



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.

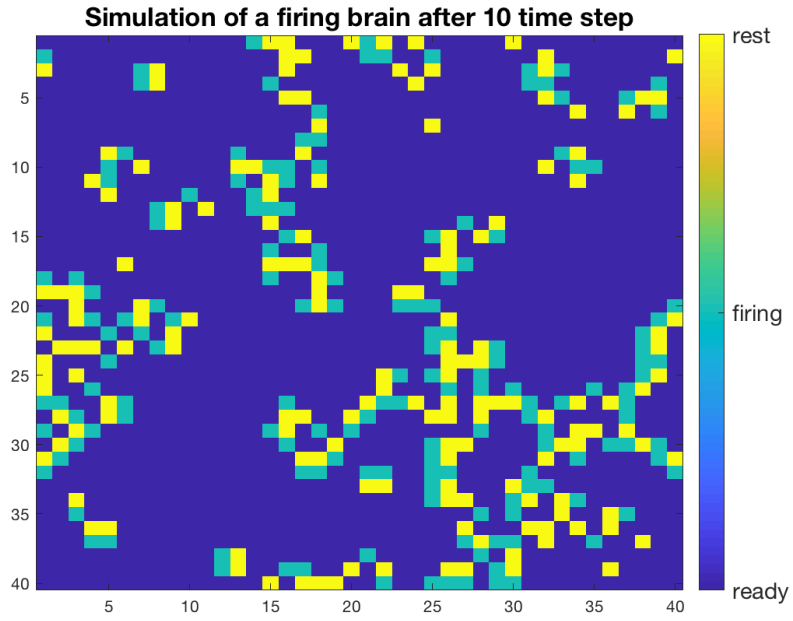


Figure 1: 40x40 cell grid at $t = 10$

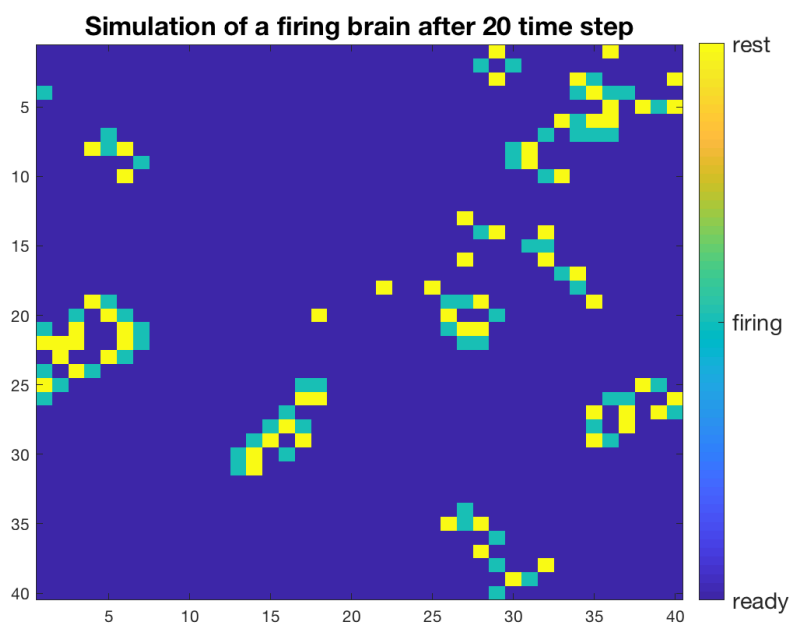


Figure 2: 40x40 cell grid at $t = 20$

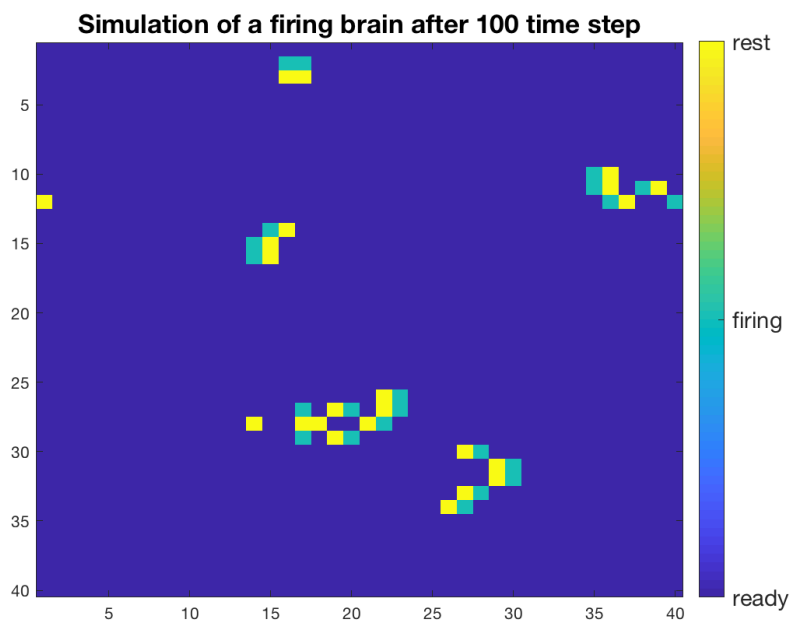


Figure 3: 40x40 cell grid at $t = 100$

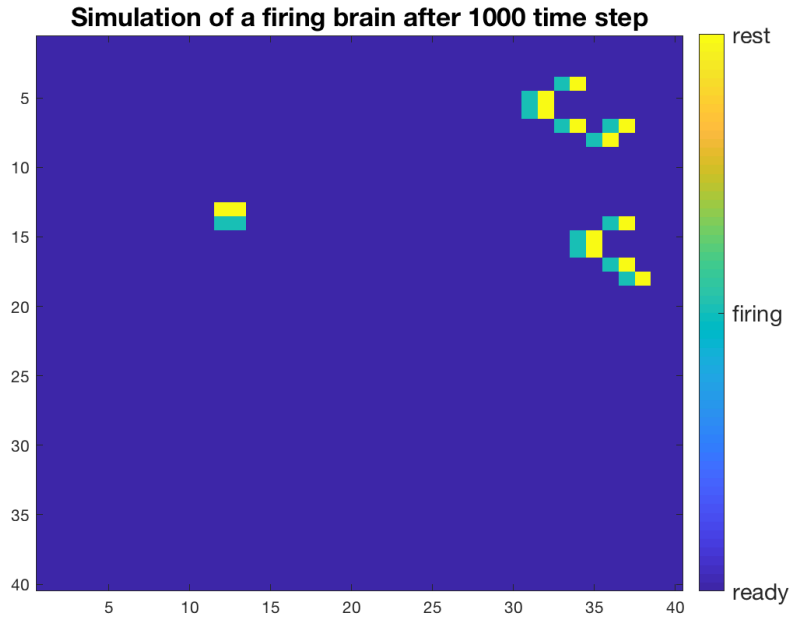


Figure 4: 40x40 cell grid at $t = 1000$

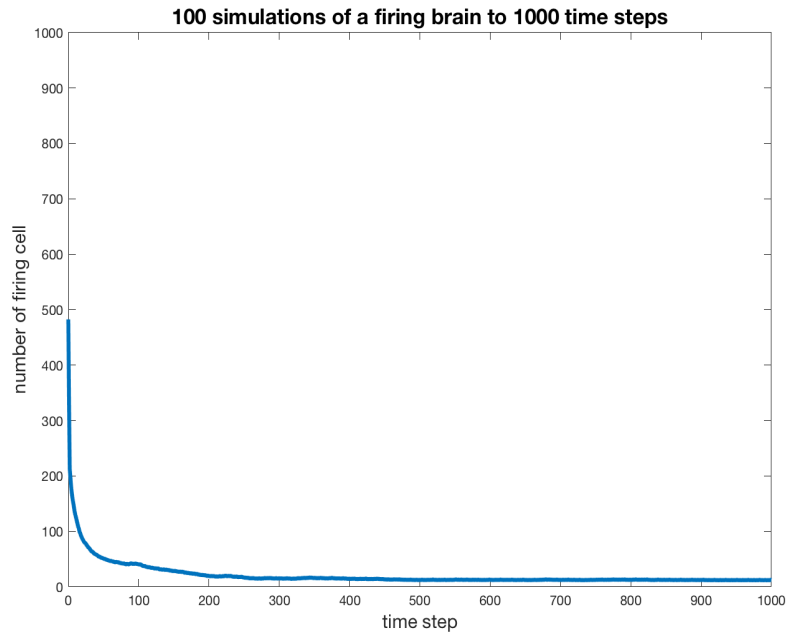


Figure 5: average number of firing cells over time with 100 simulation of different initial conditions

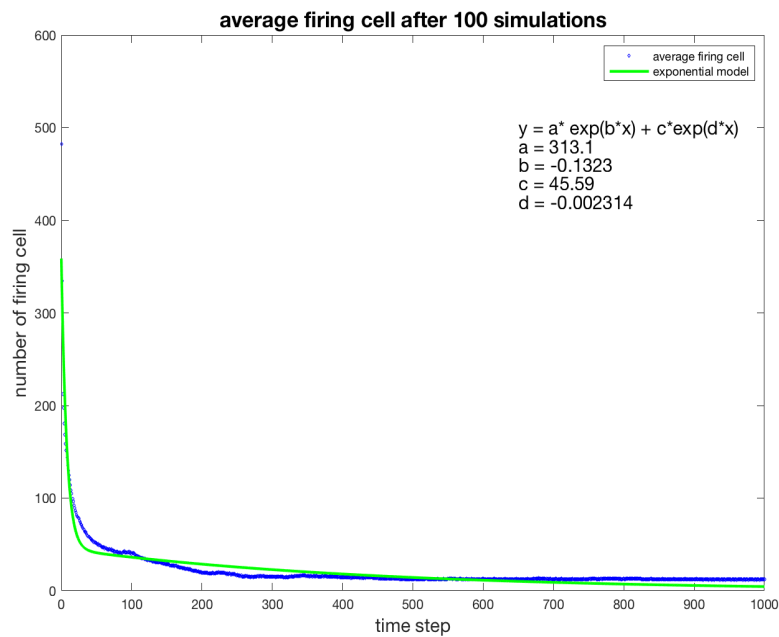


Figure 6: average number of firing cells over time with 100 simulation and exponential model fitting

