



UPPSALA UNIVERSITET

MODELLING COMPLEX SYSTEMS

Project 2

Population Dynamics

Groups of Friends

Network Epidemics

Flocks and Predators

Peili Guo

Peili.Guo.7645@student.uu.se

May 5, 2018

1 Population Dynamics

In this part, we model the population with a stochastic model. There are n resource sites in the model world, and at time $t=0$, the population is A_0 and they are assigned randomly to one resource site. at each t step, the population rules are, if there are exactly two individuals on the same site. They reproduce b offsprings and these offsprings are assigned randomly to resources sites. if the number of individuals on a resources sites are any number other than 2, no offspring reproduced.

1.1 matlab model

To begin, we can run this with different parameters and simulate in matlab and observe the total number of population at different time step. The different parameters are:

1. b : number of offspring if reproduce
2. n : total of number of resource sites
3. A_0 : initial population
4. t : the time steps we want to simulate the model

When set the initial population to a small number relative to the resource site, it would be difficult to have 2 individuals at the same site for reproduce, and even if they reproduce a large number of offspring the population dies out very quick. below are some graphs showing the total number of population at different time steps with different parameters.

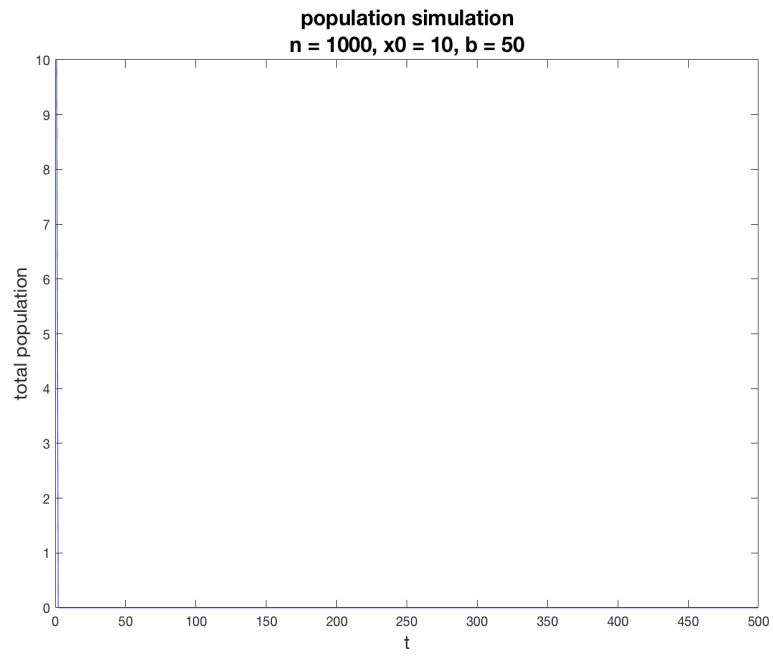


Figure 1: initial population of 10, $b = 50$ not reproducing

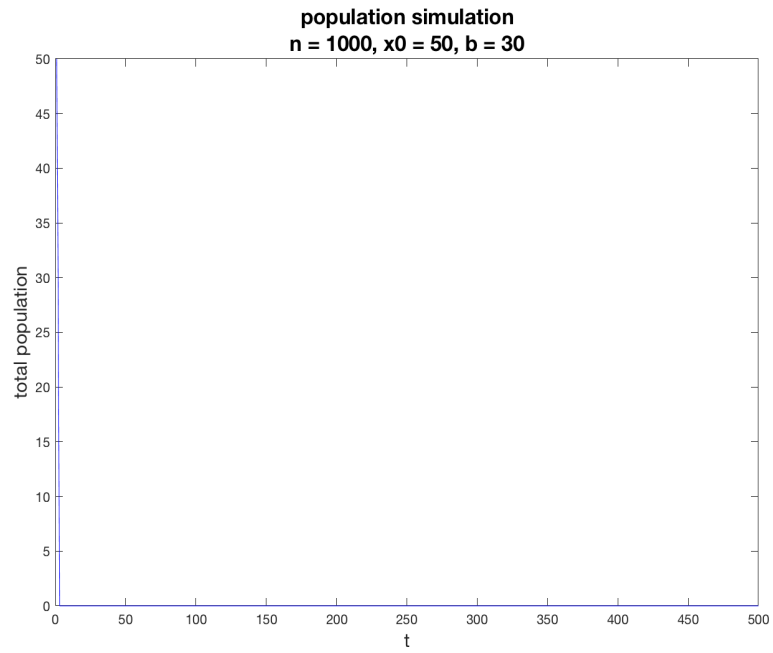


Figure 2: initial population of 50 and $b = 30$ not reproducing

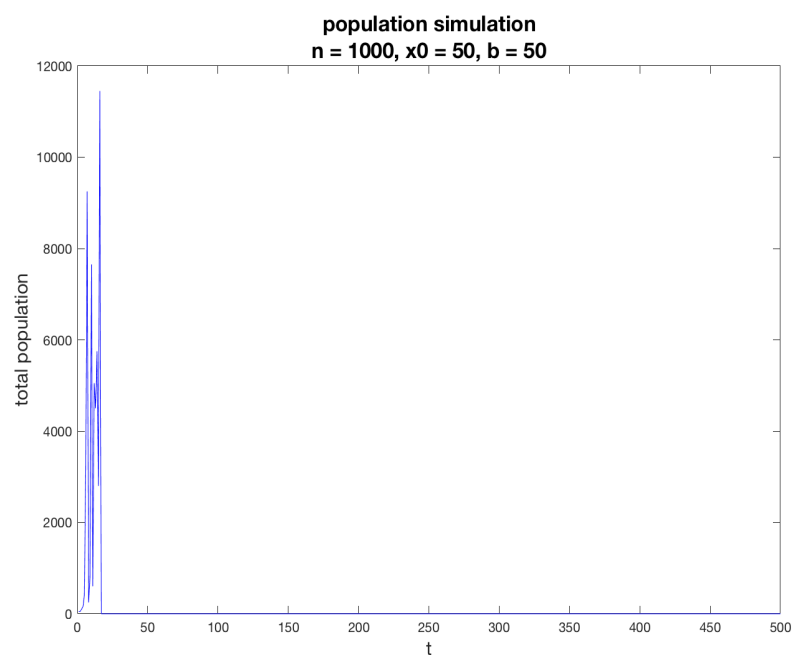


Figure 3: initial population of 50 and $b = 50$, in the beginning reproduce but quickly dies

