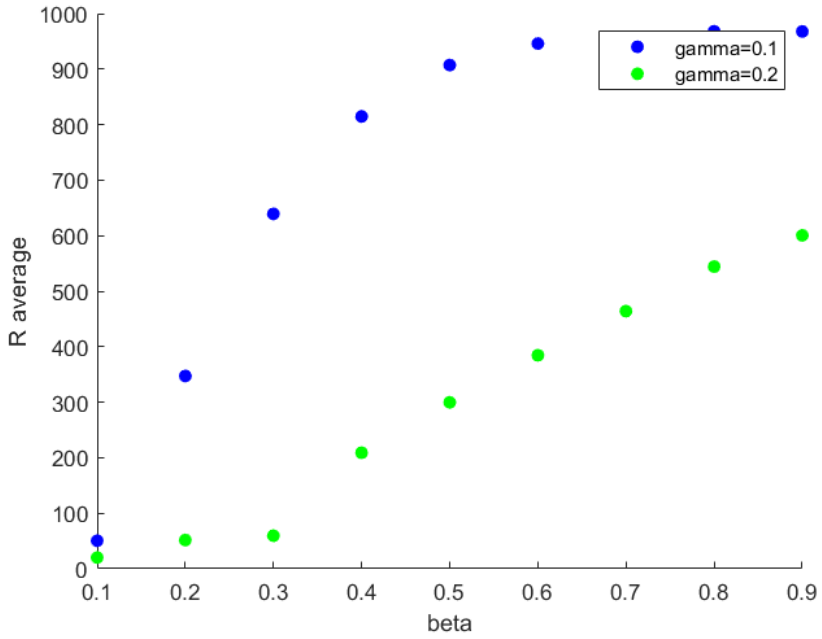

beta=0.1
beta=0.2
beta=0.4
beta=0.5
beta=0.6
beta=0.7
beta=0.8
beta=0.9
1000
500
0
0
500
1000

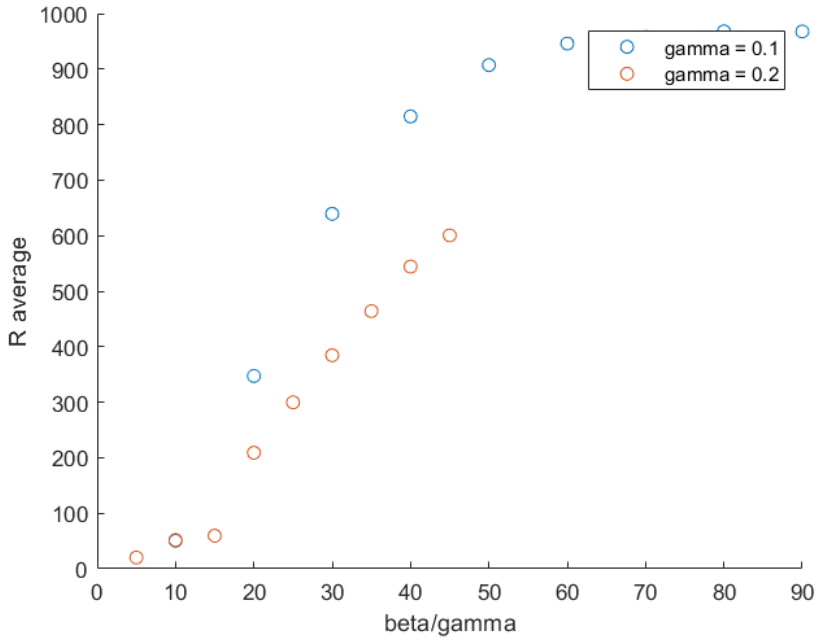
Gamma=0.03



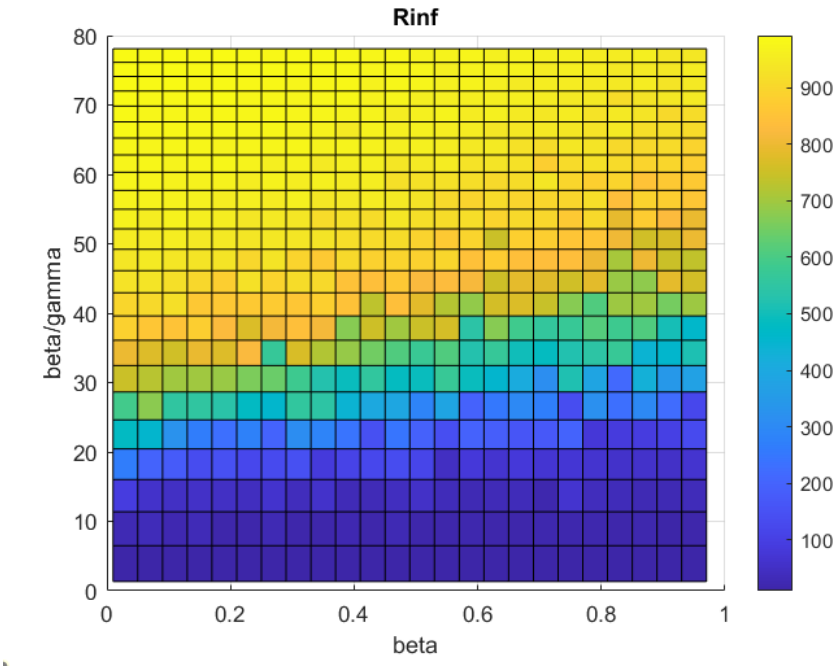
11.2

a)-b)


gamma=0.1
gamma=0.2
R average
beta
0
100
200
300
400
500
600
700
800
900
1000
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9


gamma = 0.1
gamma = 0.2
R average
beta/gamma
0
100
200
300
400
500
600
700
800
900
1000
0
10
20
30
40
50
60
70
80
90

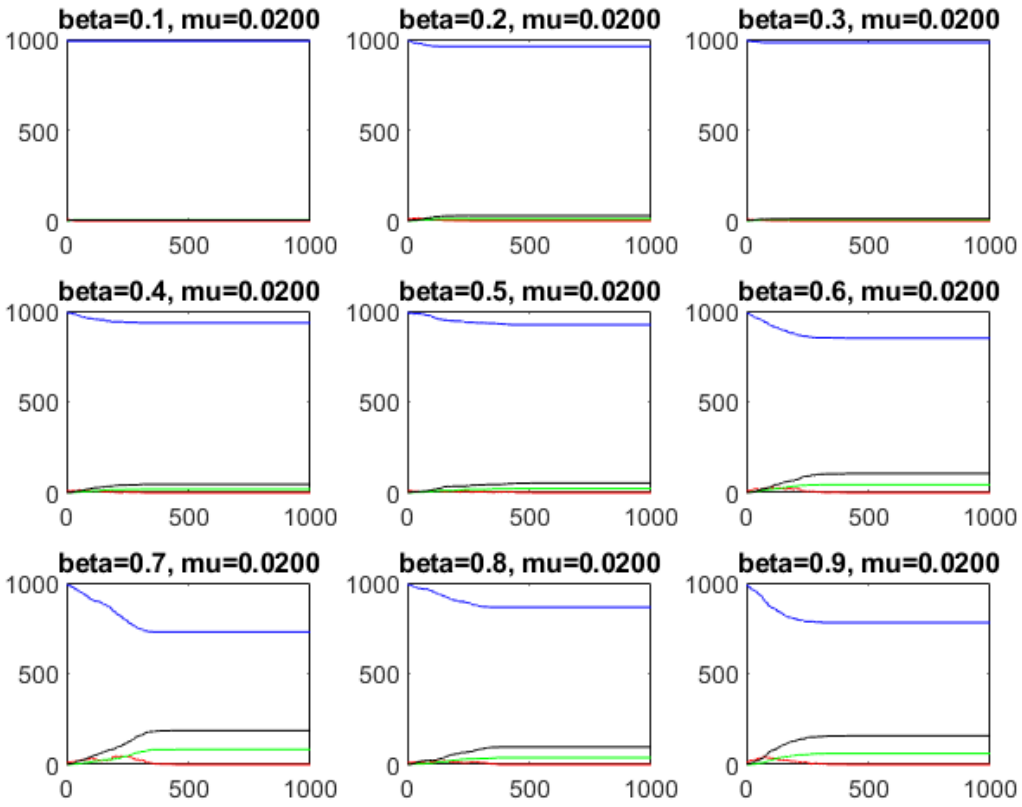
c)



