UPPSALA
UNIVERSITET

Modelling complex systems

# Project 1:
# A firing brain & Spread of memes

Peili Guo
Peili.Guo.7645@student.uu.se

April 9, 2018

# 1 A firing brain

In this part, a simple program in matlab was written to simulate a firing brain with two-dimensional cellular automata model on a N by N grid with periodic boundary conditions.

each grid represents a neuron. there are 3 different states for neuron: ready(0), firing(1), and resting(2). the rules for transit to the next time steps are:

1. a ready neuron fires on the next time step if there are exactly two neighbours that are firing. ( $0 \rightarrow 1$, if two neighbours are 1).

2. a firing neuron goes to the resting state on the next time step ($1 \rightarrow 2$).

3. a firing neuron goes to ready state on the next time step ($2 \rightarrow 0$).

## 1.1 simple simulation in matlab

Below are the results for a 40 x 40 grid where initially each cell has a probability of 0.3 being a firing(1) cell and all other neurons are ready. the figures below shows how the cells looks like after 10, 20, 100 and 1000 time steps.

The initial probability of being in a firing state is 0.3, which means at t = 0, there are around 480 cells that are being in a firing state. as we can see in figure 5, the total number of firing cell decreases over time. In the beginning, the number of cell decreases very fast, and the total number of cells gets stable around t = 300 400. When simulate 100 times to t = 1000, the average firing cell at t = 400 is around 14 and at t = 1000, the average firing cell is around 12. At the equilibrium state, the shapes that remains are travelling forward at a constant rate preserving the same shape either in the same direction(up/down or left/right), or will never interact if there are shape that travel in the up/down direction and others in left/right direction. Over 100 simulation, the curve of average firing cell decreases in an exponential model, and a exponential model of y = a* exp(b*x) + c*exp(d*x) was fitted to the curve in figure 6.

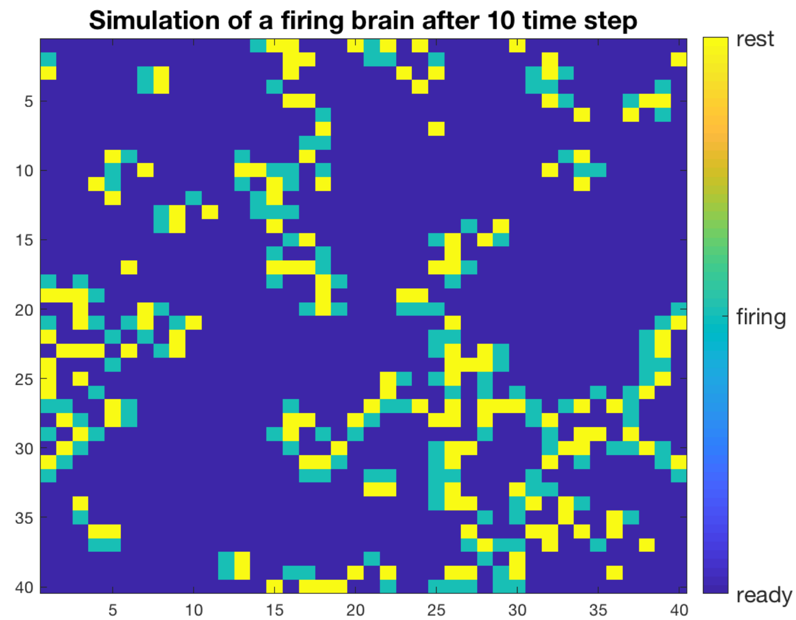


Figure 1: 40x40 cell grid at t = 10

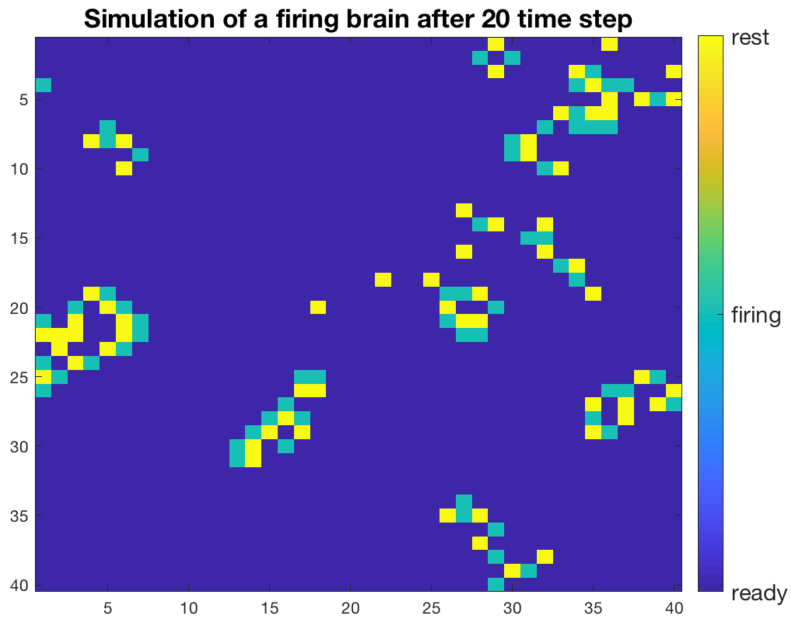


Figure 2: 40x40 cell grid at t = 20

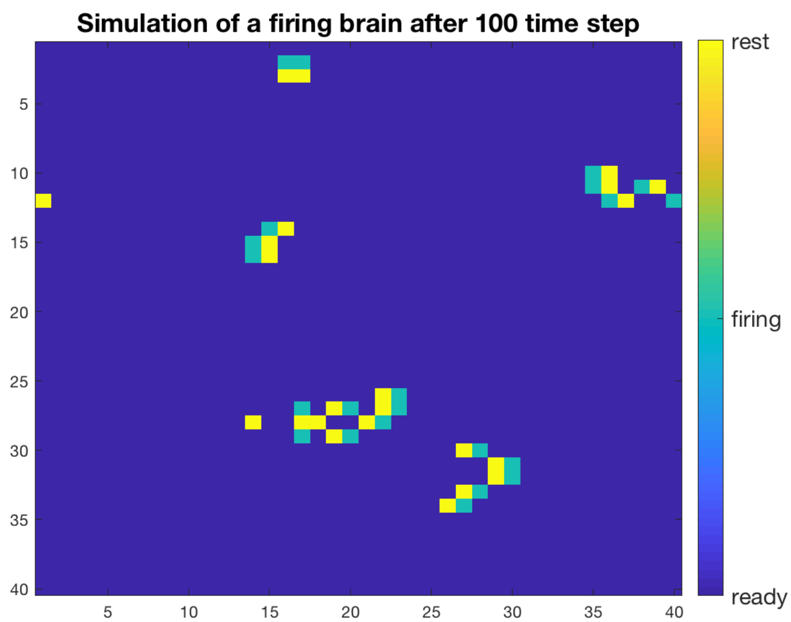


Figure 3: 40x40 cell grid at t = 100

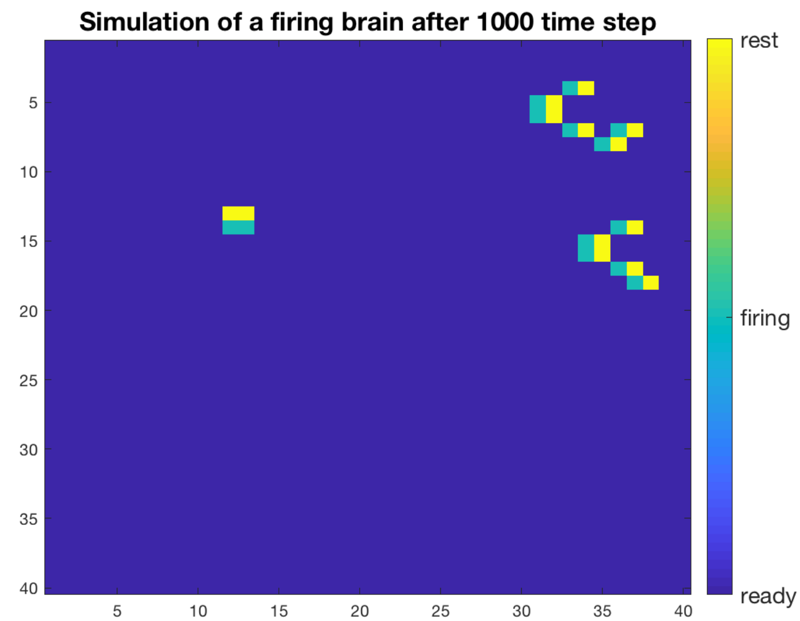


Figure 4: 40x40 cell grid at t = 1000

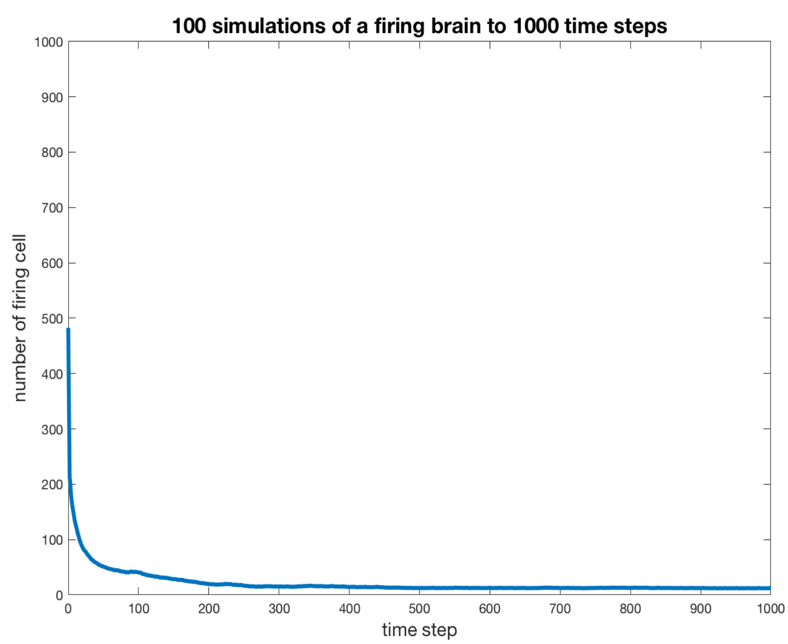


Figure 5: average number of firing cells over time with 100 simulation of different initial conditions

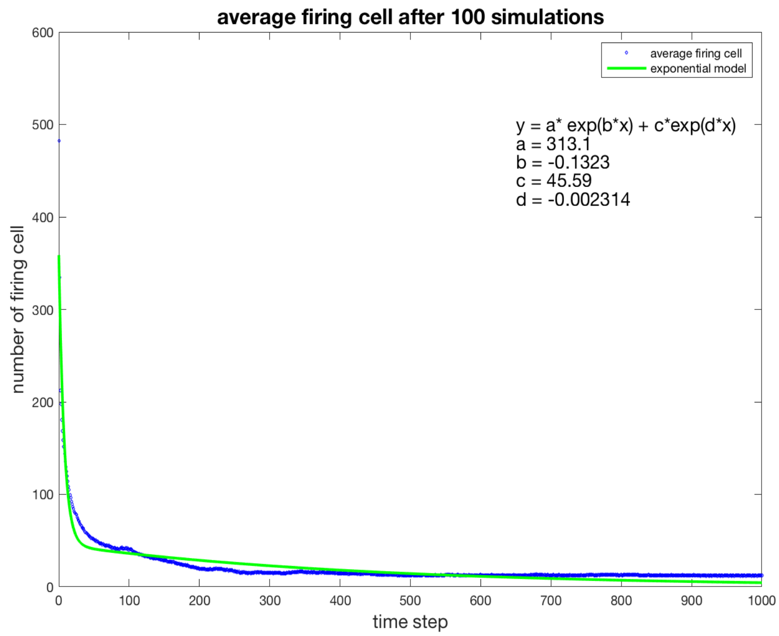


Figure 6: average number of firing cells over time with 100 simulation and exponential model fitting

and here are the links of video you can check out.

