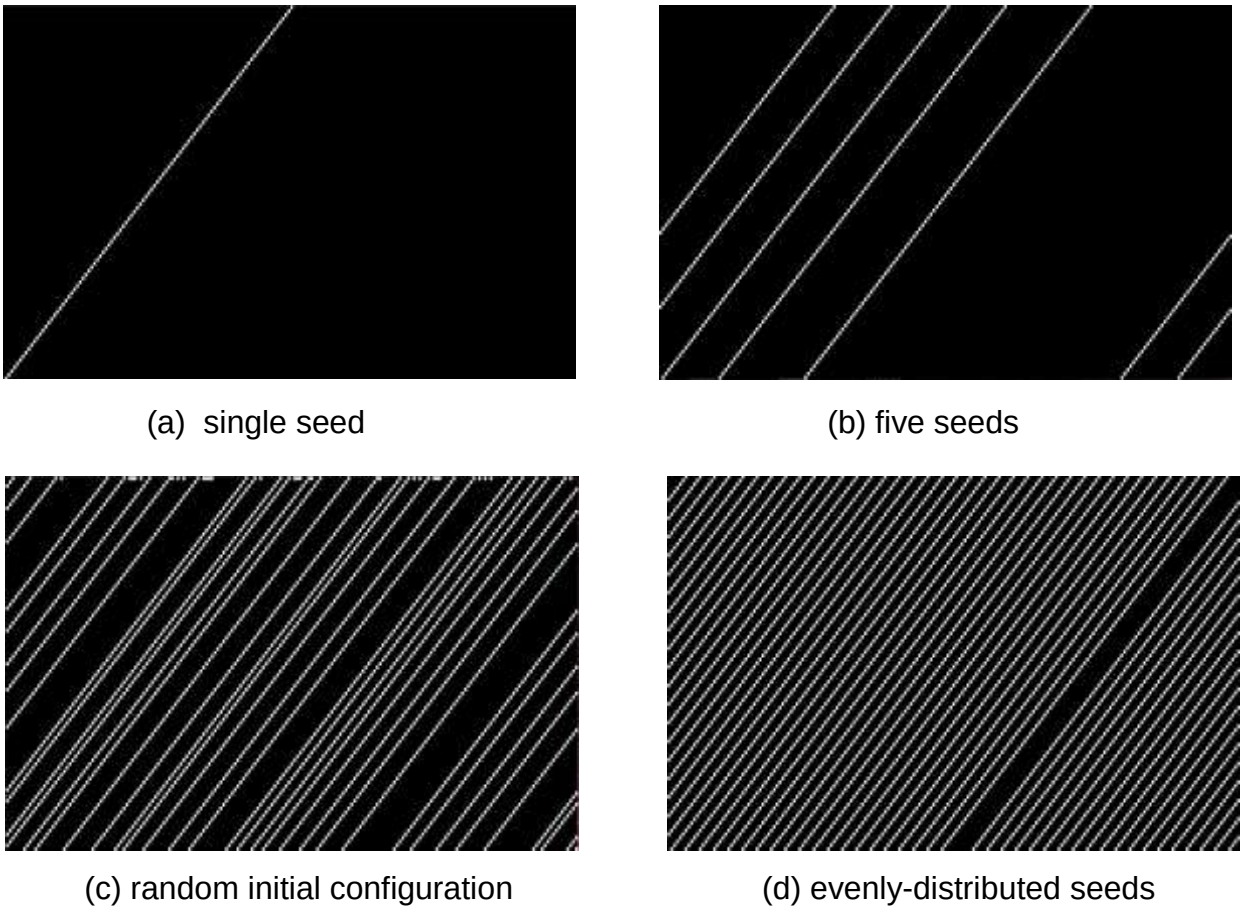
Task A - Using 1D Cellular Automata to investigate complexity

1) Classify:

We need to generate the simulations for twelve rules, which will then be used to classify each rule into the four classes suggested by Wolfram [A1].

- **Class 1**: evolves to a *homogeneous state.*
- **Class 2**: evolves to simple *separated periodic structures.*
- **Class 3**: evolves to *chaotic aperiodic patterns*.
- **Class 4**: evolves to *complex patterns of localized structures*.

Rule 2



(a) single seed (b) five seeds

(c) random initial configuration (d) evenly-distributed seeds

Figure 1: Simulations for rule 2

All configurations show similar behaviour with successive generations. Patterns are periodic. ***Class 2***.

Rule 8

(a) single seed

(b) 4-seed cluster

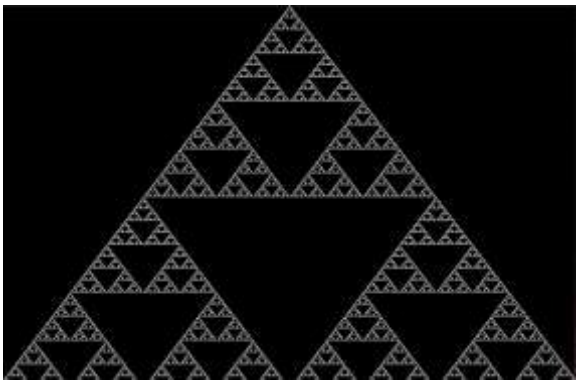
(c) random initial configuration

(d) evenly-distributed seeds

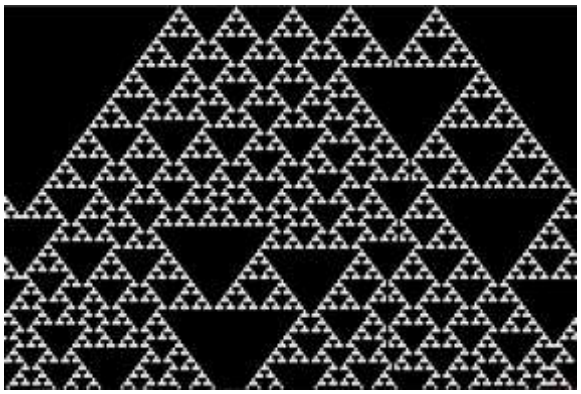
Figure 2: Simulations for rule 8

All configurations result in homogeneous final state. ***Class 1***.

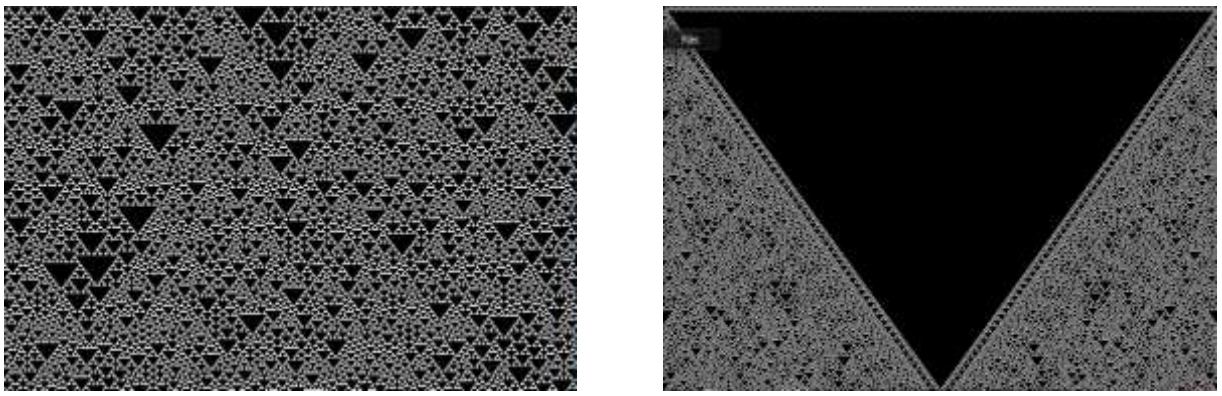
Rule 22:



(a) single seed



(b) five seeds

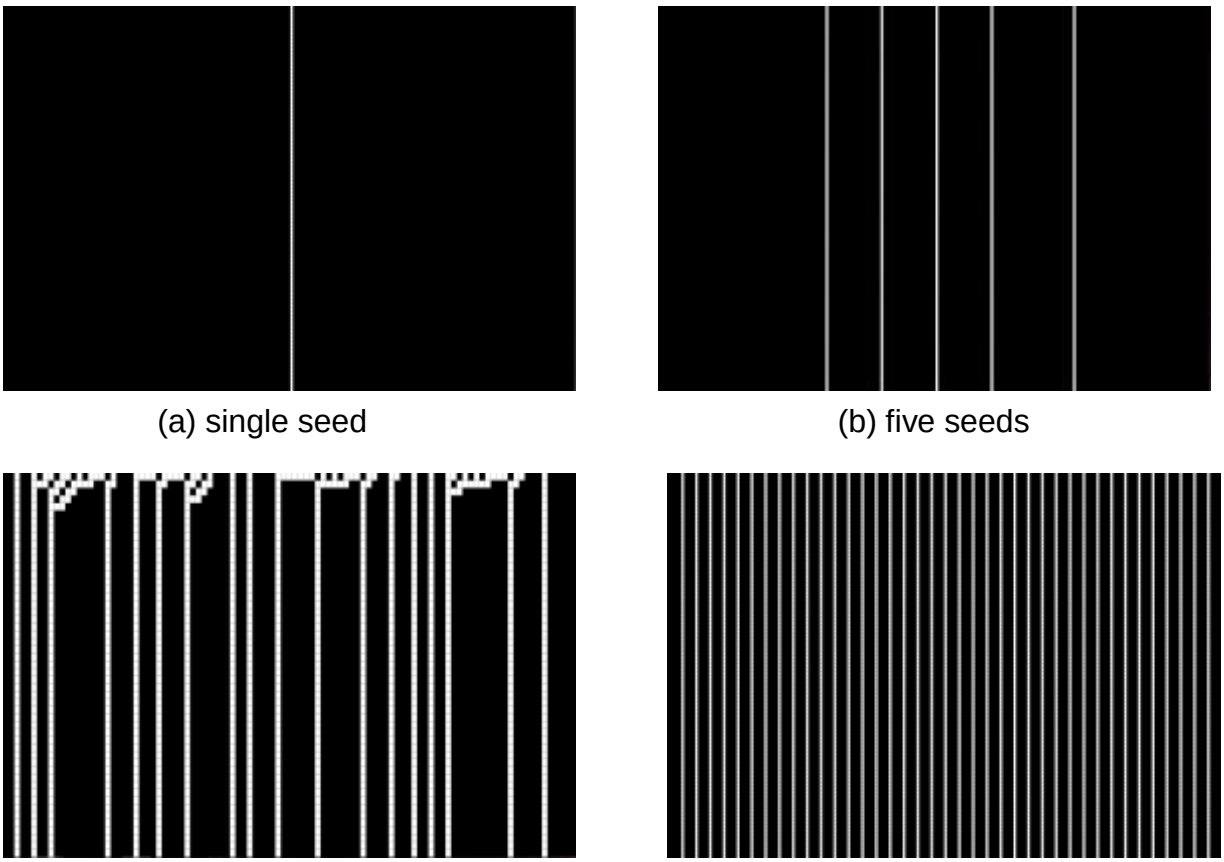


(c) random initial configuration (d) evenly-distributed seeds

Figure 3: Simulations for Rule 22

Fig. 3a shows a fractal patterns (i.e. nested patterns that emerge), while the others behave randomly, especially in random initial configuration, no predictable pattern is found. ***Class 3***.

Rule 44



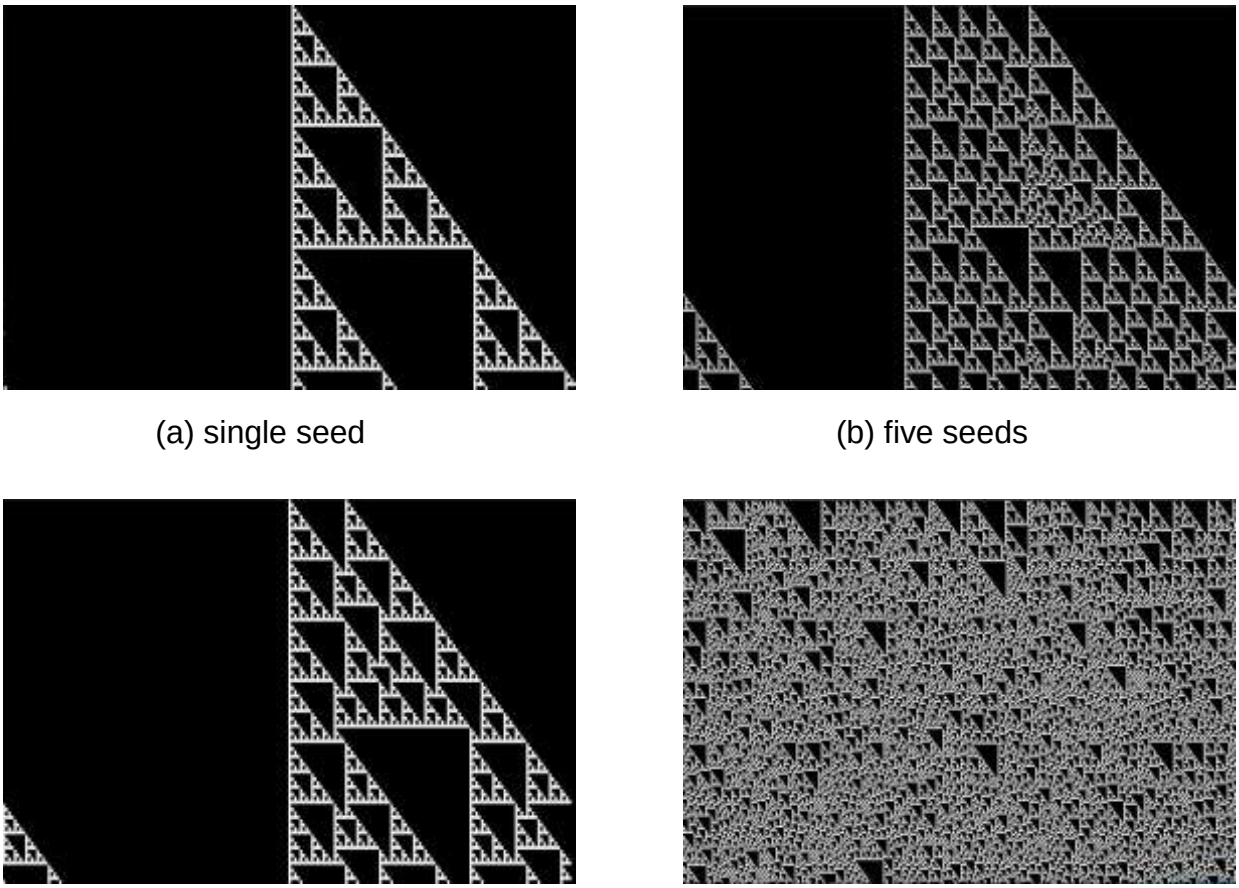
(a) single seed (b) five seeds

(c) random initial configuration (d) evenly-distributed seeds

Figure 4: Simulations for Rule 44.

