Homework 1

FRANCISCO CAETANO

SIMULATION OF COMPLEX SYSTEMS

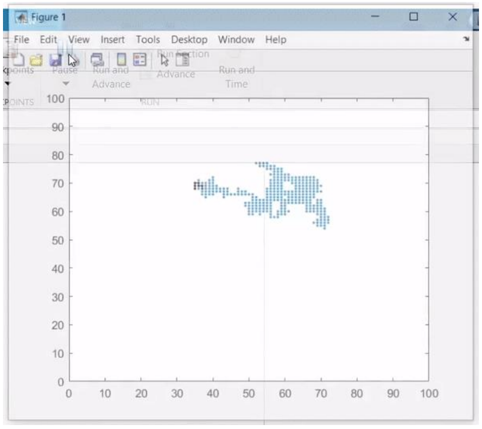
Exercise 1

□Random walk performed by a single agent
- d=1

□Modelling disease spreading on a square lattice with small dimensions
- N=40
- 20x20-lattice
- d=0.8; β=0.6; γ=0.01

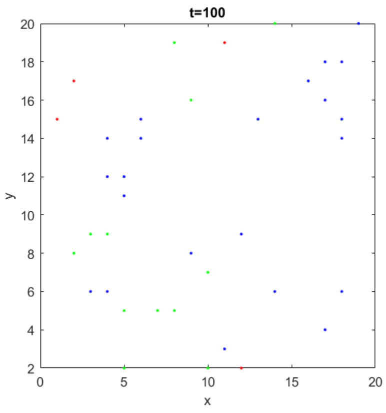
□Modelling disease spreading for 1000 agents on a 100x100–lattice
- d=0.8; β=0.6; γ=0.01

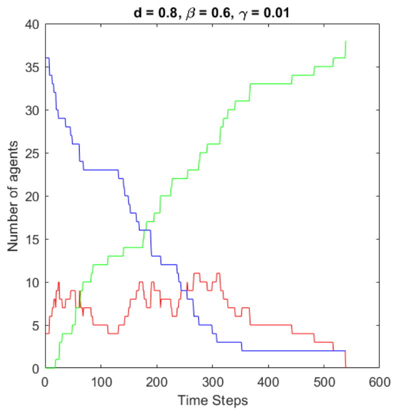
Random Walk



3

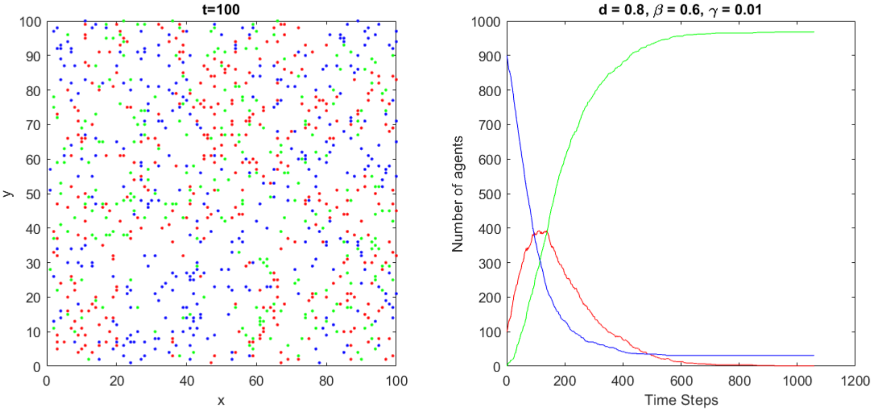
Modelling disease spreading on a 20x20-lattice





4

Modelling disease spreading on a 100x100-lattice



Exercise 2

Show that the model contains two regimes: there are parameter values for which the disease spreads to a large proportion of the population and values for which it doesn't.

□Population-wide disease spreading

□Limited disease spreading

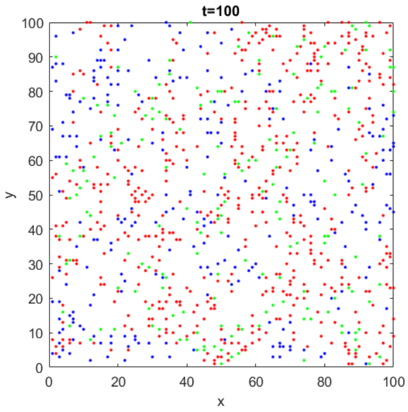
Population-wide disease spreading

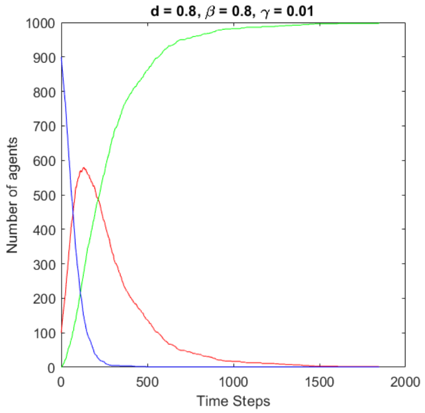
- Increase β: increases the probability of infecting all susceptibles at its current site
- Decrease γ: decreases the probability of recovering from the disease, so each infected will stay infected (and therefore contagious) for longer

Parameters used:

- N=1000
- 100x100-lattice
- d=0.8; β=0.8; γ=0.01

Population-wide disease spreading





Limited disease spreading

□Decrease β: decreases the probability of infecting all susceptibles at its current site

□Increase γ: increases the probability of recovering from the disease, so each infected will stay infected (and therefore contagious) for a shorter period of time

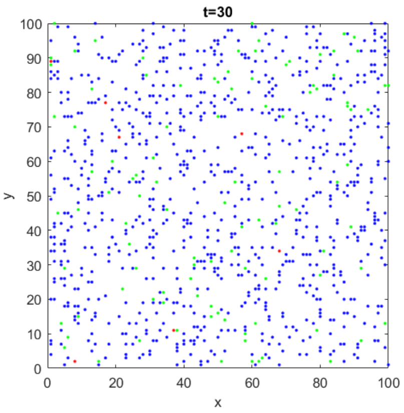
Parameters used:

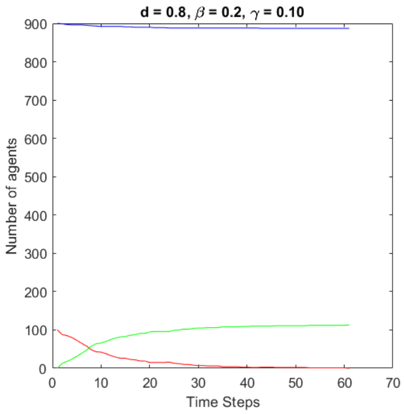
□N=1000

□100x100-lattice

□d=0.8; β=0.4; γ=0.1

Limited disease spreading





Exercise 3

Show that the epidemic threshold depends on not just the ratio k(β/γ) but on the parameters themselves. Fixing a value for β, run the model for each of several values of γ and record the final proportion of recovered agents.

Parameters used:

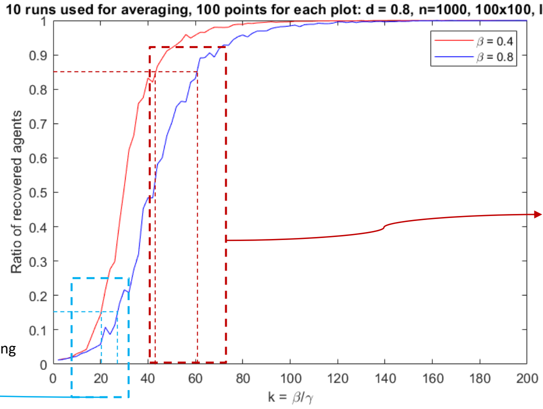
❑N=1000;

❑100x100-lattice

❑100 different values of gamma for each case (with similar spacing)

❑10 runs for each combination were used for averaging

Plot comparing the two data sets



Threshold for population-wide disease spreading

Threshold for limited disease spreading

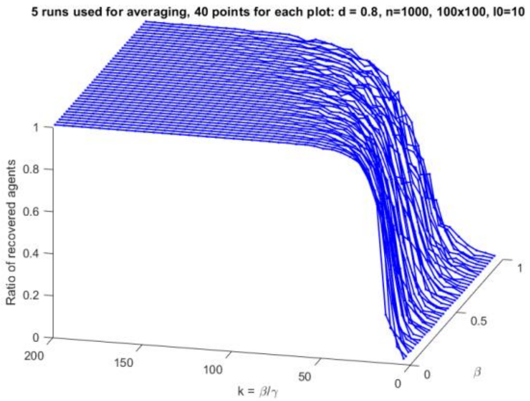
Exercise 4

Repeat the previous process for enough values of β that you can determine the important features of the phase diagram.

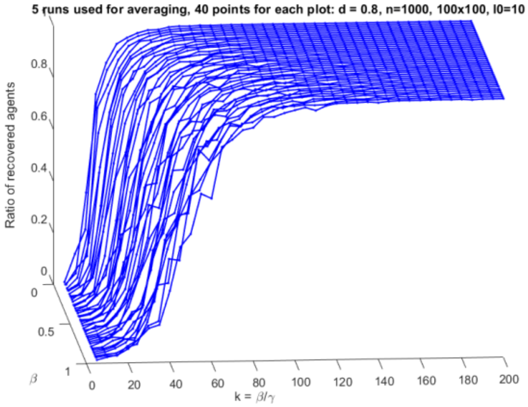
Parameters used:

- N=1000
- 100x100-lattice
- 40 different values of β and 40 different values of γ for each β (1600 points)
- 5 runs for each combination were used for averaging

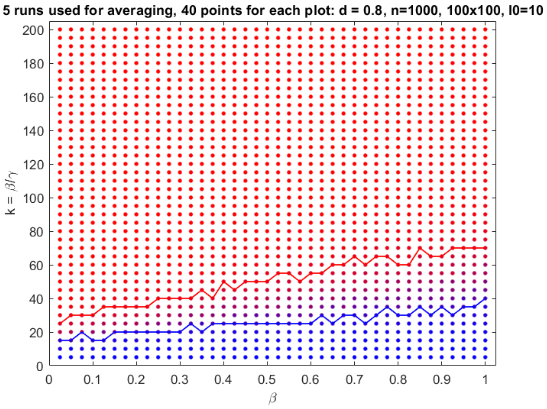
3D Model



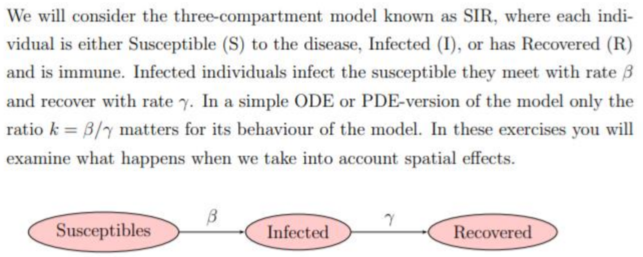
3D Model



2D Projection on β-k plane



ODE version of the SIR model



We will consider the three-compartment model known as SIR, where each individual is either Susceptible (S) to the disease, Infected (I), or has Recovered (R) and is immune. Infected individuals infect the susceptible they meet with rate β and recover with rate γ. In a simple ODE or PDE-version of the model only the ratio $k = \beta/\gamma$ matters for its behaviour of the model. In these exercises you will examine what happens when we take into account spatial effects.