for simulation to t = 100

<https://youtu.be/CFUcpGHhj0O>

for simulation to t = 1000

<https://youtu.be/8EulLy_IRmw>

## 1.2 example of shapes

### 1.2.1 move forward at a rate of one cell per time step, while preserving the same shape

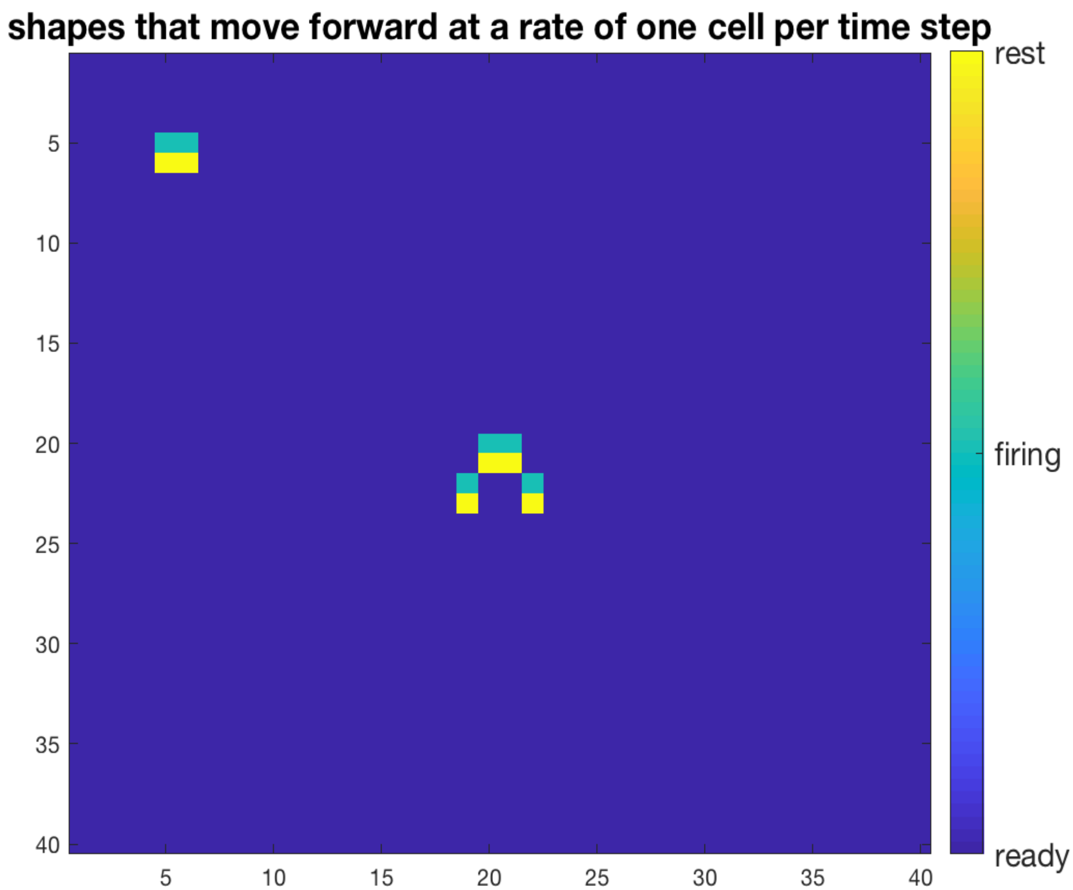


Figure 7: shapes that move forward at a rate of one cell per time step preserving the same shape

### 1.2.2 move forward at a rate of one cell per time step, launching other shapes behind them

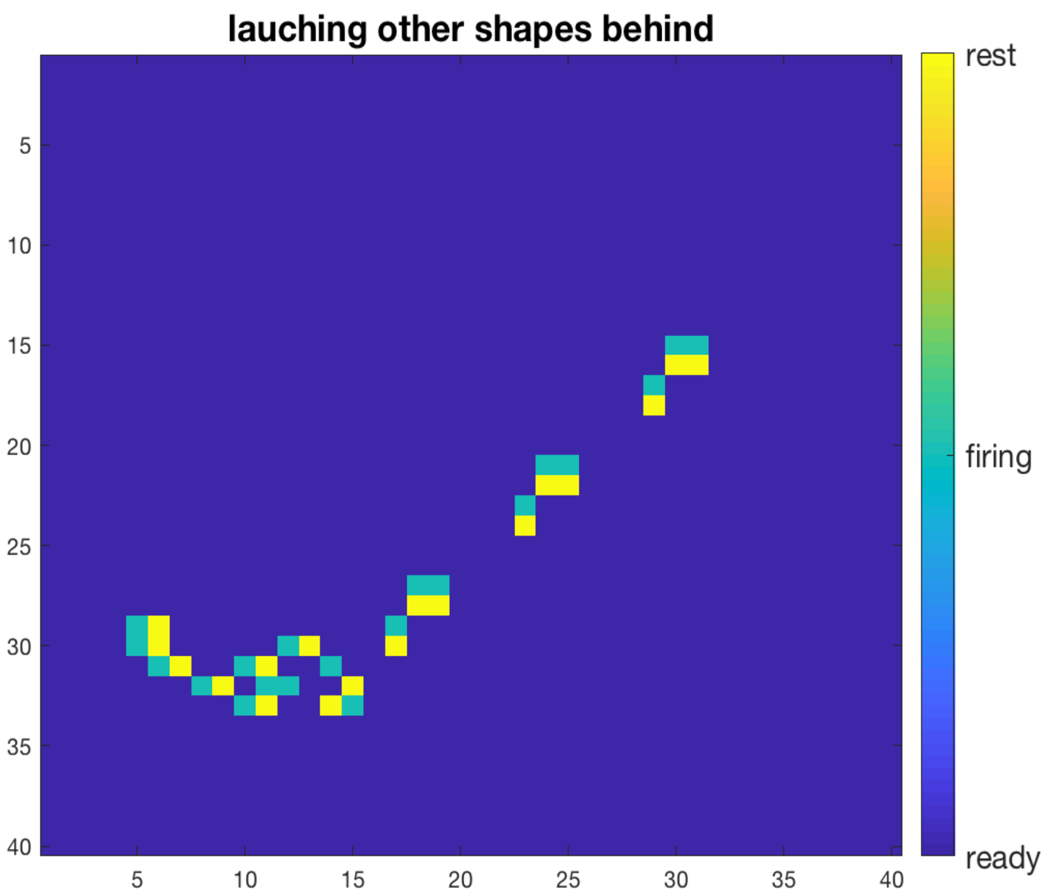


Figure 8: shapes that move forward at a rate of one cell per time step, launching other shapes behind them

### 1.2.3 move forward at a rate of less than one cell per time step, while returning to the same shape after some period

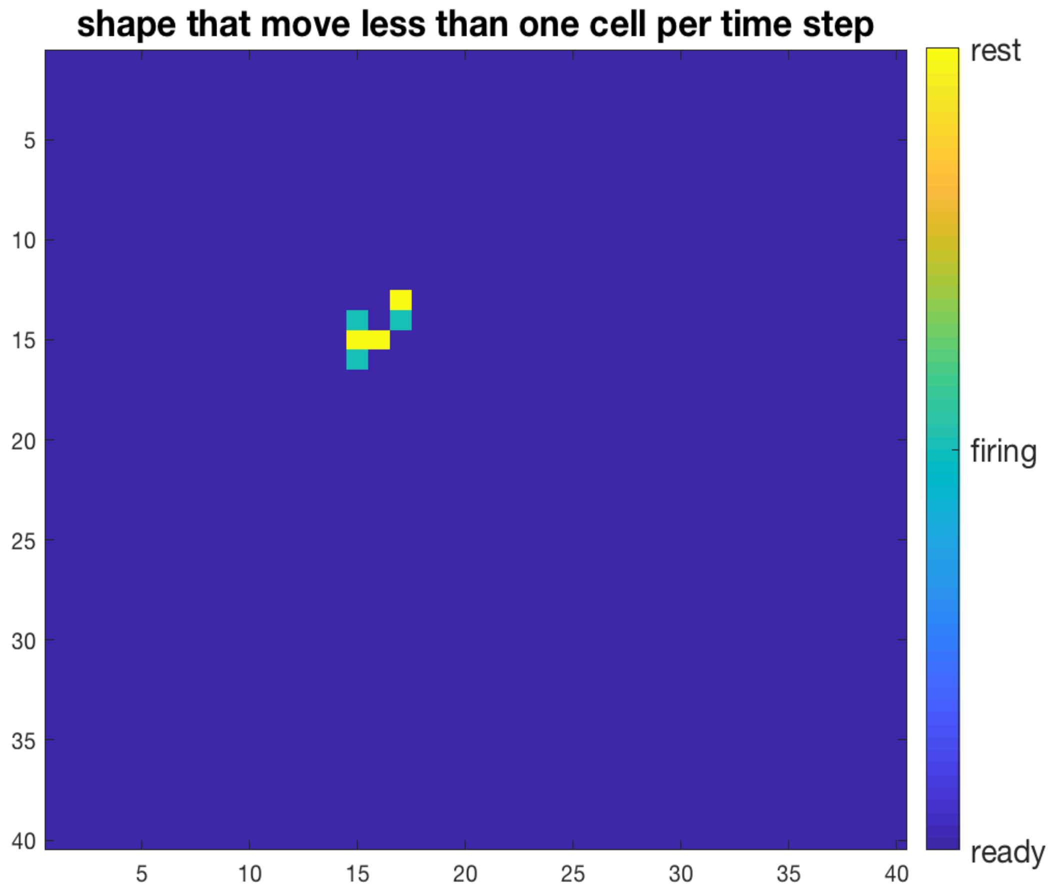


Figure 9: shapes that move less than one cell per time step, returning to same shape after some period