Rule produces periodic structures even when given random initial configuration. ***Class 2***.

Rule 60:



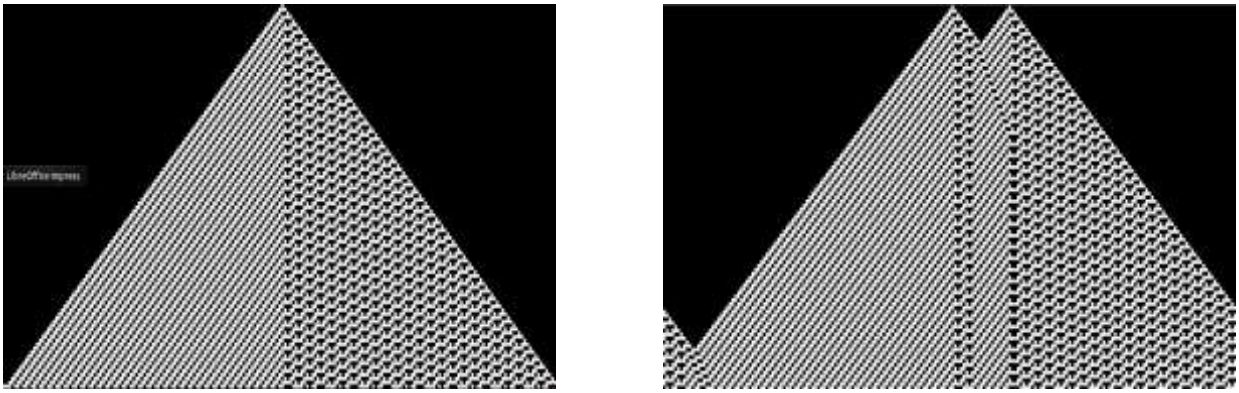
(a) single seed (b) five seeds

(c) two seeds (d) random initial configuration

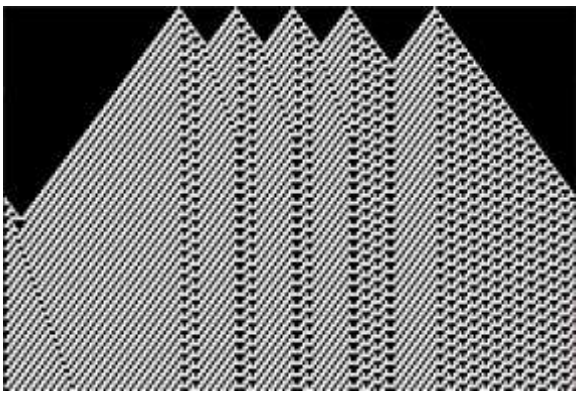
Figure 5: Simulations for Rule 60

Fig. 5a shows fractal patterns, but random initial configuration shows clearly random behaviour. ***Class 3***.

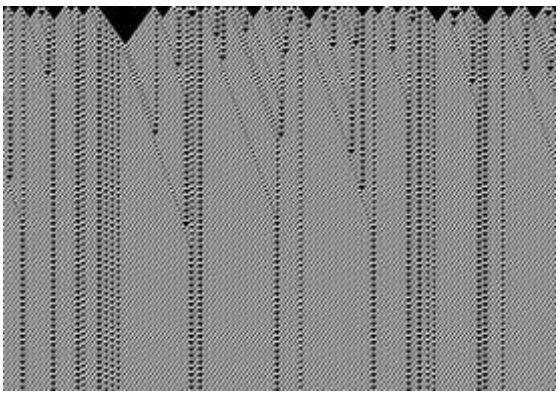
Rule 62



(a) single seed (b) two seeds



(c) five seeds

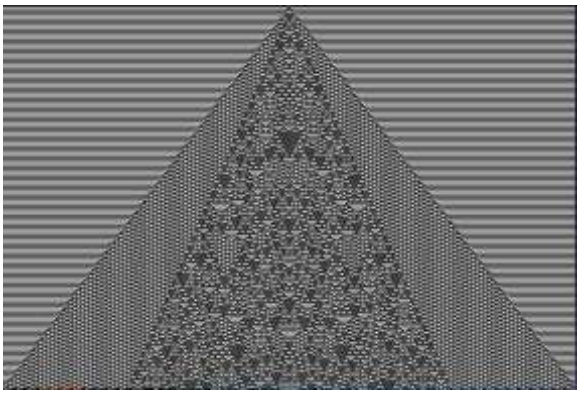


(d) random initial configuration

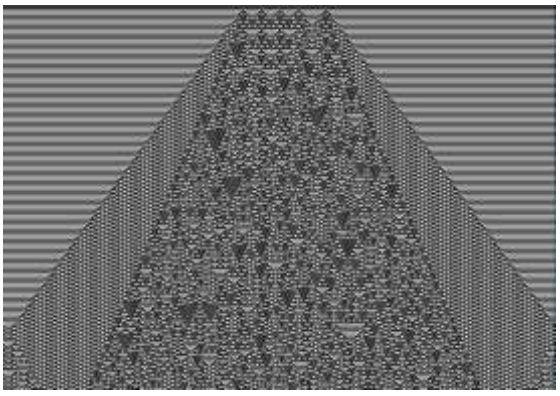
Figure 6: Simulations for Rule 62.

Fig. 6d shows separated periodic structures given random initial configuration. ***Class 2***.

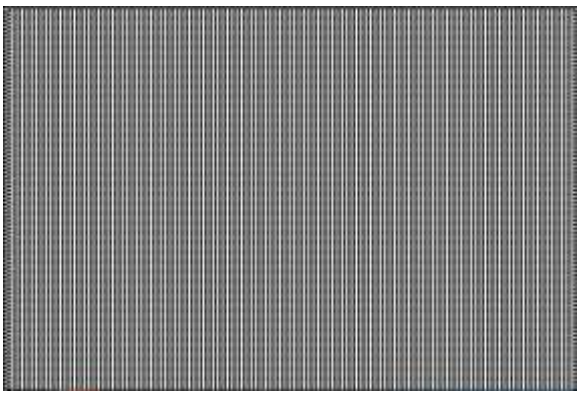
Rule 73:



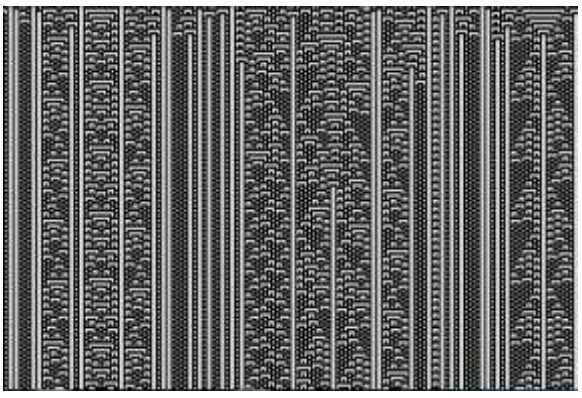
(a) single seed



(b) five seeds



(c) evenly-distributed seeds

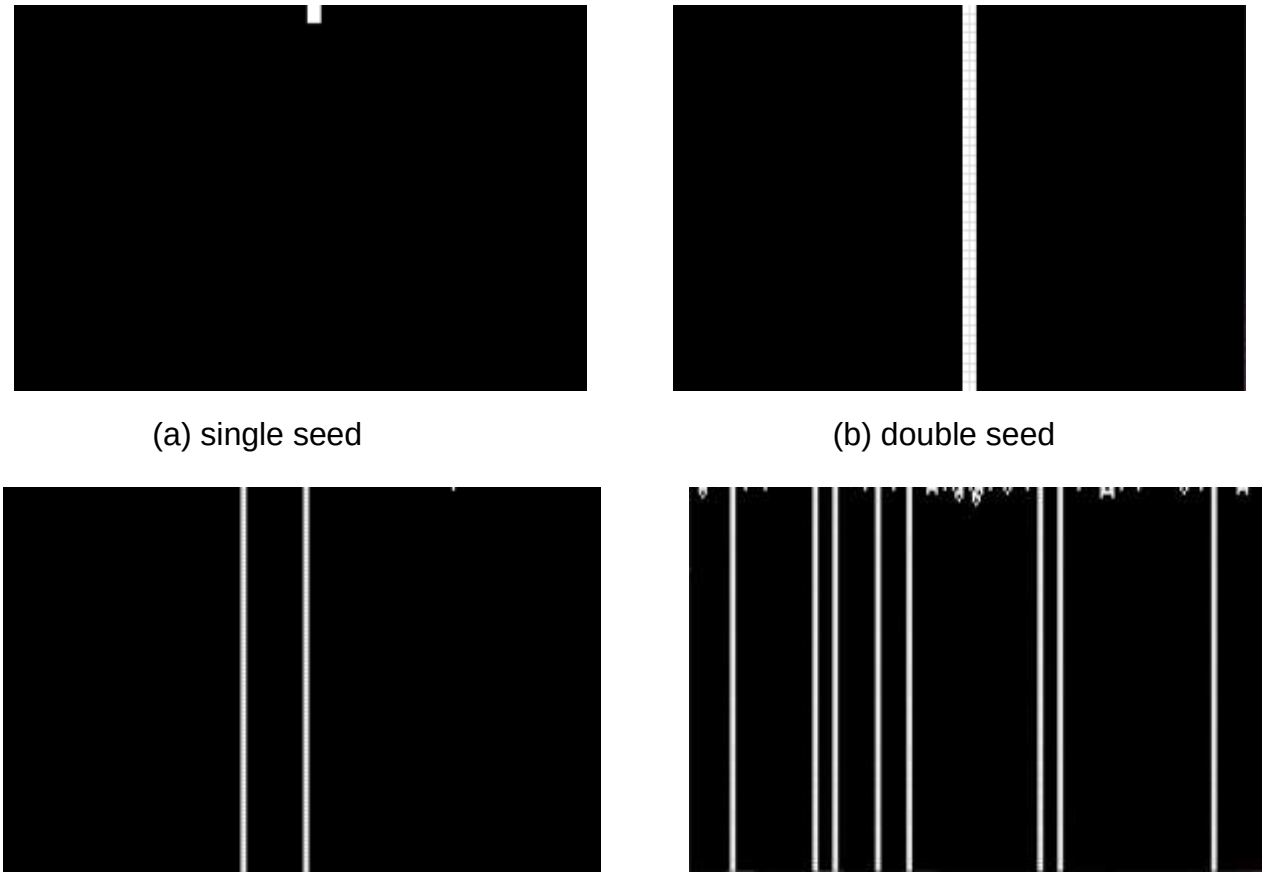


(d) random initial configuration

Figure 7: Simulations for Rule 73.

Random initial configuration produces separated periodic patterns. ***Class 2***.

Rule 104:



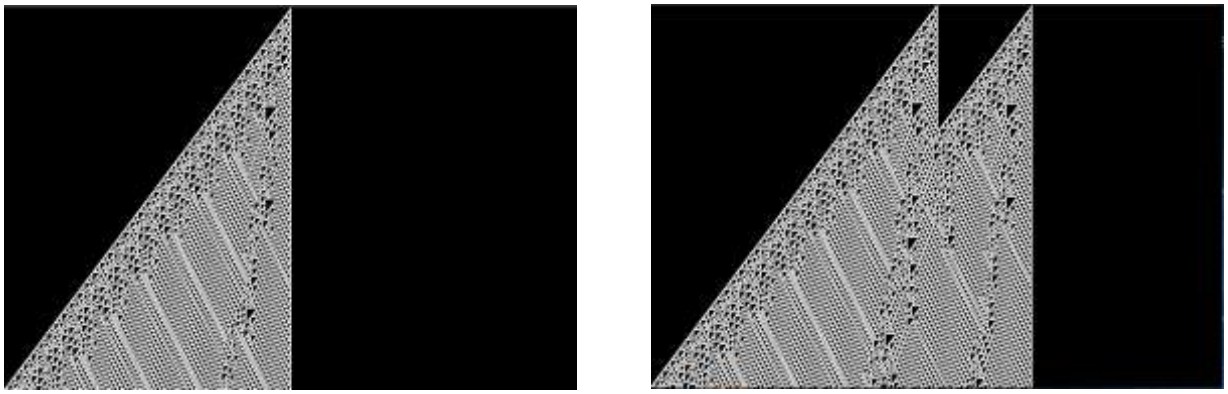
(a) single seed (b) double seed

(c) two double-seeds (d) random initial configuration

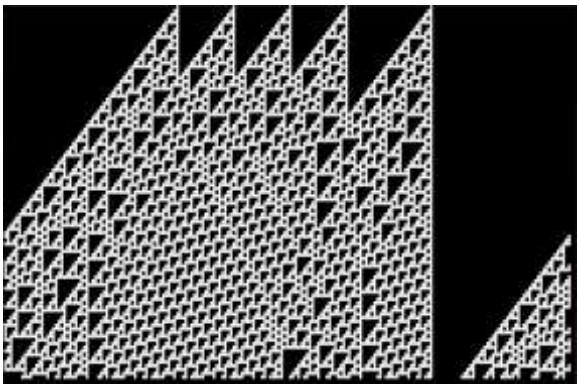
Figure 8: Simulations for Rule 104.

Initially, Fig. 8a gives an idea of homogeneous structure, but double-seed, which contain two single seeds sitting next to each other, and random configuration produce separated periodic patterns. ***Class 2***.

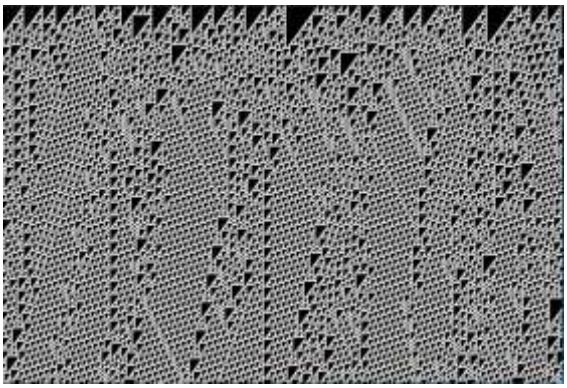
Rule 110



(a) single seed (b) two seeds



(c) five seeds



(d) random initial configuration

Figure 9: Simulations for Rule 110.

All simulations produce localized structures that interact in a complex way. ***Class 4***.