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

for simulation to $t = 100$

<<https://youtu.be/CFUcpGHhj00>>

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

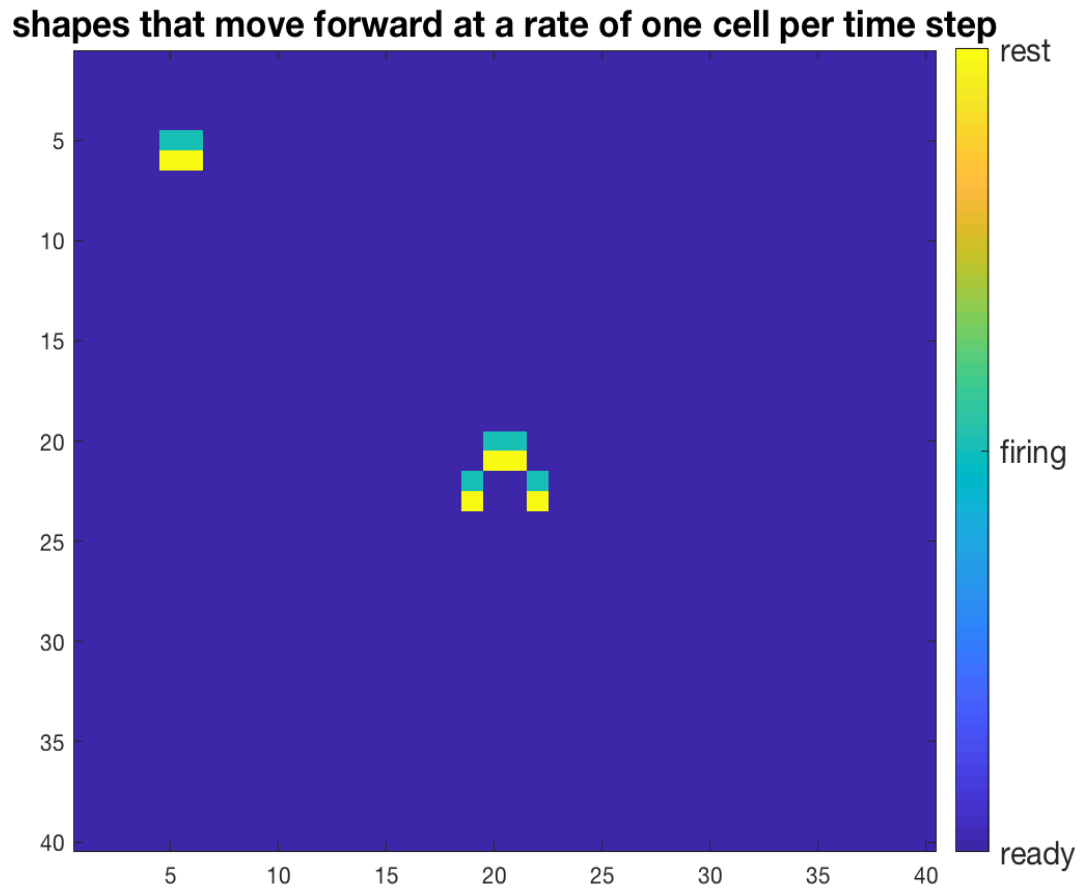


Figure 7: shapes that move forward at a rate of one cell per time step preserving the same shape

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

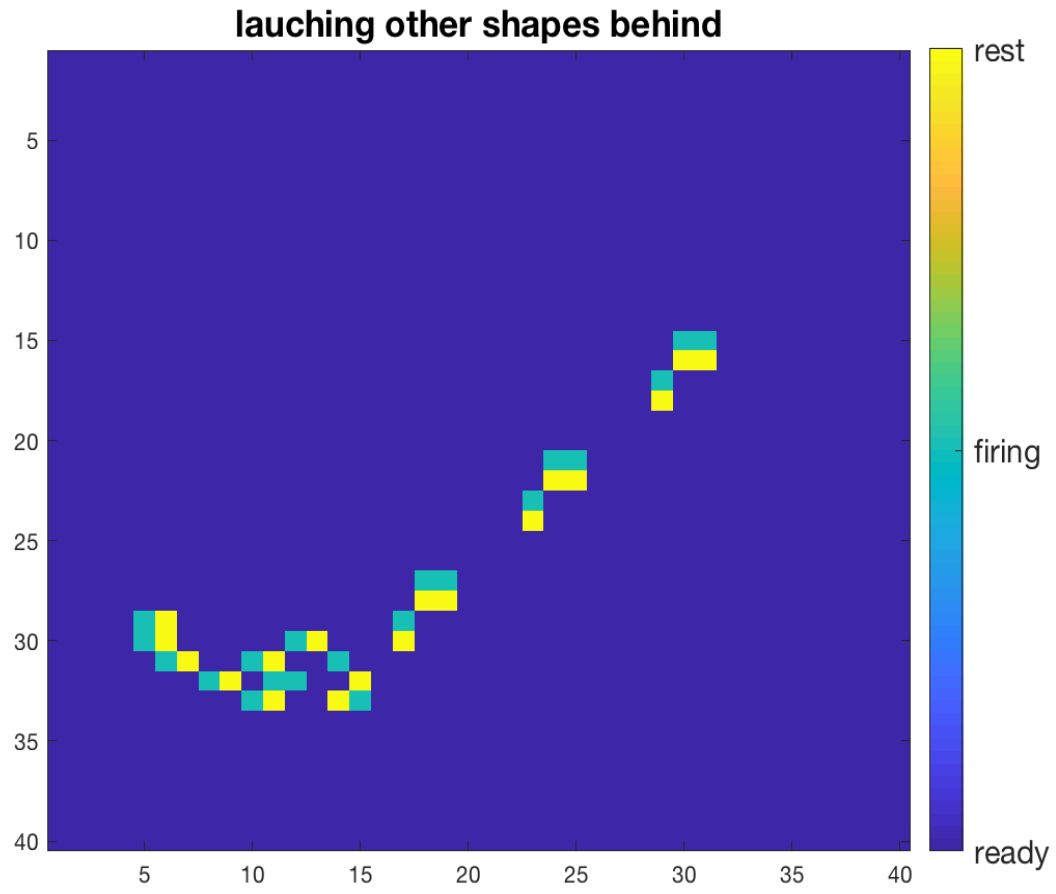


Figure 8: shapes that move forward at a rate of one cell per time step, launching other shapes behind them

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

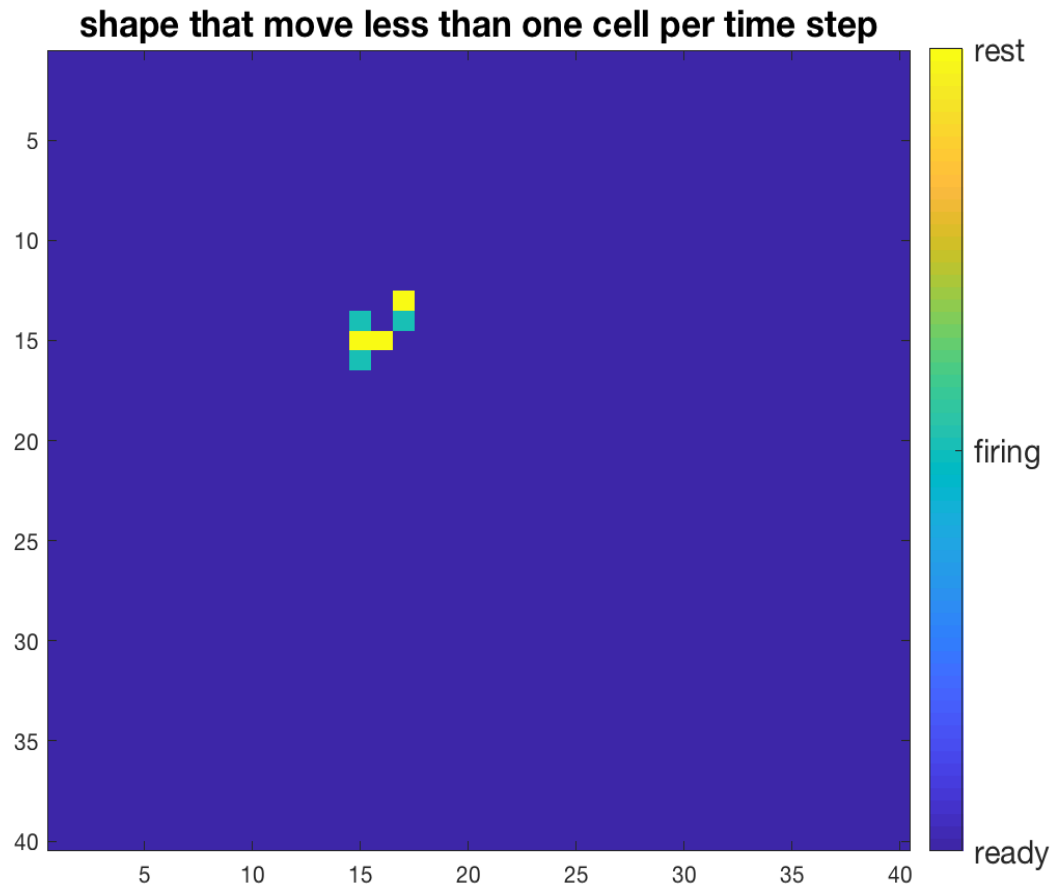


Figure 9: shapes that move less than one cell per time step, returning to same shape after some period

1.2.4 stay stationary but oscillate periodically

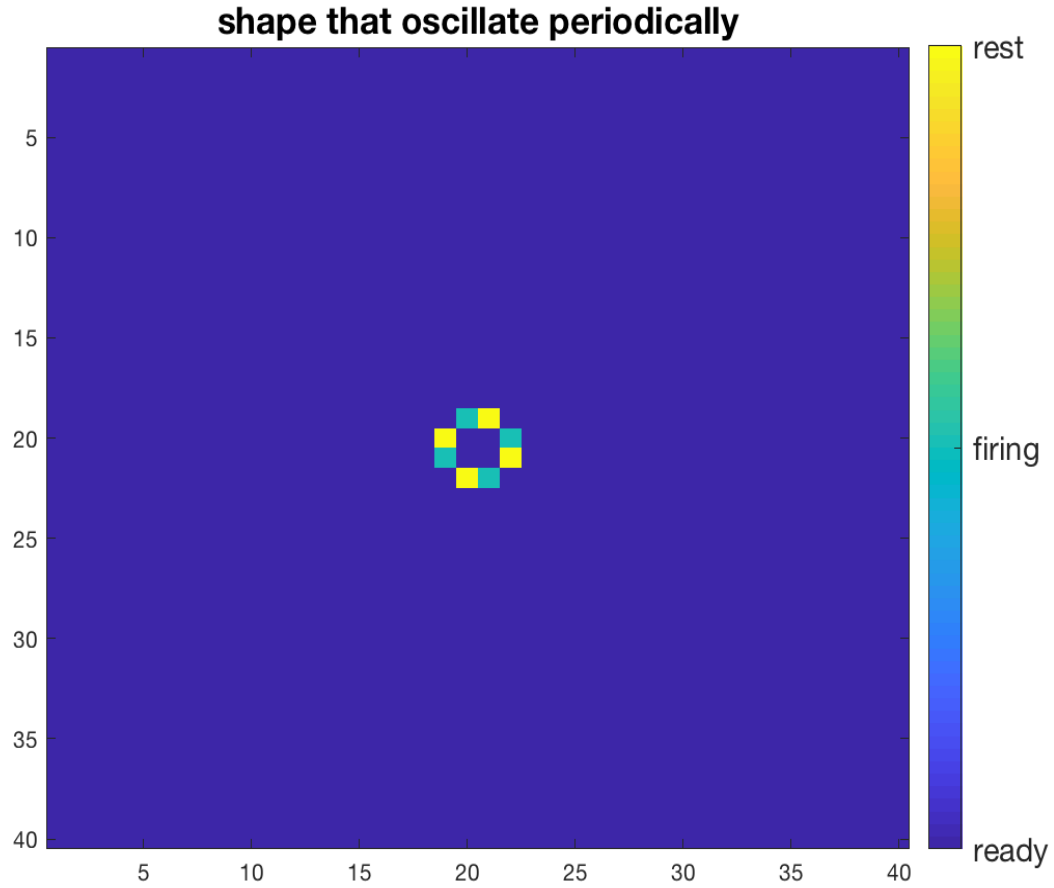


Figure 10: shapes that stay stationary but oscillate periodically

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) cell 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 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.