### 1.2.4 stay stationary but oscillate periodically

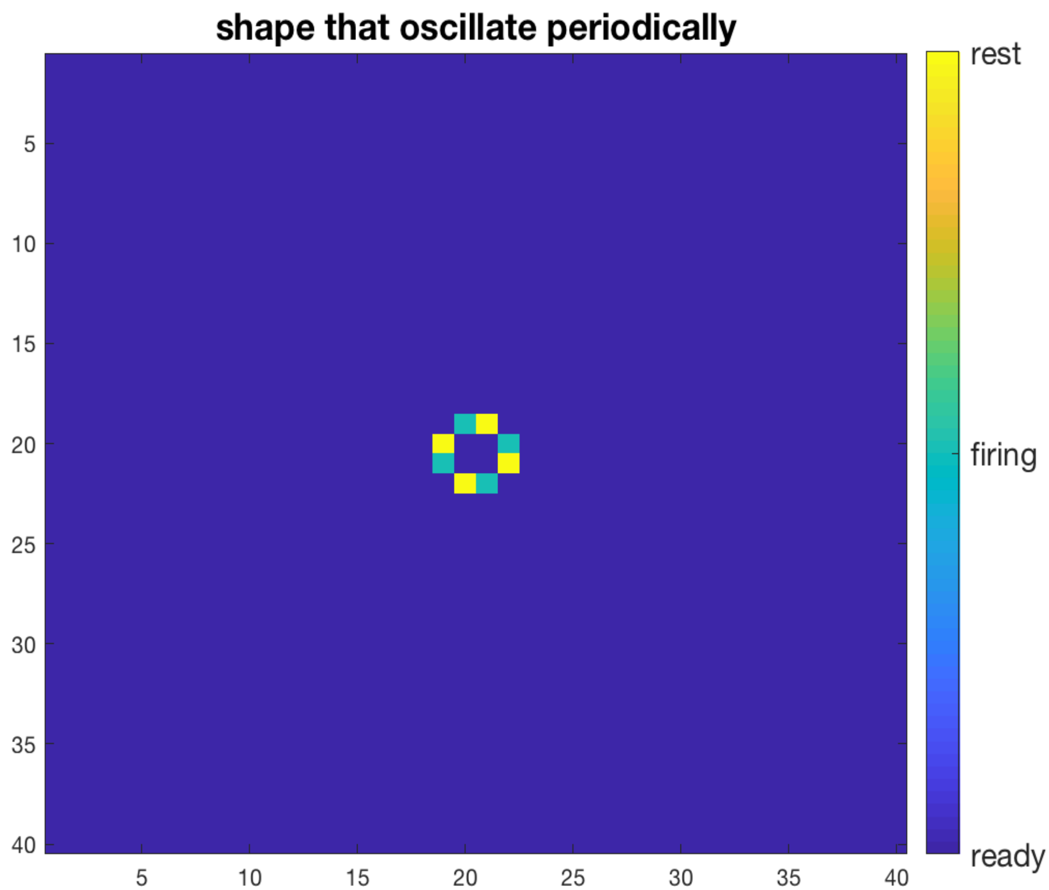


Figure 10: shapes that stay stationary but oscillate periodically

## 1.3 create cellular automata

In this part, I create my own cellular automata. I used the two model in this project (a firing brain and spread of memes), Conway's game of life and several videos I watched online showing cellular automata as reference. I create the following to simulate a population the different life stage.

The states are:

- Waiting(0)
- Growing(1)
- Reproducing(2)
- Ageing (3)
- Dead (4)

The rules for the next time steps are:

- The waiting (0) cell needs at least 2 neighbour that are in the stage of reproducing(2) to be in grow. otherwise, it stays waiting.
- The growing (1) cell needs at least 1 neighbour that is in the stage of reproducing(2) and ageing(3) to become a reproducing (2) cell in the next time step. otherwise, it stays growing.
- The reproducing(2) celll has the probability of 0.5 to be ageing and 0.5 to remain reproducing
- The ageing cell (3)has the probability of 0.5 to be dead and 0.5 to remain ageing.
- The dead cell(4) has the the probability of 0.6 to be in waiting in the next time step, and 0.4 remain dead.

To simulate, I set up the initial condition:

- each cell has the probability of 0.05 to be a growing cell(1)
- each cell has the probability of 0.05 to be a reproducing cell(2)

- each cell has the probability of 0.15 to be in ageing (3)

- each cell has the probability of 0.10 to be dead(4)

I created this to simulate life from a society that is dominated by ageing population. The population will cluster together in the beginning and then different shapes will interact with each other.

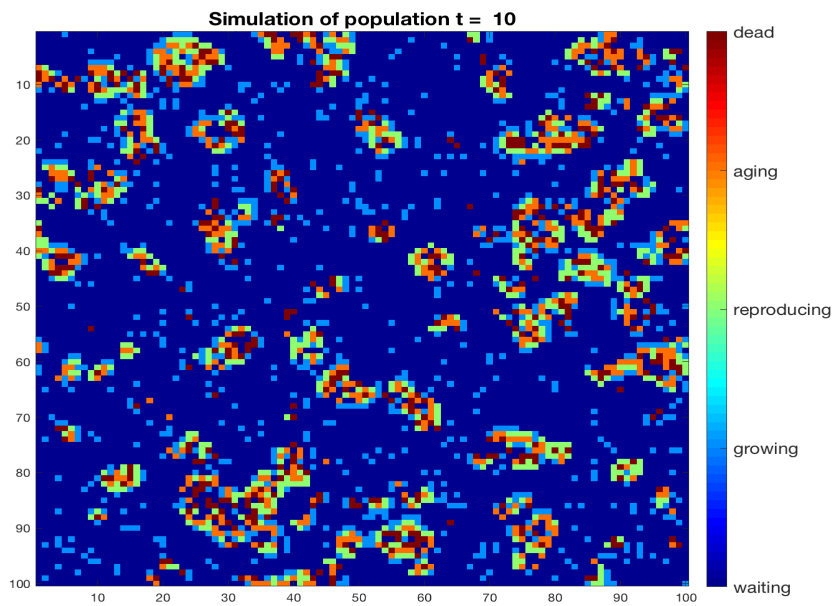


Figure 11: my cell automata at t = 10

and here are the links of video you can check out.

for simulation to t = 100

<https://youtu.be/gQ4c2FFDzDA>

for simulation to t = 1000

<https://youtu.be/OWazMF6is1c>

# 2 Spread of memes

In this part, a model is used to simulate the spread of internet memes. There are 3 different states, resting(0), sharing(1) and bored(2). The rules for the next time step are:

1. with probability p = 0.001, a person at rest will discover a new meme and become a sharer.($0 \rightarrow 1$ with p = 0.001)
2. with probability q = 0.01, a person sharing(1) will pick one person completely at random from the population to share the memes with. if the random person is at rest(0), that person will become a sharer(1), if that person is bored(2), then the sharing person will become bored(2).
3. bored(2) stays bored(2) forever. (2 is always 2).

## 2.1 some simulations in matlab

The simluation in matlab will run the model 1000 times with a population of 1000 to time at 2000 and show the change of number of resting, sharing and bored person over time. The initial condition is that there are one person sharing and one bored person. and below are the graphs showing the simulation.

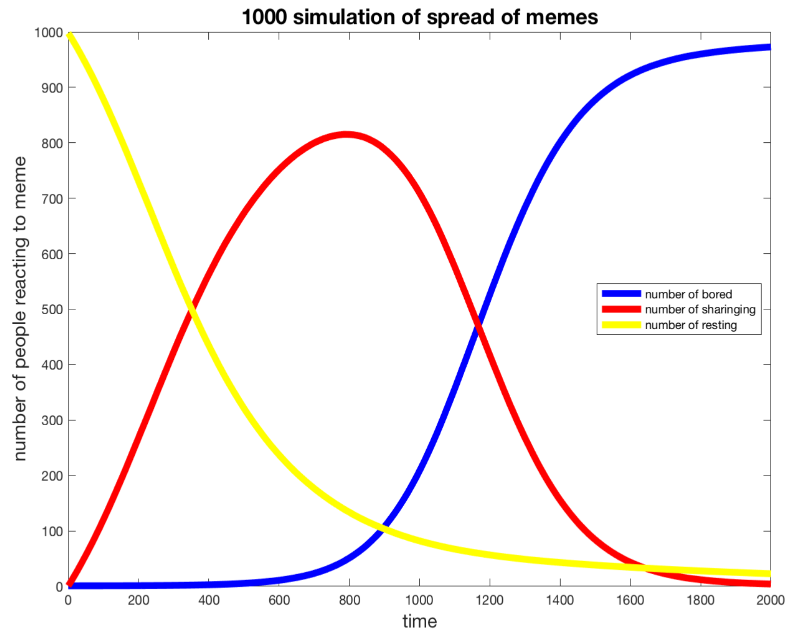


Figure 12: simulation of spread of memes showing number of bored, sharing, resting person over time

The mean field difference equation model for the sharing of meme is:

Bored(B), Sharing(S), Resting(R), population(N).

$$\begin{cases} B(t+1) = B(t) + S(t) * q * B(t)/N & (1) \\ S(t+1) = S(t) + p * R(t) - S(t) * q * B(t)/N + S(t) * q * R(t)/N & (2) \\ R(t+1) = R(t) - R(t) * p - S(t) * q * R(t)/N & (3) \end{cases}$$

The figure below shows both the simulation and the mean field model

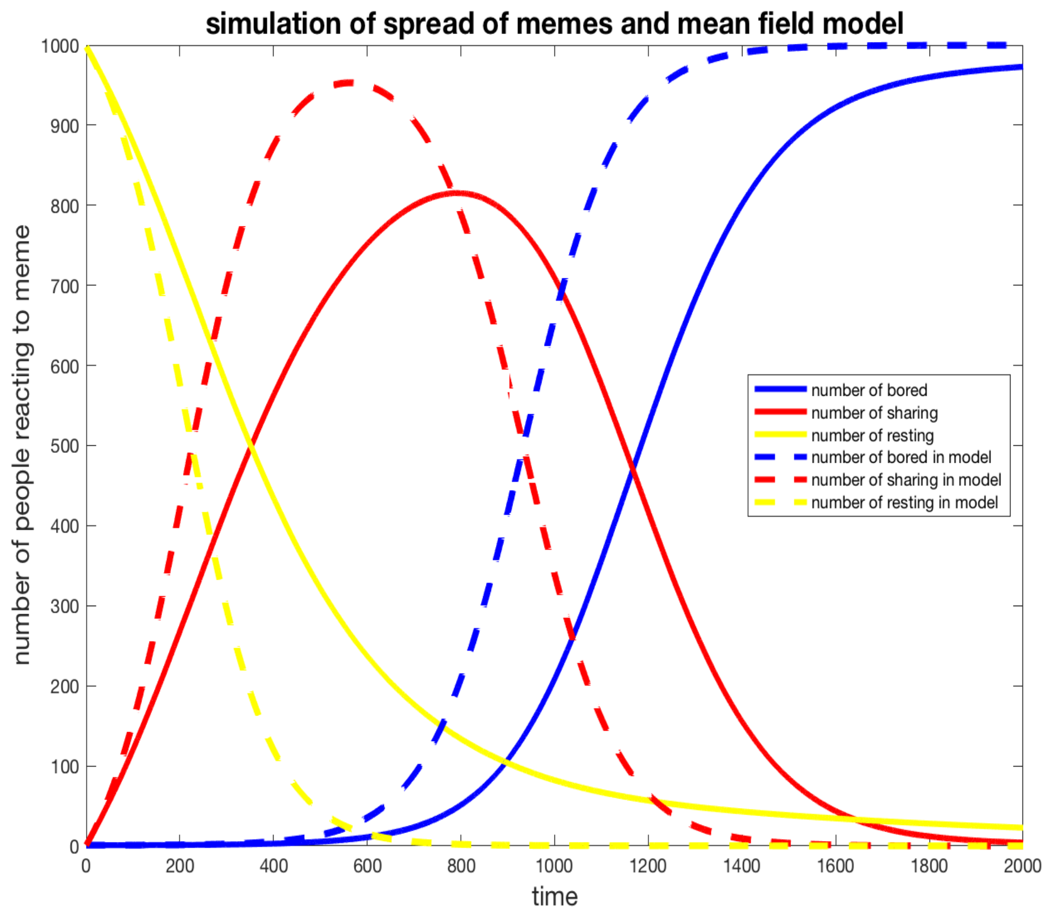


Figure 13: simulation of spread of memes showing number of bored, sharing, resting person over time with mean field model