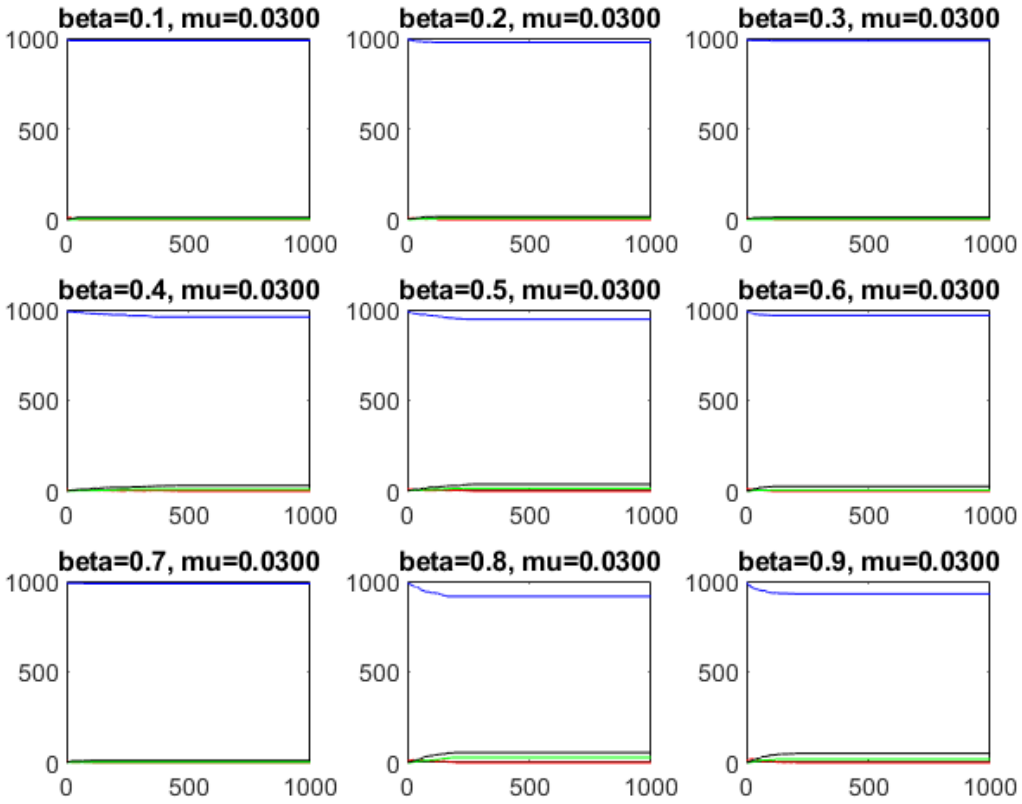
11.3)