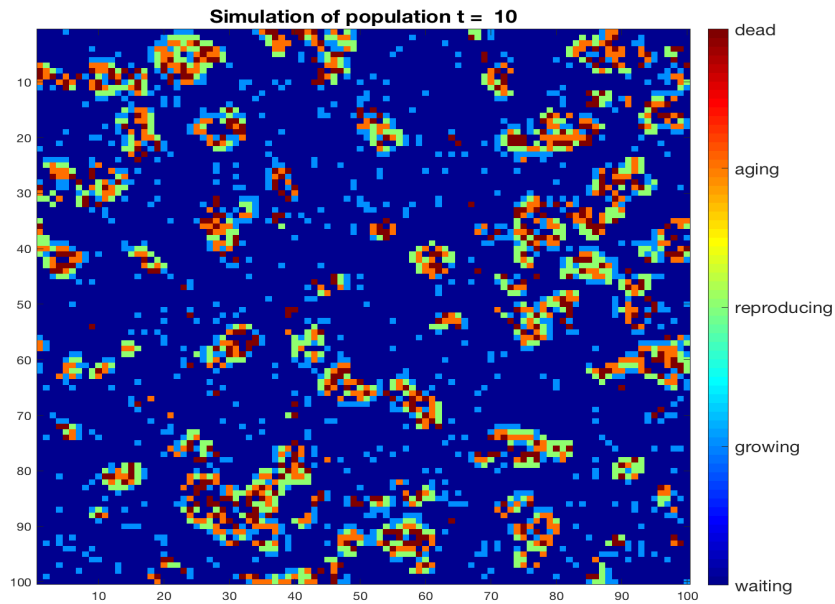


Figure 11: my cell automata at $t = 10$

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 simulation 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.

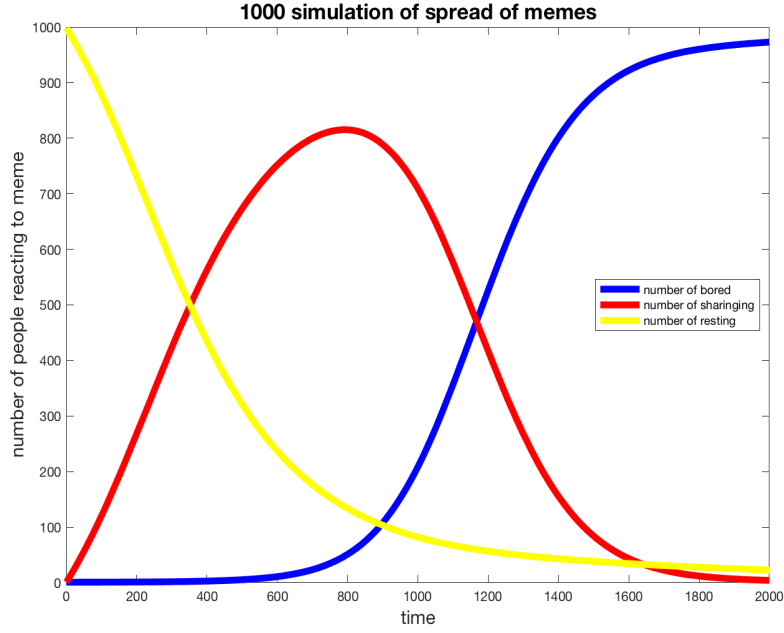


Figure 12: simulation of spread of memes showing number of bored, sharing, resting person over time

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

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

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

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

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

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

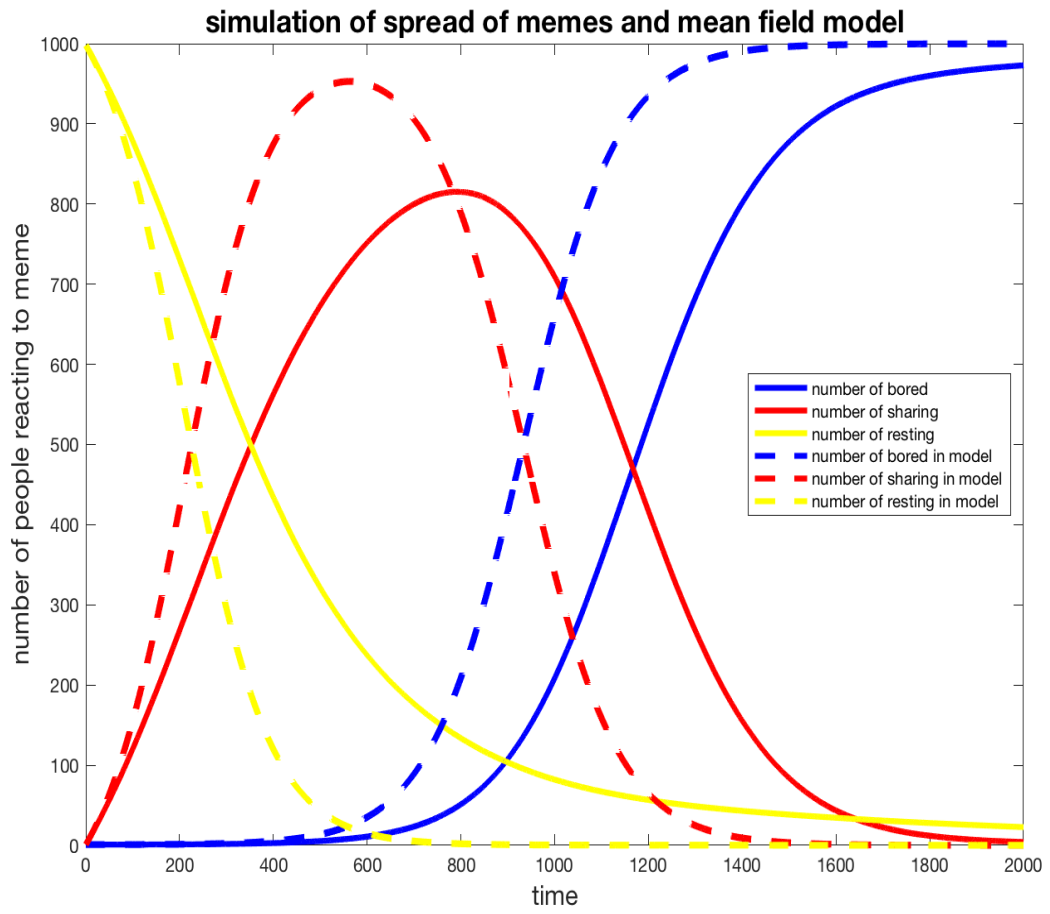


Figure 13: simulation of spread of memes showing number of bored, sharing, resting person over time with mean field model

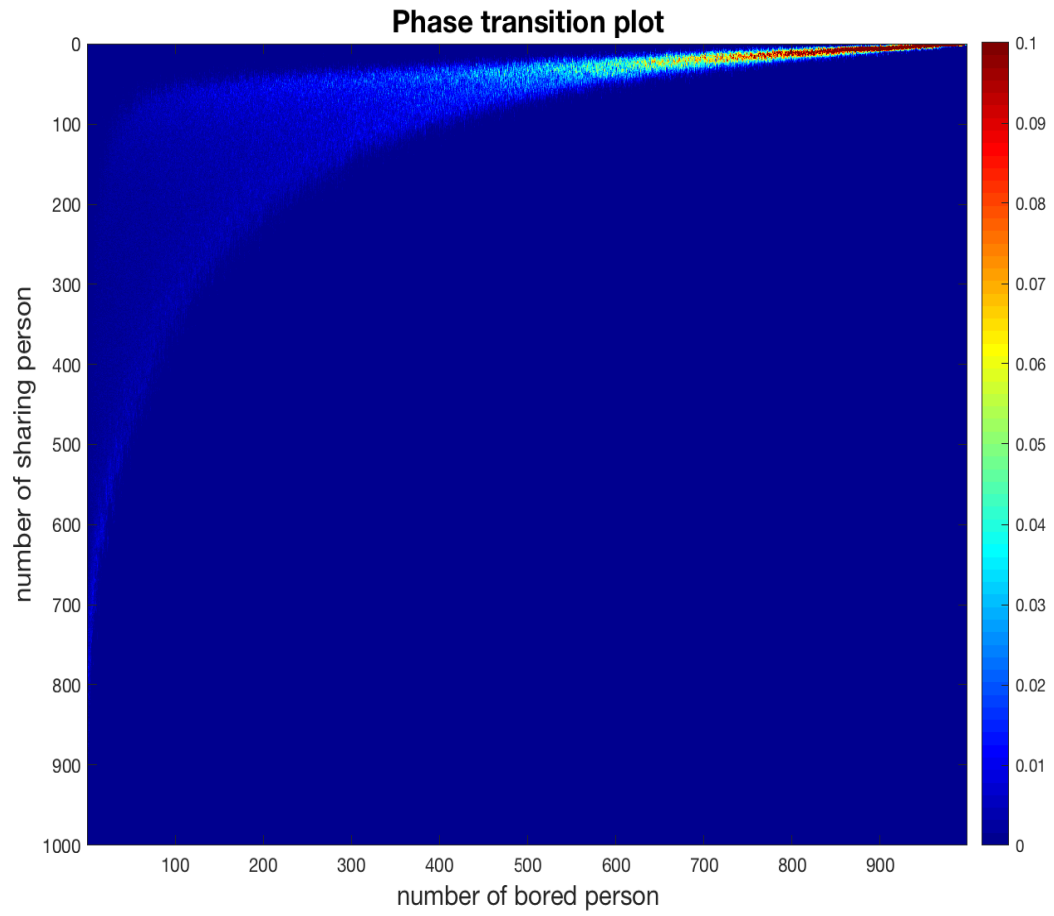


Figure 14: phase transition of total sharing person with simulation with $t = 1000$ and different $B(0)$ condition