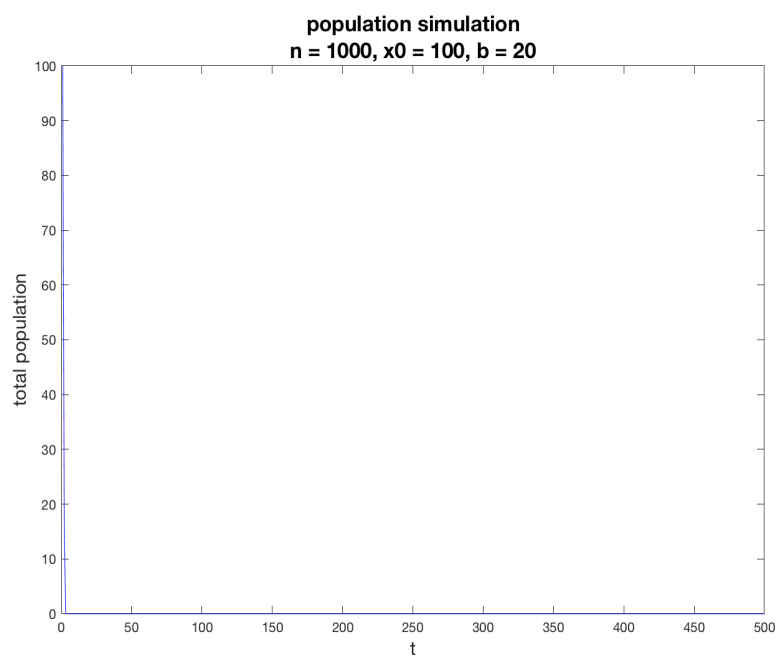


Figure 4: initial population of 100 and $b = 20$, it dies

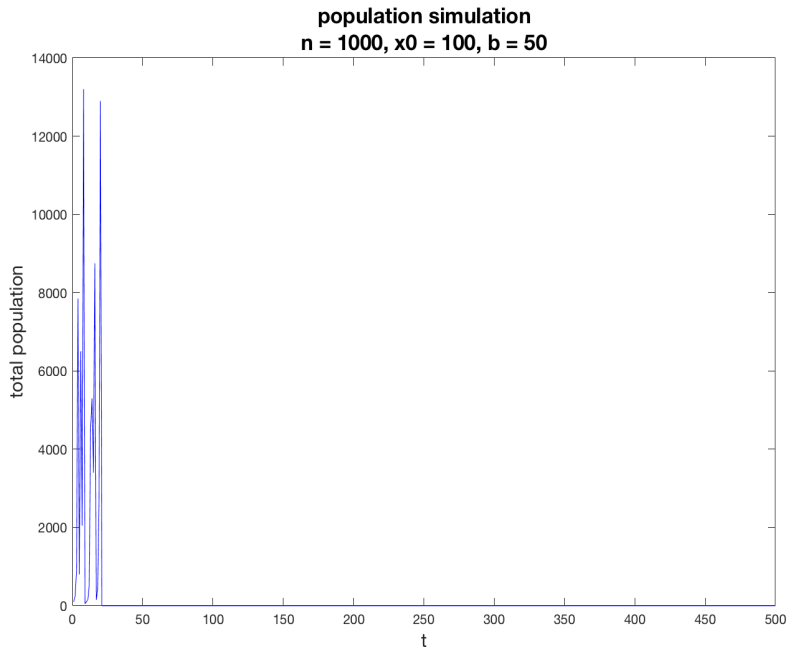


Figure 5: initial population of 100 and $b = 50$, it reproduce in the beginning but quickly dies

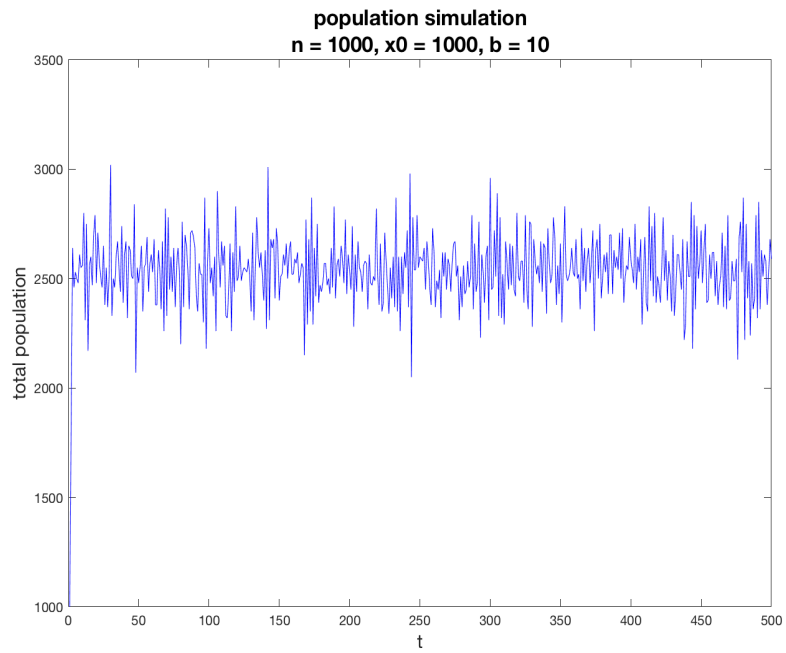


Figure 6: initial population of 1000 and $b = 10$, it show that the population oscillates around 2500

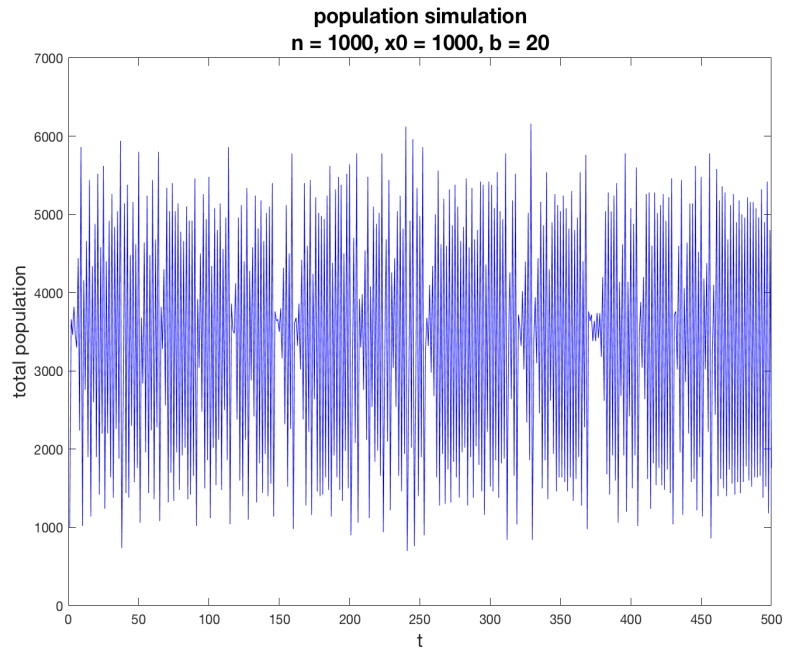


Figure 7: initial population of 1000 and $b = 20$, the populations starts to get chaotic