Gamma=0.01 for all

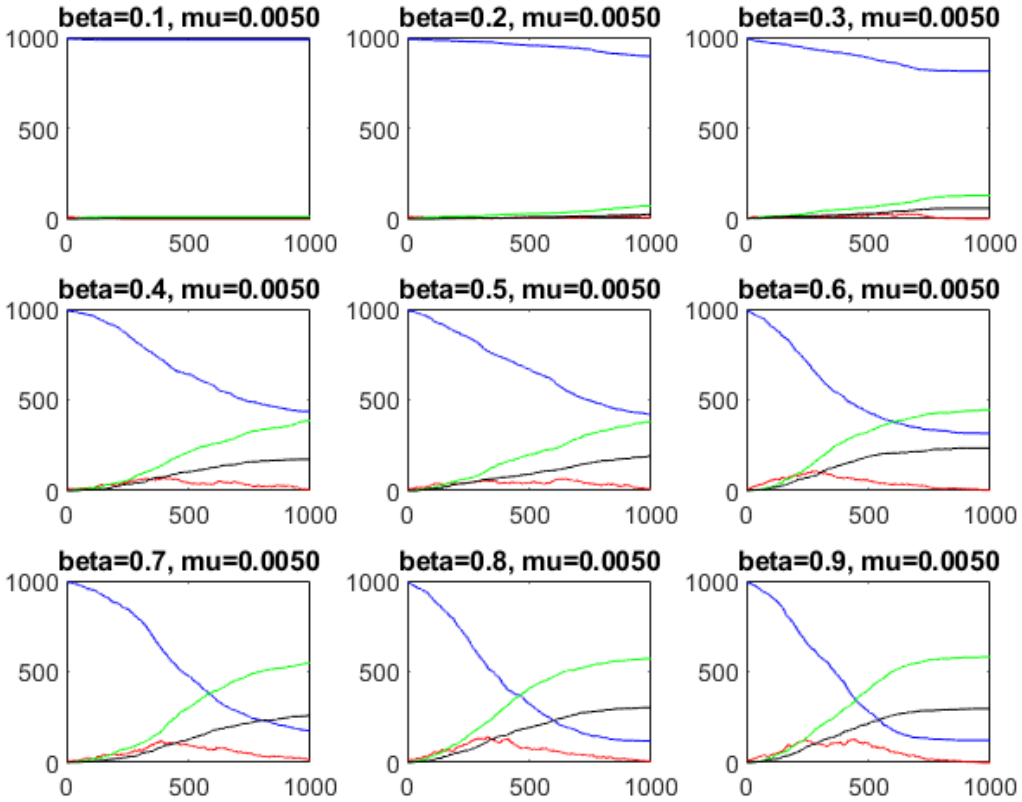
Mu=0.02



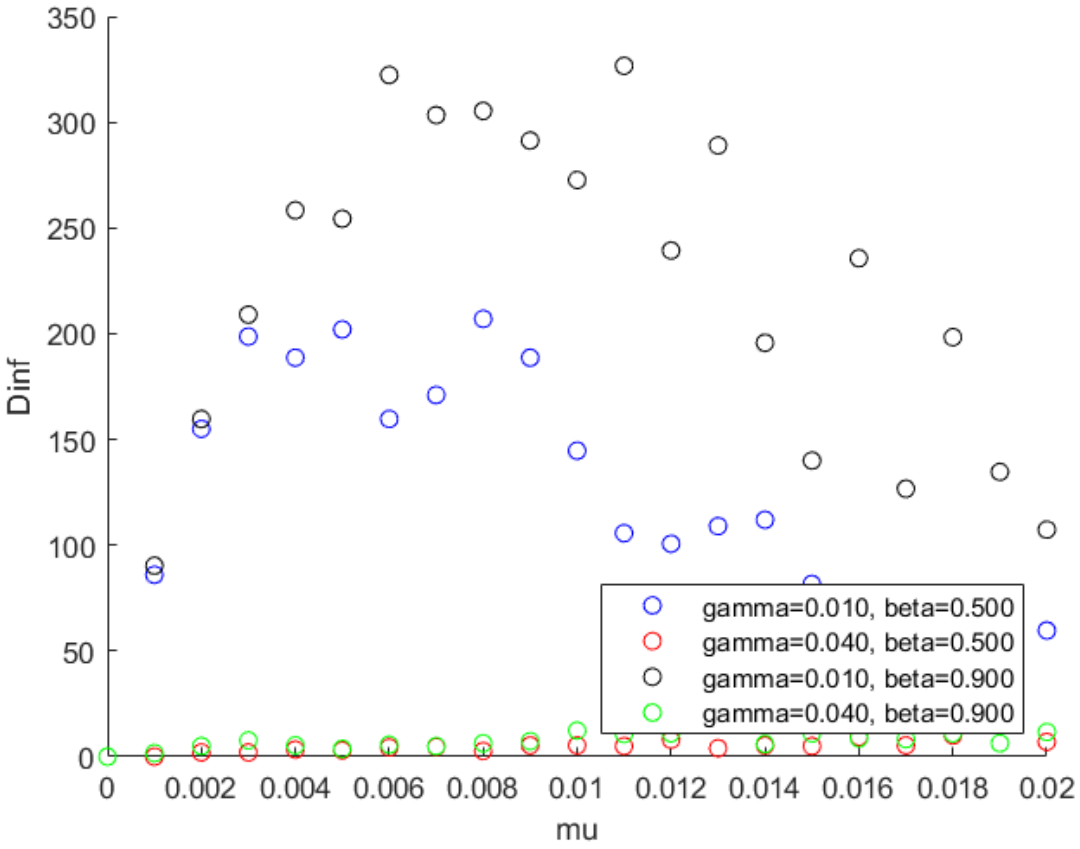
Mu=0.03

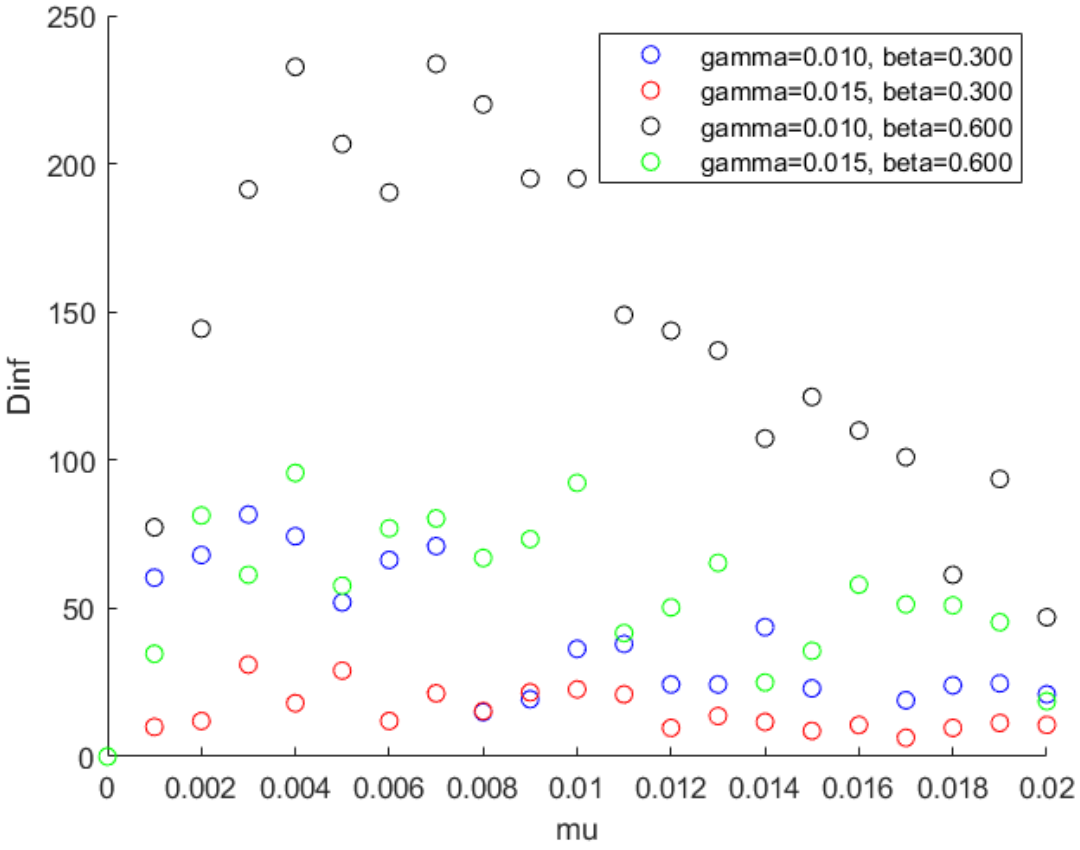

beta=0.1, mu=0.0300
beta=0.2, mu=0.0300
beta=0.3, mu=0.0300
beta=0.4, mu=0.0300
beta=0.5, mu=0.0300
beta=0.6, mu=0.0300
beta=0.7, mu=0.0300
beta=0.8, mu=0.0300
beta=0.9, mu=0.0300
0
500
1000

Mu=0.005


beta=0.1, mu=0.0050
beta=0.2, mu=0.0050
beta=0.3, mu=0.0050