Rule 136:

(a) single seed

(b) three seeds

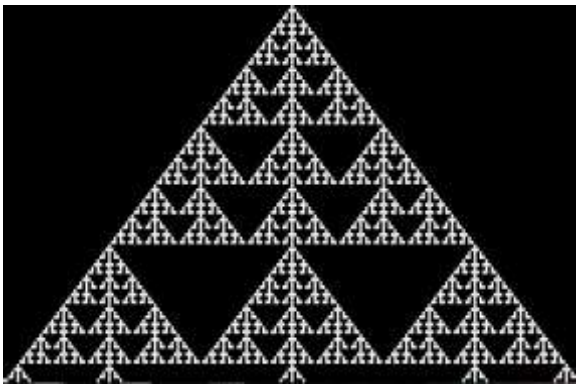
(c) random initial configuration

(d) evenly-distributed seeds

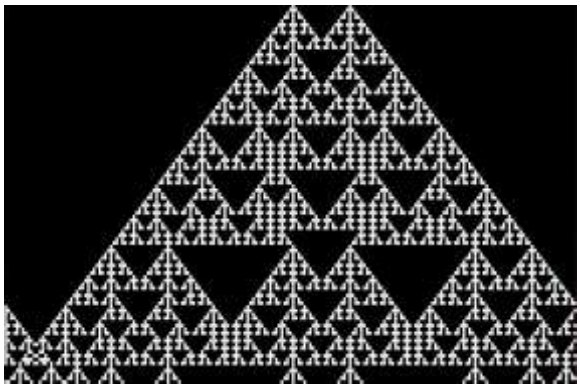
Figure 10: Simulations for Rule 136

All configurations end up with homogeneous patterns. It becomes uniform state after different time steps, 1 time step for single seed, 4 for random initial configuration. ***Class 1***.

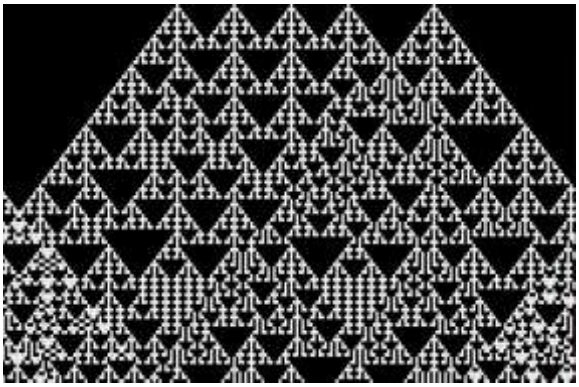
Rule 150:



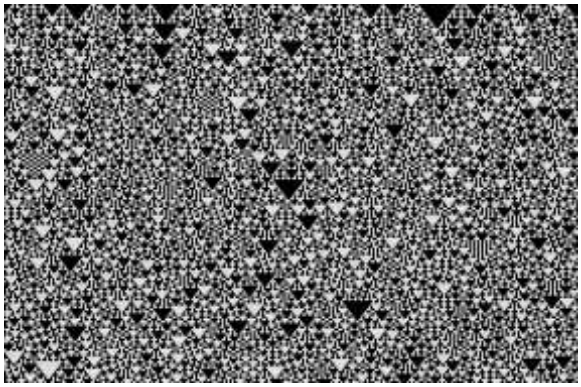
(a) single seed



(b) two seeds



(c) five seeds



(d) random initial configuration

Figure 11: Simulations for Rule 150

Fig. 11a shows fractal structure, but random configuration produces random structures. ***Class 3***.

Rule 160:

(a) single seed

(b) three seeds

(c) 4-seed cluster (d) random initial configuration

Figure 12: Simulations for Rule 160.

All simulations end up with homogeneous final state (i.e. all cells become black) after different step of times, 1 time step for single seed and 4 for random configuration. ***Class 1***.

2) Calculate Langton's parameter

Langton [A2] provided the formula to calculate Langton's parameter. Let n be the number of transitions to special *quiescent state* in a rule, K is the number of possible states each cell can have, N is the number of neighbours.

$$\boldsymbol{\lambda = \frac{K^N - n}{K^N}}$$

In this experiment:
- Each cell has 2 states, K = 2.
- Each cell has 3 neighbours (including itself), N = 3.
- Quiescent state S_q = 0.

a) Class 1:

$8_{10} = 00001000_2$

$\boldsymbol{\lambda = \frac{2^3 - 7}{2^3} = \frac{1}{8} = 0.125}$

b) Class 2:

$44_{10} = 00101100_2$

$\boldsymbol{\lambda = \frac{2^3 - 5}{2^3} = \frac{3}{8} = 0.375}$

c) Class 3:

$150_{10} = 10010110_2$

$\boldsymbol{\lambda = \frac{2^3 - 4}{2^3} = \frac{4}{8} = 0.5}$

d) Class 4:

$110_{10} = 01101110_2$

$\boldsymbol{\lambda = \frac{2^3-3}{2^3} = \frac{5}{8} = 0.625}$

Comments:

The value of Langton's parameter falls into the range from 0 to 1 as "the average behaviour of the systems goes from freezing to periodic patterns to chaos" [A3]. Langton [A2] introduced the idea of "Edge of chaos", in which almost rules with $\lambda \approx 0.5$ turns to lie on the Edge. The edge of chaos represents the phase transition in Cellular Automata, at which the system behaviour turns from chaos into complex, while rules which behave in homogeneous and separated periodic manner are expected to have value lower than this. Therefore, the values of Langton's parameter we have just measured for each class above seem to be quite reasonable, as the values are in ascending order, starting from $\frac{1}{8}$ to $\frac{3}{8}$ to $\frac{1}{2}$ to $\frac{5}{8}$ as the behaviour of classes changed from homogeneous to periodic to chaotic and localized structures. The values of Langton' parameter for classes 3 and 4 are around 0.5, which is around the edge of chaos.

Task B - Conway's Game of Life

We observed many different structures as following:

- **Static:** Beehive, Block, Boat, Loaf, Ship, Long Boat, Long Barge, Pond, Tub [B1].
- **Oscillatory:** Blinker (period=2), Toad (period=2) [B3].
- **Moving:** Glider, Glider Gun [B2].

1) Moore's neighbourhood

Starting condition	Structure(s) seen
1	Blinker
2	Beehive
3	Blinker
4	Beehive, Blinker, Loaf, Block, Toad
5	Blinker, Block, Beehive, Loaf, Glider, Ship, Boat, Tub, Pond, Long Boat, Long Barge
6	Glider, Glider Gun, Block

2) von Neuman's neighbourhood

Starting condition	Structure(s) seen
1	----
2	----
3	Fig.1
4	----
5	----
6	Block, and Fig.2

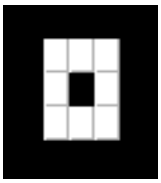


Figure 1: Condition 3

Figure 2: Condition 6

a) How to edit code to use von Neumann's neighbourhood:
We changed the condition for If-statement from **(i!=1)||(j!=1)** into **((i==1)||(j==1)) && !((i==1)&&(j==1)) in** the nested for-loops in **GOL2DCell** class**,** so that now instead of considering 8 neighbours we only take care of 4 neighbours (excluding itself).

b) Comment:
Using von Neumann's neighbourhood does not generate the same structures as those created using Moore's neighbourhood. For some conditions, such as 1, 2, 4 and 5, there is no patterns observed, the whole thing just dies out after a few time steps, giving a black screen. However, condition 3 and 6 create some different patterns. Especially, there is one block structure generated for condition 6.