Discussion

- Increasing k increases the ratio of recovered agents
- However, for higher values of β, the ratio of recovered agents increases slower
- For same values of k, different values of β will have different values of γ (proportional)
- High β increases tendency to population-wide spreading, but a high γ has a opposite effect
- It appears that the recovery speed is more effective in delaying the spreading.
- In conclusion, it is not true that the model depends only on k= β/ γ. It also depends on the parameters.

```
%%EXERCISE 1%%
%a)

%Random initial position (matrix 100x100)
p=zeros(1000,2);
p(1,:) = [randi(100);randi(100)];

%Move it and save the positions
for i=2:1000
   p(i,:) = p(i-1,:);
  if (p(i-1,1) == 1 | p(i-1,1) == 100)
     if (p(i-1,2) == 1 | p(i-1,2) == 100) %%Can only move in 2 directions
        direction = randi(2);
        p(i,direction) = p(i-1,direction)+1;
        if p(i,direction)>100
           p(i,direction) = p(i,direction)-2;
        end
     else %%Can move either way on axis 2 and to the center on axis 1 (1-> -1 in 2, 2 -> 1, 3-> 1 in 2)
        direction = randi(3);
        if direction == 1
           p(i,2) = p(i-1,2)-1;
        elseif direction == 2
           p(i,1) = p(i-1,1)+1;
           if p(i,1) > 100
              p(i,1) = p(i,1)-2;
           end
        else p(i,2) = p(i-1,2)+1;
        end
     end
  elseif(p(i-1,2) == 1 | p(i-1,2) == 100)
     %%Can move either way on axis 1 and to the center on axis 2 (1-> -1 in 2, 2 -> 1, 3-> 1 in 2)
     direction = randi(3);
     if direction == 1
        p(i,1) = p(i-1,1)-1;
     elseif direction == 2
        p(i,2) = p(i-1,2)+1;
        if p(i,2) > 100
           p(i,2) = p(i,2)-2;
        end
     else p(i,1) = p(i-1,1)+1;
     end
  else %It can move everywhere (1-> decreases axis 1, 2->increases axis 2, 3->increases axis 1, 4->
decreases axis 2)
     direction = randi(4);
     if (direction == 1)
        p(i,1) = p(i-1,1)-1;
     elseif (direction == 2)
        p(i,2) = p(i-1,2)+1;
        elseif (direction == 3)
            p(i,1) = p(i-1,1)+1;
        else
            p(i,2) = p(i-1,2)-1;
        end
    end
end

% Plot
figure (1);
plot(p(:,1),p(:,2), '.');
xlim([0 100]);
ylim([0 100]);

al = animatedline('Marker','.');
for i=1:1000
    addpoints(al,p(i,1),p(i,2));
    drawnow;
end

%%b)
n=40;
gamma = 0.01;
beta = 0.6;
d = 0.8;
I0 = 0.1*n; %%Initially infected
R=0;
Sus = zeros(2000);
In = zeros(2000);
Rec = zeros(2000);
In(1) = I0;
Rec(1) = R;
Sus(1) = n-In(1)-Rec(1);
max = 20;

%Place them in the starting  random positions
list=zeros(max,max); %% list that checks if there are anyone susceptible in one of the positions
p=zeros(n,3);
for i=1:n
    p(i,1) = randi(max);
    p(i,2) = randi(max);
    if(i<=I0)
        p(i,3) = 2; %%1 means susceptible, 2 is infected and 3 is recovered
    else
        p(i,3) = 1;
        list(p(i,1),p(i,2)) = 1;
    end
end
```

```
h=1;
%Cycle until there are no more Infected people
while(I0>0)
  %Check for new infections and recoveries
  for j=1:n
    if(p(j,3) == 2)
      if (rand(1)<= beta)
       for k=1:n
         if(p(k,1) == p(j,1) & p(k,2) == p(j,2) & p(k,3) == 1)
           p(k,3) = 2;
           list(p(j,1),p(j,2)) = 0;
           I0=I0+1;
         end
       end
      end
      if (rand(1)<= gamma)
        p(j,3) = 3;
        I0=I0-1;
        R = R+1;
      end
    end
  end

  list=zeros(max,max); %% list that checks if there are anyone susceptible in one of the positions
  %Move them
  for i=1:n
    if(rand(1)<= d)
      if (p(i,1) == 1 | p(i,1) == max)
        if (p(i,2) == 1 | p(i,2) == max) %%Can only move in 2 directions
          direction = randi(2);
          p(i,direction) = p(i,direction)+1;
          if p(i,direction)>max
            p(i,direction) = p(i,direction)-2;
          end
        else %%Can move either way on axis 2 and to the center on axis 1 (1-> -1 in 2, 2 -> 1, 3-> 1
in 2)
          direction = randi(3);
          if direction == 1
            p(i,2) = p(i,2)-1;
          elseif direction == 2
            p(i,1) = p(i,1)+1;
            if p(i,1) > max
              p(i,1) = p(i,1)-2;
            end
          else p(i,2) = p(i,2)+1;
          end
        end
      elseif(p(i,2) == 1 | p(i,2) == max)
```

```
          %%Can move either way on axis 1 and to the center on axis 2 (1-> -1 in 2, 2 -> 1, 3-> 1 in 2)
          direction = randi(3);
          if direction == 1
            p(i,1) = p(i,1)-1;
          elseif direction == 2
            p(i,2) = p(i,2)+1;
            if p(i,2) > max
              p(i,2) = p(i,2)-2;
            end
          else p(i,1) = p(i,1)+1;
          end
        else %It can move everywhere (1-> decreases axis 1, 2->increases axis 2, 3->increases axis 1, 4-> decreases axis 2)
          direction = randi(4);
          if (direction == 1)
            p(i,1) = p(i,1)-1;
          elseif (direction == 2)
            p(i,2) = p(i,2)+1;
          elseif (direction == 3)
            p(i,1) = p(i,1)+1;
          else
            p(i,2) = p(i,2)-1;
          end
        end
      end
      if (p(i,3) == 1)
        list(p(i,1),p(i,2)) = 1;
      end
    end
    h=h+1;
    In(h) = I0;
    Rec(h) = R;
    Sus(h) = n-In(h)-Rec(h);

    if(h==100)
      Inpos=zeros(I0,2);