beta=0.4, mu=0.0050
beta=0.5, mu=0.0050
beta=0.6, mu=0.0050
beta=0.7, mu=0.0050
beta=0.8, mu=0.0050
beta=0.9, mu=0.0050
0
500
1000

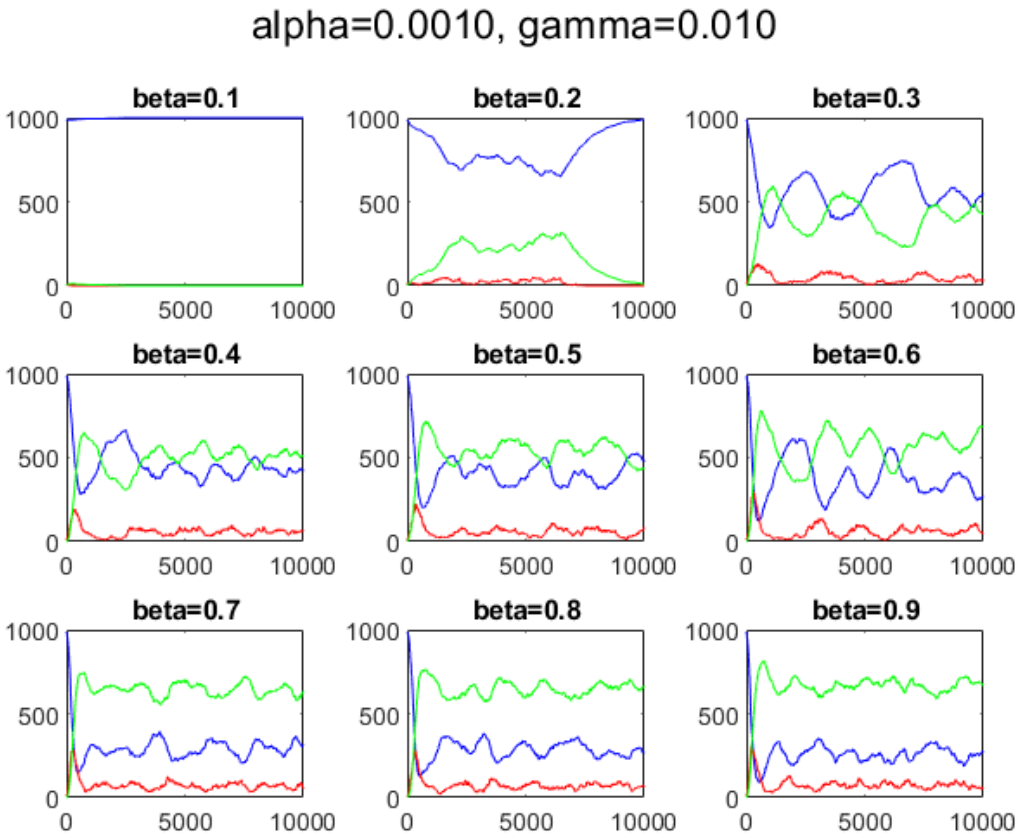
b)


Dinf
mu
gamma=0.010, beta=0.500
gamma=0.040, beta=0.500
gamma=0.010, beta=0.900
gamma=0.040, beta=0.900

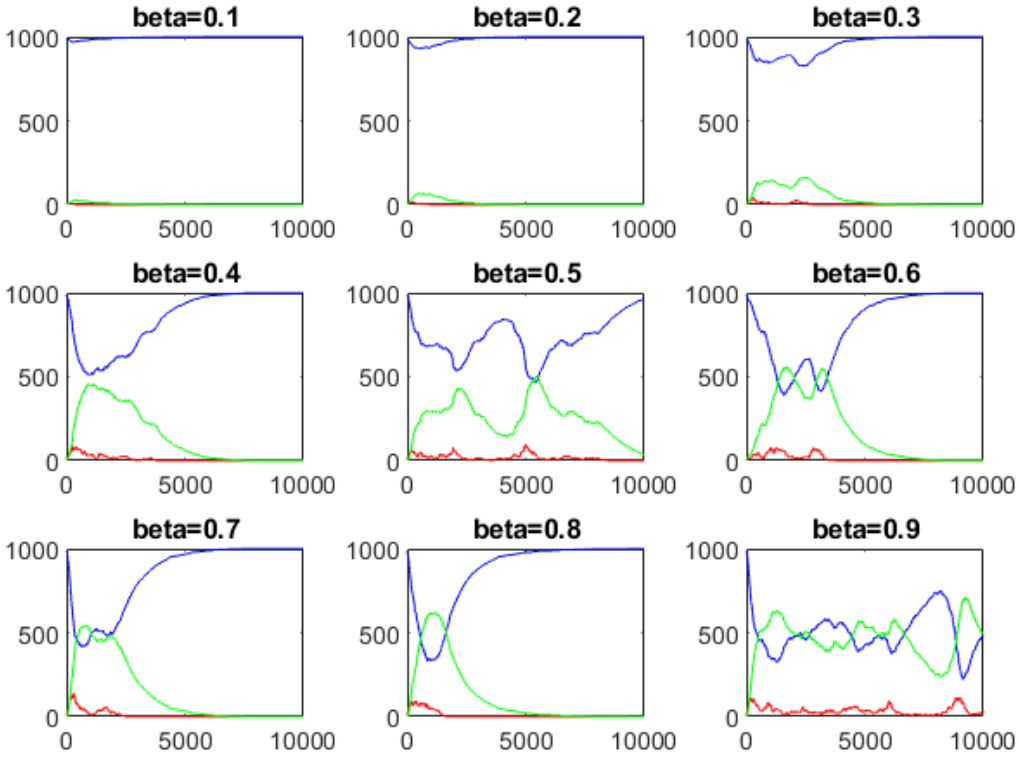

Dinf
mu
gamma=0.010, beta=0.300
gamma=0.015, beta=0.300
gamma=0.010, beta=0.600
gamma=0.015, beta=0.600