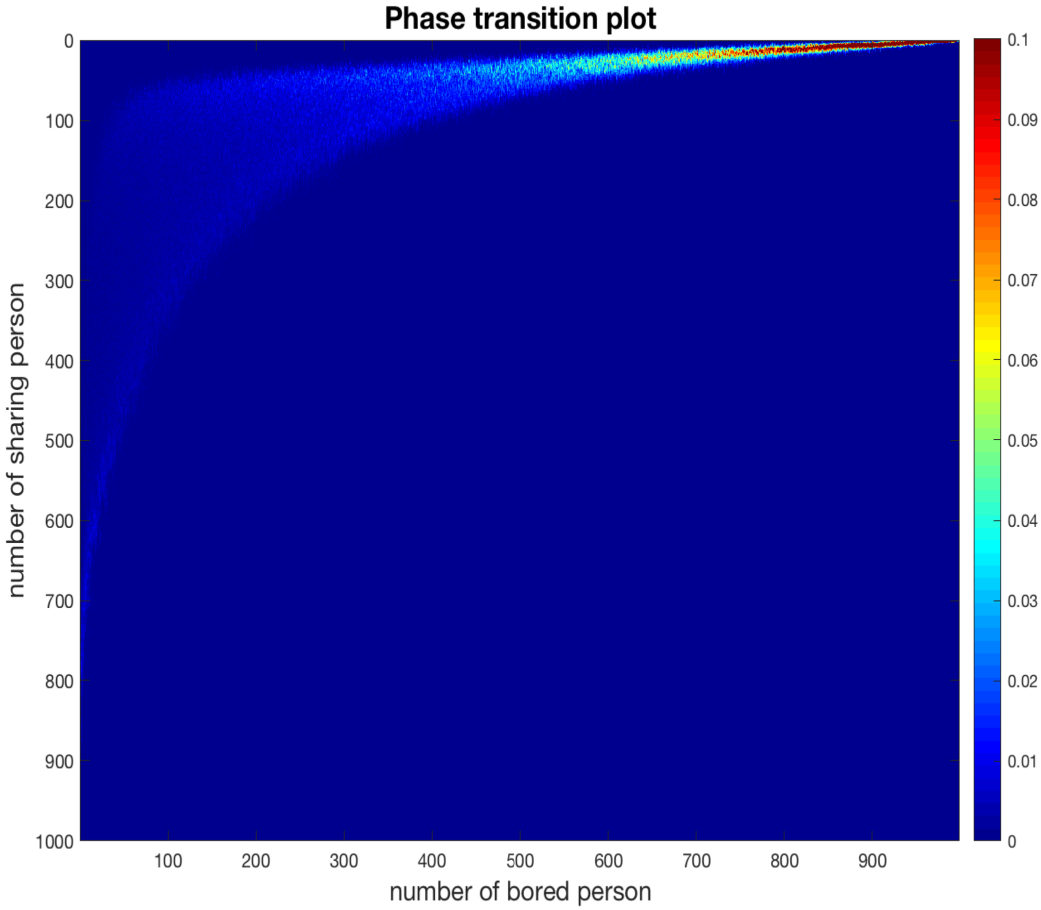


Figure 14: phase transition of total sharing person with simulation with t = 1000 and different B(0) condition

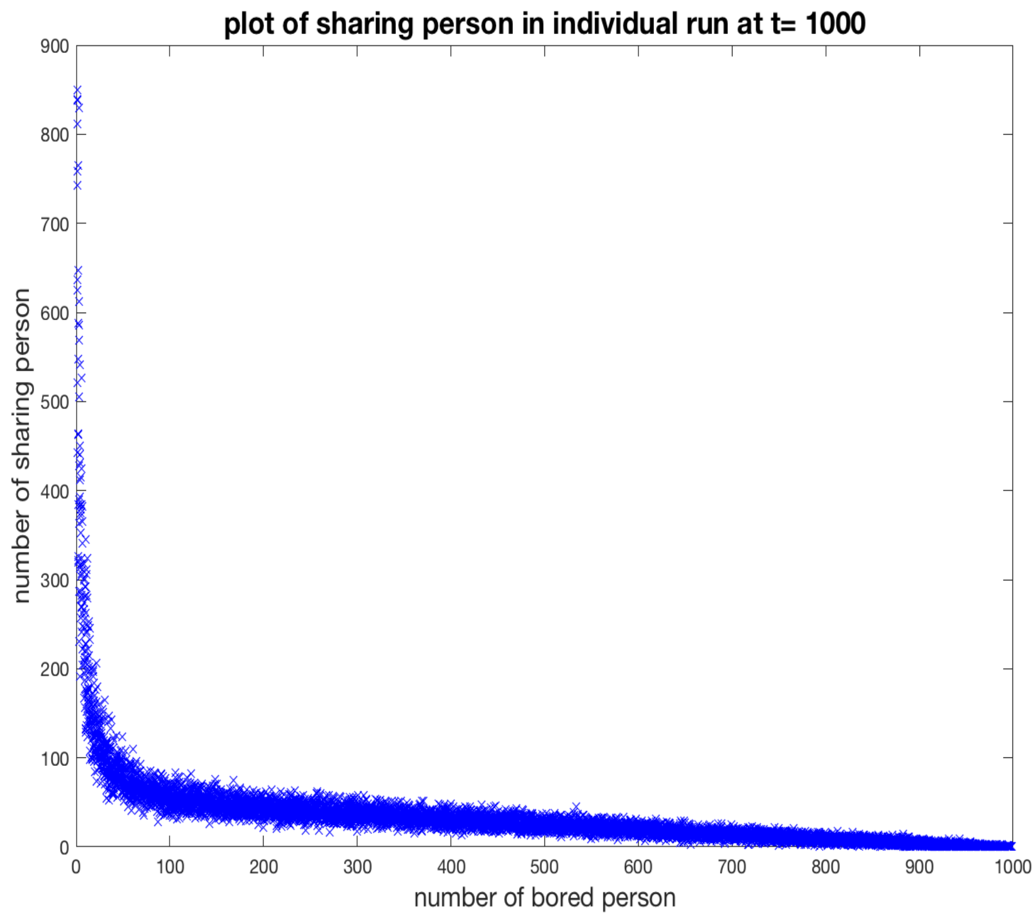


Figure 15: plot of sharing person in individual runs at t =1000 for 100 simulations of B0 = 1:1:999

To find the probability of at least 25% of the populations share a meme. I run the model with B(0) = 1:1:999 for 100 times to final time = 1000. and plot the probability in a heat map.

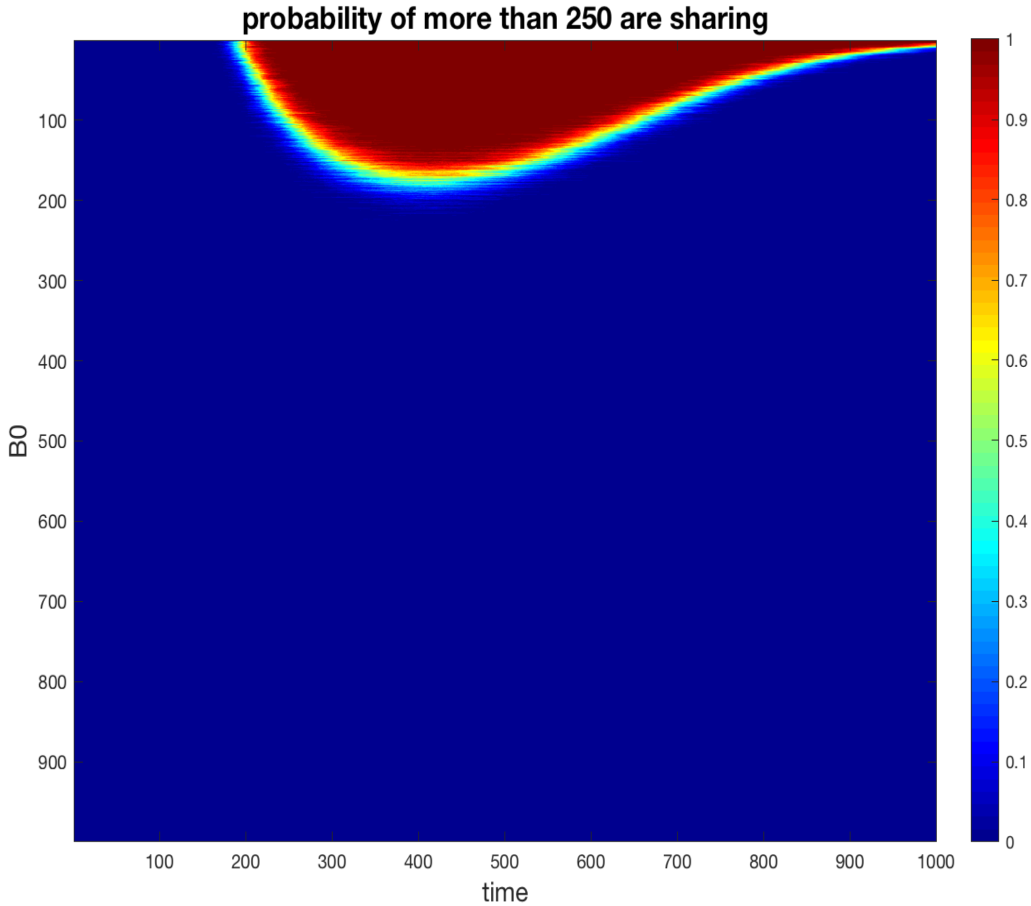


Figure 16: probability of more than 250 people are sharing with different B0 and time from 0 to 1000

## 2.2 changed the condition of a bored person

One condition was added to the bored person. A bored person will pick on person a random from the population with probability of q. If that person is resting then the bored person will become resting. otherwise she will continue to be bored. It was simulated in matlab for 1000 times with a population of 1000 to time = 2000 and at the same time plot together with the mean field model. The results were very different compared to the previous simulation. It seems that the model shows oscillation of bored person and sharing person over time but not in the simulation. In the model, we allow decimals for the number of persons. that's why the number of bored persons can slowly increase. In the simulation, it started with 1 bored person, and if that person meets someone at rest with probability of q. Then this bored person will be resting too. With this condition, a bored person soon finds a person at rest and then bored persons will be 0. The sharing person will increase gradually as a resting person will find a meme with p = 0.001. In the simulation one person can only be in one of the three states, that's why the simulations shows different results.

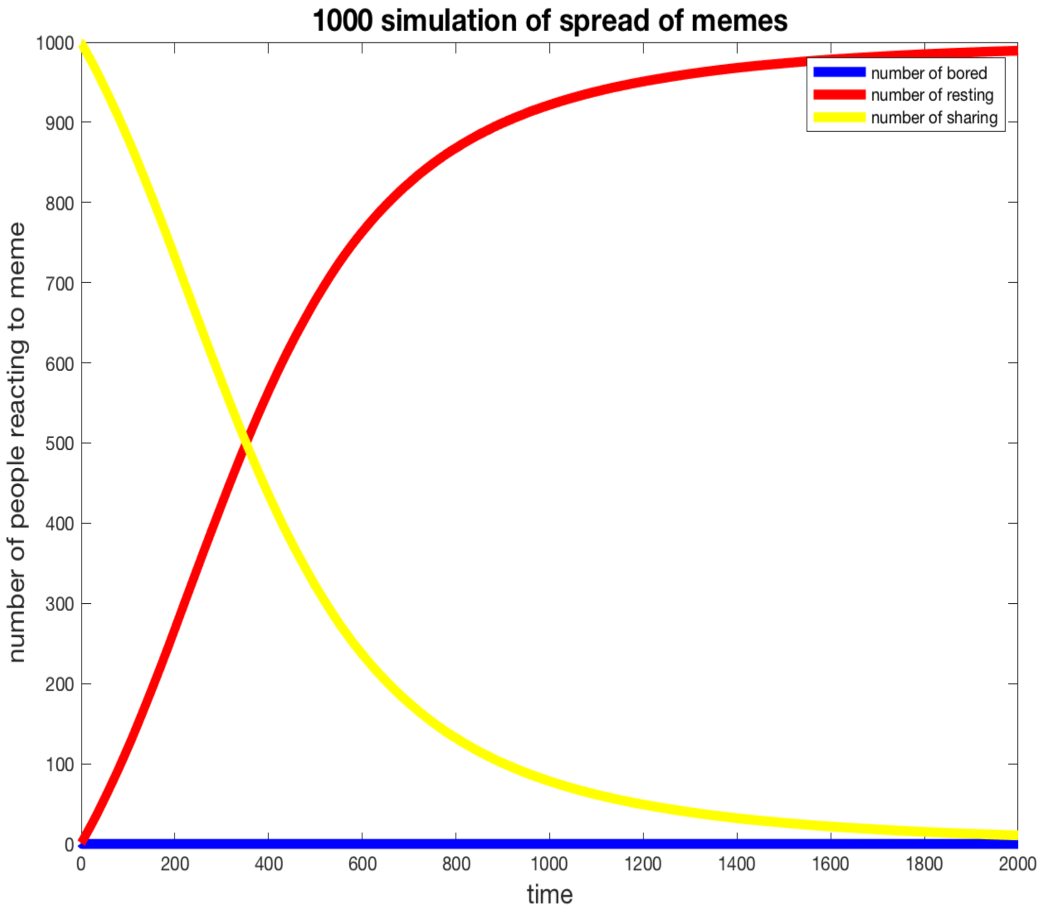


Figure 17: 100 simulations of memes with different rules for bored person over t = 2000