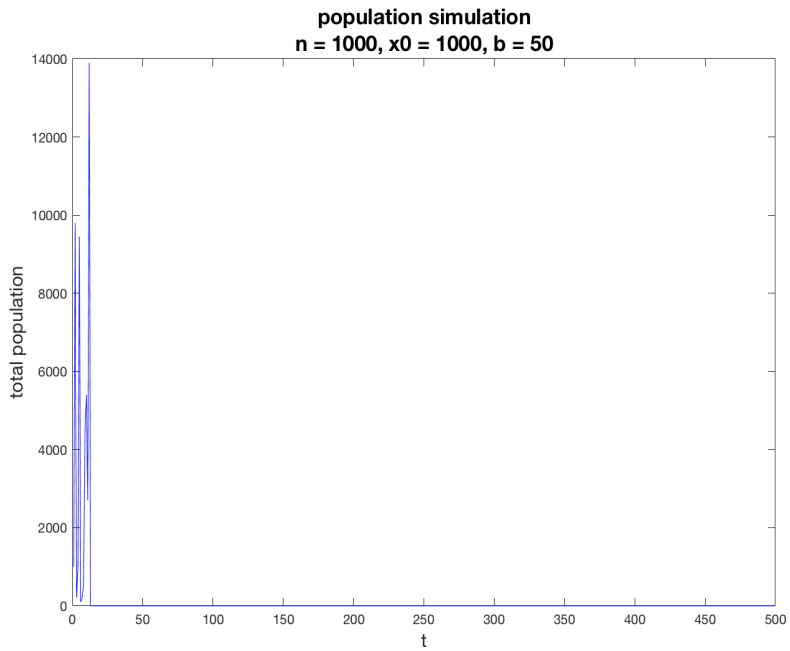


Figure 8: initial population of 1000 and $b = 500$, the populations dies very quick

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:

1.2 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 13, 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 14.