11.4

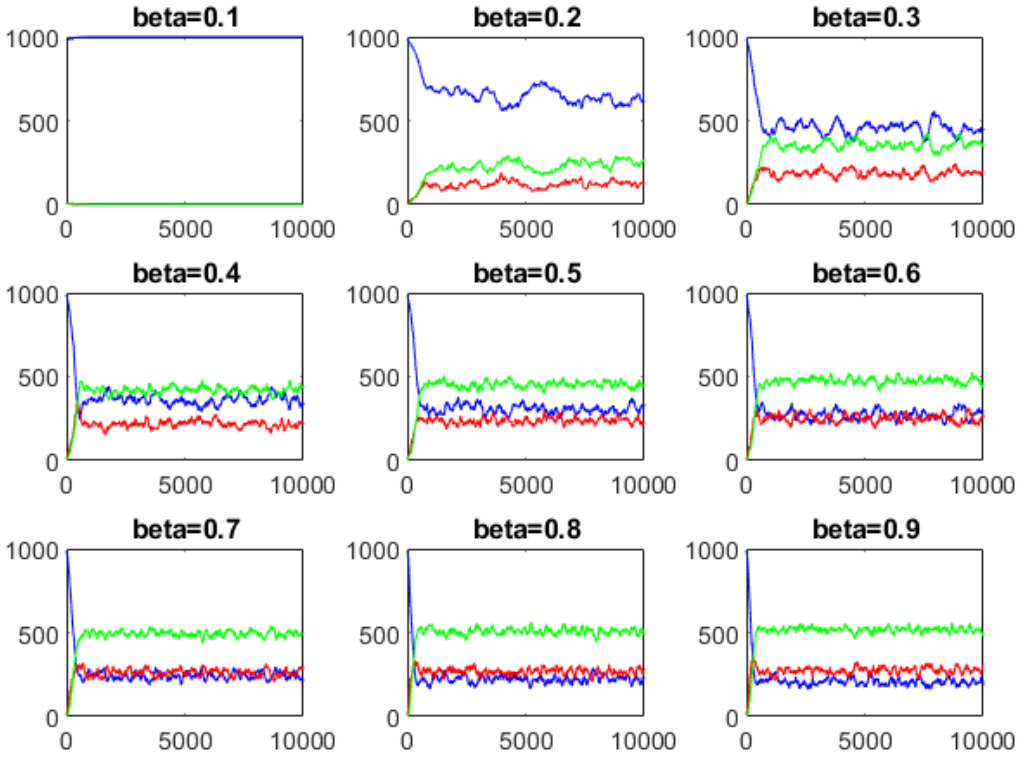
a)-b)


alpha=0.0010, gamma=0.010
beta=0.1
beta=0.2
beta=0.3
beta=0.4
beta=0.5
beta=0.6
beta=0.7
beta=0.8
beta=0.9
0
500
1000
0
5000
10000