3) Vary the Life condition for cell birth and death

For this part, we apply a ***systematic approach*** by which we change each rule in turn (perhaps other rules will be modified slightly to adapt to this change) and see the behaviours of the system.

a. Reduce number of live cells needed in birth rule

We change birth rule so that a cell is born if it has exactly 2 live neighbours. Other rules remain unchanged.

Conditions	Structures seen
1	--Fig.3--
2	-- Fig.3--
3	-- Fig.3--
4	-- Fig.3--
5	-- Fig.3--
6	-- Fig.3--

Figure 3: Similar complex structure seen in all cases

Comment:

We saw a very complex structure like the one above. It expands from a few cells in the middle to the whole screen. This is simply because when reducing the number of live cells needed for Birth rule, we increase the chance a cell can become alive. Dead cells can be reborn easily; therefore, the number of live cells increases significantly, expanding to the whole screen. Especially, the system seems to go on forever, never stabilizes.

b. Reduce number of live cells needed in birth rule

We change birth rule so that a cell is born if it has 2 or 3 live neighbours. Other rules remain unchanged.

Conditions	Structures seen
1	-- Fig.4--
2	-- Fig.4--
3	-- Fig.4--
4	-- Fig.4--
5	-- Fig.4--
6	-- Fig.4--

Figure 4: Complex structure seen in all cases

Comments:
In this case, we changed the birth rule that the dead cell can become alive if they have 2 or 3 live neighbours. It means we increase the chance for dead cell to be born, even more in comparison with the last case when we reduced birth rule from 3 down to only 2. Similarly, we would expect some complex behaviour as seen in the last case. Actually, it turns out that our prediction is correct. The picture above shows clearly the behaviour of the system which is very complex, chaotic, unpredictable, also seems to never stabilize.

c. Increasing number of live cells needed for birth rule
We change birth rule so that a cell is born if it has exactly 4 live neighbours. Other rules remain unchanged

Conditions	Structures seen
1	----
2	----
3	----
4	----
5	----
6	2 blocks

Comment:
There is not interesting things observed. In most conditions, the whole thing just dies out, giving black screen, except for condition 6, two blocks are produced.

d. Change death rule
We change death rule so that a cell becomes dead if it has less than 2 or more than 4 live neighbours. Birth rule remains unchanged, but survival rule changes to 2, 3, or 4 instead of just 2 or 3.

Conditions	Structures seen
1	1 blinker
2	1 beehive
3	----
4	Fig.5
5	Fig.6
6	Fig.6

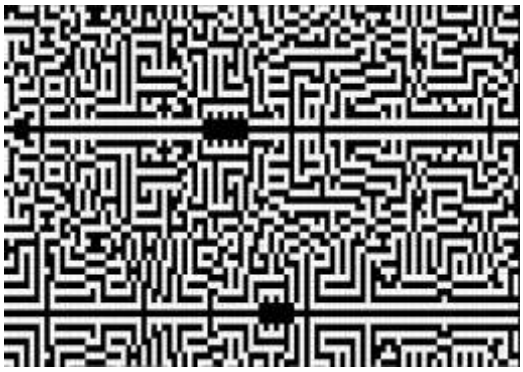


Figure 5: Simulations for Condition 4

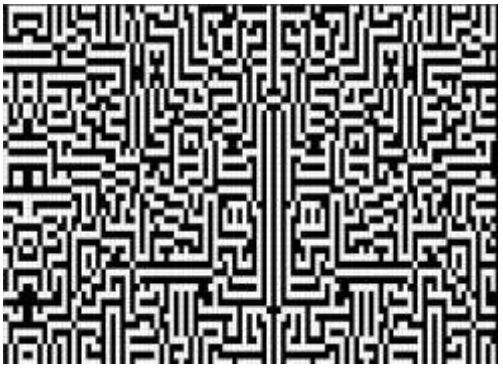


Figure 6: Simulations for Condition 5, 6

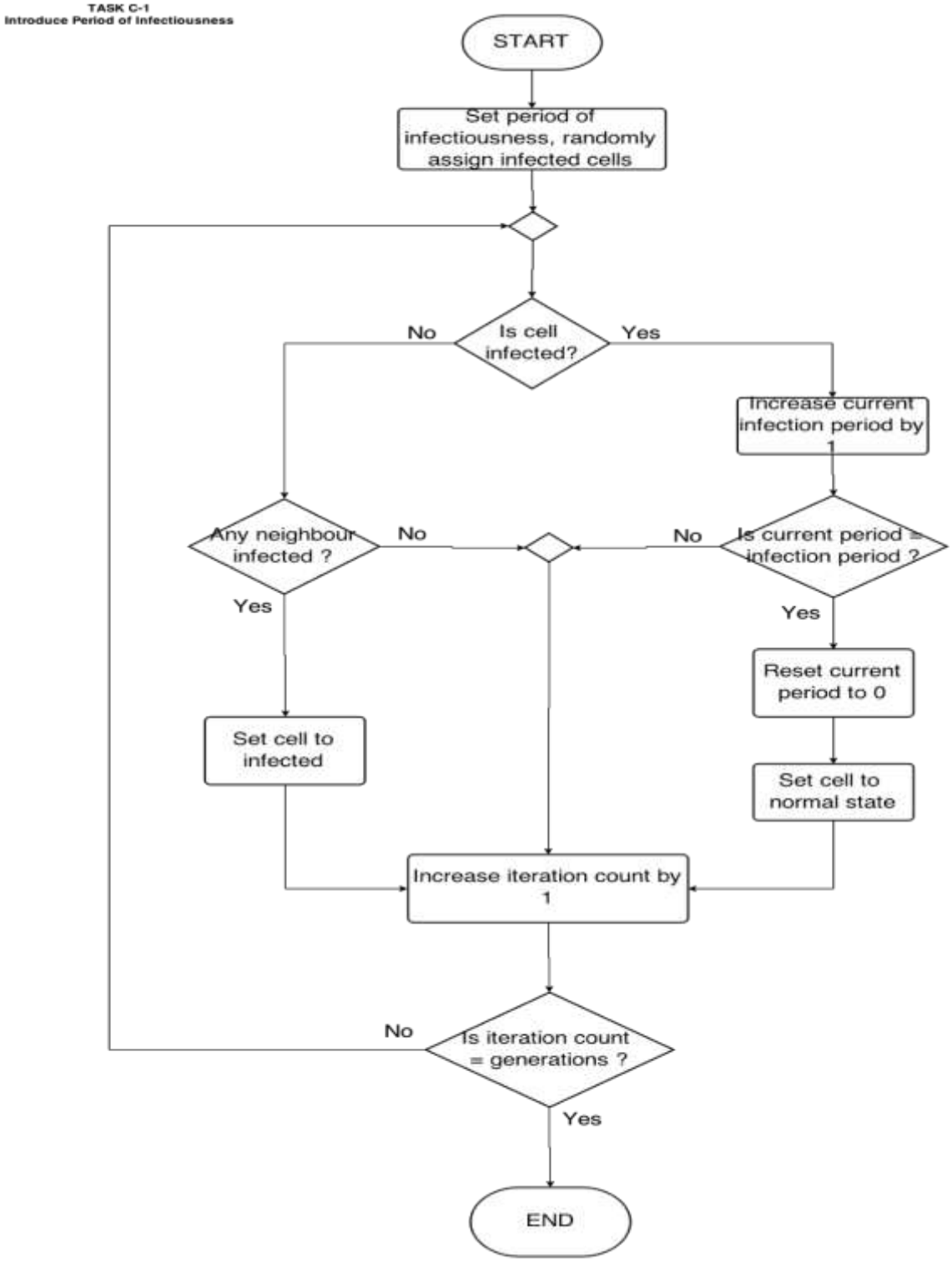
Comment:

Modification in the rule of death does give us something different. Both conditions 1 and 2 show familiar patterns (i.e. blinker and beehive). On the other hand, the difference really comes from the last 3 conditions (i.e. 4, 5, 6), which ends up with a full screen of many complex structures. The whole structures seem to be static but only a few small patterns turn out to be oscillatory with period of 1 and 5.