      Recpos=zeros(R,2);
      Suspos=zeros(n-I0-R,2);
      c1=1;
      c2=1;
      c3=1;
      for i=1:n
        switch p(i,3)
          case 1
            Suspos(c1,1) = p(i,1);
            Suspos(c1,2) = p(i,2);
            c1 = c1 + 1;
          case 2
```

```
                    Inpos(c2,1) = p(i,1);
                    Inpos(c2,2) = p(i,2);
                    c2 = c2 + 1;
                case 3
                    Recpos(c3,1) = p(i,1);
                    Recpos(c3,2) = p(i,2);
                    c3 = c3 + 1;
            end
        end
    end
end
In = In(1:h);
Rec = Rec(1:h);
Sus = Sus(1:h);

ti=sprintf('d = %.1f, \\beta = %.1f, \\gamma = %.2f', d, beta, gamma);

%Plot
figure(2);
subplot(1,2,1);
plot(Suspos(:,1),Suspos(:,2),'b.');
hold on;
plot(Inpos(:,1),Inpos(:,2),'r.');
hold on;
plot(Recpos(:,1),Recpos(:,2),'g.');
xlabel('x');
ylabel('y');
title('t=100');
subplot(1,2,2);
plot(In, 'r');
hold on;
plot(Rec, 'g');
hold on;
plot(Sus, 'b');
xlabel('Time Steps');
ylabel('Number of agents');
title(ti);

%%c)
n=1000;
gamma = 0.01;
beta = 0.6;
d = 0.8;
I0 = 0.1*n; %%Initially infected
R=0;
Sus = zeros(5000);
In = zeros(5000);
Rec = zeros(5000);
```

```
In(1) = I0;
Rec(1) = R;
Sus(1) = n-In(1)-Rec(1);
max = 100;

%Place them in the starting  random positions
list=zeros(max,max); %% list that checks if there are anyone susceptible in one of the positions
p=zeros(n,3);
for i=1:n
  p(i,1) = randi(max);
  p(i,2) = randi(max);
  if(i<=I0)
    p(i,3) = 2; %%1 means susceptible, 2 is infected and 3 is recovered
  else
    p(i,3) = 1;
    list(p(i,1),p(i,2)) = 1;
  end
end
h=1;
%Cycle until there are no more Infected people
while(I0>0)
  %Check for new infections and recoveries
  for j=1:n
    if(p(j,3) == 2)
      if (rand(1)<= beta)
       for k=1:n
         if(p(k,1) == p(j,1) & p(k,2) == p(j,2) & p(k,3) == 1)
           p(k,3) = 2;
           list(p(j,1),p(j,2)) = 0;
           I0=I0+1;
         end
       end
      end
      if (rand(1)<= gamma)
        p(j,3) = 3;
        I0=I0-1;
        R = R+1;
      end
    end
  end

  list=zeros(max,max); %% list that checks if there are anyone susceptible in one of the positions
  %Move them
  for i=1:n
    if(rand(1)<= d)
      if (p(i,1) == 1 | p(i,1) == max)
        if (p(i,2) == 1 | p(i,2) == max) %%Can only move in 2 directions
          direction = randi(2);
```

```
                   p(i,direction) = p(i,direction)+1;
                   if p(i,direction)>max
                       p(i,direction) = p(i,direction)-2;
                   end
               else %%Can move either way on axis 2 and to the center on axis 1 (1-> -1 in 2, 2 -> 1, 3-> 1
in 2)
                   direction = randi(3);
                   if direction == 1
                       p(i,2) = p(i,2)-1;
                   elseif direction == 2
                       p(i,1) = p(i,1)+1;
                       if p(i,1) > max
                           p(i,1) = p(i,1)-2;
                       end
                   else p(i,2) = p(i,2)+1;
                   end
               end
           elseif(p(i,2) == 1 | p(i,2) == max)
               %%Can move either way on axis 1 and to the center on axis 2 (1-> -1 in 2, 2 -> 1, 3-> 1 in 2)
               direction = randi(3);
               if direction == 1
                   p(i,1) = p(i,1)-1;
               elseif direction == 2
                   p(i,2) = p(i,2)+1;
                   if p(i,2) > max
                       p(i,2) = p(i,2)-2;
                   end
               else p(i,1) = p(i,1)+1;
               end
           else %It can move everywhere (1-> decreases axis 1, 2->increases axis 2, 3->increases axis 1,
4-> decreases axis 2)
               direction = randi(4);
               if (direction == 1)
                   p(i,1) = p(i,1)-1;
               elseif (direction == 2)
                   p(i,2) = p(i,2)+1;
               elseif (direction == 3)
                   p(i,1) = p(i,1)+1;
               else
                   p(i,2) = p(i,2)-1;
               end
            end
        end
        if (p(i,3) == 1)
            list(p(i,1),p(i,2)) = 1;
        end
    end
    h=h+1;
    In(h) = I0;
    Rec(h) = R;
    Sus(h) = n-In(h)-Rec(h);
    if(h==100)
        Inpos=zeros(I0,2);
        Recpos=zeros(R,2);
        Suspos=zeros(n-I0-R,2);
        c1=1;
        c2=1;
        c3=1;
        for i=1:n
            switch p(i,3)
                case 1
                    Suspos(c1,1) = p(i,1);
                    Suspos(c1,2) = p(i,2);
                    c1 = c1 + 1;
                case 2
                    Inpos(c2,1) = p(i,1);
                    Inpos(c2,2) = p(i,2);
                    c2 = c2 + 1;
                case 3
                    Recpos(c3,1) = p(i,1);
                    Recpos(c3,2) = p(i,2);
                    c3 = c3 + 1;
            end
        end
    end
end
In = In(1:h);
Rec = Rec(1:h);
Sus = Sus(1:h);

ti=sprintf('d = %.1f, \\beta = %.1f, \\gamma = %.2f', d, beta, gamma);

%Plot
figure(3);
subplot(1,2,1);
plot(Suspos(:,1),Suspos(:,2),'b.');
hold on;
plot(Inpos(:,1),Inpos(:,2),'r.');
hold on;
plot(Recpos(:,1),Recpos(:,2),'g.');
xlabel('x');
ylabel('y');
title('t=100');
subplot(1,2,2);
plot(In, 'r');
hold on;
```

```
plot(Rec, 'g');