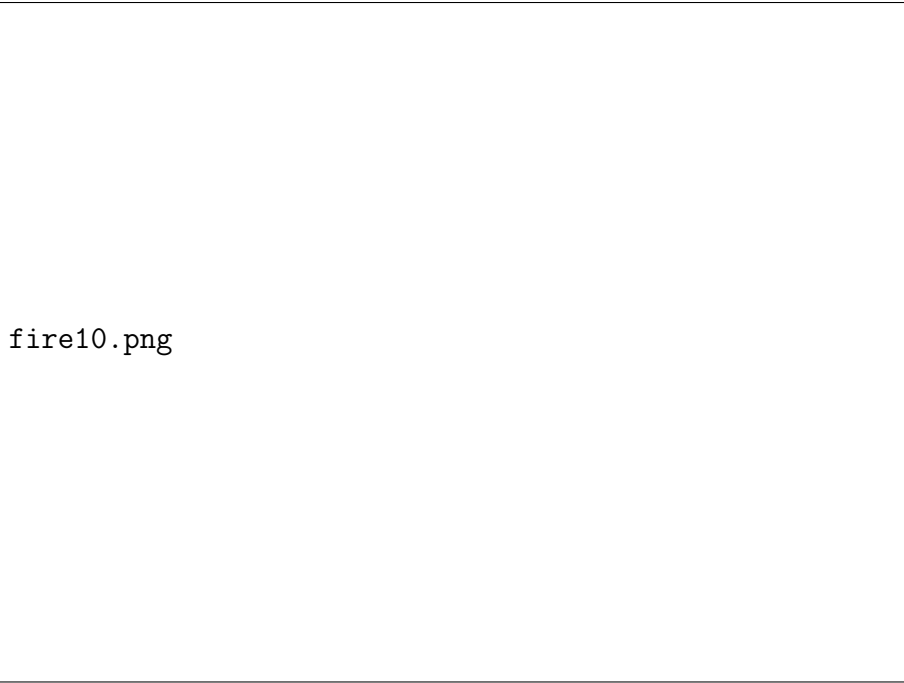


Figure 9: 40x40 cell grid at $t = 10$

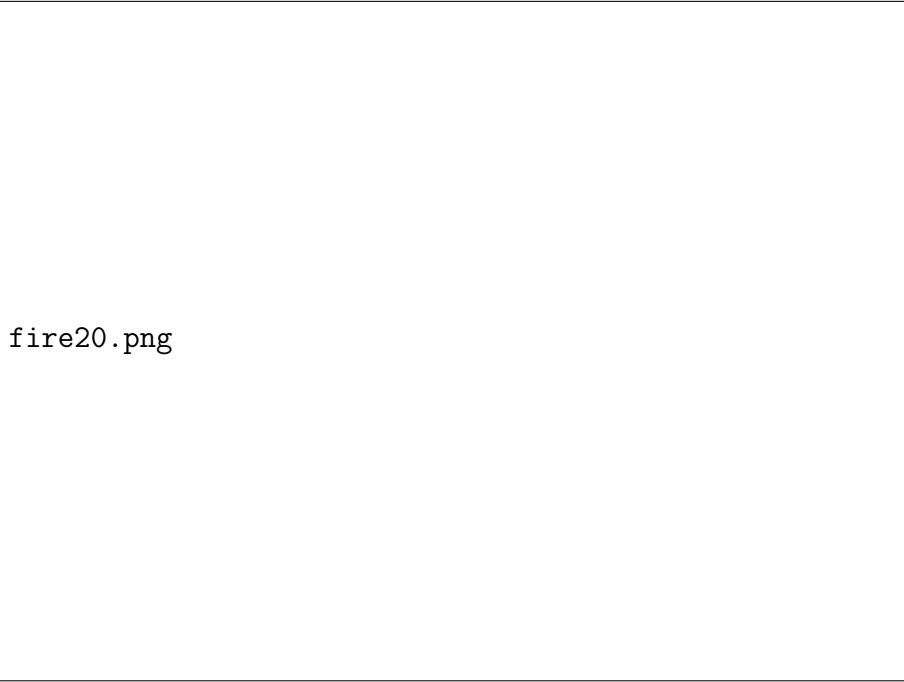


Figure 10: 40x40 cell grid at $t = 20$

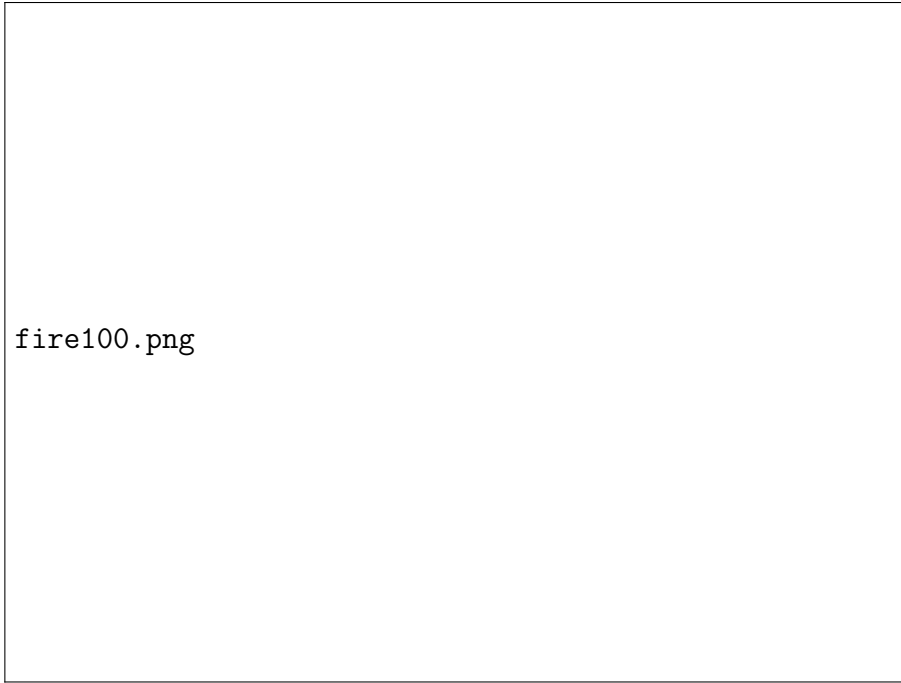


Figure 11: 40x40 cell grid at $t = 100$

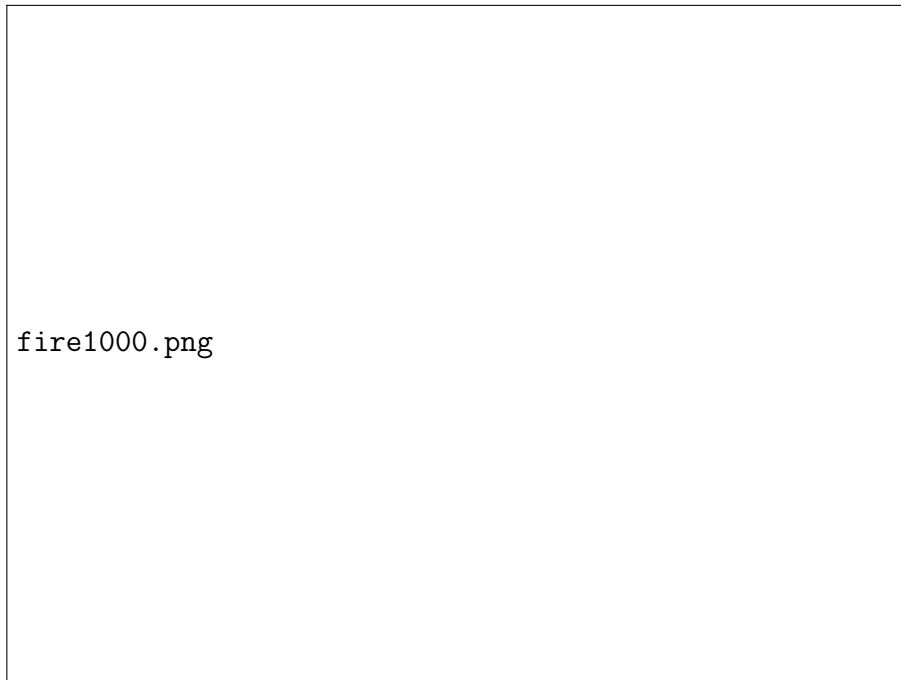


Figure 12: 40x40 cell grid at $t = 1000$



Figure 13: average number of firing cells over time with 100 simulation of different initial conditions

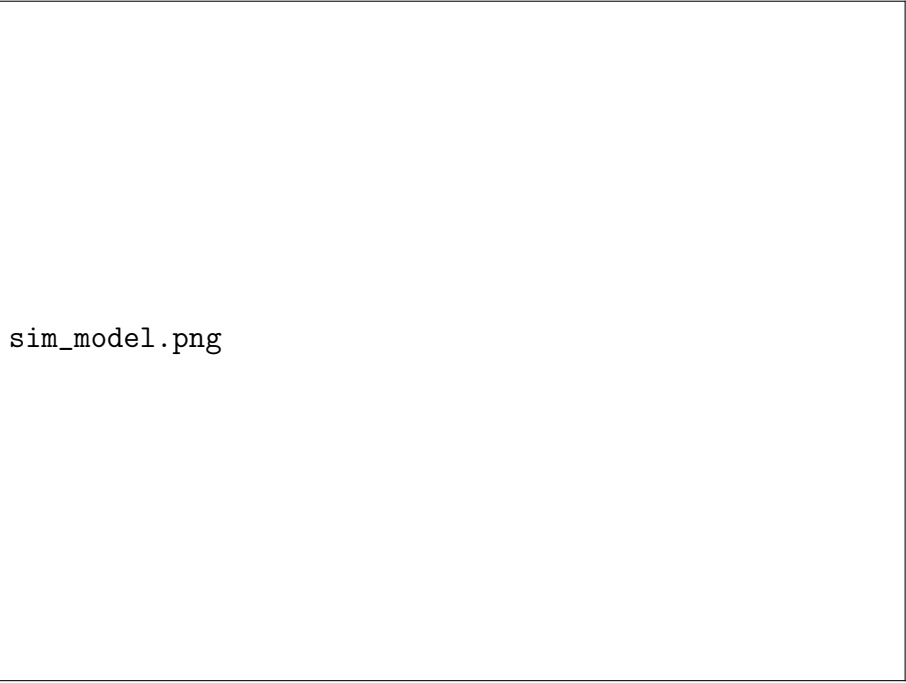


Figure 14: average number of firing cells over time with 100 simulation and exponential model fitting