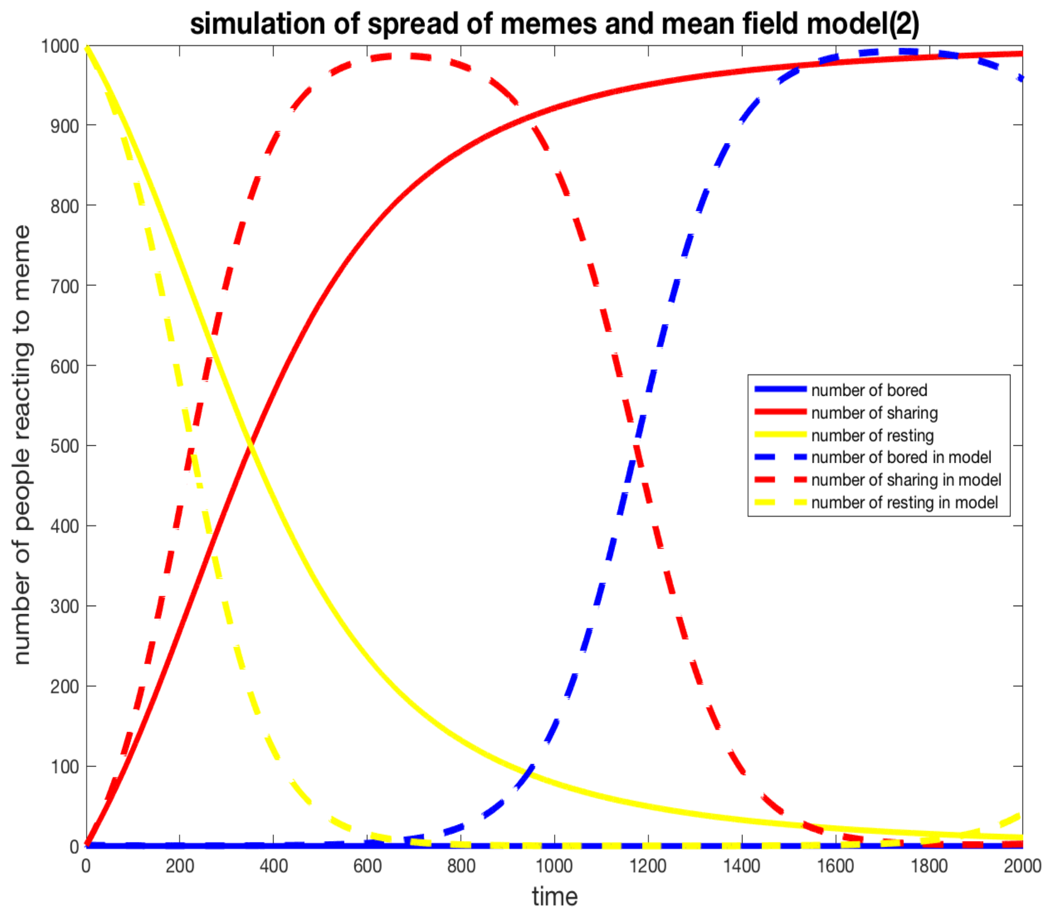


Figure 18: 100 simulations of memes with different rules for bored person over t = 2000 with mean field model plot

A phase transtion over q = 0.01:0.01:1 was made for the simulation. it is interesting to see that with probability q a person interact with another either to share or transtion from bored to rest. from q = 0.01 to q = 0.1, the total number of sharing persons at t = 1000 increase rapidly and almost everyone was sharing in the end when p > 0.1.

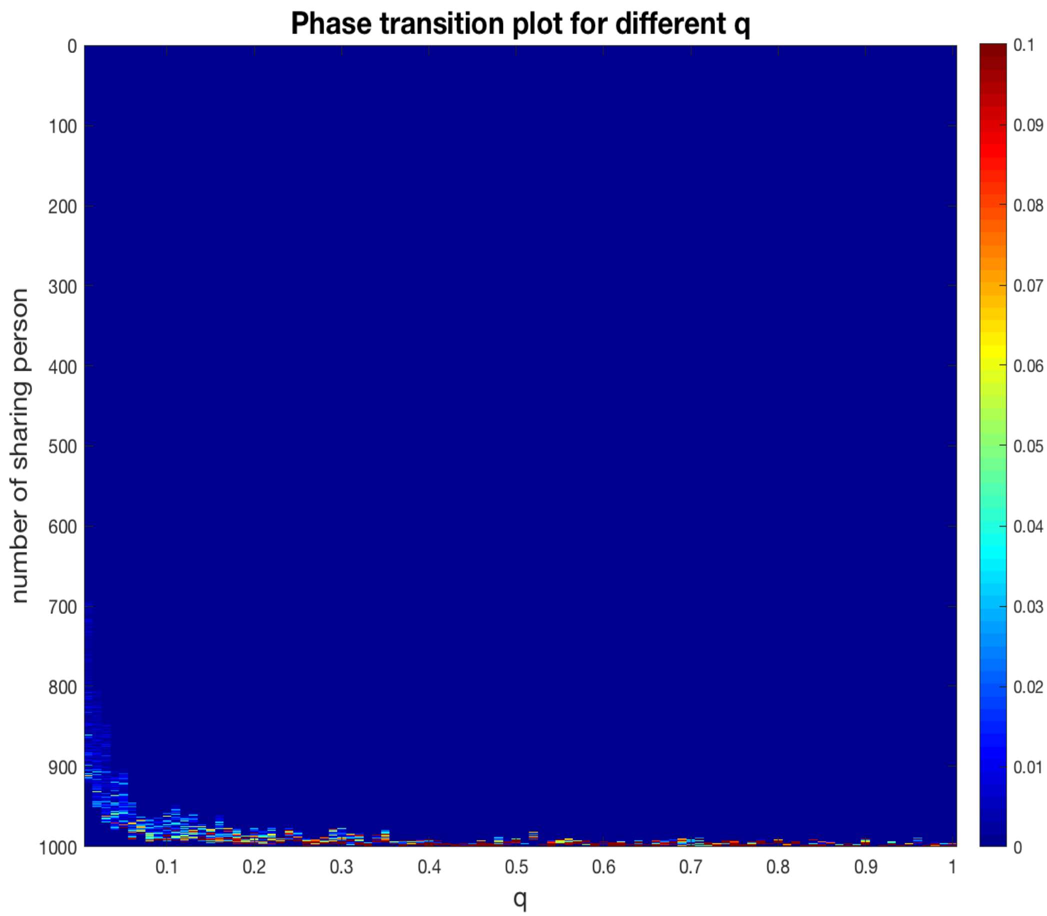


Figure 19: phase transition of total sharing person with simulation with t = 1000 and different q condition

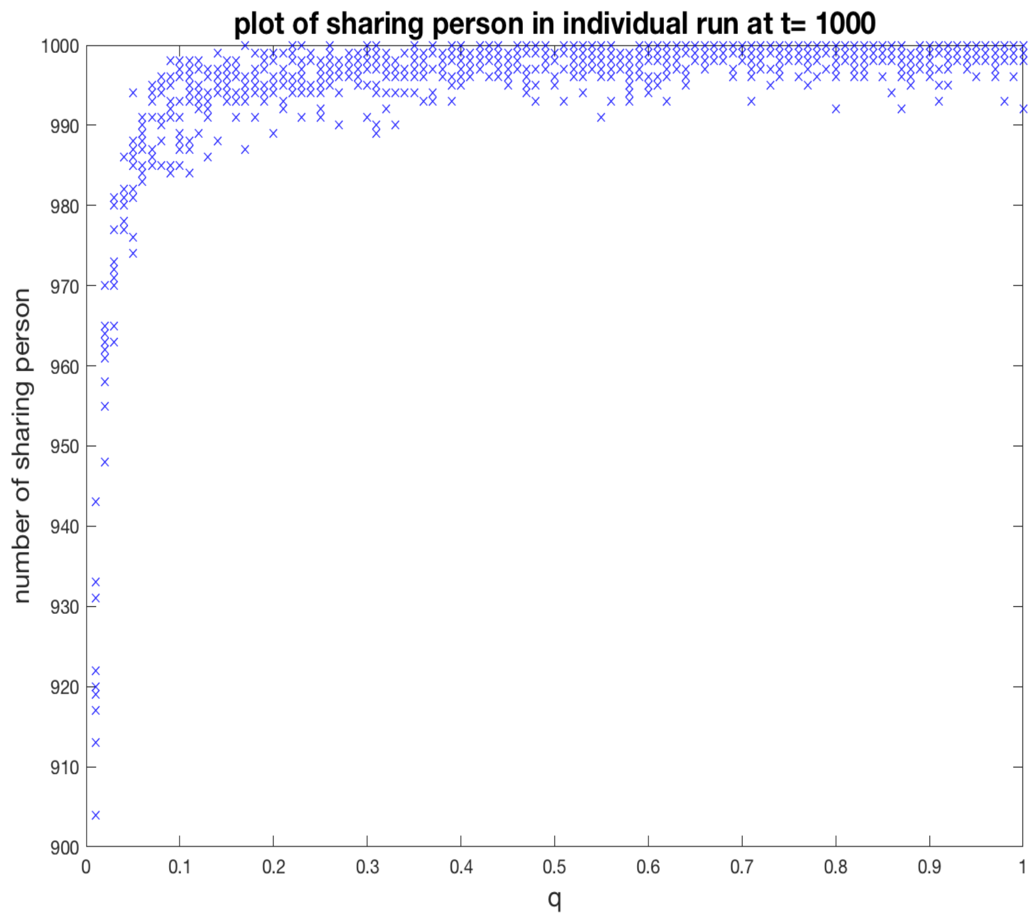


Figure 20: plot of sharing person in individual run at t = 1000

## 2.3 simulate on grids of 40x40 for spread of memes

In this part, we use the model above and in 2.2 to simulate that the person can only interact with the neighbours. the boundary conditions are set to be periodic. so that it goes from left to right, right to left, up to bottom, and bottom to up. After several runs. it can be observed that the number of sharer are increasing steadily. while the number of bored people tend to remain around at 1.the only way that bored person increase is that a sharing person meets a bored person. in theory the probability is p

= 1* 0.01*(1/1600). For the majority resting population, with a chance of 0.001 to discover a memes and become a sharer. in the initial state, there were 1588 resting people, that means that in theory 1.5 people will become a sharer. That's why it grows very fast, when t gets to 1000. almost everyone is sharing.