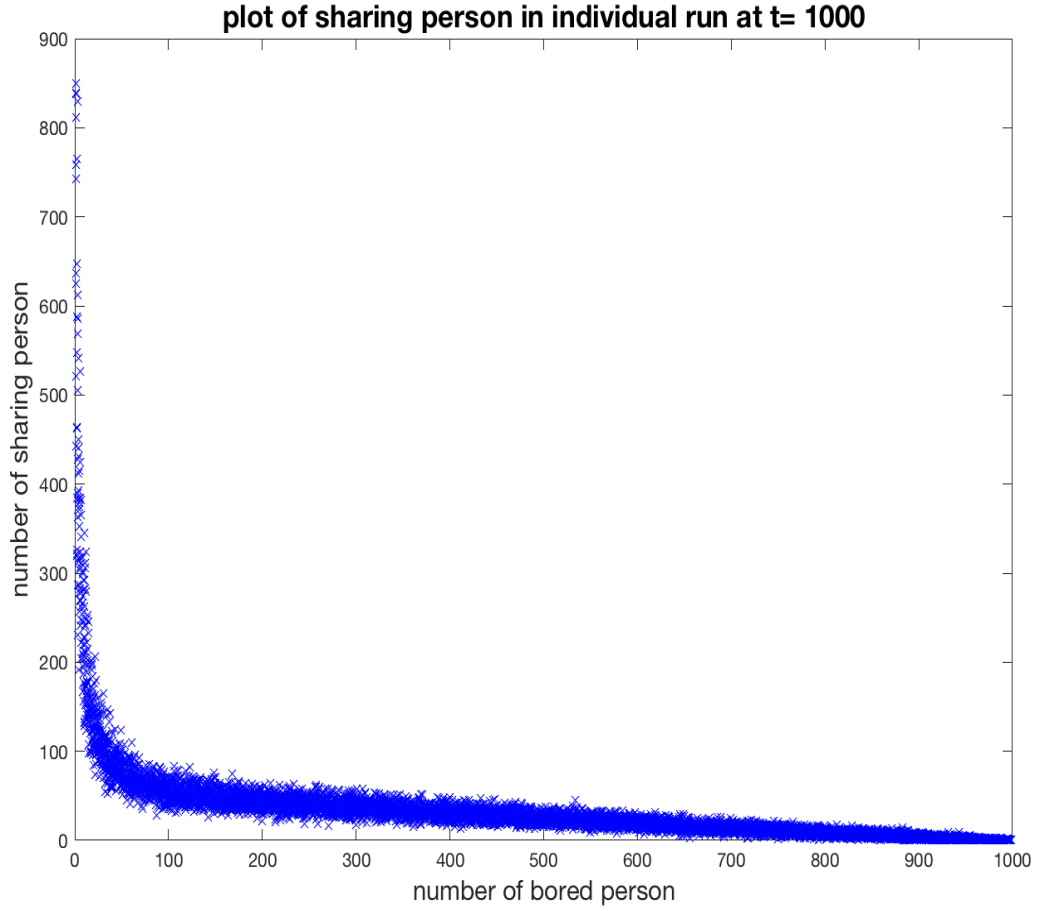


Figure 15: plot of sharing person in individual runs at $t = 1000$ for 100 simulations of $B_0 = 1:1:999$

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.

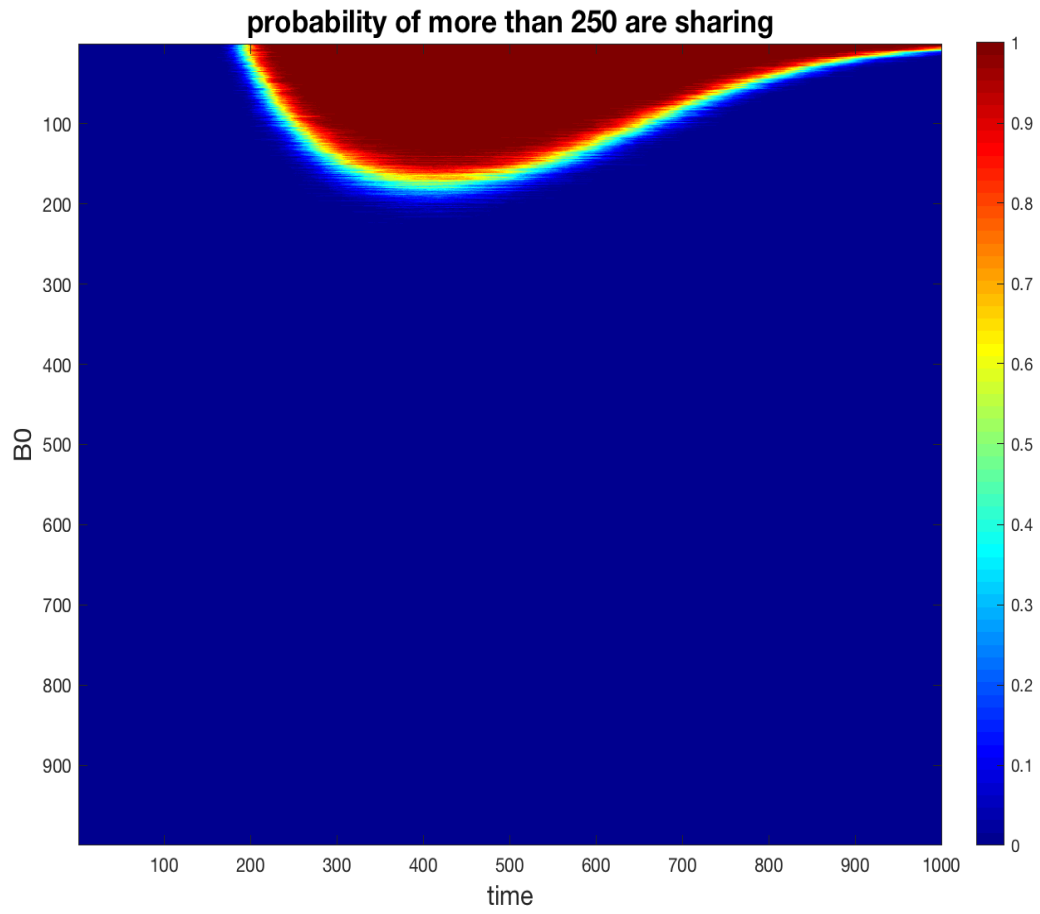


Figure 16: probability of more than 250 people are sharing with different B0 and time from 0 to 1000

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.

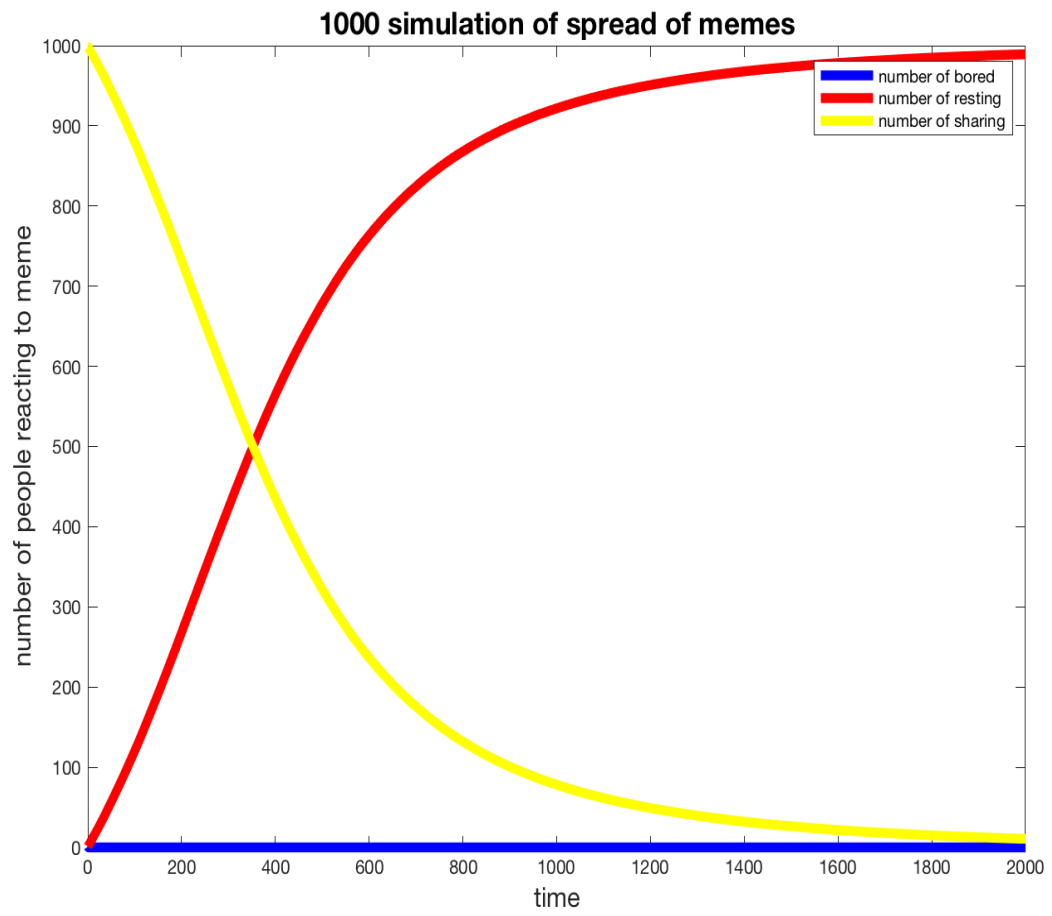


Figure 17: 100 simulations of memes with different rules for bored person over $t = 2000$

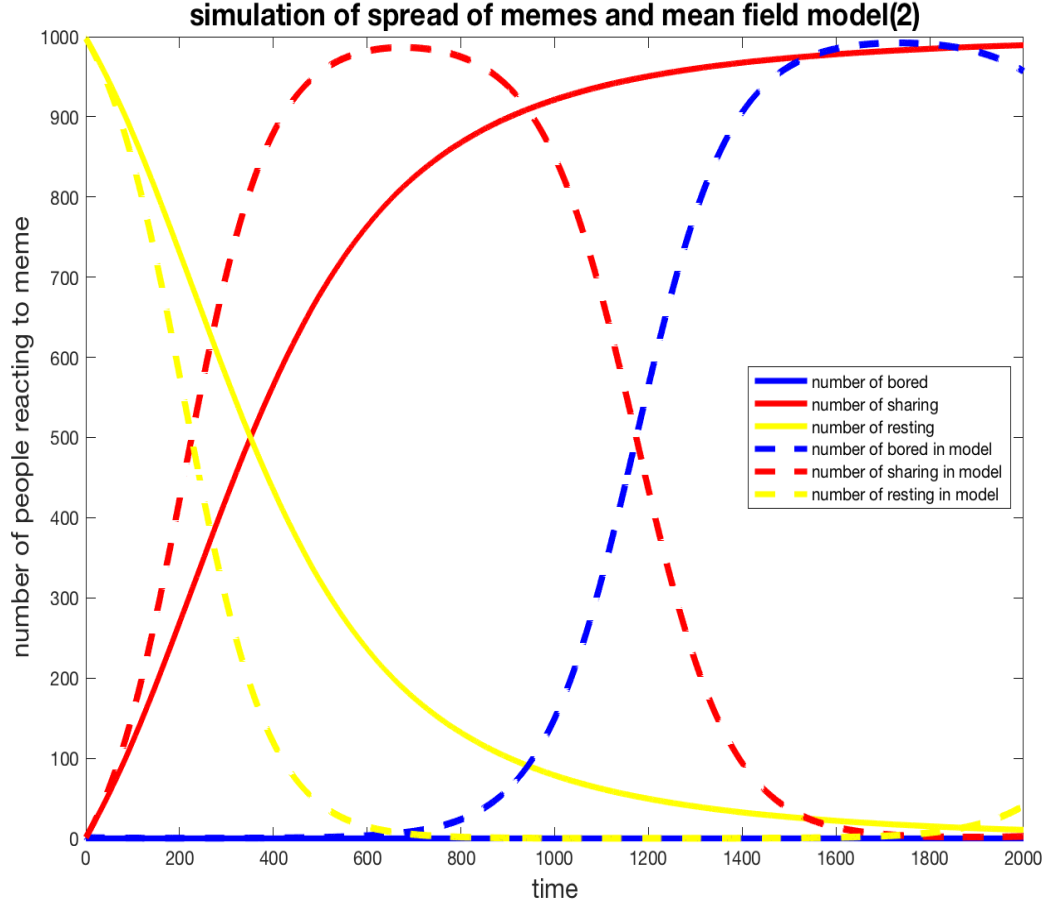


Figure 18: 100 simulations of memes with different rules for bored person over $t = 2000$ with mean field model plot

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 $p > 0.1$.