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>

1.3 example of shapes

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

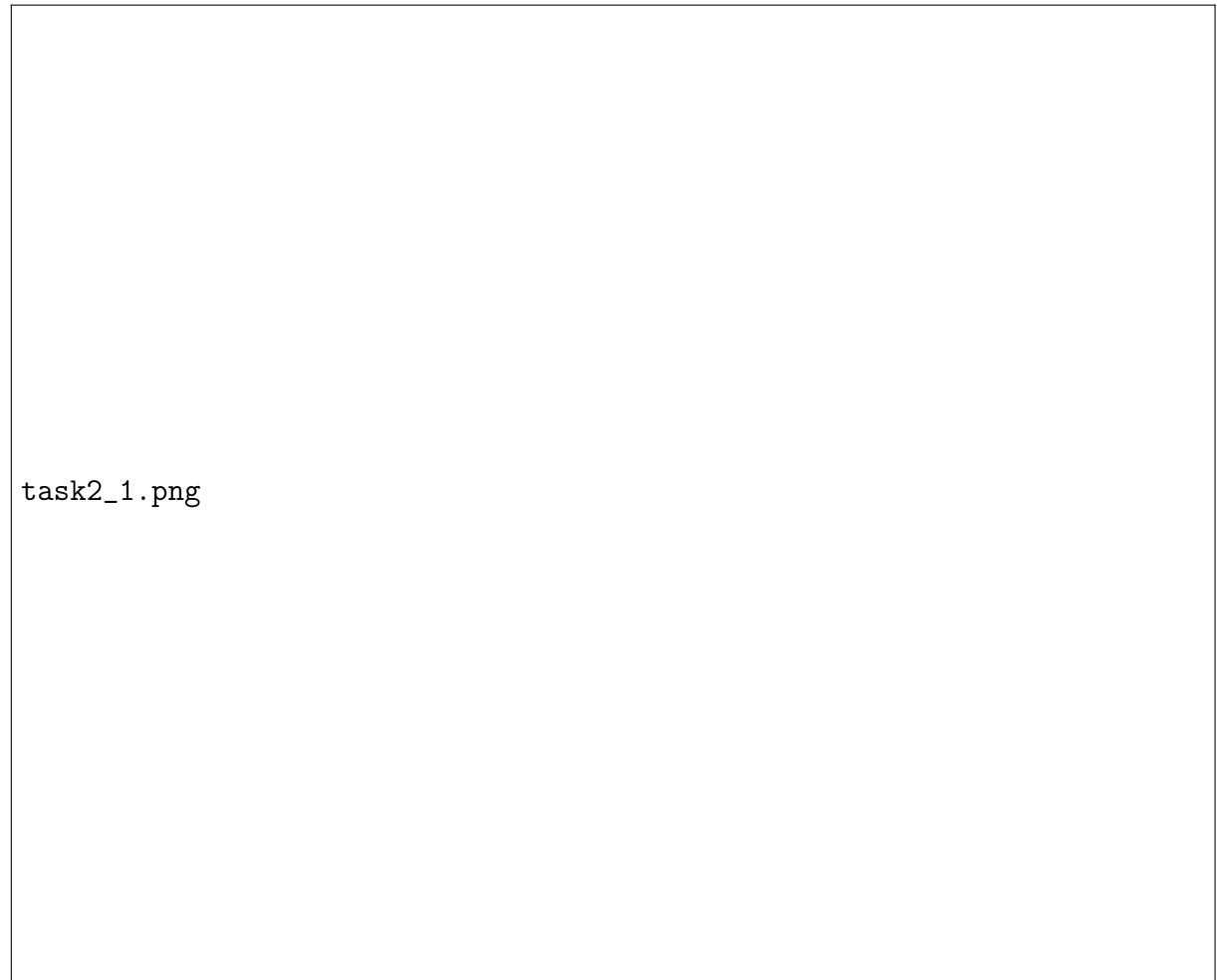


Figure 15: shapes that move forward at a rate of one cell per time step preserving the same shape

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

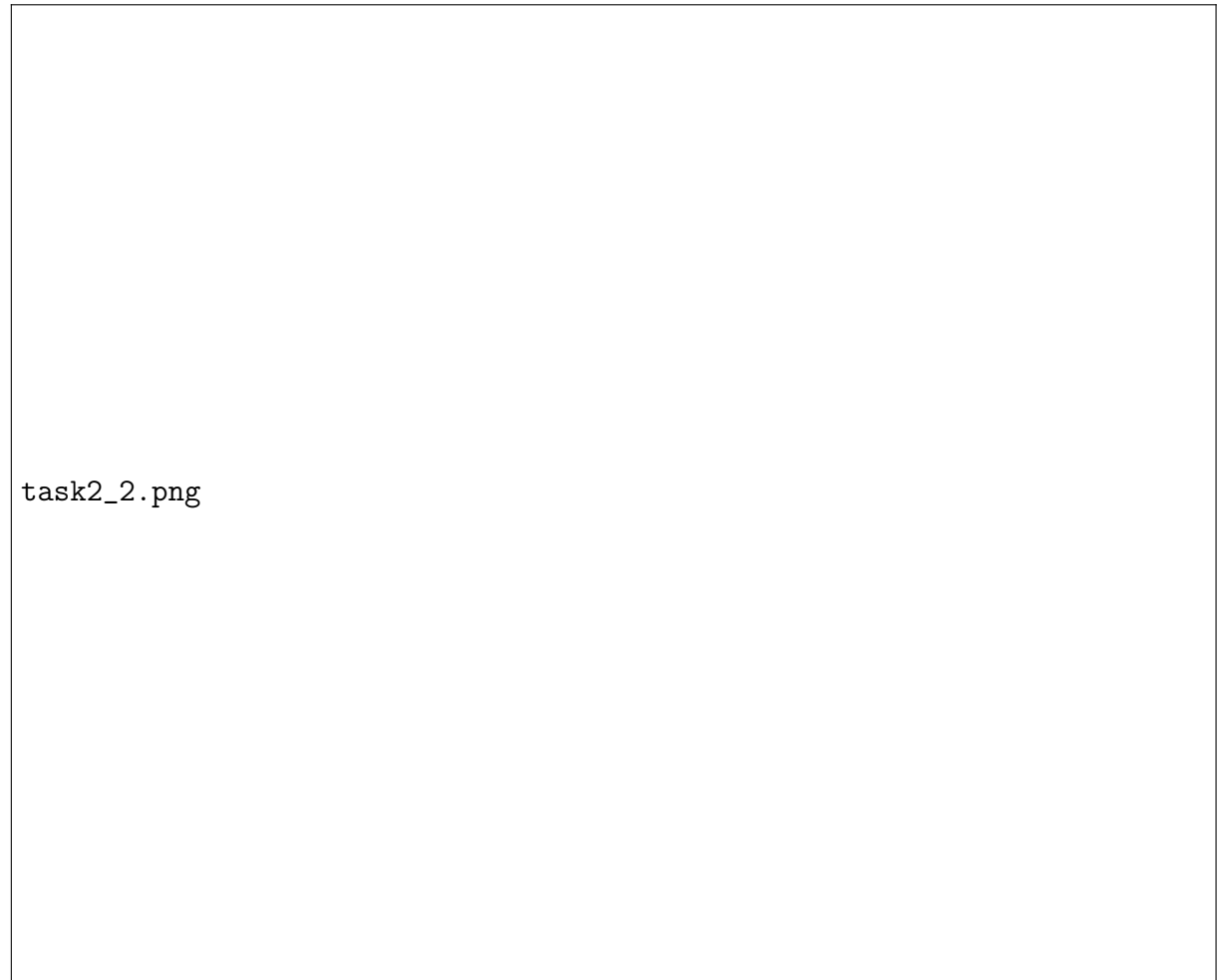


Figure 16: shapes that move forward at a rate of one cell per time step, launching other shapes behind them