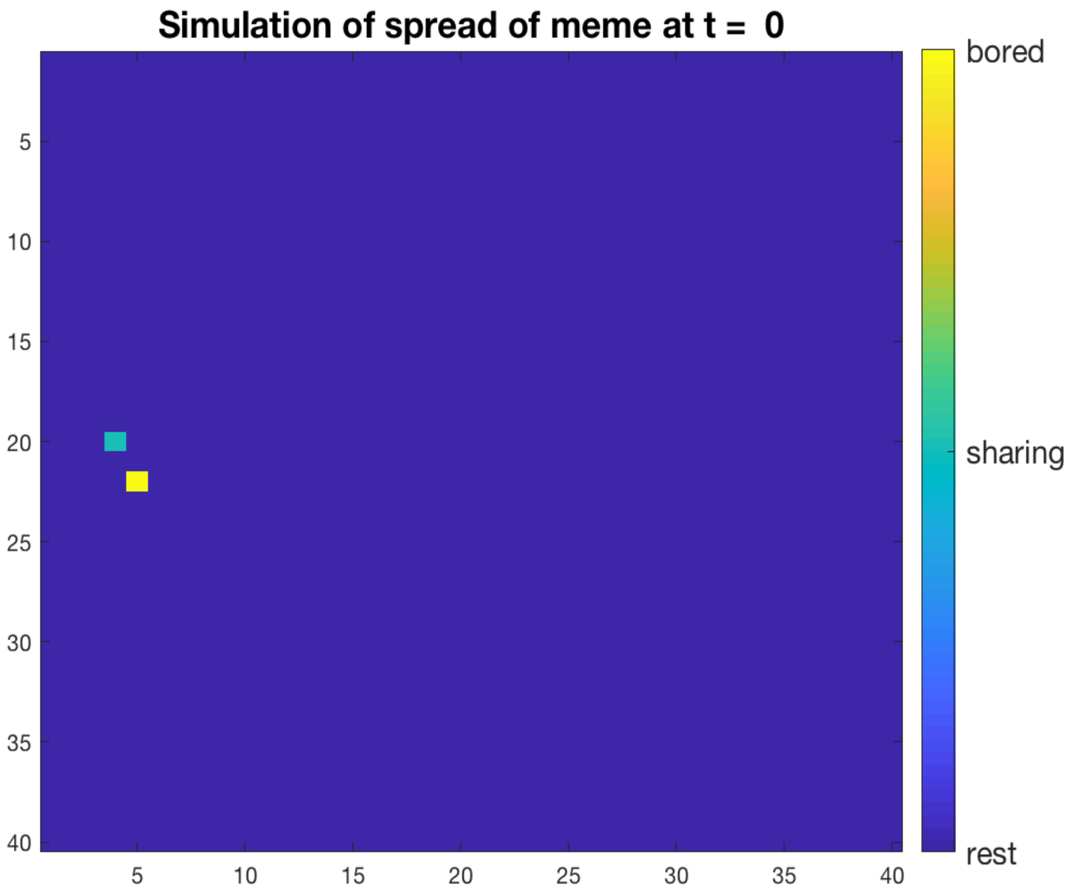


Figure 21: initial condition for simulation

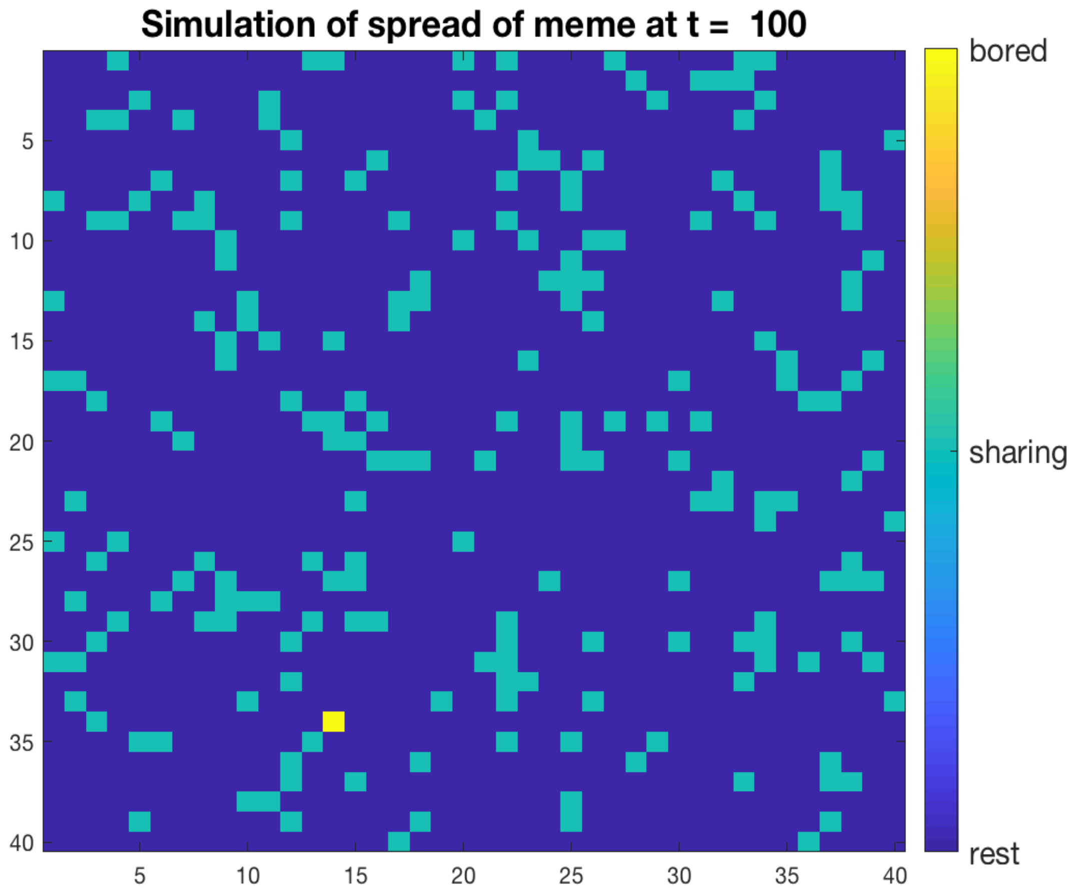


Figure 22: simulation of spread of meme to t = 100

The video of the simulation can be found at <https://www.youtube.com/watch?v=WMfI2P52ros>

# 3 Appendix

## 3.1 1firing brain code in matlab

- simulate single time of fire brain

```
clear all;
%this is to simualte 1 time at different time steps of a
    firing brain

n = 40;%input grid number
p = 0.3;
I = random_start(n,p);%generate initial state

t = 0;

v = VideoWriter('firebrain_1000.avi');
v.FrameRate = 10;
open(v);
figure('position', [0, 0, 700, 600])
imagesc(I);%plot
caxis([0 2]);
colorbar('Ticks',[0,1,2],'TickLabels',{'ready','firing',
    'rest'},'FontSize',14)
title(sprintf('Simulation of a firing brain at t = %s',
    string(t)),'FontSize',16)

axis tight manual
set(gca,'nextplot','replacechildren');
frame = getframe(gcf);
writeVideo(v,frame);

waitforbuttonpress;

endtime = 1000;


for t = 1:endtime

        I = transit(n,I);

```

```
        imagesc(I);
        caxis([0 2]);
        colorbar('Ticks',[0,1,2],'TickLabels',{'ready','firing','rest'},'FontSize',14)
        title(sprintf('Simulation of a firing brain at t = %s', string(t)),'FontSize',16)

        frame = getframe(gcf);
        writeVideo(v,frame);

        %pause(0.0025)

end
close(v)

%{
to save the final state
fig1 = imagesc(I);
caxis([0 2]);
colorbar('Ticks',[0,1,2],'TickLabels',{'ready','firing','rest'},'FontSize',14)
title('Simulation of a firing brain after 1000 time step','FontSize',16)
%saveas(fig1,'fire1000.png');
%}
```

- transition function

```
function y = transit(n, I)
%for firebrain
%ready 0, fire 1, rest 2

y_initial = I; %initial our transit value
y = zeros(n); %create matrix for next state with all zeros ready to write

for i = 1:n
    for k = 1:n

        %if fire go to rest
        if y_initial(i,k) == 1
```

```
            y(i,k) = 2;
        end


        % if rest go to ready
        if y_initial(i,k) == 2
            y(i,k) = 0;
        end

        % if ready go to fire if two neighbour are firing
        if y_initial(i,k) == 0
            count = 0;

            for p = (i-1) : (i+1)
                for q = (k-1) : (k+1)
                    %hard reset boundary condition to periodic
                    if p == 0
                        p = n;
                    end

                    if q == 0
                         q = n;
                    end

                    if p == n + 1
                         p = 1;
                    end

                    if q == n + 1
                         q = 1;
                    end


                    if y_initial(p,q) == 1
                         count = count + 1;
                    end
                end
            end

            if count == 2
```

```
                y(i,k) = 1;
            end
        end
    end
end
end
```

- initial state

```
function I = random_start(n, p)
I = zeros(n);
for i = 1:n
        for j = 1:n
            initial_state = rand;
            if initial_state < p
                I(i,j) = 1;
            end
        end
end
end
```

- simulate 100 times of firing brain

```
clear all;

n = 40;%input grid number
p = 0.3;

endtime = 1000;

for s_time = 1:100
    I = random_start(n,p);%generate initial state
%{
    imagesc(I);%plot
    caxis([0 2]);
    colorbar('Ticks',[0,1,2],'TickLabels',{'ready','
       firing','rest'},'FontSize',14)
    title('Simulation of a firing brain','FontSize',16)
%}
```

```
    for t = 1:endtime+1
        %waitforbuttonpress;

        %plot
        %{
        imagesc(I)
        caxis([0 2]);
        colorbar('Ticks',[0,1,2],'TickLabels',{'ready','firing','rest'},'FontSize',14)
        title('Simulation of a firing brain','FontSize',16)
        %}

        %pause(0.0025)

        firecell(s_time, t) = 0;
        for i = 1:n
            for j = 1:n
                if I(i,j) == 1
                    firecell(s_time,t) = firecell(s_time,t) + 1;
                end
            end
        end

        I = transit(n,I);
    end
end

%to plot

fig2 = figure('position', [0, 0, 700, 500]);
x = 0:1:1000;
plot(x,mean(firecell),'LineWidth',3);
ylim([0 600])
xlabel('time step','FontSize',14);
ylabel('number of firing cell','FontSize',14);
ylim([0 1001]);
title('100 simulations of a firing brain to 1000 time steps','FontSize',16);
%legend('total number of bored','total number of resting','total number of sharing');
```

```
saveas(fig2,'firebrain100sim.png');
```

- cell that move forward at one cell per time preserving the same shape

```
clear all;

%input grid number
n = 40;

I = zeros(n); %create grid

I(5,5) = 1;
I(5,6) = 1;
I(6,5) = 2;
I(6,6) = 2;

I(20,21) = 1;
I(20,20) = 1;
I(21,20) = 2;
I(21,21) = 2;
I(22,19) = 1;
I(22,22) = 1;
I(23,19) = 2;
I(23,22) = 2;

imagesc(I); %plot
fig1 = imagesc(I);
caxis([0 2]);
colorbar('Ticks',[0,1,2],'TickLabels',{'ready','firing',
    'rest'},'FontSize',14)
title('shapes that move forward at a rate of one cell
    per time step','FontSize',16)
%saveas(fig1,'task2_1.png');


waitforbuttonpress;
%for tentime = 1:10
for t = 1:200

    I = transit(n,I);
```

```
    imagesc(I)
    %pause(0.25)

end
```