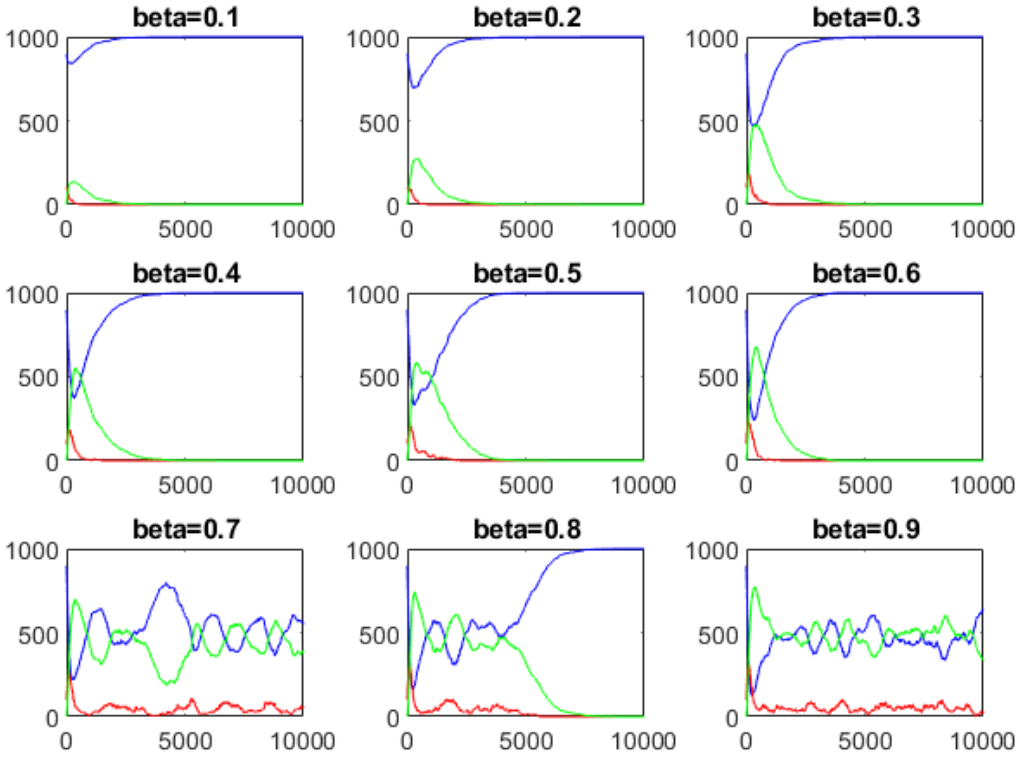
alpha=0.0010, gamma=0.015



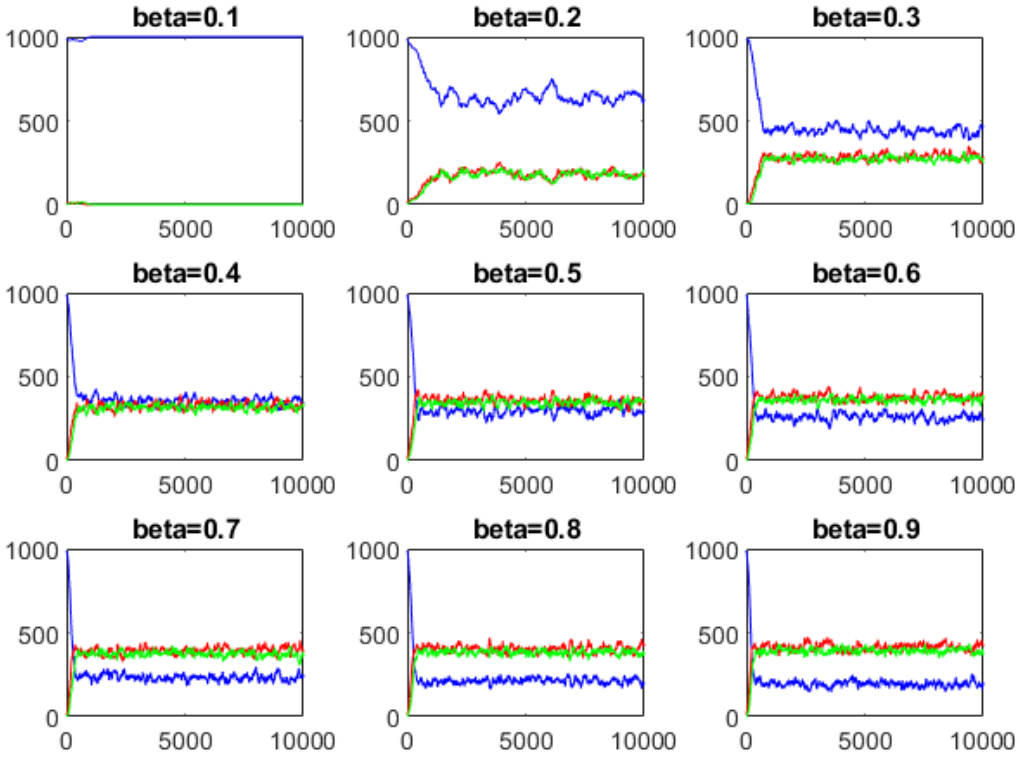
alpha=0.0050, gamma=0.010

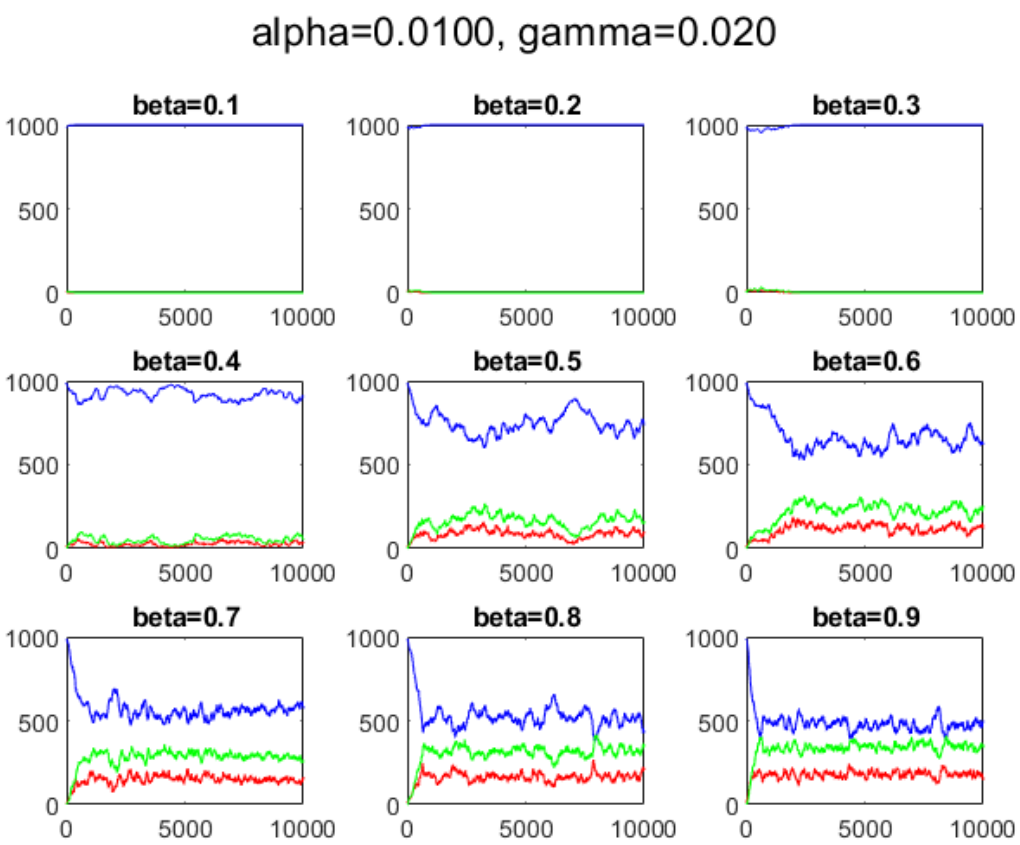


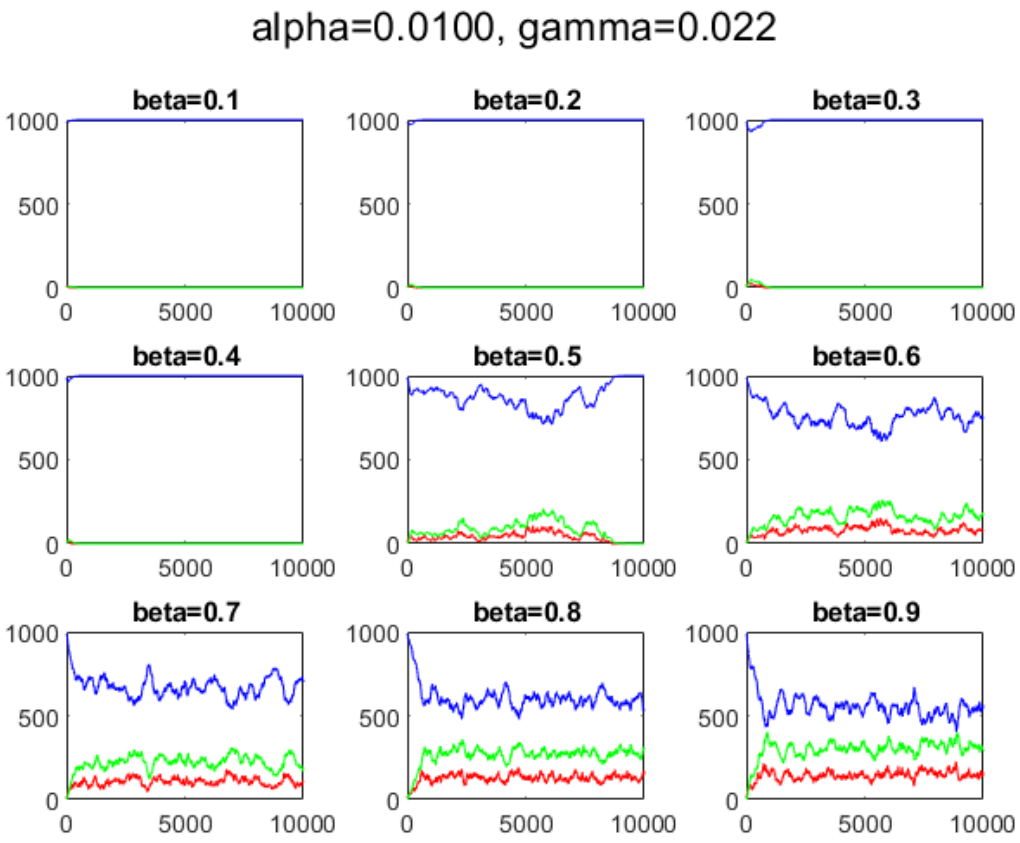
alpha=0.0015, gamma=0.015

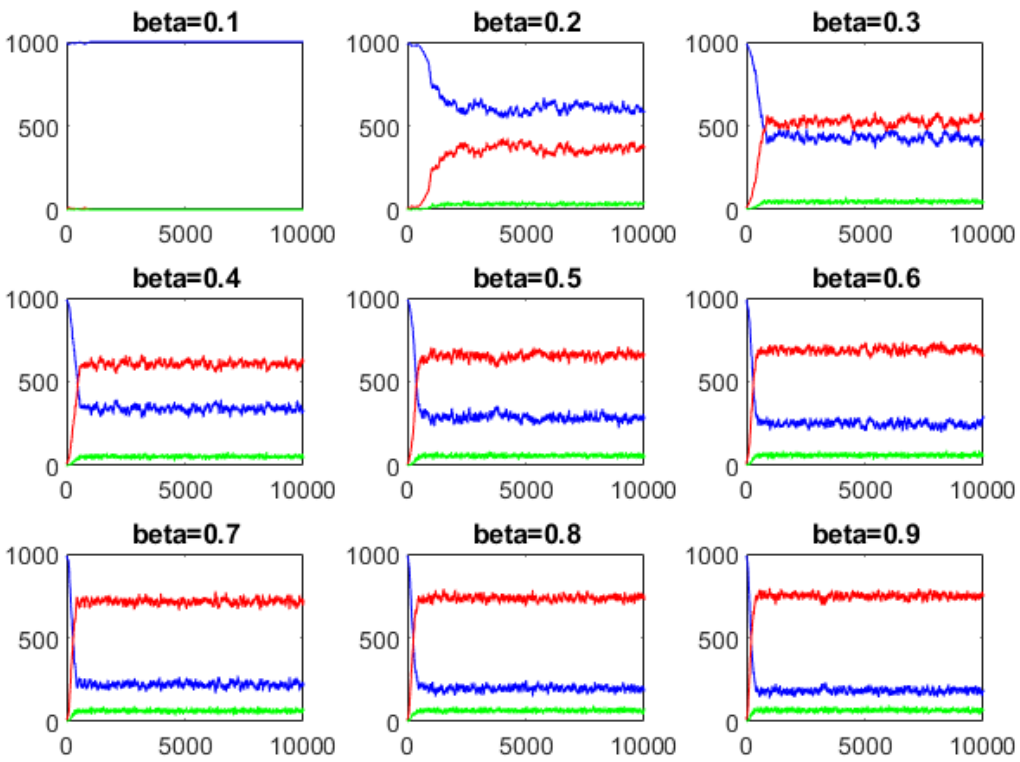


alpha=0.0100, gamma=0.010




alpha=0.0100, gamma=0.020
beta=0.1
beta=0.2
beta=0.3
beta=0.4
beta=0.5
beta=0.6
beta=0.7
beta=0.8
beta=0.9
0
500
1000
0
5000
10000


alpha=0.0100, gamma=0.022
beta=0.1
beta=0.2
beta=0.3
beta=0.4
beta=0.5
beta=0.6
beta=0.7
beta=0.8
beta=0.9
0
500
1000
0
5000
10000


alpha=0.1000, gamma=0.010
beta=0.1
beta=0.2
beta=0.3
beta=0.4
beta=0.5
beta=0.6
beta=0.7
beta=0.8
beta=0.9
1000
500
0
0
5000
10000