Task C – Using CAs to investigate Epidemiology

Task C-1: A simple model

Cellular Automata (CA) can be used to model real world systems. The two-dimensional array of cells in the CA can be used to represent geographical space and the 'on/off' states of each cell can be used in a way that models the spread of infectious disease. Using a Moore's neighbourhood, the state of a cell is set to "on" only if it currently has at least one infected neighbour.

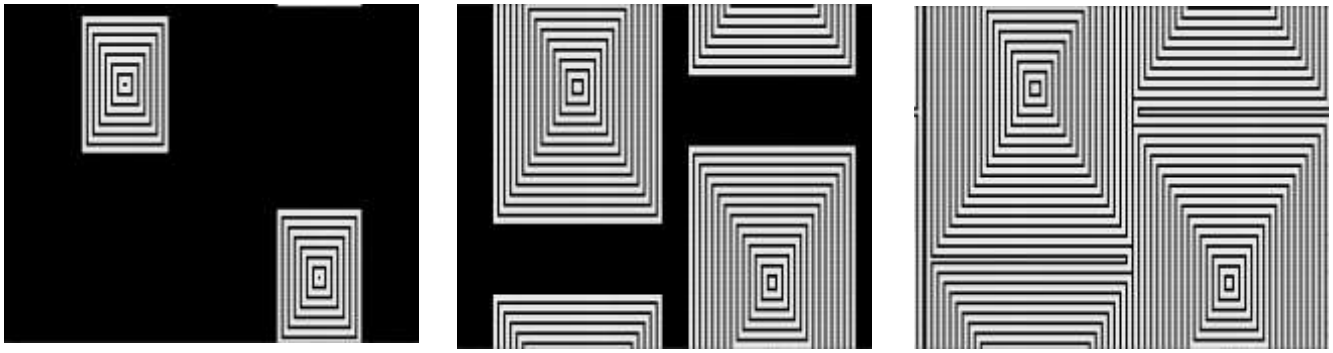


The flowchart above represents the simple model used in a simple implementation of this idea.

Experiment Results:

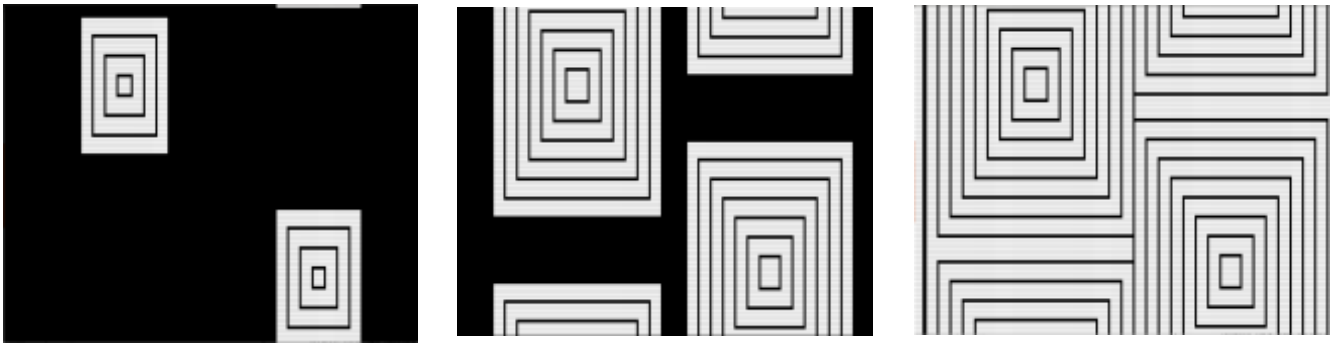
a) PERIOD_OF_INFECTIOUSNESS=2

20-generations 40-generations 100-generations



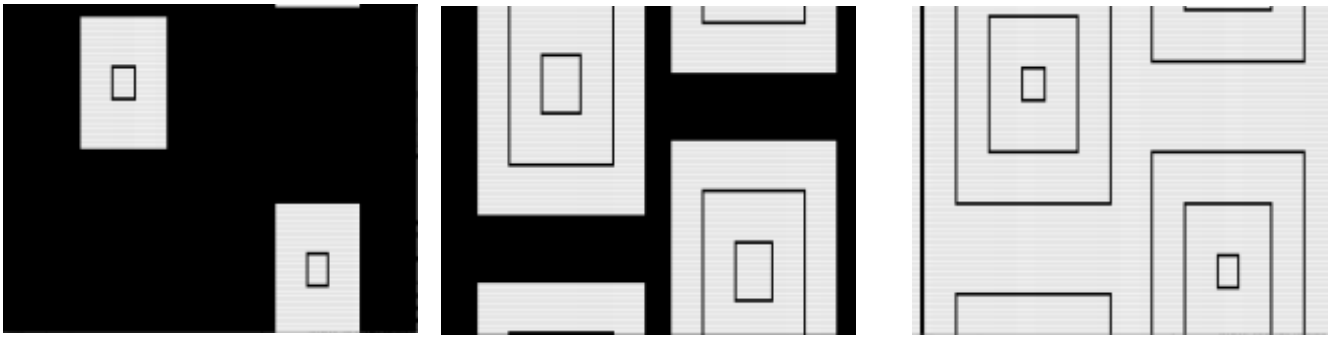
b) PERIOD_OF_INFECTIOUSNESS=5

20-generations 40-generations 100-generations



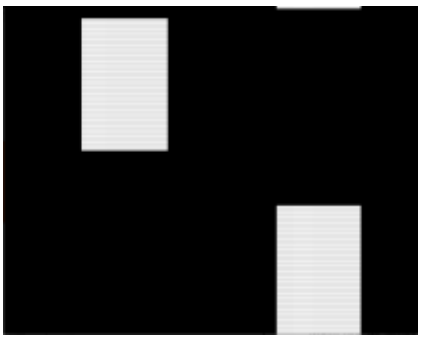
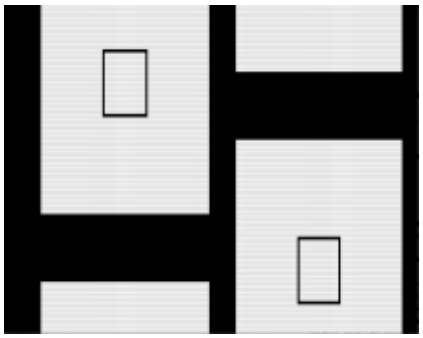
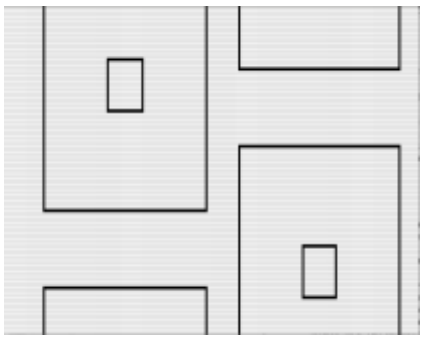
c) PERIOD_OF_INFECTIOUSNESS=15