- cell that move forwad at one cell per time, launching other shapes behind them

```
clear all;

%I = random_start(40,0.2);
%if you want a random start

%plot fireing number over time

%input grid number
n = 40;

I = zeros(n); %create grid

I(29,20) = 1;
I(29,21) = 2;
I(30,20) = 1;
I(30,21) = 2;
I(30,27) = 1;
I(30,28) = 2;
I(30,30) = 1;
I(30,31) = 1;
I(31,21) = 1;
I(31,22) = 2;
I(31,25) = 1;
I(31,26) = 2;
I(31,30) = 2;
I(31,31) = 2;
I(32,23) = 1;
I(32,24) = 2;
I(32,27) = 2;
I(32,28) = 2;
I(32,29) = 1;
I(32,30) = 1;
I(33,25) = 1;
```

```
I(33,26) = 2;
I(33,28) = 1;
I(33,32) = 2;
%{
I(21,20) = 1;
I(21,21) = 1;
I(19,20) = 2;
I(20,22) = 2;
I(22,21) = 2;
I(21,19) = 2;
%}

imagesc(I); %plot
waitforbuttonpress;

for t = 1:15

    I = transit(n,I);

    imagesc(I)
    pause(0.25)

end

fig1 = imagesc(I);
caxis([0 2]);
colorbar('Ticks',[0,1,2],'TickLabels',{'ready','firing',
    'rest'},'FontSize',14)
title('lauching other shapes behind','FontSize',16)
%saveas(fig1,'task2_2.png');
```

- move forward at a rate of less than one cell per time step

```
clear all;

%input grid number
n = 40;

I = zeros(n); %create grid

I(20,20) = 2;
```

```
I(20,21) = 2;
I(18,22) = 2;
I(19,20) = 1;
I(19,22) = 1;
I(21,20) = 1;
%{
I(21,20) = 1;
I(21,21) = 1;
I(19,20) = 2;
I(20,22) = 2;
I(22,21) = 2;
I(21,19) = 2;
%}

imagesc(I); %plot
waitforbuttonpress;

for t = 1:20

    I = transit(n,I);
   %waitforbuttonpress;
    imagesc(I)
    pause(0.2)
end

fig1 = imagesc(I);
caxis([0 2]);
colorbar('Ticks',[0,1,2],'TickLabels',{'ready','firing',
    'rest'},'FontSize',14)
title('shape that move less than one cell per time step'
    ,'FontSize',16)
saveas(fig1,'task2_3.png');
```

- oscillate shape

```
clear all;

%input grid number
n = 40;

I = zeros(n); %create grid
```

```
I(20,20) = 1;
I(20,21) = 1;
I(21,20) = 1;
I(21,21) = 1;
I(19,20) = 2;
I(20,22) = 2;
I(22,21) = 2;
I(21,19) = 2;

imagesc(I); %plot
waitforbuttonpress;

for t = 1:20

    I = transit(n,I);
    imagesc(I)
    pause(0.25)
end
fig1 = imagesc(I);
caxis([0 2]);
colorbar('Ticks',[0,1,2],'TickLabels',{'ready','firing',
    'rest'},'FontSize',14)
title('shape that oscillate periodically','FontSize',16)
saveas(fig1,'task2_4.png');
```

- my cellular automata

```
clear all;
%input grid number
n = 100;

I = zeros(n); %create grid

imagesc(I);

%generate initial state
for i = 1:n
    for j = 1:n
        initial_state = rand;
        if initial_state < 0.05
```

```
                I(i,j) = 1;
            elseif initial_state < 0.1
                I(i,j) = 2;
            elseif initial_state< 0.25
                I(i,j) = 3;
            elseif initial_state< 0.35
                I(i,j) = 4;
            end


    end
end
%


t = 0;

v = VideoWriter('myca_t1000.avi');
v.FrameRate = 10;
open(v);
figure('position', [0, 0, 700, 600])
imagesc(I);
colormap jet
caxis([0 4]);
colorbar('Ticks',[0,1,2,3,4],'TickLabels',{'waiting','
    growing','reproducing','aging','dead'},'FontSize',14)
title(sprintf('Simulation of population t = %s', string
    (t)),'FontSize',16)
axis tight manual
set(gca,'nextplot','replacechildren');
frame = getframe(gcf);
writeVideo(v,frame);


waitforbuttonpress;

%for tentime = 1:10
for t = 1:1000

    I = transitown(n,I);

    imagesc(I)
```

```
    colorbar
    pause(0.25)


    imagesc(I);
    colormap jet
    caxis([0 4]);
    colorbar('Ticks',[0,1,2,3,4],'TickLabels',{'waiting'
      ,'growing','reproducing','aging','dead'},'
      FontSize',14)
    title(sprintf('Simulation of population t = %s',
      string(t)),'FontSize',16)
    frame = getframe(gcf);
    writeVideo(v,frame);

    %pause(0.1)
end

close(v)
```

- my cellular automata transit

```
function y = transitown(n, I)
%waiting 0 grow 1 reproduce 2 mature 3 dead 4

y_initial = I; %initial our transit value
y = zeros(n); %create matrix for next state with all
   zeros ready to write

for i = 1:n
    for k = 1:n

        %if dead 0.2 chance go to wait
        if y_initial(i,k) == 4
            if rand < 0.6
                y(i,k) = 0;
            else
                y(i,k) = 4;
            end
        end

```

```
        if y_initial(i,k) == 3 %if mature 50% go to dead
            if rand < 0.5
                y(i,k) = 4;
            else
                y(i,k) = 3;
            end
        end


         % if 0 need to start grow
        if y_initial(i,k) == 0
            count = 0;

            for p = (i-1) : (i+1)
                for q = (k-1) : (k+1)
                    %hard reset boundary condition to
                        periodic >_<
                    if p == 0
                        p = n;
                    end

                    if q == 0
                         q = n;
                    end

                    if p == n + 1
                         p = 1;
                    end

                    if q == n + 1
                         q = 1;
                    end


                    if y_initial(p,q) == 2
                         count = count + 1;
                    end
                end
            end

            if count >1
                y(i,k) = 1;
```

```
            else
                y(i,k) = 0;
            end
        end

        % if 1 need to get at one reproduce and mature to survice
        if y_initial(i,k) == 1
            count = 0;
            count2 = 0;

            for p = (i-1) : (i+1)
                for q = (k-1) : (k+1)
                    %hard reset boundary condition to periodic >_<
                    if p == 0
                        p = n;
                    end

                    if q == 0
                        q = n;
                    end

                    if p == n + 1
                        p = 1;
                    end

                    if q == n + 1
                        q = 1;
                    end


                    if y_initial(p,q) == 2
                        count = count + 1;
                    end
                    if y_initial(p,q) == 3
                        count2 = count2 +1;
                    end
                end
            end

            if count >=1 && count2 >= 1
```

```
                    y(i,k) = 2;
               else
                    y(i,k) = 1;
               end
          end
                    % if 2 need to reproduce
          if y_initial(i,k) == 2
               if rand < 0.5
                    y(i,k) = 3;
               else
                    y(i,k) = 2;
               end

          end


     end
end
end
```

### 3.2 Spread of memes

- simulation of spread of memes

```
clear all;

n = 1000;
p = 0.001;%discover new meme
q = 0.01; %share a meme
total_time_step = 2000;
sim_number = 1000;
Bt = 0;
St = 0;
Rt = 0;


for rep = 1:sim_number
     rep

     [B,S,R] = runmeme(n,total_time_step,p,q);
```

```
    Bt = Bt + B;
    St = St + S;
    Rt = Rt + R;
end

result_B = Bt/sim_number;
result_S = St/sim_number;
result_R = Rt/sim_number;
x = 0:1:total_time_step;

my_fig = figure('position', [0, 0, 700, 500]);
plot(x,result_B ,'b','LineWidth',5);
hold on;
plot(x,result_S ,'r','LineWidth',5);
plot(x,result_R ,'y','LineWidth',5);
xlabel('time','FontSize',14);
ylabel('number of people reacting to meme','FontSize',14);
title('1000 simulation of spread of memes','FontSize',16);
legend('number of bored','number of resting','number of sharing');
saveas(my_fig ,'memes_sim_1000times.png');
hold off;
```

- run single spread of memes

```
function [B,S,R] = runmeme(n,total_time_step ,p,q)

%n = 1000;
%p = 0.001;%discover new meme
%q = 0.01; %share a meme

N = zeros(total_time_step ,n);


start = randperm(1000,2); %generate the person who (bored, share)

N(1,start(1)) = 2;%bored 2
N(1,start(2)) = 1; %sharing
```

```
for t = 1:1:total_time_step

    for i = 1:n

        if N(t,i) == 2%if bored stay bored forever
            N(t+1, i) = 2;

        end

        if N(t,i) == 0 %if rest will discover a new meme
            if rand < p % found a new meme at
               probability p and be sharer
                N(t+1, i) = 1;
            else
                N(t+1, i) = 0;
            end
        end

        if N(t,i) == 1
            if rand < q
                target = randi([1 1000]);
                if N(t,target) == 2
                    N(t+1, i) = 2;
                else
                    N(t+1, target) = 1;
                    N(t+1, i) = 1;
                end
            else
                N(t+1,i) = 1;
            end
        end
    end
end


for t = 1:1:total_time_step+1
    B(t) = 0;
    S(t) = 0;
    R(t) = 0;
    for i = 1:n
```

```
        if N(t,i) == 2
            B(t) = B(t) + 1;
        end
        if N(t,i) == 1
            S(t) = S(t) + 1;
        end
        if N(t,i) == 0
            R(t) = R(t) + 1;
        end
    end
end

end
```

- mean field model

```
clear all;

n = 1000;
p = 0.001;%discover new meme
q = 0.01; %share a meme
total_time_step = 2000;
sim_number = 1;

for rep = 1:sim_number
    [B,S,R] = mememodel(n,total_time_step,p,q);

    m_result_B(rep,:) = B;
    m_result_S(rep,:) = S;
    m_result_R(rep,:) = R;
end

load('1000_sim_memes.mat');
x = 0:1:total_time_step;

my_fig = figure('position', [0, 0, 700, 500]);
plot(x,result_B,'b','LineWidth',3);
hold on;
plot(x,result_S,'r','LineWidth',3);
plot(x,result_R,'y','LineWidth',3);
plot(x,m_result_B,'--b','LineWidth',3);
```

```
plot(x,m_result_S,'r--','LineWidth',3);
plot(x,m_result_R,'y--','LineWidth',3);

xlabel('time','FontSize',14);
ylabel('number of people reacting to meme','FontSize',14);
title('simulation of spread of memes and mean field model','FontSize',16);
legend({'number of bored','number of sharing','number of resting','number of bored in model','number of sharing in model','number of resting in model'},'Location','east');
saveas(my_fig,'memes_withmodel1000.png');
hold off;
```