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

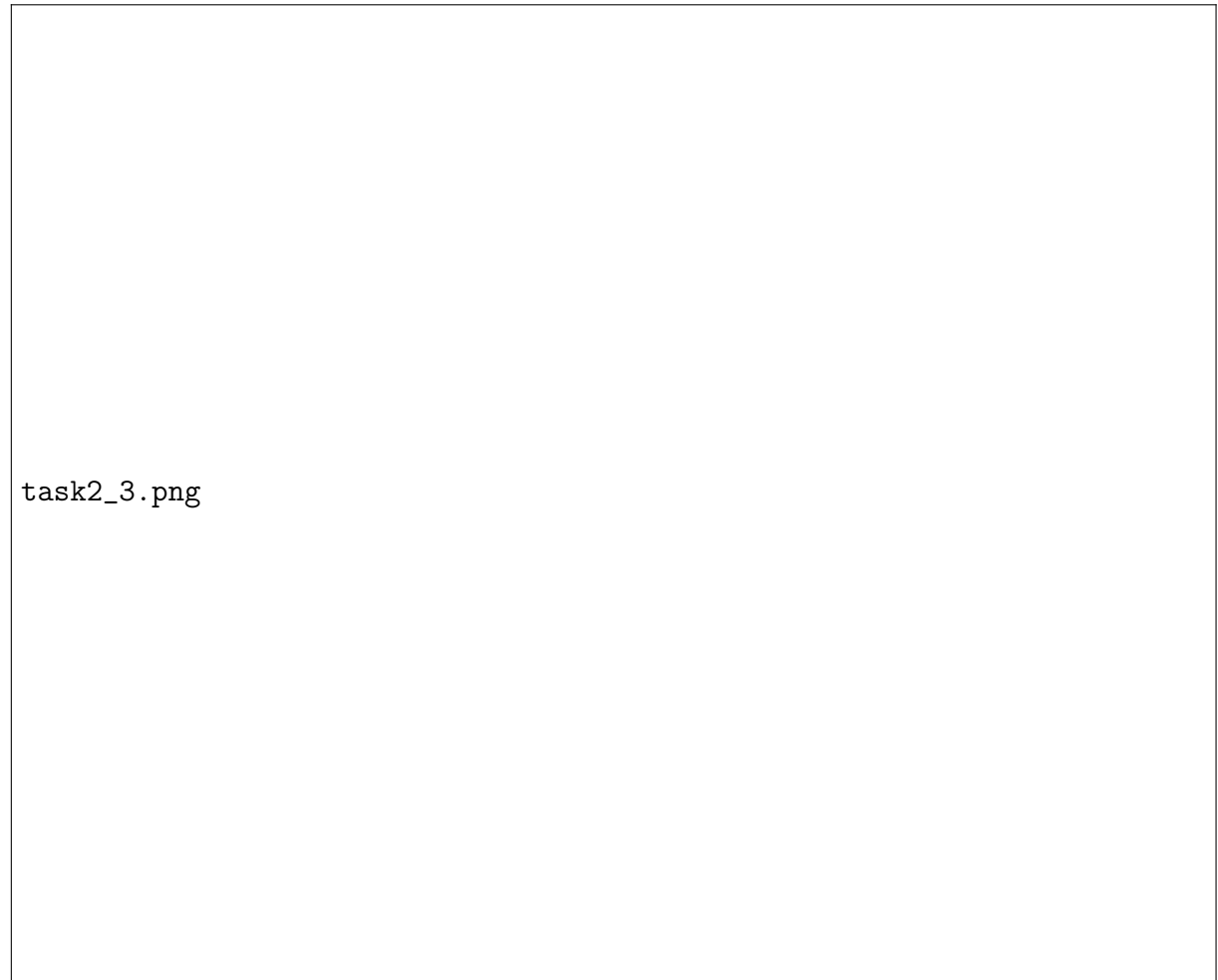


Figure 17: shapes that move less than one cell per time step, returning to same shape after some period

1.3.4 stay stationary but oscillate periodically



Figure 18: shapes that stay stationary but oscillate periodically

1.4 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.

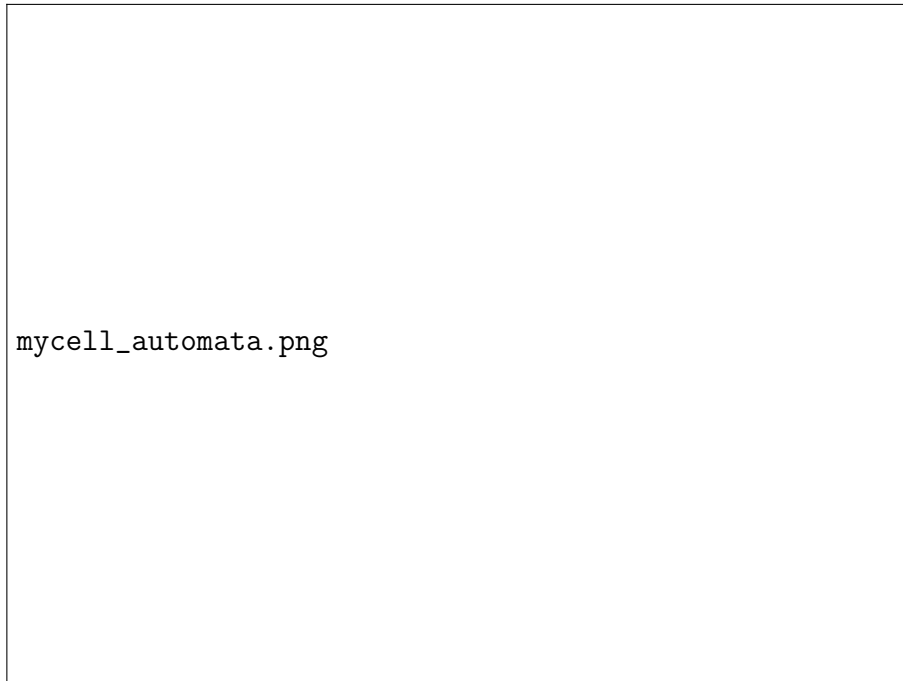


Figure 19: my cell automata at $t = 10$

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.