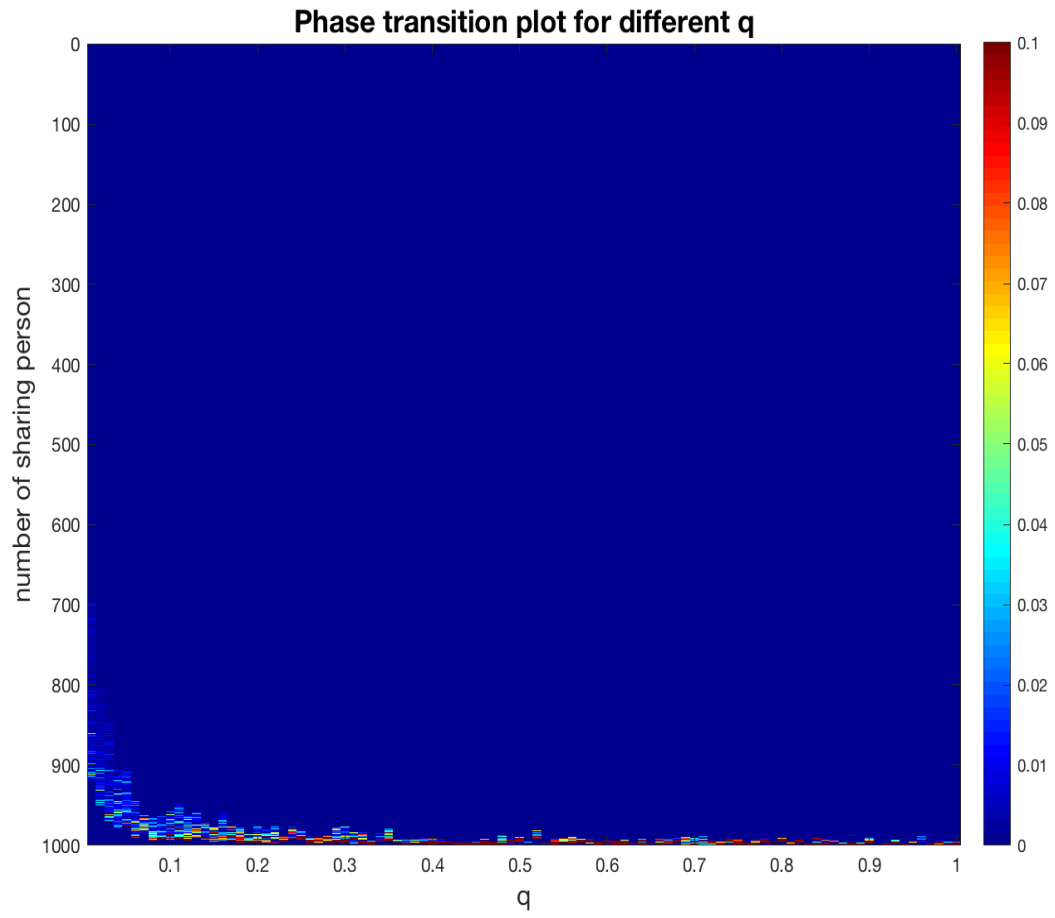


Figure 19: phase transition of total sharing person with simulation with $t = 1000$ and different q condition

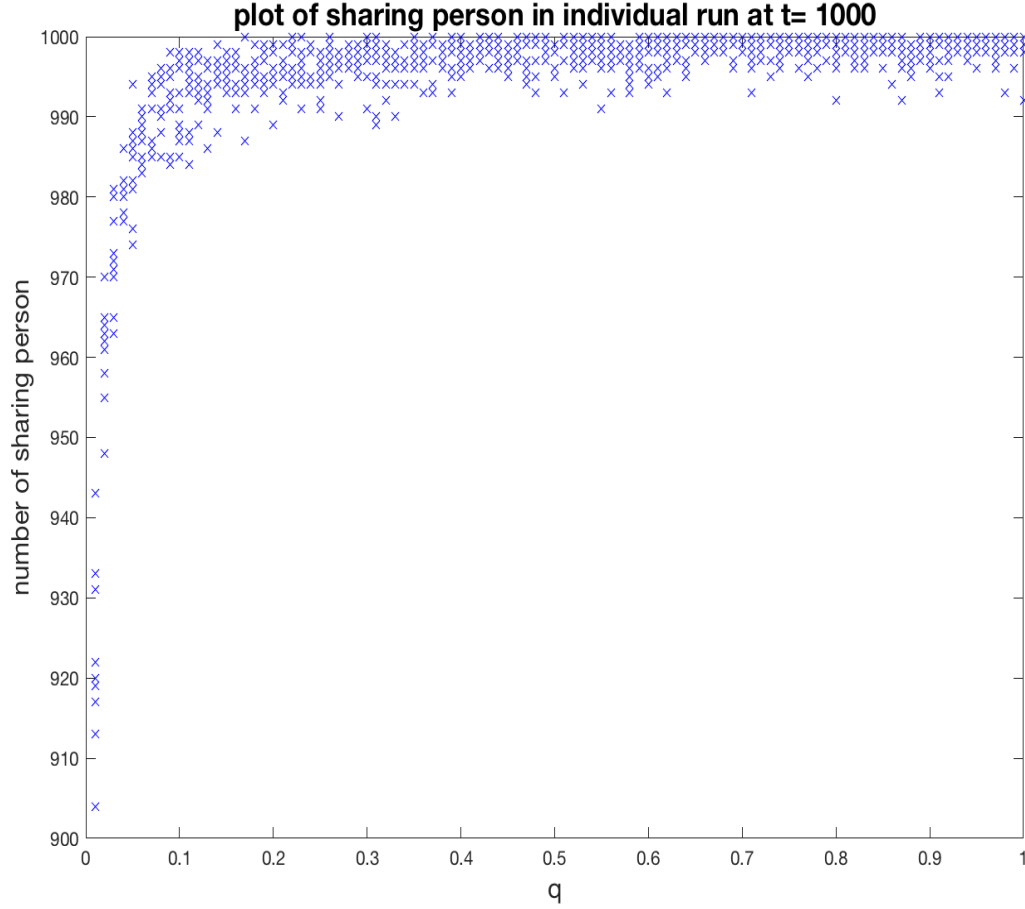


Figure 20: plot of sharing person in individual run at $t = 1000$

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.