- phase transition

```
clear all;

n = 1000;
pp = 0.001;%discover new meme
q = 0.01; %share a meme
total_time_step = 1000;
%sim_number = 1000;%a lot @.@ for my computer
Bt = 0;
St = 0;
Rt = 0;

Bvals = [1:1:999];
numreps = 10;
hrange = [0:1:n];

histu = zeros(length(Bvals), length(hrange));
count = 0;

for b = Bvals
    Sall = [];
    count = count + 1

    for rep = 1:numreps
        [B,S,R] = runmeme_phase(n,total_time_step,pp,q,b
```

```
            );
        S = S';

        finalS(count,rep) = S(end);
        maxS(count,rep) = max(S);
        Sall = [Sall;S((end-499):end)];
    end
    %take the histogram
    histu(count,:) = hist(Sall,hrange);
end

figure2 = figure('position', [0, 0, 700, 500]);
imagesc(Bvals,hrange,histu'/(numreps*500),[0 0.1])
colormap jet
hlx=xlabel('number of bored person','FontSize',14)
hlx=ylabel('number of sharing person','FontSize',14)
title('Phase transition plot', 'FontSize', 16)
colorbar
saveas(figure2,'phasetransition.png');

figure3 = figure('position', [0, 0, 700, 500]);
%Plot the results
count=0;
for pp=Bvals
    count=count+1;
    pp;
    for rep=1:numreps
        plot(pp,finalS(count,rep),'bx')
        hold on
    end
end

xlabel('number of bored person','FontSize',14)
ylabel('number of sharing person','FontSize',14)
title('plot of sharing person in individual run at t=1000', 'FontSize', 16)
saveas(figure3,'indi_phase.png');
```

- probability for at least 25% are sharing

```
clear all;
```

```
n = 1000;
p = 0.001;%discover new meme
q = 0.01; %share a meme
total_time_step = 1000;

Bvals = [1:1:999];
numreps = 100;
tval = [1:1:total_time_step];

p25 = zeros(999, total_time_step);
count = 0;

for b = Bvals
    b
    for i = 1:1:numreps
        [B,S,R] = runmeme_phase(n,total_time_step,p,q,b)
            ;
        for t = 1:total_time_step
            if S(t) > 250
                p25(b,t) = p25(b,t) + 0.01;
            end
        end
    end
end
figure2 = figure('position', [0, 0, 700, 500]);
imagesc(p25)
colormap jet
hlx=xlabel('time','FontSize',14)
hlx=ylabel('B0','FontSize',14)
title('probability of more than 250 are sharing', '
    FontSize', 16)
colorbar
%saveas(figure2,'probablity250_100sim.png');
```

- simulation of spread of memes with new rules

```
clear all;

n = 1000;
p = 0.001;%discover new meme
q = 0.01; %share a meme
```

```
total_time_step = 2000;
sim_number = 1000;
Bt = 0;
St = 0;
Rt = 0;


for rep = 1:sim_number
    rep

    [B,S,R] = runmeme_newbored(n,total_time_step,p,q);
    Bt = Bt + B;
    St = St + S;
    Rt = Rt + R;
end

result_B = Bt/sim_number;
result_S = St/sim_number;
result_R = Rt/sim_number;
x = 0:1:total_time_step;

my_fig = figure('position', [0, 0, 700, 500]);
plot(x,result_B,'b','LineWidth',5);
hold on;
plot(x,result_S,'r','LineWidth',5);
plot(x,result_R,'y','LineWidth',5);
xlabel('time','FontSize',14);
ylabel('number of people reacting to meme','FontSize'
    ,14);
title('1000 simulation of spread of memes','FontSize'
    ,16);
legend('number of bored','number of resting','number of
    sharing');
%saveas(my_fig,'memes2_sim_1000times.png');
hold off;
save('meme2data.mat','result_B','result_R','result_S')
```

- run single spread of memes

```
function [B,S,R] = runmeme_newbored(n,total_time_step,p,
    q)
```

```
%n = 1000;
%p = 0.001;%discover new meme
%q = 0.01; %share a meme

N = zeros(total_time_step,n);

 start = randperm(1000,2); %generate the person who (bored, share)

N(1,start(1)) = 2;%bored 2
N(1,start(2)) = 1;%share 1

 for t = 1:1:total_time_step

     for i = 1:n

         if N(t,i) == 2%if bored will find someone
             if rand < q
                 target = randi([1 1000]);
                 if N(t,target) == 0
                     N(t+1,i) = 0;
                 else
                     N(t+1,i) = 2;
                 end
             end

         end

         if N(t,i) == 0 %if rest will discover a new meme
             if rand < p % found a new meme at probability p and be sharer
                 N(t+1, i) = 1;
             else
                 N(t+1, i) = 0;
             end
         end

         if N(t,i) == 1
             if rand < q
                 target = randi([1 1000]);
                 if N(t,target) == 2
```

```
                        N(t+1, i) = 2;
                    else
                        N(t+1, target) = 1;
                        N(t+1, i) = 1;
                    end
                else
                    N(t+1,i) = 1;
                end
            end
        end
    end

for t = 1:1:total_time_step+1
    B(t) = 0;
    S(t) = 0;
    R(t) = 0;
    for i = 1:n
        if N(t,i) == 2
            B(t) = B(t) + 1;
        end
        if N(t,i) == 1
            S(t) = S(t) + 1;
        end
        if N(t,i) == 0
            R(t) = R(t) + 1;
        end
    end
end

end
```

- mean field model

```
clear all;

n = 1000;
p = 0.001;%discover new meme
q = 0.01; %share a meme
total_time_step = 2000;
sim_number = 1;
```

```

for rep = 1:sim_number
    [B,S,R] = mememodel2(n,total_time_step,p,q);

    m_result_B(rep,:) = B;
    m_result_S(rep,:) = S;
    m_result_R(rep,:) = R;
end

load('meme2data.mat');
x = 0:1:total_time_step;

my_fig = figure('position', [0, 0, 700, 500]);
plot(x,result_B,'b','LineWidth',3);
hold on;
plot(x,result_S,'r','LineWidth',3);
plot(x,result_R,'y','LineWidth',3);
plot(x,m_result_B,'--b','LineWidth',3);
plot(x,m_result_S,'r--','LineWidth',3);
plot(x,m_result_R,'y--','LineWidth',3);

xlabel('time','FontSize',14);
ylabel('number of people reacting to meme','FontSize',14);
title('simulation of spread of memes and mean field model(2)','FontSize',16);
legend({'number of bored','number of sharing','number of resting','number of bored in model','number of sharing in model','number of resting in model'},'Location','east');
saveas(my_fig,'memes2_withmodel1000.png');
hold off;
```

- phase transition for new rules

```
clear all;

n = 1000;
pp = 0.001;%discover new meme
q = 0.01; %share a meme
total_time_step = 1000;
```

```
%sim_number = 1000;%a lot @.@ for my computer
Bt = 0;
St = 0;
Rt = 0;

qvals = [0.01:0.01:1];
numreps = 10;
hrange = [0:1:n];

histu = zeros(length(qvals), length(hrange));
count = 0;

for b = qvals
    Sall = [];
    count = count + 1

    for rep = 1:numreps
        [B,S,R] = runmeme_newbored(n,total_time_step,pp,b);
        S = S';
        finalS(count,rep) = S(end);
        Sall = [Sall;S((end-499):end)];
    end
    %take the histogram
    histu(count,:) = hist(Sall,hrange);
end

figure2 = figure('position', [0, 0, 700, 500]);
imagesc(qvals,hrange,histu'/(numreps*500),[0 0.1])
colormap jet
hlx=xlabel('q','FontSize',14)
hlx=ylabel('number of sharing person','FontSize',14)
title('Phase transition plot for different q', 'FontSize', 16)
colorbar
%saveas(figure2,'phasetransition2t1000withB.png');

figure3 = figure('position', [0, 0, 700, 500]);
%Plot the results
count=0;
for pp=qvals
    count=count+1;
```

```
    pp;
    for rep=1:numreps
        plot(pp,finalS(count,rep),'bx')
        hold on
    end
end

xlabel('q','FontSize',14)
ylabel('number of sharing person','FontSize',14)
title('plot of sharing person in individual run at t=
   1000', 'FontSize', 16)
saveas(figure3,'indi_phase2t1000withq.png');
```

- lattice simulation for memes

```
clear all;
%input grid number
n = 40;
p = 0.001;
q = 0.01;
I = zeros(n); %create grid
imagesc(I);
initial_x = randperm(n,2);
initial_y = randperm(n,2);


%1, sharing person
I(initial_x(1),initial_y(1)) = 1;
%2. bored person
I(initial_x(2),initial_y(2)) = 2;

t = 0;

v = VideoWriter('memes_3_t1000.avi');
v.FrameRate = 10;
open(v);
figure('position', [0, 0, 700, 600])
imagesc(I);
caxis([0 2]);
colorbar('Ticks',[0,1,2],'TickLabels',{'rest','sharing',
    'bored'},'FontSize',14)
```

```
title(sprintf('Simulation of spread of meme at t = %s', string(t)),'FontSize',16)
axis tight manual
set(gca,'nextplot','replacechildren');
frame = getframe(gcf);
writeVideo(v,frame);


waitforbuttonpress;

for t = 1:1000
    I = transit_meme(n,I,p,q);

    %figure('position', [0, 0, 700, 600])
    imagesc(I);
    caxis([0 2]);
    colorbar('Ticks',[0,1,2],'TickLabels',{'rest','sharing','bored'},'FontSize',14)
    title(sprintf('Simulation of spread of meme at t = %s', string(t)),'FontSize',16)

    frame = getframe(gcf);
    writeVideo(v,frame);

    %pause(0.1)


end

close(v)
```

- transition function for memes

```
function y = transit_meme(n,I,p,q)
%rest0, share1, bored2
%n grid set to 40*40
%I matrix
%p probability find a new meme for a rest
%q probability for share and bored to find a person

y_initial = I; %initial our transit value
```

```
y = zeros(n); %create matrix for next state with all zeros ready to write

for i = 1:n
    for k = 1:n

        %%%%%%%bored
        %if bored will find someone
        if y_initial(i,k) == 2
            if rand < q
                target = randi(8);
                if target == 1
                    loc = [i-1,k-1];
                elseif target == 2
                    loc = [i-1,k];
                elseif target == 3
                    loc = [i-1,k+1];
                elseif target == 4
                    loc = [i,k-1];
                elseif target == 5
                    loc = [i,k+1];
                elseif target == 6
                    loc = [i+1,k-1];
                elseif target == 7
                    loc = [i+1,k];
                else
                    loc = [i+1,k+1];
                end


                %hard reset boundary
                if loc(1) ==0
                    loc(1) = n;
                end
                if loc(1) == n+1
                    loc(1) = 1;
                end
                if loc(2) ==0
                    loc(2) = n;
                end
                if loc(2) ==n+1
                    loc(2) = 1;
```

```
                end

                if y_initial(loc(1),loc(2)) == 0
                    y(i,k) = 0;
                else
                    y(i,k) = 2;
                end
            else
                y(i,k) = 2;
            end
        end


        if y_initial(i,k) == 0%if rest go find new meme
            if rand < p %p to find a meme and become a sharer
                y(i,k) = 1;
            else
                y(i,k) = 0; %stay rest
            end
        end

        if y_initial(i,k) == 1%if share go find people to share
            if rand < q
                target = randi(8);
                if target == 1
                    loc = [i-1,k-1];
                elseif target == 2
                    loc = [i-1,k];
                elseif target == 3
                    loc = [i-1,k+1];
                elseif target == 4
                    loc = [i,k-1];
                elseif target == 5
                    loc = [i,k+1];
                elseif target == 6
                    loc = [i+1,k-1];
                elseif target == 7
                    loc = [i+1,k];
                else
                    loc = [i+1,k+1];
```

```
                end


                %hard reset boundary
                if loc(1) ==0
                    loc(1) = n;
                end
                if loc(1) == n+1
                    loc(1) = 1;
                end
                if loc(2) ==0
                    loc(2) = n;
                end
                if loc(2) ==n+1
                    loc(2) = 1;
                end

                if y_initial(loc(1),loc(2)) == 0
                    y(loc(1),loc(2)) = 1;
                    y(i,k) = 1;
                elseif y_initial(loc(1),loc(2)) == 2
                    y(i,k) = 2;
                else
                    y(i,k) = 1;
                end


            else
                y(i,k) = 1;
            end
        end

    end
end
end
```