hold on;
plot(Sus, 'b');
xlabel('Time Steps');
ylabel('Number of agents');
title(ti);



%%Exercise 2
%Population wise disease-spreading: high beta and low gamma
n=1000;
gamma = 0.005;
beta = 0.8;
d = 0.8;
I0 = 0.1*n; %%Initially infected
R=0;
Sus = zeros(5000);
In = zeros(5000);
Rec = zeros(5000);
In(1) = I0;
Rec(1) = R;
Sus(1) = n-In(1)-Rec(1);
max = 100;

%Place them in the starting  random positions
list=zeros(max,max); %% list that checks if there are anyone susceptible in one of the positions
p=zeros(n,3);
for i=1:n
  p(i,1) = randi(max);
  p(i,2) = randi(max);
  if(i<=I0)
    p(i,3) = 2; %%1 means susceptible, 2 is infected and 3 is recovered
  else
    p(i,3) = 1;
    list(p(i,1),p(i,2)) = 1;
  end
end
h=1;
%Cycle until there are no more Infected people
while(I0>0)
  %Check for new infections and recoveries
  for j=1:n
    if(p(j,3) == 2)
      if (rand(1)<= beta)
       for k=1:n
         if(p(k,1) == p(j,1) & p(k,2) == p(j,2) & p(k,3) == 1)
           p(k,3) = 2;
```

```
                      list(p(j,1),p(j,2)) = 0;
                      I0=I0+1;
                  end
              end
            end
            if (rand(1)<= gamma)
                p(j,3) = 3;
                I0=I0-1;
                R = R+1;
            end
        end
    end

    list=zeros(max,max); %% list that checks if there are anyone susceptible in one of the positions
    %Move them
    for i=1:n
        if(rand(1)<= d)
            if (p(i,1) == 1 | p(i,1) == max)
               if (p(i,2) == 1 | p(i,2) == max) %%Can only move in 2 directions
                   direction = randi(2);
                   p(i,direction) = p(i,direction)+1;
                   if p(i,direction)>max
                       p(i,direction) = p(i,direction)-2;
                   end
               else %%Can move either way on axis 2 and to the center on axis 1 (1-> -1 in 2, 2 -> 1, 3-> 1
in 2)
                   direction = randi(3);
                   if direction == 1
                       p(i,2) = p(i,2)-1;
                   elseif direction == 2
                       p(i,1) = p(i,1)+1;
                       if p(i,1) > max
                          p(i,1) = p(i,1)-2;
                       end
                   else p(i,2) = p(i,2)+1;
                   end
               end
           elseif(p(i,2) == 1 | p(i,2) == max)
               %%Can move either way on axis 1 and to the center on axis 2 (1-> -1 in 2, 2 -> 1, 3-> 1 in 2)
               direction = randi(3);
               if direction == 1
                   p(i,1) = p(i,1)-1;
               elseif direction == 2
                   p(i,2) = p(i,2)+1;
                   if p(i,2) > max
                       p(i,2) = p(i,2)-2;
                   end
               else p(i,1) = p(i,1)+1;
               end
           else %It can move everywhere (1-> decreases axis 1, 2->increases axis 2, 3->increases axis 1,
4-> decreases axis 2)
               direction = randi(4);
               if (direction == 1)
                   p(i,1) = p(i,1)-1;
               elseif (direction == 2)
                   p(i,2) = p(i,2)+1;
               elseif (direction == 3)
                   p(i,1) = p(i,1)+1;
               else
                   p(i,2) = p(i,2)-1;
               end
            end
        end
        if (p(i,3) == 1)
            list(p(i,1),p(i,2)) = 1;
        end
    end
    h=h+1;
    In(h) = I0;
    Rec(h) = R;
    Sus(h) = n-In(h)-Rec(h);
    if(h==100)
        Inpos=zeros(I0,2);
        Recpos=zeros(R,2);
        Suspos=zeros(n-I0-R,2);
        c1=1;
        c2=1;
        c3=1;
        for i=1:n
            switch p(i,3)
                case 1
                    Suspos(c1,1) = p(i,1);
                    Suspos(c1,2) = p(i,2);
                    c1 = c1 + 1;
                case 2
                    Inpos(c2,1) = p(i,1);
                    Inpos(c2,2) = p(i,2);
                    c2 = c2 + 1;
                case 3
                    Recpos(c3,1) = p(i,1);
                    Recpos(c3,2) = p(i,2);
                    c3 = c3 + 1;
            end
        end
    end
end
```

```
In = In(1:h);
Rec = Rec(1:h);
Sus = Sus(1:h);

ti=sprintf('d = %.1f, \\beta = %.1f, \\gamma = %.2f', d, beta, gamma);

%Plot
figure(4);
subplot(1,2,1);
plot(Suspos(:,1),Suspos(:,2),'b.');
hold on;
plot(Inpos(:,1),Inpos(:,2),'r.');
hold on;
plot(Recpos(:,1),Recpos(:,2),'g.');
xlabel('x');
ylabel('y');
title('t=100');
subplot(1,2,2);
plot(In, 'r');
hold on;
plot(Rec, 'g');
hold on;
plot(Sus, 'b');
xlabel('Time Steps');
ylabel('Number of agents');
title(ti);

%Limited disease spreading: low beta and high gamma
n=1000;
gamma = 0.1;
beta = 0.2;
d = 0.8;
I0 = 0.1*n; %%Initially infected
R=0;
Sus = zeros(5000);
In = zeros(5000);
Rec = zeros(5000);
In(1) = I0;
Rec(1) = R;
Sus(1) = n-In(1)-Rec(1);
max = 100;

%Place them in the starting  random positions
list=zeros(max,max); %% list that checks if there are anyone susceptible in one of the positions
p=zeros(n,3);
for i=1:n
   p(i,1) = randi(max);
   p(i,2) = randi(max);
   if(i<=I0)
      p(i,3) = 2; %%1 means susceptible, 2 is infected and 3 is recovered
   else
      p(i,3) = 1;
      list(p(i,1),p(i,2)) = 1;
   end
end
h=1;
%Cycle until there are no more Infected people
while(I0>0)