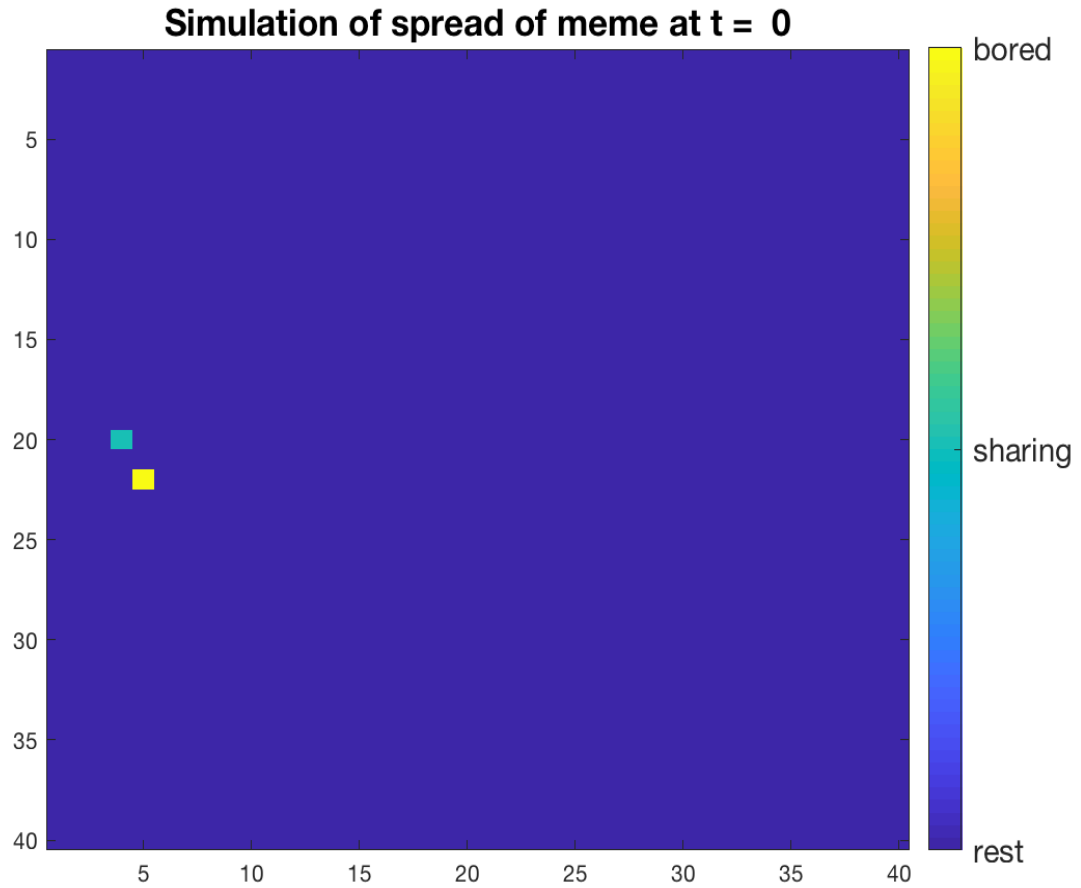


Figure 21: initial condition for simulation

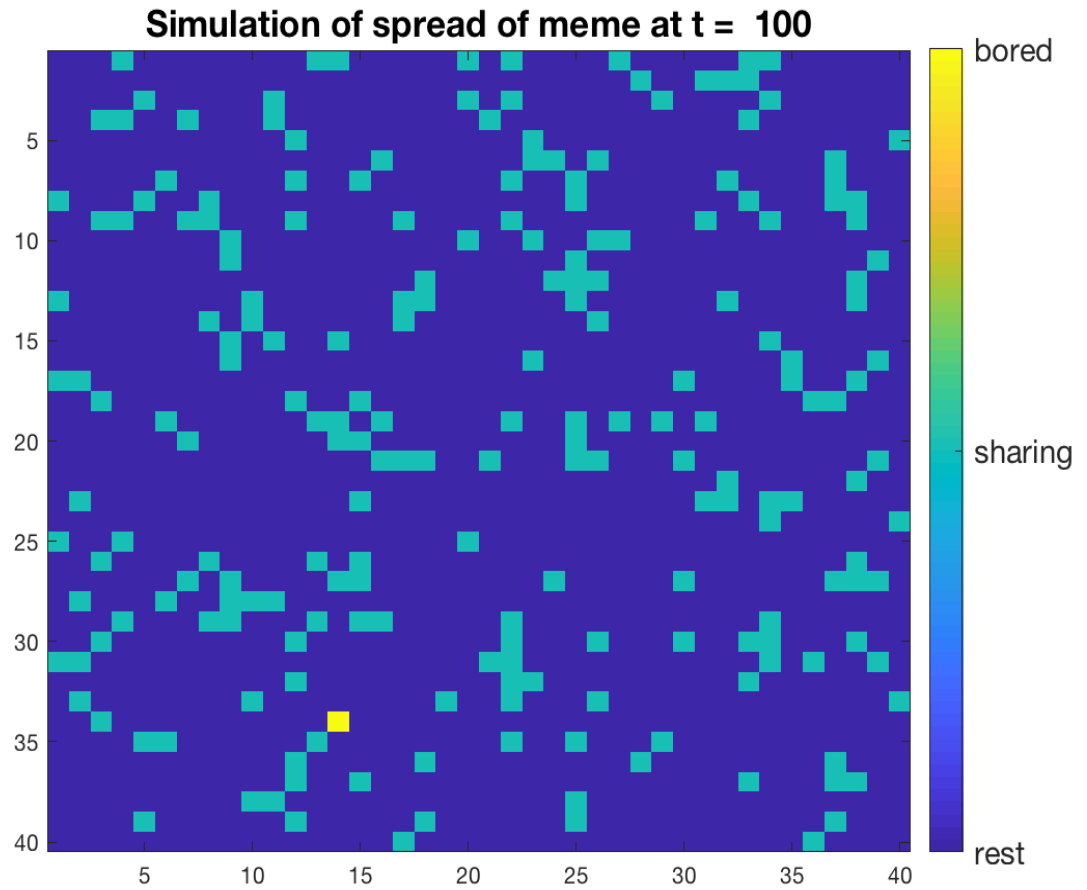


Figure 22: simulation of spread of meme to $t = 100$

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

```
1 clear all;
2 %this is to simualte 1 time at different time steps of a
   firing brain
3
4 n = 40;%input grid number
5 p = 0.3;
6 I = random_start(n,p);%generate initial state
7
8 t = 0;
9
10 v = VideoWriter('firebrain_1000.avi');
11 v.FrameRate = 10;
12 open(v);
13 figure('position',[0, 0, 700, 600])
14 imagesc(I);%plot
15 caxis([0 2]);
16 colorbar('Ticks',[0,1,2],'TickLabels',{'ready','firing',
   'rest'},'FontSize',14)
17 title(sprintf('Simulation of a firing brain at t = %s',
   string(t)),'FontSize',16)
18
19 axis tight manual
20 set(gca,'nextplot','replacechildren');
21 frame = getframe(gcf);
22 writeVideo(v,frame);
23
24 waitforbuttonpress;
25
26 endtime = 1000;
27
28
29 for t = 1:endtime
30
31     I = transit(n,I);
32
```

```

33     imagesc(I);
34     caxis([0 2]);
35     colorbar('Ticks',[0,1,2],'TickLabels',{'ready','firing','rest'},'FontSize',14)
36     title(sprintf('Simulation of a firing brain at t = %s', string(t)),'FontSize',16)
37
38     frame = getframe(gcf);
39     writeVideo(v,frame);
40
41     %pause(0.0025)
42
43 end
44 close(v)
45
46 %{
47 to save the final state
48 fig1 = imagesc(I);
49 caxis([0 2]);
50 colorbar('Ticks',[0,1,2],'TickLabels',{'ready','firing','rest'},'FontSize',14)
51 title('Simulation of a firing brain after 1000 time step', 'FontSize',16)
52 %saveas(fig1,'fire1000.png');
53 %}

```

- transition function

```

1 function y = transit(n, I)
2 %for firebrain
3 %ready 0, fire 1, rest 2
4
5 y_initial = I; %initial our transit value
6 y = zeros(n); %create matrix for next state with all
   zeros ready to write
7
8 for i = 1:n
9     for k = 1:n
10
11         %if fire go to rest
12         if y_initial(i,k) == 1

```

```

13         y(i,k) = 2;
14     end
15
16
17     % if rest go to ready
18     if y_initial(i,k) == 2
19         y(i,k) = 0;
20     end
21
22     % if ready go to fire if two neighbour are
23     firing
24     if y_initial(i,k) == 0
25         count = 0;
26
27         for p = (i-1) : (i+1)
28             for q = (k-1) : (k+1)
29                 %hard reset boundary condition to
30                 periodic
31                 if p == 0
32                     p = n;
33                 end
34
35                 if q == 0
36                     q = n;
37                 end
38
39                 if p == n + 1
40                     p = 1;
41                 end
42
43                 if q == n + 1
44                     q = 1;
45                 end
46
47                 if y_initial(p,q) == 1
48                     count = count + 1;
49                 end
50             end
51         end
52     if count == 2

```

```

53         y(i,k) = 1;
54     end
55 end
56 end
57 end
58 end

```

- initial state

```

1 function I = random_start(n, p)
2 I = zeros(n);
3 for i = 1:n
4     for j = 1:n
5         initial_state = rand;
6         if initial_state < p
7             I(i,j) = 1;
8         end
9     end
10 end
11 end

```

- simulate 100 times of firing brain

```

1 clear all;
2
3 n = 40;%input grid number
4 p = 0.3;
5
6 endtime = 1000;
7
8 for s_time = 1:100
9     I = random_start(n,p);%generate initial state
10    %{
11        imagesc(I);%plot
12        caxis([0 2]);
13        colorbar('Ticks',[0,1,2], 'TickLabels',{ 'ready', 'firing', 'rest'}, 'FontSize',14)
14        title('Simulation of a firing brain', 'FontSize',16)
15    %}

```

```

16     for t = 1:endtime+1
17         %waitforbuttonpress;
18
19         %plot
20         %{
21         imagesc(I)
22         caxis([0 2]);
23         colorbar('Ticks',[0,1,2],'TickLabels',{ 'ready', '
           firing', 'rest' }, 'FontSize',14)
24         title('Simulation of a firing brain','FontSize'
           ,16)
25         %}
26
27         %pause(0.0025)
28
29         firecell(s_time, t) = 0;
30         for i = 1:n
31             for j = 1:n
32                 if I(i,j) == 1
33                     firecell(s_time,t) = firecell(s_time,t)
34                         + 1;
35                 end
36             end
37         end
38         I = transit(n,I);
39     end
40 end
41
42 %to plot
43
44 fig2 = figure('position',[0, 0, 700, 500]);
45 x = 0:1:1000;
46 plot(x,mean(firecell),'LineWidth',3);
47 ylim([0 600])
48 xlabel('time step','FontSize',14);
49 ylabel('number of firing cell','FontSize',14);
50 ylim([0 1001]);
51 title('100 simulations of a firing brain to 1000 time
           steps','FontSize',16);
52 %legend('total number of bored','total number of resting
           ','total number of sharing');

```

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

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

```
1 clear all;
2
3 %input grid number
4 n = 40;
5
6 I = zeros(n); %create grid
7
8 I(5,5) = 1;
9 I(5,6) = 1;
10 I(6,5) = 2;
11 I(6,6) = 2;
12
13 I(20,21) = 1;
14 I(20,20) = 1;
15 I(21,20) = 2;
16 I(21,21) = 2;
17 I(22,19) = 1;
18 I(22,22) = 1;
19 I(23,19) = 2;
20 I(23,22) = 2;
21
22 imagesc(I); %plot
23 fig1 = imagesc(I);
24 caxis([0 2]);
25 colorbar('Ticks',[0,1,2], 'TickLabels',{'ready','firing',
    'rest'}, 'FontSize',14)
26 title('shapes that move forward at a rate of one cell
    per time step', 'FontSize',16)
27 %saveas(fig1, 'task2_1.png');
28
29
30 waitforbuttonpress;
31 %for tentime = 1:10
32 for t = 1:200
33
34     I = transit(n,I);
```

```

35     imagesc(I)
36     %pause(0.25)
37
38 end

```

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

```

1  clear all;
2
3  %I = random_start(40,0.2);
4  %if you want a random start
5
6  %plot fireing number over time
7
8  %input grid number
9  n = 40;
10
11 I = zeros(n); %create grid
12
13 I(29,20) = 1;
14 I(29,21) = 2;
15 I(30,20) = 1;
16 I(30,21) = 2;
17 I(30,27) = 1;
18 I(30,28) = 2;
19 I(30,30) = 1;
20 I(30,31) = 1;
21 I(31,21) = 1;
22 I(31,22) = 2;
23 I(31,25) = 1;
24 I(31,26) = 2;
25 I(31,30) = 2;
26 I(31,31) = 2;
27 I(32,23) = 1;
28 I(32,24) = 2;
29 I(32,27) = 2;
30 I(32,28) = 2;
31 I(32,29) = 1;
32 I(32,30) = 1;
33 I(33,25) = 1;

```



```

34 I(33,26) = 2;
35 I(33,28) = 1;
36 I(33,32) = 2;
37 %{
38 I(21,20) = 1;
39 I(21,21) = 1;
40 I(19,20) = 2;
41 I(20,22) = 2;
42 I(22,21) = 2;
43 I(21,19) = 2;
44 %}
45
46 imagesc(I); %plot
47 waitforbuttonpress;
48
49 for t = 1:15
50
51     I = transit(n,I);
52
53     imagesc(I)
54     pause(0.25)
55
56 end
57
58 fig1 = imagesc(I);
59 caxis([0 2]);
60 colorbar('Ticks',[0,1,2], 'TickLabels',{ 'ready', 'firing',
        'rest'}, 'FontSize',14)
61 title('lauching other shapes behind', 'FontSize',16)
62 %saveas(fig1, 'task2_2.png');

```

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

```

1 clear all;
2
3 %input grid number
4 n = 40;
5
6 I = zeros(n); %create grid
7
8 I(20,20) = 2;

```

```

 9  I(20,21) = 2;
10  I(18,22) = 2;
11  I(19,20) = 1;
12  I(19,22) = 1;
13  I(21,20) = 1;
14  %{
15  I(21,20) = 1;
16  I(21,21) = 1;
17  I(19,20) = 2;
18  I(20,22) = 2;
19  I(22,21) = 2;
20  I(21,19) = 2;
21  %}
22
23  imagesc(I); %plot
24  waitforbuttonpress;
25
26  for t = 1:20
27
28      I = transit(n,I);
29      %waitforbuttonpress;
30      imagesc(I)
31      pause(0.2)
32  end
33
34  fig1 = imagesc(I);
35  caxis([0 2]);
36  colorbar('Ticks',[0,1,2], 'TickLabels',{ 'ready', 'firing',
37      'rest'}, 'FontSize',14)
38  title('shape that move less than one cell per time step',
39      'FontSize',16)
40  saveas(fig1, 'task2_3.png');

```

- oscillate shape

```

1  clear all;
2
3  %input grid number
4  n = 40;
5
6  I = zeros(n); %create grid

```

```

7
8 I(20,20) = 1;
9 I(20,21) = 1;
10 I(21,20) = 1;
11 I(21,21) = 1;
12 I(19,20) = 2;
13 I(20,22) = 2;
14 I(22,21) = 2;
15 I(21,19) = 2;
16
17 imagesc(I); %plot
18 waitforbuttonpress;
19
20 for t = 1:20
21
22     I = transit(n,I);
23     imagesc(I)
24     pause(0.25)
25 end
26 fig1 = imagesc(I);
27 caxis([0 2]);
28 colorbar('Ticks',[0,1,2], 'TickLabels',{'ready','firing',
    'rest'}, 'FontSize',14)
29 title('shape that oscillate periodically', 'FontSize',16)
30 saveas(fig1, 'task2_4.png');

```

- my cellular automata

```

1 clear all;
2 %input grid number
3 n = 100;
4
5 I = zeros(n); %create grid
6
7 imagesc(I);
8
9 %generate initial state
10 for i = 1:n
11     for j = 1:n
12         initial_state = rand;
13         if initial_state < 0.05