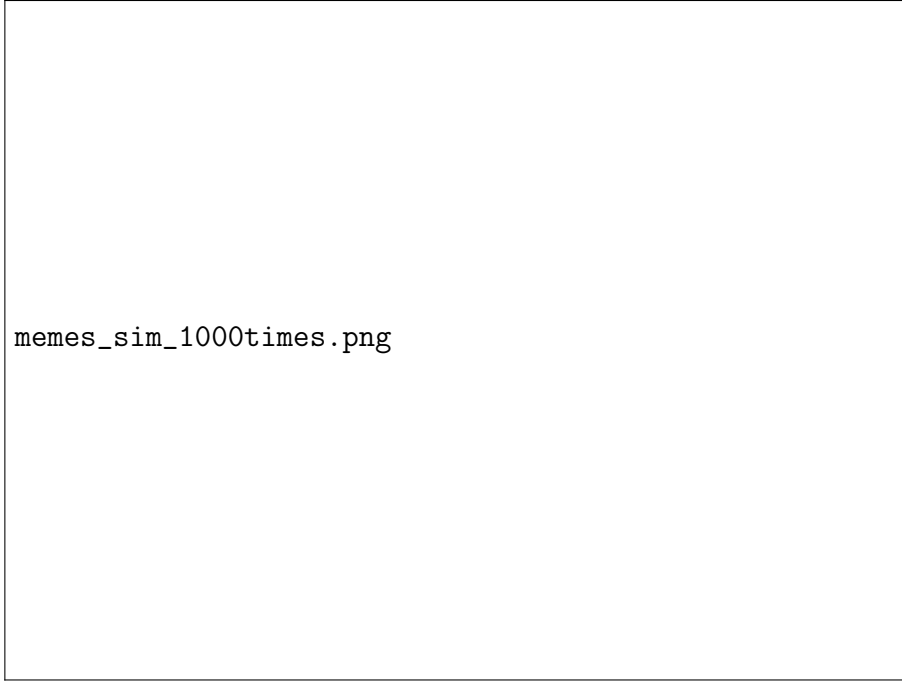


Figure 20: simulation of spread of memes showing number of bored, sharing, resting person over time

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

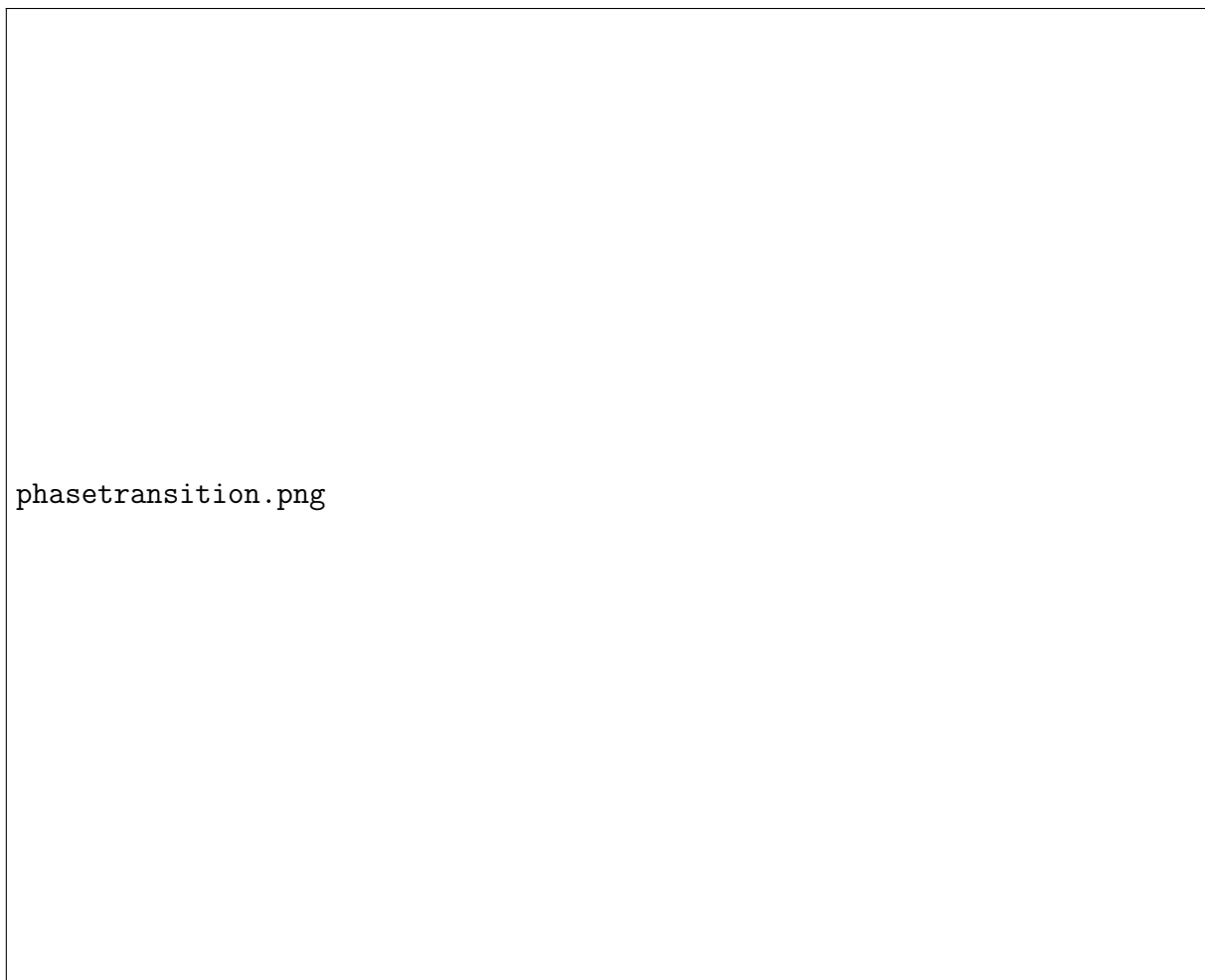


Figure 22: phase transition of total sharing person with simulation with $t = 1000$ and different $B(0)$ condition