   %Check for new infections and recoveries
   for j=1:n
      if(p(j,3) == 2)
         if (rand(1)<= beta)
          for k=1:n
             if(p(k,1) == p(j,1) & p(k,2) == p(j,2) & p(k,3) == 1)
                p(k,3) = 2;
                list(p(j,1),p(j,2)) = 0;
                I0=I0+1;
             end
          end
         end
         if (rand(1)<= gamma)
            p(j,3) = 3;
            I0=I0-1;
            R = R+1;
         end
      end
   end

   list=zeros(max,max); %% list that checks if there are anyone susceptible in one of the positions
   %Move them
   for i=1:n
      if(rand(1)<= d)
         if (p(i,1) == 1 | p(i,1) == max)
           if (p(i,2) == 1 | p(i,2) == max) %%Can only move in 2 directions
              direction = randi(2);
              p(i,direction) = p(i,direction)+1;
              if p(i,direction)>max
                 p(i,direction) = p(i,direction)-2;
              end
           else %%Can move either way on axis 2 and to the center on axis 1 (1-> -1 in 2, 2 -> 1, 3-> 1
in 2)
              direction = randi(3);
              if direction == 1
                 p(i,2) = p(i,2)-1;
              elseif direction == 2
                 p(i,1) = p(i,1)+1;
```

```
                     if p(i,1) > max
                        p(i,1) = p(i,1)-2;
                     end
                  else p(i,2) = p(i,2)+1;
                  end
               end
            elseif(p(i,2) == 1 | p(i,2) == max)
               %%Can move either way on axis 1 and to the center on axis 2 (1-> -1 in 2, 2 -> 1, 3-> 1 in 2)
               direction = randi(3);
               if direction == 1
                  p(i,1) = p(i,1)-1;
               elseif direction == 2
                  p(i,2) = p(i,2)+1;
                  if p(i,2) > max
                     p(i,2) = p(i,2)-2;
                  end
               else p(i,1) = p(i,1)+1;
               end
            else %It can move everywhere (1-> decreases axis 1, 2->increases axis 2, 3->increases axis 1,
4-> decreases axis 2)
               direction = randi(4);
               if (direction == 1)
                  p(i,1) = p(i,1)-1;
               elseif (direction == 2)
                  p(i,2) = p(i,2)+1;
               elseif (direction == 3)
                  p(i,1) = p(i,1)+1;
               else
                  p(i,2) = p(i,2)-1;
               end
            end
         end
         if (p(i,3) == 1)
            list(p(i,1),p(i,2)) = 1;
         end
      end
      h=h+1;
      In(h) = I0;
      Rec(h) = R;
      Sus(h) = n-In(h)-Rec(h);
      if(h==30)
         Inpos=zeros(I0,2);
         Recpos=zeros(R,2);
         Suspos=zeros(n-I0-R,2);
         c1=1;
         c2=1;
         c3=1;
         for i=1:n
```

```
            switch p(i,3)
                case 1
                    Suspos(c1,1) = p(i,1);
                    Suspos(c1,2) = p(i,2);
                    c1 = c1 + 1;
                case 2
                    Inpos(c2,1) = p(i,1);
                    Inpos(c2,2) = p(i,2);
                    c2 = c2 + 1;
                case 3
                    Recpos(c3,1) = p(i,1);
                    Recpos(c3,2) = p(i,2);
                    c3 = c3 + 1;
            end
        end
    end
end
In = In(1:h);
Rec = Rec(1:h);
Sus = Sus(1:h);

ti=sprintf('d = %.1f, \\beta = %.1f, \\gamma = %.2f', d, beta, gamma);

%Plot
figure(5);
subplot(1,2,1);
plot(Suspos(:,1),Suspos(:,2),'b.');
hold on;
plot(Inpos(:,1),Inpos(:,2),'r.');
hold on;
plot(Recpos(:,1),Recpos(:,2),'g.');
xlabel('x');
ylabel('y');
title('t=30');
subplot(1,2,2);
plot(In, 'r');
hold on;
plot(Rec, 'g');
hold on;
plot(Sus, 'b');
xlabel('Time Steps');
ylabel('Number of agents');
title(ti);

%{
%%Exercise 3
%1st run
n=1000;
```

```
b_g = 2:2:200;
beta = 0.4;
d = 0.8;
I0 = floor(0.01*n); %%Initially infected
R=0;
max = 100;
F_R_1 = zeros(100,1);
avg = 0;

for z=1:100
  for w=1:10
    %Place them in the starting  random positions
    R=0;
    I0 = floor(0.01*n);
    p=zeros(n,3);
    for i=1:n
      p(i,1) = randi(max);
      p(i,2) = randi(max);
      if(i<=I0)
        p(i,3) = 2; %%1 means susceptible, 2 is infected and 3 is recovered
      else
        p(i,3) = 1;
        list(p(i,1),p(i,2)) = 1;
      end
    end
    h=1;
    %Cycle until there are no more Infected people
    while(I0>0)
      %Check for new infections and recoveries
      for j=1:n
        if(p(j,3) == 2)
          if (rand(1)<= beta)
           for k=1:n
             if(p(k,1) == p(j,1) & p(k,2) == p(j,2) & p(k,3) == 1)
               p(k,3) = 2;
               list(p(j,1),p(j,2)) = 0;
               I0=I0+1;
             end
           end
          end
          if (rand(1)<= beta/b_g(z))
            p(j,3) = 3;
            I0=I0-1;
            R = R+1;
          end
        end
      end
```

```
        %Move them
        for i=1:n
           if(rand(1)<= d)
               if (p(i,1) == 1 | p(i,1) == max)
                 if (p(i,2) == 1 | p(i,2) == max) %%Can only move in 2 directions
                    direction = randi(2);
                    p(i,direction) = p(i,direction)+1;
                    if p(i,direction)>max
                       p(i,direction) = p(i,direction)-2;
                    end