Code

```
close all
clear all
clc
L = 100;
N = 1000;
infectedProb = 0.01;

% l = find(M==1);
% k = find(M==2);
% length(l)+length(k)
% %
% [x1, y1] = find(M==1);
% [x2, y2] = find(M==2);
% scatter(x1, y1, 10, "filled", MarkerFaceColor='r')
% hold on
% scatter(x2, y2, 10, "filled", MarkerFaceColor='g')


% Initialize lattice
initLattice = InitLatticeWithAgents(L,N,infectedProb);
initData = Data(initLattice);
nrOfInitInfected = initData(2);



beta = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9];  % infected probability
gamma = 0.03; % recover probability
d = 0.8;
for b = beta
    time = [];
    t = 0;
    nSuceptibleList = [];
    nInfectedList = [];
    nRecoveredList = [];
    lattice = initLattice;
%     while(FindInfected(lattice))
    for h = 1:1000
        data = Data(lattice);
        nSuceptibleList(end+1) = data(1);
        nInfectedList(end+1) = data(2);
        nRecoveredList(end+1) = data(3);
%         clf
        lattice = Move(lattice, d);
        lattice = infection(lattice, b);
        lattice = Recover(lattice, gamma);
    %     PlotModel(lattice)
```

```
    %     drawnow
        t = t + 1;
        time(end+1) = t;
%         if (nSuceptibleList(t) + nInfectedList(t) + nRecoveredList(t)) ~= N
%             disp('something wrong')
%             break
%         end
    end
    if b == 0.1
        b1Suceptible = nSuceptibleList;
        b1Infected = nInfectedList;
        b1Recovered = nRecoveredList;
    elseif b== 0.2
        b2Suceptible = nSuceptibleList;
        b2Infected = nInfectedList;
        b2Recovered = nRecoveredList;
    elseif b== 0.3
        b3Suceptible = nSuceptibleList;
        b3Infected = nInfectedList;
        b3Recovered = nRecoveredList;
    elseif b== 0.4
        b4Suceptible = nSuceptibleList;
        b4Infected = nInfectedList;
        b4Recovered = nRecoveredList;
    elseif b== 0.5
        b5Suceptible = nSuceptibleList;
        b5Infected = nInfectedList;
        b5Recovered = nRecoveredList;
    elseif b== 0.6
        b6Suceptible = nSuceptibleList;
        b6Infected = nInfectedList;
        b6Recovered = nRecoveredList;
    elseif b== 0.7
        b7Suceptible = nSuceptibleList;
        b7Infected = nInfectedList;
        b7Recovered = nRecoveredList;
    elseif b== 0.8
        b8Suceptible = nSuceptibleList;
        b8Infected = nInfectedList;
        b8Recovered = nRecoveredList;
    elseif b== 0.9
        b9Suceptible = nSuceptibleList;
        b9Infected = nInfectedList;
        b9Recovered = nRecoveredList;
    end

end
```

```
subplot(3,3,1)
plot(time, b1Suceptible,'b')
hold on
plot(time, b1Recovered,'g')
plot(time, b1Infected,'r')
title('beta=0.1')

subplot(3,3,2)
plot(time, b2Suceptible,'b')
hold on
plot(time, b2Recovered,'g')
plot(time, b2Infected,'r')
title('beta=0.2')

subplot(3,3,3)
plot(time, b3Suceptible,'b')
hold on
plot(time, b3Recovered,'g')
plot(time, b3Infected,'r')
title('beta=0.3')

subplot(3,3,4)
plot(time, b4Suceptible,'b')
hold on
plot(time, b4Recovered,'g')
plot(time, b4Infected,'r')
title('beta=0.4')

subplot(3,3,5)
plot(time, b5Suceptible,'b')
hold on
plot(time, b5Recovered,'g')
plot(time, b5Infected,'r')
title('beta=0.5')

subplot(3,3,6)
plot(time, b6Suceptible,'b')
hold on
plot(time, b6Recovered,'g')
plot(time, b6Infected,'r')
title('beta=0.6')

subplot(3,3,7)
plot(time, b7Suceptible,'b')
hold on
plot(time, b7Recovered,'g')
plot(time, b7Infected,'r')
title('beta=0.7')
```

```
subplot(3,3,8)
plot(time, b8Suceptible,ld on
plot(time, b8Recovered,'g')
plot(time, b8Infected,'r')
title('beta=0.8')

subplot(3,3,9)
plot(time, b9Suceptible,'b')
hold on
plot(time, b9Recovered,'g')
plot(time, b9Infected,'r')
title('beta=0.9')


%% check to see that we initialize N agents in the grid
nrOfA = 0;
for j= 1:length(lattice)
    for i= 1:length(lattice)
        a = lattice{j, i};
        nrElements = length(a);
        nrOfA = nrOfA + nrElements;
    end
end


close all
clear all
clc
L = 100;
N = 1000;
infectedProb = 0.01;

% l = find(M==1);
% k = find(M==2);
% length(l)+length(k)
% %
% [x1, y1] = find(M==1);
% [x2, y2] = find(M==2);
% scatter(x1, y1, 10, "filled", MarkerFaceColor='r')
% hold on
% scatter(x2, y2, 10, "filled", MarkerFaceColor='g')


% Initialize lattice
initLattice = InitLatticeWithAgents(L,N,infectedProb);
initData = Data(initLattice);
nrOfInitInfected = initData(2);
```

```
betas = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9];  % infected probability
gammas = [0.01, 0.02]; % recover probability
d = 0.8;
trials = 4;

RList1 = zeros(trials, length(betas));
RList2 = zeros(trials, length(betas));
for trial = 1:trials
    for g = 1:length(gammas)
        gamma = gammas(g);
        for l = 1:length(betas)
            beta = betas(l);
            time = [];
            t = 0;
            lattice = initLattice;
        %     while(FindInfected(lattice))
            for h = 1:1000
                data = Data(lattice);
        %         clf
                lattice = Move(lattice, d);
                lattice = infection(lattice, beta);
                lattice = Recover(lattice, gamma);
            %     PlotModel(lattice)
            %     drawnow
                t = t + 1;
                time(end+1) = t;

        %         if (nSuceptibleList(t) + nInfectedList(t) + nRecoveredList(t)) ~= N
        %             disp('something wrong')
        %             break
        %         end
            end
            data = Data(lattice);
            if g == 1
                RList1(trial, l) = data(3);
            else
                RList2(trial, l) = data(3);
            end
        end
    end
end

averageR1 = sum(RList1,1)/trials;
averageR2 = sum(RList2,1)/trials;

scatter(betas, averageR1,'filled','b')
```

```
hold on
scatter(betas, averageR2,'filled','g')
legend('gamma=0.1', 'gamma=0.2')



%% store
save('R1average',"averageR1")
save('Raverage2','averageR2')
save('betas',"betas")
save('gammas',"gammas")
betaGamma1 = betas/gammas(1);
betaGamma2 = betas/gammas(2);

figure(1)
scatter(betaGamma1, averageR1)
hold on
scatter(betaGamma2, averageR2)
xlabel('beta/gamma')
ylabel('R average')
legend('gamma = 0.1', 'gamma = 0.2')
hold off

figure(2)
scatter(betas, averageR1,'filled','b')
hold on
scatter(betas, averageR2,'filled','g')
legend('gamma=0.1', 'gamma=0.2')
xlabel('beta')
ylabel('R average')
hold off

figure(3)
imagesc(betas, betaGamma1, averageR1)
colorbar

%% check to see that we initialize N agents in the grid
nrOfA = 0;
for j= 1:length(lattice)
    for i= 1:length(lattice)
        a = lattice{j, i};
        nrElements = length(a);
        nrOfA = nrOfA + nrElements;
    end
end


    function PlotModel(lattice)
```

```
    positions = zeros(length(lattice), length(lattice));

    for i = 1:length(lattice)
        for j = 1:length(lattice)
            mostFrequentAgent = mode(lattice{i,j});
            if (mostFrequentAgent == 1)
                positions(i,j) = 1;
            elseif (mostFrequentAgent == 2)
                positions(i,j) = 2;
            elseif (mostFrequentAgent == 3)
                positions(i,j) = 3;
            else
                positions(i,j) = 0;
            end
        end
    end

    [x1,y1] = find(positions == 1);
    [x2, y2] = find(positions == 2);
    [x3, y3] = find(positions == 3);
    scatter(x1, y1, 10, "filled", MarkerFaceColor='b')  % suceptible
    hold on
    scatter(x2, y2, 10, "filled", MarkerFaceColor='r')  % infected
    scatter(x3, y3, 10, "filled", MarkerFaceColor='g')  % recovered
end

    function lattice = Move(lattice, d)
    newLattice = lattice;
    for i = 1:length(lattice)
        for j = 1:length(lattice)
            numberOfAgents = size(lattice{i,j}, 2);
            for n = 1:numberOfAgents
                r = rand();
                if (d > r)
                    savedAgent = lattice{i,j}(1);
                    lattice{i,j}(1) = [];
                    newLattice{i, j}(1) = [];
                    dir = randi(4);
                    if (dir == 1)
                        if (j == 1)
                            newLattice{i, end} = [newLattice{i, end}, savedAgent];
                        else
                            newLattice{i, j-1} = [newLattice{i, j-1}, savedAgent];
                        end
                    elseif (dir == 2)
                        if (j == length(lattice))
                            newLattice{i, 1} = [newLattice{i, 1}, savedAgent];
                        else
                            newLattice{i, j+1} = [newLattice{i, j+1}, savedAgent];
```

```
                    end
                elseif (dir == 3)
                    if (i == 1)
                        newLattice{end, j} = [newLattice{end, j}, savedAgent];
                    else
                        newLattice{i-1, j} = [newLattice{i-1, j}, savedAgent];
                    end
                elseif (dir == 4)
                    if (i == length(lattice))
                        newLattice{1, j} = [newLattice{1, j}, savedAgent];
                    else
                        newLattice{i+1, j} = [newLattice{i+1, j}, savedAgent];
                    end
                end

                 newLattice{x, y} = [newLattice{x, y}, savedAgent];
            end
        end
    end
    lattice = newLattice;
end


    function lattice = InitLatticeWithAgents(L, N, infectedProb)

    lattice = cell(L);
    for n = 1:(N - N*infectedProb)
        coord = randi(L,1,2);
        x = coord(1);
        y = coord(2);
        lattice{x, y} = [lattice{x, y}, 1];
        if (n > (N - 2*N*infectedProb))
            lattice{x, y} = [lattice{x, y}, 2];
        end
    end

end

    function newLattice = infection(lattice, beta)
    latticeLength = length(lattice);
    for i = 1:latticeLength
        for j = 1:latticeLength
            indexInfected = find(lattice{i, j} == 2);
            if isempty(indexInfected)
                continue
            end
            indexSuceptible = find(lattice{i, j} == 1);
            if isempty(indexSuceptible)
```

```
                continue
            end
            for index=indexInfected
                randomNr = rand();
                if randomNr < beta
                    for indexS = indexSuceptible
                        lattice{i, j}(indexS) = 2; % suceptible gets infected
                    end
                end
            end
        end
    end

    newLattice = lattice;

end

function data = Data(lattice)

    nSuceptible = 0;
    nInfected = 0;
    nRecovered = 0;
    latticeLength = length(lattice);
     for i = 1:latticeLength
        for j = 1:latticeLength
            site = lattice{i, j};
            for agent = site
                if agent == 1
                    nSuceptible = nSuceptible + 1;
                elseif agent == 2
                    nInfected = nInfected + 1;
                elseif agent == 3
                    nRecovered = nRecovered + 1;
                end
            end
        end
     end

     data = [nSuceptible, nInfected, nRecovered];

end
```