```

```

14         I(i,j) = 1;
15         elseif initial_state < 0.1
16             I(i,j) = 2;
17         elseif initial_state < 0.25
18             I(i,j) = 3;
19         elseif initial_state < 0.35
20             I(i,j) = 4;
21         end
22
23
24     end
25 end
26 %
27
28
29 t = 0;
30
31 v = VideoWriter('myca_t1000.avi');
32 v.FrameRate = 10;
33 open(v);
34 figure('position', [0, 0, 700, 600])
35 imagesc(I);
36 colormap jet
37 caxis([0 4]);
38 colorbar('Ticks',[0,1,2,3,4], 'TickLabels',{ 'waiting', '
    growing', 'reproducing', 'aging', 'dead' }, 'FontSize',14)
39 title(sprintf('Simulation of population t = %s', string
    (t)), 'FontSize',16)
40 axis tight manual
41 set(gca, 'nextplot', 'replacechildren');
42 frame = getframe(gcf);
43 writeVideo(v, frame);
44
45
46 waitforbuttonpress;
47
48 %for tentime = 1:10
49 for t = 1:1000
50
51     I = transitown(n,I);
52
53     imagesc(I)

```

```

54     colorbar
55     pause(0.25)
56
57
58     imagesc(I);
59     colormap jet
60     caxis([0 4]);
61     colorbar('Ticks',[0,1,2,3,4],'TickLabels',{'waiting',
        'growing','reproducing','aging','dead'},'
        FontSize',14)
62     title(sprintf('Simulation of population t = %s',
        string(t)),'FontSize',16)
63     frame = getframe(gcf);
64     writeVideo(v,frame);
65
66     %pause(0.1)
67 end
68
69 close(v)

```

- my cellular automata transit

```

1 function y = transitown(n, I)
2 %waiting 0 grow 1 reproduce 2 mature 3 dead 4
3
4 y_initial = I; %initial our transit value
5 y = zeros(n); %create matrix for next state with all
    zeros ready to write
6
7 for i = 1:n
8     for k = 1:n
9
10         %if dead 0.2 chance go to wait
11         if y_initial(i,k) == 4
12             if rand < 0.6
13                 y(i,k) = 0;
14             else
15                 y(i,k) = 4;
16             end
17         end
18

```

```

19     if y_initial(i,k) == 3 %if mature 50% go to dead
20         if rand < 0.5
21             y(i,k) = 4;
22         else
23             y(i,k) = 3;
24         end
25     end
26
27
28     % if 0 need to start grow
29     if y_initial(i,k) == 0
30         count = 0;
31
32         for p = (i-1) : (i+1)
33             for q = (k-1) : (k+1)
34                 %hard reset boundary condition to
35                 periodic >_<
36                 if p == 0
37                     p = n;
38                 end
39
40                 if q == 0
41                     q = n;
42                 end
43
44                 if p == n + 1
45                     p = 1;
46                 end
47
48                 if q == n + 1
49                     q = 1;
50                 end
51
52                 if y_initial(p,q) == 2
53                     count = count + 1;
54                 end
55             end
56         end
57
58         if count >1
59             y(i,k) = 1;

```

```

60         else
61             y(i,k) = 0;
62         end
63     end
64
65     % if 1 need to get at one reproduce and mature
66     % to survice
67     if y_initial(i,k) == 1
68         count = 0;
69         count2 = 0;
70
71         for p = (i-1) : (i+1)
72             for q = (k-1) : (k+1)
73                 %hard reset boundary condition to
74                 %periodic >_<
75                 if p == 0
76                     p = n;
77                 end
78
79                 if q == 0
80                     q = n;
81                 end
82
83                 if p == n + 1
84                     p = 1;
85                 end
86
87                 if q == n + 1
88                     q = 1;
89                 end
90
91                 if y_initial(p,q) == 2
92                     count = count + 1;
93                 end
94                 if y_initial(p,q) == 3
95                     count2 = count2 + 1;
96                 end
97             end
98         end
99
100     if count >= 1 && count2 >= 1

```

```

100         y(i,k) = 2;
101     else
102         y(i,k) = 1;
103     end
104 end
105     % if 2 need to reproduce
106     if y_initial(i,k) == 2
107         if rand < 0.5
108             y(i,k) = 3;
109         else
110             y(i,k) = 2;
111         end
112     end
113 end
114
115
116     end
117 end
118 end

```

3.2 Spread of memes

- simulation of spread of memes

```

1  clear all;
2
3  n = 1000;
4  p = 0.001;%discover new meme
5  q = 0.01;%share a meme
6  total_time_step = 2000;
7  sim_number = 1000;
8  Bt = 0;
9  St = 0;
10 Rt = 0;
11
12
13 for rep = 1:sim_number
14     rep
15
16     [B,S,R] = runmeme(n,total_time_step,p,q);

```



```

17     Bt = Bt + B;
18     St = St + S;
19     Rt = Rt + R;
20 end
21
22 result_B = Bt/sim_number;
23 result_S = St/sim_number;
24 result_R = Rt/sim_number;
25 x = 0:1:total_time_step;
26
27 my_fig = figure('position', [0, 0, 700, 500]);
28 plot(x,result_B,'b','LineWidth',5);
29 hold on;
30 plot(x,result_S,'r','LineWidth',5);
31 plot(x,result_R,'y','LineWidth',5);
32 xlabel('time','FontSize',14);
33 ylabel('number of people reacting to meme','FontSize',
    ,14);
34 title('1000 simulation of spread of memes','FontSize',
    ,16);
35 legend('number of bored','number of resting','number of
    sharing');
36 saveas(my_fig,'memes_sim_1000times.png');
37 hold off;

```

- run single spread of memes

```

1 function [B,S,R] = runmeme(n,total_time_step,p,q)
2
3 %n = 1000;
4 %p = 0.001;%discover new meme
5 %q = 0.01; %share a meme
6
7 N = zeros(total_time_step,n);
8
9
10 start = randperm(1000,2); %generate the person who (
    bored, share)
11
12 N(1,start(1)) = 2;%bored 2
13 N(1,start(2)) = 1; %sharing

```

```

14
15
16 for t = 1:1:total_time_step
17
18     for i = 1:n
19
20         if N(t,i) == 2%if bored stay bored forever
21             N(t+1, i) = 2;
22
23         end
24
25         if N(t,i) == 0 %if rest will discover a new meme
26             if rand < p % found a new meme at
27                 probability p and be sharer
28                 N(t+1, i) = 1;
29             else
30                 N(t+1, i) = 0;
31             end
32         end
33
34         if N(t,i) == 1
35             if rand < q
36                 target = randi([1 1000]);
37                 if N(t,target) == 2
38                     N(t+1, i) = 2;
39                 else
40                     N(t+1, target) = 1;
41                     N(t+1, i) = 1;
42                 end
43             else
44                 N(t+1,i) = 1;
45             end
46         end
47     end
48
49
50 for t = 1:1:total_time_step+1
51     B(t) = 0;
52     S(t) = 0;
53     R(t) = 0;
54     for i = 1:n

```

```

55         if N(t,i) == 2
56             B(t) = B(t) + 1;
57         end
58         if N(t,i) == 1
59             S(t) = S(t) + 1;
60         end
61         if N(t,i) == 0
62             R(t) = R(t) + 1;
63         end
64     end
65 end
66
67 end

```

- mean field model

```

1  clear all;
2
3  n = 1000;
4  p = 0.001;%discover new meme
5  q = 0.01;%share a meme
6  total_time_step = 2000;
7  sim_number = 1;
8
9  for rep = 1:sim_number
10     [B,S,R] = mememodel(n,total_time_step,p,q);
11
12     m_result_B(rep,:) = B;
13     m_result_S(rep,:) = S;
14     m_result_R(rep,:) = R;
15 end
16
17 load('1000_sim_memes.mat');
18 x = 0:1:total_time_step;
19
20 my_fig = figure('position',[0, 0, 700, 500]);
21 plot(x,result_B,'b','LineWidth',3);
22 hold on;
23 plot(x,result_S,'r','LineWidth',3);
24 plot(x,result_R,'y','LineWidth',3);
25 plot(x,m_result_B,'—b','LineWidth',3);

```

```

26 plot(x,m_result_S,'r—','LineWidth',3);
27 plot(x,m_result_R,'y—','LineWidth',3);
28
29 xlabel('time','FontSize',14);
30 ylabel('number of people reacting to meme','FontSize',
    ,14);
31 title('simulation of spread of memes and mean field
    model','FontSize',16);
32 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');
33 saveas(my_fig,'memes-withmodel1000.png');
34 hold off;

```

- phase transition

```

1 clear all;
2
3 n = 1000;
4 pp = 0.001;%discover new meme
5 q = 0.01; %share a meme
6 total_time_step = 1000;
7 %sim_number = 1000;%a lot @.@ for my computer
8 Bt = 0;
9 St = 0;
10 Rt = 0;
11
12 Bvals = [1:1:999];
13 numreps = 10;
14 hrange = [0:1:n];
15
16 histu = zeros(length(Bvals), length(hrange));
17 count = 0;
18
19 for b = Bvals
20     Sall = [];
21     count = count + 1
22
23     for rep = 1:numreps
24         [B,S,R] = runmeme_phase(n,total_time_step,pp,q,b

```

```

24         );
25     S = S';
26
27     finalS(count,rep) = S(end);
28     maxS(count,rep) = max(S);
29     Sall = [Sall;S((end-499):end)];
30 end
31 %take the histogram
32 histu(count,:) = hist(Sall,hrange);
33 end
34
35 figure2 = figure('position',[0, 0, 700, 500]);
36 imagesc(Bvals,hrange,histu'/(numreps*500),[0 0.1])
37 colormap jet
38 hlx=xlabel('number of bored person','FontSize',14)
39 hly=ylabel('number of sharing person','FontSize',14)
40 title('Phase transition plot','FontSize',16)
41 colorbar
42 saveas(figure2,'phasetransition.png');
43
44 figure3 = figure('position',[0, 0, 700, 500]);
45 %Plot the results
46 count=0;
47 for pp=Bvals
48     count=count+1;
49     pp;
50     for rep=1:numreps
51         plot(pp,finalS(count,rep),'bx')
52         hold on
53     end
54 end
55
56 xlabel('number of bored person','FontSize',14)
57 ylabel('number of sharing person','FontSize',14)
58 title('plot of sharing person in individual run at t=
    1000','FontSize',16)
59 saveas(figure3,'indi-phase.png');

```

- probability for at least 25% are sharing

```
1 clear all;
```

```

2  n = 1000;
3  p = 0.001;%discover new meme
4  q = 0.01; %share a meme
5  total_time_step = 1000;
6
7  Bvals = [1:1:999];
8  numreps = 100;
9  tval = [1:1:total_time_step];
10
11 p25 = zeros(999, total_time_step);
12 count = 0;
13
14 for b = Bvals
15     b
16     for i = 1:1:numreps
17         [B,S,R] = runmeme_phase(n,total_time_step ,p,q,b)
18         ;
19         for t = 1:total_time_step
20             if S(t) > 250
21                 p25(b,t) = p25(b,t) + 0.01;
22             end
23         end
24     end
25 figure2 = figure('position', [0, 0, 700, 500]);
26 imagesc(p25)
27 colormap jet
28 hlx=xlabel('time','FontSize',14)
29 hlx=ylabel('B0','FontSize',14)
30 title('probability of more than 250 are sharing', '
    FontSize', 16)
31 colorbar
32 %saveas(figure2,'probability250_100sim.png');