                 else %%Can move either way on axis 2 and to the center on axis 1 (1-> -1 in 2, 2 -> 1,
3-> 1 in 2)
                    direction = randi(3);
                    if direction == 1
                       p(i,2) = p(i,2)-1;
                    elseif direction == 2
                       p(i,1) = p(i,1)+1;
                       if p(i,1) > max
                          p(i,1) = p(i,1)-2;
                       end
                    else p(i,2) = p(i,2)+1;
                    end
                 end
              elseif(p(i,2) == 1 | p(i,2) == max)
                 %%Can move either way on axis 1 and to the center on axis 2 (1-> -1 in 2, 2 -> 1, 3-> 1
in 2)
                 direction = randi(3);
                 if direction == 1
                    p(i,1) = p(i,1)-1;
                 elseif direction == 2
                    p(i,2) = p(i,2)+1;
                    if p(i,2) > max
                       p(i,2) = p(i,2)-2;
                    end
                 else p(i,1) = p(i,1)+1;
                 end
              else %It can move everywhere (1-> decreases axis 1, 2->increases axis 2, 3->increases
axis 1, 4-> decreases axis 2)
                 direction = randi(4);
                 if (direction == 1)
                    p(i,1) = p(i,1)-1;
                 elseif (direction == 2)
                    p(i,2) = p(i,2)+1;
                 elseif (direction == 3)
                    p(i,1) = p(i,1)+1;
                 else
                    p(i,2) = p(i,2)-1;
                 end
```

```
                    end
                end
                if (p(i,3) == 1)
                    list(p(i,1),p(i,2)) = 1;
                end
            end
            h=h+1;
        end
        avg = avg + R;
    end
    F_R_1(z) = avg/10;
    avg=0;
end

%2nd run
n=1000;
b_g = 2:2:200;
beta = 0.8;
d = 0.8;
I0 = floor(0.01*n); %%Initially infected
R=0;
max = 100;
F_R_2 = zeros(100,1);
avg = 0;

for z=1:100
    for w=1:10
        %Place them in the starting  random positions
        R=0;
        I0 = floor(0.01*n);
        p=zeros(n,3);
        for i=1:n
            p(i,1) = randi(max);
            p(i,2) = randi(max);
            if(i<=I0)
                p(i,3) = 2; %%1 means susceptible, 2 is infected and 3 is recovered
            else
                p(i,3) = 1;
                list(p(i,1),p(i,2)) = 1;
            end
        end
        h=1;
        %Cycle until there are no more Infected people
        while(I0>0)
            %Check for new infections and recoveries
            for j=1:n
                if(p(j,3) == 2)
                    if (rand(1)<= beta)
```

```
          for k=1:n
              if(p(k,1) == p(j,1) & p(k,2) == p(j,2) & p(k,3) == 1)
                  p(k,3) = 2;
                  list(p(j,1),p(j,2)) = 0;
                  I0=I0+1;
              end
          end
        end
        if (rand(1)<= beta/b_g(z))
            p(j,3) = 3;
            I0=I0-1;
            R = R+1;
        end
    end
end

%Move them
for i=1:n
    if(rand(1)<= d)
        if (p(i,1) == 1 | p(i,1) == max)
           if (p(i,2) == 1 | p(i,2) == max) %%Can only move in 2 directions
               direction = randi(2);
               p(i,direction) = p(i,direction)+1;
               if p(i,direction)>max
                   p(i,direction) = p(i,direction)-2;
               end
           else %%Can move either way on axis 2 and to the center on axis 1 (1-> -1 in 2, 2 -> 1, 3-> 1 in 2)
               direction = randi(3);
               if direction == 1
                   p(i,2) = p(i,2)-1;
               elseif direction == 2
                   p(i,1) = p(i,1)+1;
                   if p(i,1) > max
                       p(i,1) = p(i,1)-2;
                   end
               else p(i,2) = p(i,2)+1;
               end
           end
       elseif(p(i,2) == 1 | p(i,2) == max)
           %%Can move either way on axis 1 and to the center on axis 2 (1-> -1 in 2, 2 -> 1, 3-> 1 in 2)
           direction = randi(3);
           if direction == 1
               p(i,1) = p(i,1)-1;
           elseif direction == 2
               p(i,2) = p(i,2)+1;
               if p(i,2) > max
```

```
                               p(i,2) = p(i,2)-2;
                           end
                       else p(i,1) = p(i,1)+1;
                       end
                   else %It can move everywhere (1-> decreases axis 1, 2->increases axis 2, 3->increases
axis 1, 4-> decreases axis 2)
                       direction = randi(4);
                       if (direction == 1)
                           p(i,1) = p(i,1)-1;
                       elseif (direction == 2)
                           p(i,2) = p(i,2)+1;
                       elseif (direction == 3)
                           p(i,1) = p(i,1)+1;
                       else
                           p(i,2) = p(i,2)-1;
                       end
                    end
                end
                if (p(i,3) == 1)
                    list(p(i,1),p(i,2)) = 1;
                end
            end
            h=h+1;
        end
        avg = avg + R;
    end
    F_R_2(z) = avg/10;
    avg=0;
end

figure(6);
plot(b_g,F_R_1./n,'r','MarkerSize',10);
hold on;
plot(b_g,F_R_2./n,'b','MarkerSize',10);
legend({'\beta = 0.4','\beta = 0.8'});