I Found out that my code was ineffective for the rest of the assignments so I rewrote to a much more effective code that is shown below

```
    clear all
close all
clc
L = 100;
```

```
N = 1000;
nrOfInfected = 10;
d = 0.8;
beta = 0.6;
gamma = 0.01;
time = 1000;
betas = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9];
initializeAgents = InitializeAgentsWithPos(L, N, nrOfInfected);


dataArray = zeros(3*length(betas),time);
for i = 1:length(betas)
    beta = betas(i);
    agents = initializeAgents;
    for t = 1:time
        dataArray(1+3*(i-1):3+3*(i-1),t) =  Data(agents);
        agents = Move(agents, d, L);
        agents = Recover(agents,gamma);
        agents = Infected(agents, beta);
    end
end


% timeVector = [1:time];
% length(timeVector)
% plot(timeVector, dataArray(1,:))
% hold on
% plot(timeVector, dataArray(2,:))
% plot(timeVector,dataArray(3,:))

for i = 1:length(betas)
    beta = betas(i);
    subplot(3,3,i)
    plot(dataArray(1+3*(i-1),:),'b')
    hold on
    plot(dataArray(2+3*(i-1),:),'r')
    plot(dataArray(3+3*(i-1),:),'g')
    title(sprintf('beta=%0.1f',beta))
end

    clear all
close all
clc
L = 100;
N = 1000;
nrOfInfected = 10;
d = 0.8;
beta = 0.6;
gamma = 0.01;
```

```
time = 1000;
betas = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9];
gammas = [0.01, 0.02];
initializeAgents = InitializeAgentsWithPos(L, N, nrOfInfected);

trials = 4;
RArray1 = zeros(trials, length(betas));
RArray2 = zeros(trials, length(betas));
for j = 1:length(gammas)
    gamma = gammas(j);
    for trial =  1:trials
        for i = 1:length(betas)
            beta = betas(i);
            agents = initializeAgents;
            for t = 1:time
                agents = Move(agents, d, L);
                agents = Recover(agents,gamma);
                agents = Infected(agents, beta);
                data = Data(agents);
            end
            if j == 1
                RArray1(trial,i) = data(3);
            elseif j == 2
                RArray2(trial,i) = data(3);
            end
        end
    end
end

R1Averages = sum(RArray1,1) / trials;
R2Averages = sum(RArray2,1)/ trials;
scatter(betas, R1Averages,'filled','b')
hold on
scatter(betas, R2Averages,'filled','g')
legend(sprintf('gamma=%0.3f',gammas(1)), sprintf('gamma=%0.3f', gammas(2)),'Location','southeast

    clear all
close all
clc
L = 100;
N = 1000;
nrOfInfected = 10;
d = 0.8;
beta = 0.6;
gamma = 0.01;
time = 1000;
betas = [0.01:0.04:1];
gammas = [0.0075:0.000205:0.0125];
initializeAgents = InitializeAgentsWithPos(L, N, nrOfInfected);
```

```
trials = 4;
% RArray1 = zeros(trials, length(betas));
% RArray2 = zeros(trials, length(betas));
RArray = zeros(length(gammas), length(betas));
for j = 1:length(gammas)
    gamma = gammas(j);
    R = zeros(trials, length(betas));
    for trial =  1:trials
        for i = 1:length(betas)
            beta = betas(i);
            agents = initializeAgents;
            for t = 1:time
                agents = Move(agents, d, L);
                agents = Recover(agents,gamma);
                agents = Infected(agents, beta);
                data = Data(agents);
            end
            R(trial, i) = data(3);
        end
    end
    RAverages = sum(R,1) / trials;
    RArray(j, :) = RAverages;
end

% R1Averages = sum(RArray1,1) / trials;
% R2Averages = sum(RArray2,1)/ trials;
% scatter(betas, R1Averages,'filled','b')
% hold on
% scatter(betas, R2Averages,'filled','g')
% legend(sprintf('gamma=%0.3f',gammas(1)), sprintf('gamma=%0.3f', gammas(2)),'Location','southeas


%% Saving RArray and plot phase diagram

save('RAverages','RArray')
gammaBetas = betas ./ gammas;
[x, y] = meshgrid(betas,gammaBetas);
surf(x, y, RArray')
xlabel('beta')
ylabel('beta/gamma')
title('Rinf')
view(2)
colorbar

    clear all
close all
clc
L = 100;
```

```
N = 1000;
nrOfInfected = 10;
d = 0.8;
beta = 0.6;
gamma = 0.01;
mu = 0.00007;
time = 1000;
betas = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9];
initializeAgents = InitializeAgentsWithPos(L, N, nrOfInfected);


% dataArray = zeros(4*length(betas),time);
% for i = 1:length(betas)
%     beta = betas(i);
%     agents = initializeAgents;
%     for t = 1:time
%         dataArray(1+4*(i-1):4+4*(i-1),t) =  DataMortality(agents);
%         agents = Move(agents, d, L);
%         agents = Recover(agents,gamma);
%         agents = Infected(agents, beta);
%         agents = Death(agents, mu);
%     end
% end
%
%
%
% for i = 1:length(betas)
%     beta = betas(i);
%     subplot(3,3,i)
%     plot(dataArray(1+4*(i-1),:),'b')
%     hold on
%     plot(dataArray(2+4*(i-1),:),'r')
%     plot(dataArray(3+4*(i-1),:),'g')
%     plot(dataArray(4+4*(i-1),:),'k')
%     title(sprintf('beta=%0.1f, mu=%0.4f',beta,mu))
% end


%% 11.3b)
betas = [0.3, 0.6];
gammas = [0.01, 0.015];
trials = 3;
mus = [0:0.001:0.02];
DArray1 = zeros(trials, length(mus));
DArray2 = zeros(trials, length(mus));
DArray3 = zeros(trials, length(mus));
DArray4 = zeros(trials, length(mus));
DArray5 = zeros(trials, length(mus));
```

```
DArray6 = zeros(trials, length(mus));
DArray7 = zeros(trials, length(mus));
DArray8 = zeros(trials, length(mus));
for m = 1:length(mus)
    mu = mus(m);
    for j = 1:length(gammas)
        gamma = gammas(j);
        for i = 1:length(betas)
            beta = betas(i);
            for trial =  1:trials
                agents = initializeAgents;
                for t = 1:time
                    agents = Move(agents, d, L);
                    agents = Recover(agents,gamma);
                    agents = Infected(agents, beta);
                    agents = Death(agents,mu);
                    data = DataMortality(agents);
                end
                if i==1 && j==1
                    DArray1(trial,m) = data(4);  % beta1 & gamma1
                elseif i==1 && j==2
                    DArray2(trial,m) = data(4);   %beta1 & gamma2
                elseif i==2 && j==1
                    DArray3(trial,m) = data(4);    %beta2 & gamma1
                elseif i==2 && j==2
                    DArray4(trial,m) = data(4);    %beta2 & gamma2
                end
            end
        end
    end
end


D1Averages = sum(DArray1,1) / trials;
D2Averages = sum(DArray2,1)/ trials;
D3Averages = sum(DArray3,1)/ trials;
D4Averages = sum(DArray4,1)/ trials;
scatter(mus, D1Averages,'b')
hold on
scatter(mus, D2Averages,'r')
scatter(mus, D3Averages,'k')
scatter(mus, D4Averages,'g')
prints1 = sprintf('gamma=%0.3f, beta=%0.3f',gammas(1),betas(1));
prints2 = sprintf('gamma=%0.3f, beta=%0.3f',gammas(2),betas(1));
prints3 = sprintf('gamma=%0.3f, beta=%0.3f',gammas(1),betas(2));
prints4 = sprintf('gamma=%0.3f, beta=%0.3f',gammas(2),betas(2));
legend(prints1,prints2,prints3,prints4,'Location','best');
xlabel('mu')
```

```
ylabel('Dinf')









%                 if j == 1 && i==1
%                     DArray1(trial,i) = data(4);
%                 elseif j==1 && i==2
%                     DArray3(trial,i) = data(4);
%                 elseif j==1 && i==3
%                     DArray4(trial,i) = data(4)
%
%                 elseif j == 2
%                     DArray2(trial,i) = data(4);
%                 end
%             end
%         end
%     end
%
% R1Averages = sum(DArray1,1) / trials;
% R2Averages = sum(DArray2,1)/ trials;
% scatter(betas, R1Averages,'filled','b')
% hold on
% scatter(betas, R2Averages,'filled','g')
% legend(sprintf('gamma=%0.3f',gammas(1)), sprintf('gamma=%0.3f', gammas(2)),'Location','southeas

    function agents = Recover(agents, gamma)

    indexOfInfected = find(agents(:, 3)==2);
    randomNumbers = rand(length(indexOfInfected),1);

    indexGettingRecovered = indexOfInfected(find(randomNumbers<gamma));
    if~isempty(indexGettingRecovered)
        agents(indexGettingRecovered,3) = 3;
    end

end

    function agents = Move(agents, d, L)
    N = length(agents);    %number of agents
    randomNumbers = rand(N,1);
    indexForMove = find(randomNumbers < d);

    nMovingAgents = length(indexForMove);   %nr of agents that will move
```

```
    randomMoveIndex = randi(4, nMovingAgents, 1);
    Movements = [0, 1; 0, -1; 1, 0; -1, 0];

    move = Movements(randomMoveIndex, :);
    agents(indexForMove, 1:2) = agents(indexForMove, 1:2) + move;

    % coordinates that gets outside grid gets to other side (PBC)
    tooSmallCoordinates = find(agents(:, 1:2) < 1);
    tooBigCoordinates = find(agents(:, 1:2) > L);
    if ~isempty(tooBigCoordinates)
        agents(tooBigCoordinates) = agents(tooBigCoordinates) - L;
    end
    if ~isempty(tooSmallCoordinates)
        agents(tooSmallCoordinates) = agents(tooSmallCoordinates) + L;

    end
end

    function agents = InitializeAgentsWithPos(L, N, nrOfInfected)
    agents = zeros(N,3);
    for n = 1:N
        coord = randi(L,1,2);
        x = coord(1);
        y = coord(2);
        agents(n,1) = x;
        agents(n,2) = y;
        agents(n,3) = 1;
        if (n > (N - nrOfInfected))
            agents(n,3) = 2;
        end
    end
end

    function agents = Infected(agents, beta)

    % Find index of suceptible and infected
    indexOfInfected = find(agents(:, 3) == 2);
    coordsOfInfected = agents(indexOfInfected, 1:2);

    % use index to find pos of suceptible and infected
    indexOfSuceptible = find(agents(:,3)==1);
    coordsOfSuceptible = agents(indexOfSuceptible,1:2);

    % find sites where there is suceptible and infected
    InfectedWithSuceptibleIndex = find(ismember(coordsOfSuceptible, coordsOfInfected, 'rows'));
    if isempty(InfectedWithSuceptibleIndex)
        return
    else
```

```
        % Suceptible gets infected with probability beta
        randomNumbers = rand(length(InfectedWithSuceptibleIndex),1);
        suceptibleGetInfectedIndex = InfectedWithSuceptibleIndex(find(randomNumbers<beta));
        if ~isempty(suceptibleGetInfectedIndex)
            coordsOfGettingInfected = coordsOfSuceptible(suceptibleGetInfectedIndex,:);
            gettingInfectedIndex = find(ismember(agents(:,1:2),coordsOfGettingInfected,"rows"));
            agents(gettingInfectedIndex,3) = 2;

        end
    end
end

    function agents = Immunity(agents, alpha)

    indexRecovered = find(agents(:,3)==3);
    randomNumbers = rand(length(indexRecovered),1);

    indexGettingSuceptible = indexRecovered(find(randomNumbers<alpha));
    if ~isempty(indexGettingSuceptible)
        agents(indexGettingSuceptible,3) = 1;
    end

end

    function agents = Death(agents, mu)

    indexOfInfected = find(agents(:, 3) == 2);
    randomNr = rand(length(indexOfInfected),1);
    indexOfGettingDead = indexOfInfected(find(randomNr<mu));
    if ~isempty(indexOfGettingDead)
        agents(indexOfGettingDead,3) = 0;
    end
end

    function data = DataMortality(agents)

   % Find index of the Three different types of agents
   indexOfSuceptible = find(agents(:,3)==1);
   indexOfInfected = find(agents(:,3)==2);
   indexOfRecovered = find(agents(:,3)==3);
   indexOfDead = find(agents(:,3)==0);

   nrOfSuceptible = length(indexOfSuceptible);
   nrOfInfected = length(indexOfInfected);
   nrOfRecovered = length(indexOfRecovered);
   nrOfDead = length(indexOfDead);
   data = [nrOfSuceptible; nrOfInfected; nrOfRecovered; nrOfDead];
```

```
 function data = Data(agents)

% Find index of the Three different types of agents
indexOfSuceptible = find(agents(:,3)==1);
indexOfInfected = find(agents(:,3)==2);
indexOfRecovered = find(agents(:,3)==3);


nrOfSuceptible = length(indexOfSuceptible);
nrOfInfected = length(indexOfInfected);
nrOfRecovered = length(indexOfRecovered);

data = [nrOfSuceptible; nrOfInfected; nrOfRecovered];
```