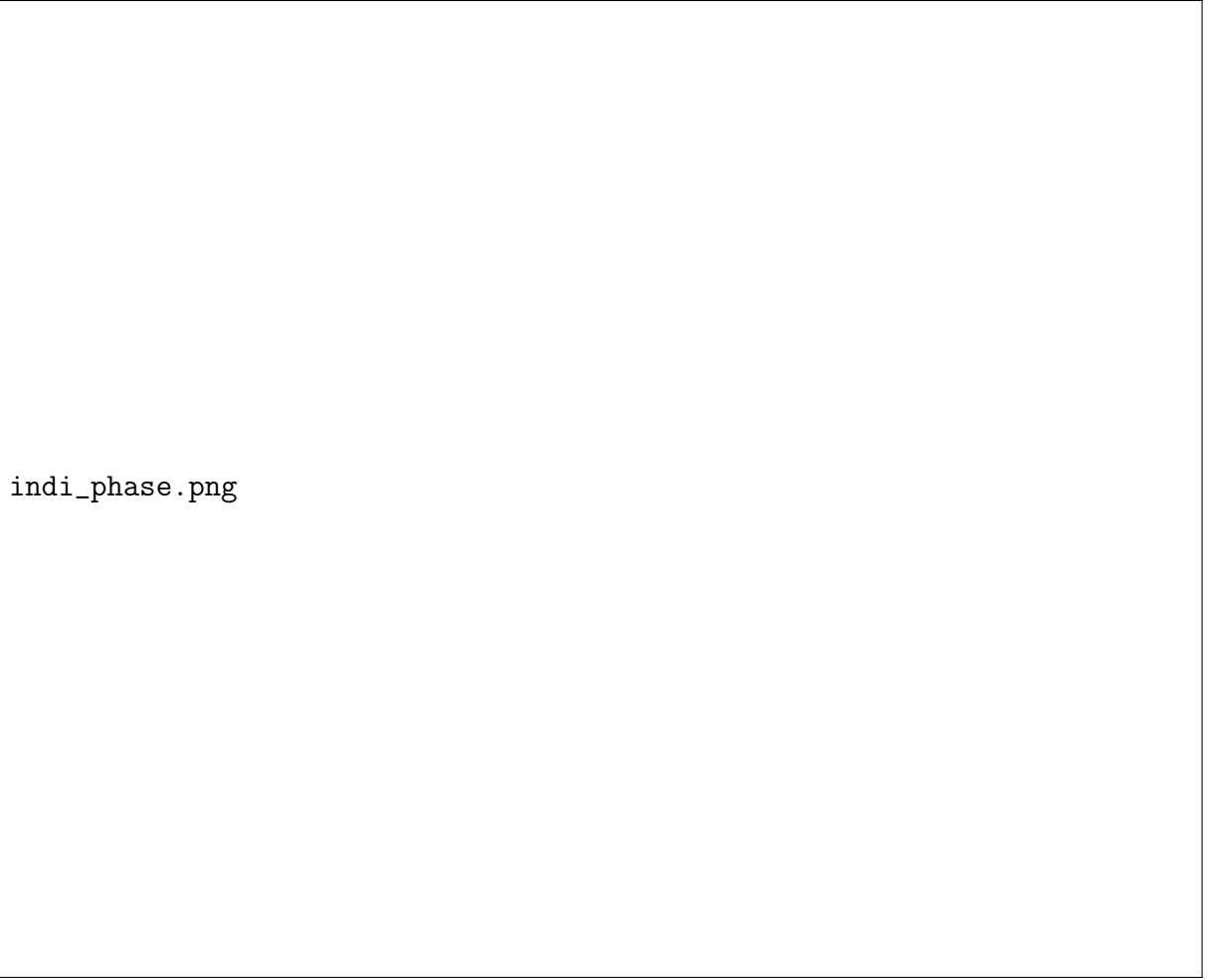


Figure 23: plot of sharing person in individual runs at $t = 1000$ for 100 simulations of $B_0 = 1:1:999$

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 24: 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 25: 100 simulations of memes with different rules for bored person over $t = 2000$



Figure 26: 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 27: phase transition of total sharing person with simulation with $t = 1000$ and different q condition



Figure 28: 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.

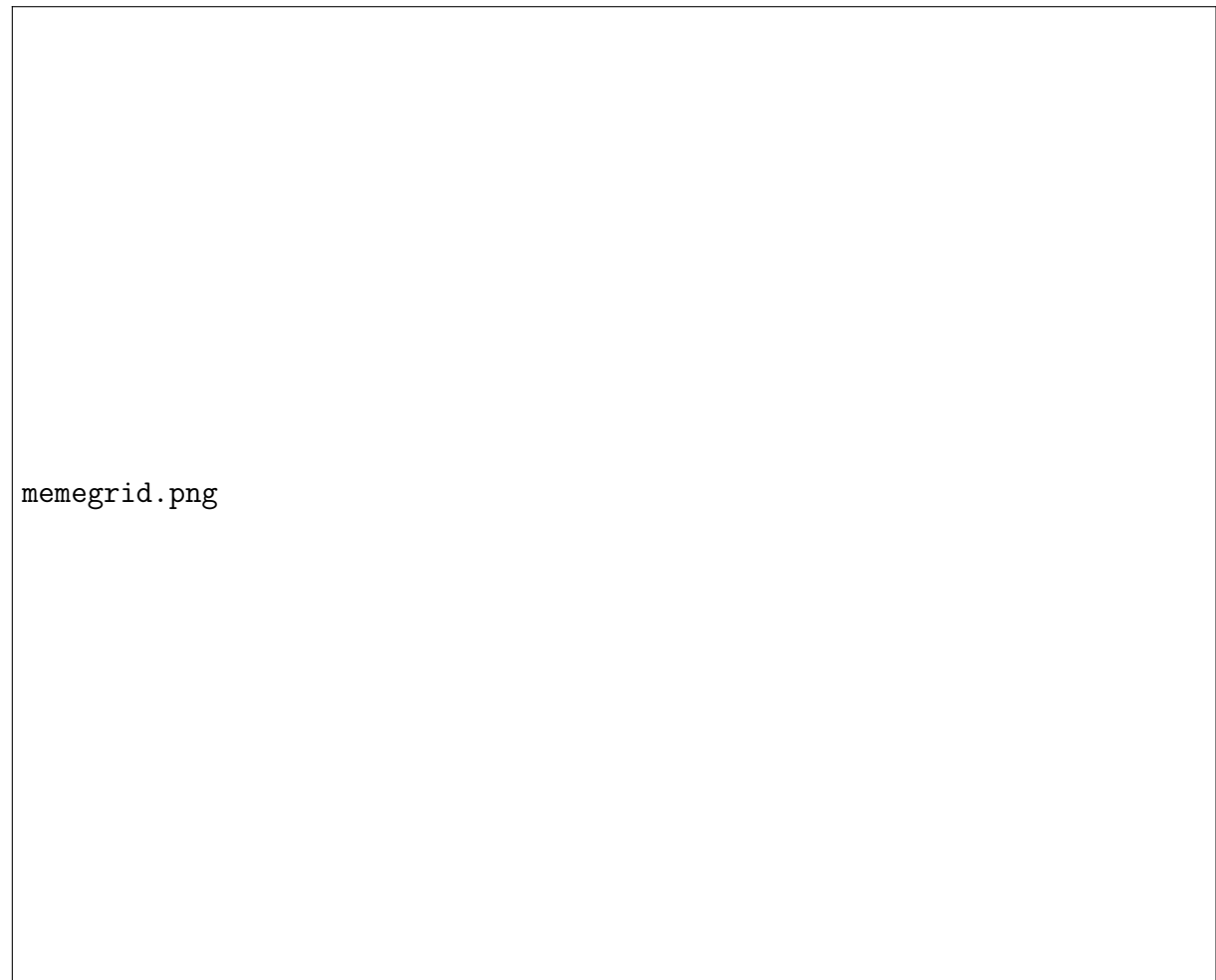


Figure 29: initial condition for simulation



Figure 30: 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
- transition function
- initial state
- simulate 100 times of firing brain
- cell that move forward at one cell per time preserving the same shape
- cell that move forwad at one cell per time, launching other shapes behind them
- move forward at a rate of less than one cell per time step
- oscillate shape
- my cellular automata
- my cellular automata transit

3.2 Spread of memes

- simulation of spread of memes
- run single spread of memes
- mean field model
- phase transition
- probability for at least 25% are sharing
- simulation of spread of memes with new rules
- run single spread of memes
- mean field model
- phase transition for new rules
- lattice simulation for memes
- transition function for memes