```

- simulation of spread of memes with new rules

```

1  clear all;
2
3  n = 1000;
4  p = 0.001;%discover new meme
5  q = 0.01; %share a meme

```

```

6 total_time_step = 2000;
7 sim_number = 1000;
8 Bt = 0;
9 St = 0;
10 Rt = 0;
11
12
13 for rep = 1:sim_number
14     rep
15
16     [B,S,R] = runmeme_newbored(n,total_time_step,p,q);
17     Bt = Bt + B;
18     St = St + S;
19     Rt = Rt + R;
20 end
21
22 result_B = Bt/sim_number;
23 result_S = St/sim_number;
24 result_R = Rt/sim_number;
25 x = 0:1:total_time_step;
26
27 my_fig = figure('position',[0,0,700,500]);
28 plot(x,result_B,'b','LineWidth',5);
29 hold on;
30 plot(x,result_S,'r','LineWidth',5);
31 plot(x,result_R,'y','LineWidth',5);
32 xlabel('time','FontSize',14);
33 ylabel('number of people reacting to meme','FontSize',14);
34 title('1000 simulation of spread of memes','FontSize',16);
35 legend('number of bored','number of resting','number of sharing');
36 %saveas(my_fig,'memes2_sim_1000times.png');
37 hold off;
38 save('meme2data.mat','result_B','result_R','result_S')

```

- run single spread of memes

```

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

```

```

2
3 %n = 1000;
4 %p = 0.001;%discover new meme
5 %q = 0.01; %share a meme
6
7 N = zeros(total_time_step ,n);
8
9 start = randperm(1000,2); %generate the person who (
    bored , share)
10
11 N(1,start(1)) = 2;%bored 2
12 N(1,start(2)) = 1;%share 1
13
14 for t = 1:1:total_time_step
15
16     for i = 1:n
17
18         if N(t,i) == 2%if bored will find someone
19             if rand < q
20                 target = randi([1 1000]);
21                 if N(t,target) == 0
22                     N(t+1,i) = 0;
23                 else
24                     N(t+1,i) = 2;
25                 end
26             end
27
28         end
29
30         if N(t,i) == 0 %if rest will discover a new meme
31             if rand < p % found a new meme at
                probability p and be sharer
32                 N(t+1, i) = 1;
33             else
34                 N(t+1, i) = 0;
35             end
36         end
37
38         if N(t,i) == 1
39             if rand < q
40                 target = randi([1 1000]);
41                 if N(t,target) == 2

```



```

42             N(t+1, i) = 2;
43         else
44             N(t+1, target) = 1;
45             N(t+1, i) = 1;
46         end
47     else
48         N(t+1,i) = 1;
49     end
50 end
51 end
52 end
53
54
55 for t = 1:1:total_time_step+1
56     B(t) = 0;
57     S(t) = 0;
58     R(t) = 0;
59     for i = 1:n
60         if N(t,i) == 2
61             B(t) = B(t) + 1;
62         end
63         if N(t,i) == 1
64             S(t) = S(t) + 1;
65         end
66         if N(t,i) == 0
67             R(t) = R(t) + 1;
68         end
69     end
70 end
71
72 end

```

- mean field model

```

1  clear all;
2
3  n = 1000;
4  p = 0.001;%discover new meme
5  q = 0.01; %share a meme
6  total_time_step = 2000;
7  sim_number = 1;

```

```

8
9  for rep = 1:sim_number
10     [B,S,R] = mememodel2(n,total_time_step,p,q);
11
12     m_result_B(rep,:) = B;
13     m_result_S(rep,:) = S;
14     m_result_R(rep,:) = R;
15 end
16
17 load('meme2data.mat');
18 x = 0:1:total_time_step;
19
20 my_fig = figure('position',[0,0,700,500]);
21 plot(x,result_B,'b','LineWidth',3);
22 hold on;
23 plot(x,result_S,'r','LineWidth',3);
24 plot(x,result_R,'y','LineWidth',3);
25 plot(x,m_result_B,'—b','LineWidth',3);
26 plot(x,m_result_S,'r—','LineWidth',3);
27 plot(x,m_result_R,'y—','LineWidth',3);
28
29 xlabel('time','FontSize',14);
30 ylabel('number of people reacting to meme','FontSize',
    14);
31 title('simulation of spread of memes and mean field
    model(2)','FontSize',16);
32 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');
33 saveas(my_fig,'memes2-withmodel1000.png');
34 hold off;

```

- phase transition for new rules

```

1  clear all;
2
3  n = 1000;
4  pp = 0.001;%discover new meme
5  q = 0.01;%share a meme
6  total_time_step = 1000;

```

```

7 %sim_number = 1000;%a lot @@ for my computer
8 Bt = 0;
9 St = 0;
10 Rt = 0;
11
12 qvals = [0.01:0.01:1];
13 numreps = 10;
14 hrange = [0:1:n];
15
16 histu = zeros(length(qvals), length(hrange));
17 count = 0;
18
19 for b = qvals
20     Sall = [];
21     count = count + 1
22
23     for rep = 1:numreps
24         [B,S,R] = runmeme_newbored(n,total_time_step,pp,
25             b);
26         S = S';
27         finalS(count,rep) = S(end);
28         Sall = [Sall;S((end-499):end)];
29     end
30     %take the histogram
31     histu(count,:) = hist(Sall,hrange);
32
33 end
34
35 figure2 = figure('position',[0, 0, 700, 500]);
36 imagesc(qvals,hrange,histu'/(numreps*500),[0 0.1])
37 colormap jet
38 hlx=xlabel('q','FontSize',14)
39 hly=ylabel('number of sharing person','FontSize',14)
40 title('Phase transition plot for different q','FontSize',16)
41
42 colorbar
43 %saveas(figure2,'phasetransition2t1000withB.png');
44
45 figure3 = figure('position',[0, 0, 700, 500]);
46 %Plot the results
47 count=0;
48 for pp=qvals
49     count=count+1;

```

```

47     pp;
48     for rep=1:numreps
49         plot(pp,finalS(count,rep),'bx')
50         hold on
51     end
52 end
53
54 xlabel('q','FontSize',14)
55 ylabel('number of sharing person','FontSize',14)
56 title('plot of sharing person in individual run at t=
      1000','FontSize',16)
57 saveas(figure3,'indi_phase2t1000withq.png');

```

- lattice simulation for memes

```

1  clear all;
2  %input grid number
3  n = 40;
4  p = 0.001;
5  q = 0.01;
6  I = zeros(n); %create grid
7  imagesc(I);
8  initial_x = randperm(n,2);
9  initial_y = randperm(n,2);
10
11
12 %1, sharing person
13 I(initial_x(1),initial_y(1)) = 1;
14 %2. bored person
15 I(initial_x(2),initial_y(2)) = 2;
16
17 t = 0;
18
19 v = VideoWriter('memes_3_t1000.avi');
20 v.FrameRate = 10;
21 open(v);
22 figure('position',[0,0,700,600])
23 imagesc(I);
24 caxis([0 2]);
25 colorbar('Ticks',[0,1,2],'TickLabels',{'rest','sharing',
      'bored'},'FontSize',14)

```

```

26 title(sprintf('Simulation of spread of meme at t = %s',
    string(t)), 'FontSize',16)
27 axis tight manual
28 set(gca, 'nextplot', 'replacechildren');
29 frame = getframe(gcf);
30 writeVideo(v,frame);
31
32
33 waitforbuttonpress;
34
35 for t = 1:1000
36     I = transit_meme(n,I,p,q);
37
38     %figure('position', [0, 0, 700, 600])
39     imagesc(I);
40     caxis([0 2]);
41     colorbar('Ticks',[0,1,2], 'TickLabels',{ 'rest', '
        sharing', 'bored' }, 'FontSize',14)
42     title(sprintf('Simulation of spread of meme at t =
        %s', string(t)), 'FontSize',16)
43
44     frame = getframe(gcf);
45     writeVideo(v,frame);
46
47     %pause(0.1)
48
49
50 end
51
52 close(v)

```

- transition function for memes

```

1 function y = transit_meme(n,I,p,q)
2 %rest0, share1, bored2
3 %n grid set to 40*40
4 %I matrix
5 %p probability find a new meme for a rest
6 %q probability for share and bored to find a person
7
8 y_initial = I; %initial our transit value

```

```

9  y = zeros(n); %create matrix for next state with all
    zeros ready to write
10
11  for i = 1:n
12      for k = 1:n
13
14          %%%%%%%%%bored
15          %if bored will find someone
16          if y_initial(i,k) == 2
17              if rand < q
18                  target = randi(8);
19                  if target == 1
20                      loc = [i-1,k-1];
21                  elseif target == 2
22                      loc = [i-1,k];
23                  elseif target == 3
24                      loc = [i-1,k+1];
25                  elseif target == 4
26                      loc = [i,k-1];
27                  elseif target == 5
28                      loc = [i,k+1];
29                  elseif target == 6
30                      loc = [i+1,k-1];
31                  elseif target == 7
32                      loc = [i+1,k];
33                  else
34                      loc = [i+1,k+1];
35                  end
36
37
38          %hard reset boundary
39          if loc(1) == 0
40              loc(1) = n;
41          end
42          if loc(1) == n+1
43              loc(1) = 1;
44          end
45          if loc(2) == 0
46              loc(2) = n;
47          end
48          if loc(2) == n+1
49              loc(2) = 1;

```

```

50         end
51
52         if y_initial(loc(1),loc(2)) == 0
53             y(i,k) = 0;
54         else
55             y(i,k) = 2;
56         end
57     else
58         y(i,k) = 2;
59     end
60 end
61
62
63 if y_initial(i,k) == 0%if rest go find new meme
64     if rand < p %p to find a meme and become a
65         sharer
66             y(i,k) = 1;
67     else
68         y(i,k) = 0; %stay rest
69     end
70 end
71
72 if y_initial(i,k) == 1%if share go find people
73     to share
74     if rand < q
75         target = randi(8);
76         if target == 1
77             loc = [i-1,k-1];
78         elseif target == 2
79             loc = [i-1,k];
80         elseif target == 3
81             loc = [i-1,k+1];
82         elseif target == 4
83             loc = [i,k-1];
84         elseif target == 5
85             loc = [i,k+1];
86         elseif target == 6
87             loc = [i+1,k-1];
88         elseif target == 7
89             loc = [i+1,k];
90         else
91             loc = [i+1,k+1];

```

```

90         end
91
92
93         %hard reset boundary
94         if loc(1) == 0
95             loc(1) = n;
96         end
97         if loc(1) == n+1
98             loc(1) = 1;
99         end
100        if loc(2) == 0
101            loc(2) = n;
102        end
103        if loc(2) == n+1
104            loc(2) = 1;
105        end
106
107        if y_initial(loc(1),loc(2)) == 0
108            y(loc(1),loc(2)) = 1;
109            y(i,k) = 1;
110        elseif y_initial(loc(1),loc(2)) == 2
111            y(i,k) = 2;
112        else
113            y(i,k) = 1;
114        end
115
116
117        else
118            y(i,k) = 1;
119        end
120    end
121
122    end
123 end
124 end

```