xlabel('k = \beta/\gamma');
ylabel('Ratio of recovered agents');
title('10 runs used for averaging, 100 points for each plot: d = 0.8, n=1000, 100x100, I0=10');

%}

n=1000;
b_g = 5:5:200;
beta = 0.025:0.025:1;
d = 0.8;
I0 = floor(0.01*n); %%Initially infected
R=0;
max = 100;
```

```
%{
F_R = zeros(40,40);
avg = 0;
x=zeros(1600,1);
y=[b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g
b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g b_g];
y=transpose(y);
cc=1;
for i=1:40
    for j=1:40
        x(cc)= beta(i);
        cc = cc + 1;
    end
end

for b=1:40
  for z=1:40
    for w=1:5
        %Place them in the starting  random positions
        R=0;
        I0 = floor(0.01*n);
        p=zeros(n,3);
        for i=1:n
            p(i,1) = randi(max);
            p(i,2) = randi(max);
            if(i<=I0)
                p(i,3) = 2; %%1 means susceptible, 2 is infected and 3 is recovered
            else
                p(i,3) = 1;
            end
        end
        h=1;
        %Cycle until there are no more Infected people
        while(I0>0)
            %Check for new infections and recoveries
            for j=1:n
                if(p(j,3) == 2)
                    if (rand(1)<= beta(b))
                      for k=1:n
                          if(p(k,1) == p(j,1) & p(k,2) == p(j,2) & p(k,3) == 1)
                              p(k,3) = 2;
                              I0=I0+1;
                          end
                      end
                    end
                    if (rand(1)<= (beta(b)/b_g(z)))
                        p(j,3) = 3;
                        I0=I0-1;
```

```
                       R = R+1;
                   end
               end
            end

            %Move them
            for i=1:n
               if(rand(1)<= d)
                   if (p(i,1) == 1 | p(i,1) == max)
                      if (p(i,2) == 1 | p(i,2) == max) %%Can only move in 2 directions
                          direction = randi(2);
                          p(i,direction) = p(i,direction)+1;
                          if p(i,direction)>max
                              p(i,direction) = p(i,direction)-2;
                          end
                      else %%Can move either way on axis 2 and to the center on axis 1 (1-> -1 in 2, 2 -> 1,
3-> 1 in 2)
                          direction = randi(3);
                          if direction == 1
                              p(i,2) = p(i,2)-1;
                          elseif direction == 2
                              p(i,1) = p(i,1)+1;
                              if p(i,1) > max
                                  p(i,1) = p(i,1)-2;
                              end
                          else p(i,2) = p(i,2)+1;
                          end
                      end
                  elseif(p(i,2) == 1 | p(i,2) == max)
                      %%Can move either way on axis 1 and to the center on axis 2 (1-> -1 in 2, 2 -> 1, 3-> 1
in 2)
                      direction = randi(3);
                      if direction == 1
                          p(i,1) = p(i,1)-1;
                      elseif direction == 2
                          p(i,2) = p(i,2)+1;
                          if p(i,2) > max
                              p(i,2) = p(i,2)-2;
                          end
                      else p(i,1) = p(i,1)+1;
                      end
                  else %It can move everywhere (1-> decreases axis 1, 2->increases axis 2, 3->increases
axis 1, 4-> decreases axis 2)
                      direction = randi(4);
                      if (direction == 1)
                          p(i,1) = p(i,1)-1;
                      elseif (direction == 2)
                          p(i,2) = p(i,2)+1;
            elseif (direction == 3)
              p(i,1) = p(i,1)+1;
            else
              p(i,2) = p(i,2)-1;
            end
          end
        end
      end
      h=h+1;
    end
    avg = avg + R;
  end
  F_R(z,b) = avg/5;
  avg=0;
 end
 disp(b);
end

cc = 1;
z = zeros(1600,1);
for i=1:40
  for j=1:40
    z(cc)= F_R(j,i);
    cc = cc + 1;
  end
end

z = z./n;
%}
figure(1);
for i=1:40
 a = x(40*i-39 : 40*i);
 b = y(40*i-39 : 40*i);
 c = z(40*i-39 : 40*i);
 plot3(a,b,c,'.-b', 'LineWidth', 1.2);
 hold on;
end
xlabel('\beta');
ylabel('k = \beta/\gamma');
zlabel('Ratio of recovered agents');
title('5 runs used for averaging, 40 points for each plot: d = 0.8, n=1000, 100x100, I0=10');

figure(2);
for i=1:1600
 plot(x(i),y(i),'.','MarkerSize',10,'MarkerEdgeColor',[z(i),0,1-z(i)]);
 hold on;
end
```

```
red=zeros(40,2);
blue=zeros(40,2);

for i=1:40
  for j=1:40
    if (blue(i,1) == 0 & z((i-1)*40+j)>=0.15)
     blue(i,1) = x((i-1)*40+j);
     blue(i,2) = y((i-1)*40+j);
    elseif (red(i,1) == 0 & z((i-1)*40+j)>=0.85)
     red(i,1) = x((i-1)*40+j);
     red(i,2) = y((i-1)*40+j);
    end
  end
end
plot(blue(:,1),blue(:,2),'b','LineWidth',1,'MarkerSize',20);
hold on;
plot(red(:,1),red(:,2),'r','LineWidth',1,'MarkerSize',20);
xlabel('\beta');
ylabel('k = \beta/\gamma');
title('5 runs used for averaging, 40 points for each plot: d = 0.8, n=1000, 100x100, I0=10');
xlim([0 1.025]);
ylim([0 205]);




%{
figure(6);
plot(0.4./gamma,F_R_1./n,'r');
hold on;
plot(0.8./gamma,F_R./n,'b');
xlim([10 250]);
%}
```