20-generations 40-generations 100-generations



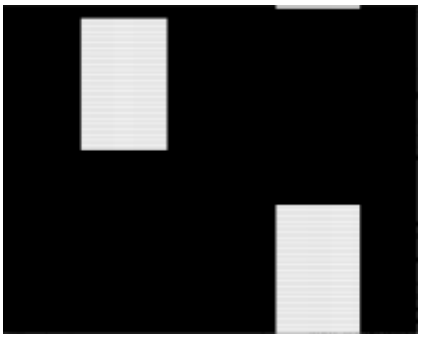
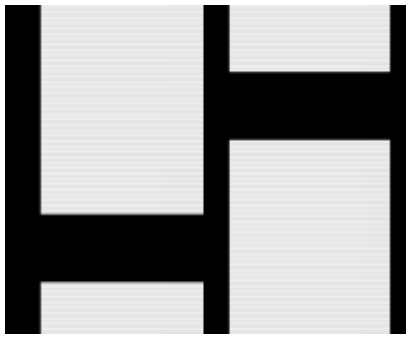
d) PERIOD_OF_INFECTIOUSNESS=30

20-generations 40-generations 100-generations

e) PERIOD_OF_INFECTIOUSNESS=50

20-generations 40-generations 100-generations


Comments:

Using the same initial configuration, in 40 generations, an infection period of 2 produces many wave fronts, whereas period of 50 is still on the way to produce its first wave front. In 100 generations, the larger infection period the more the number of infected cells. Increasing the infection period makes the epidemics wave front thicker.

The simulations produce epidemics wave front because this simple model is homogeneous in properties [C5], such as period of infectiousness, local rules used to determine next state, number of possible states each cell can have.

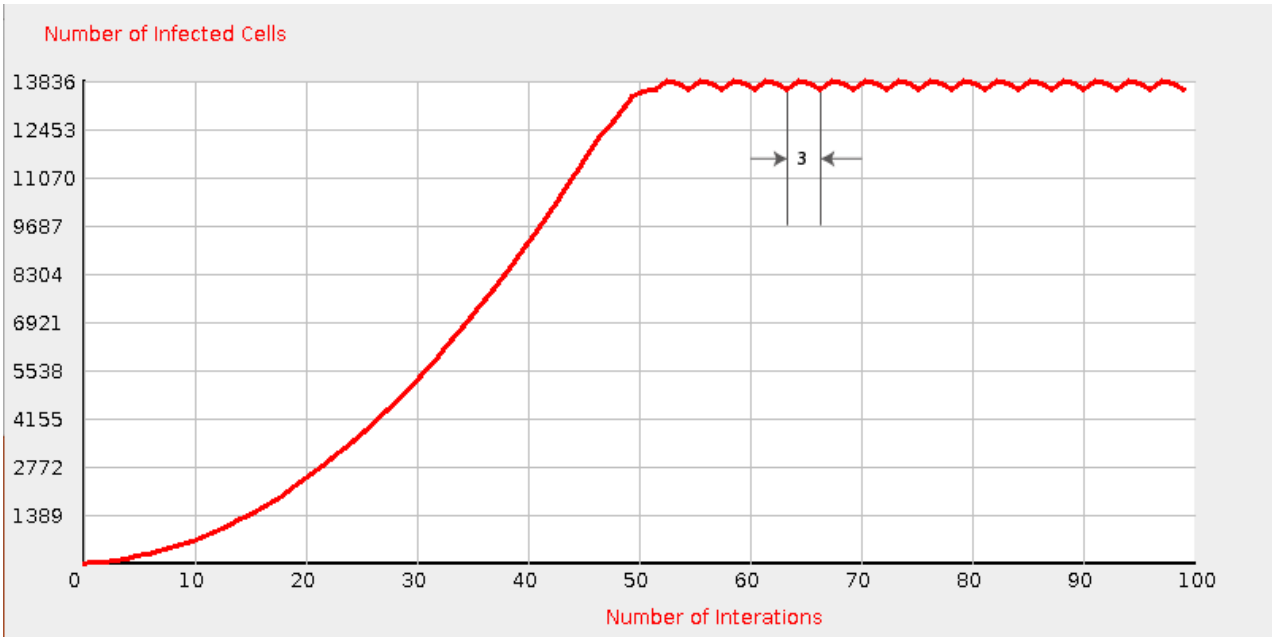
Task C-2: Quantitative representations

We can quantify the findings and present them on a line graph, where the x-axis represents the number of iterations and y-axis the number of infected cells.

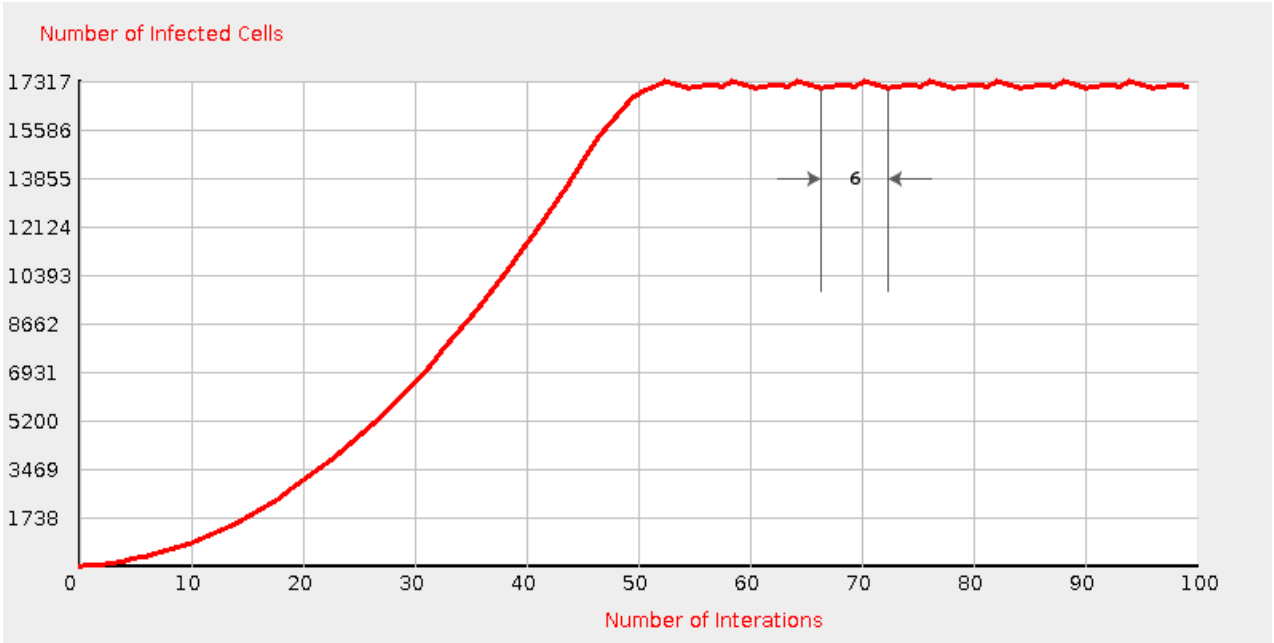
At each time step we count the number of infected cells then record the collected data in a text file, the text file can then be used by an additional class that reads the file and plots the desired graph which will show our quantity of infected cells over time.

Experiment Results:

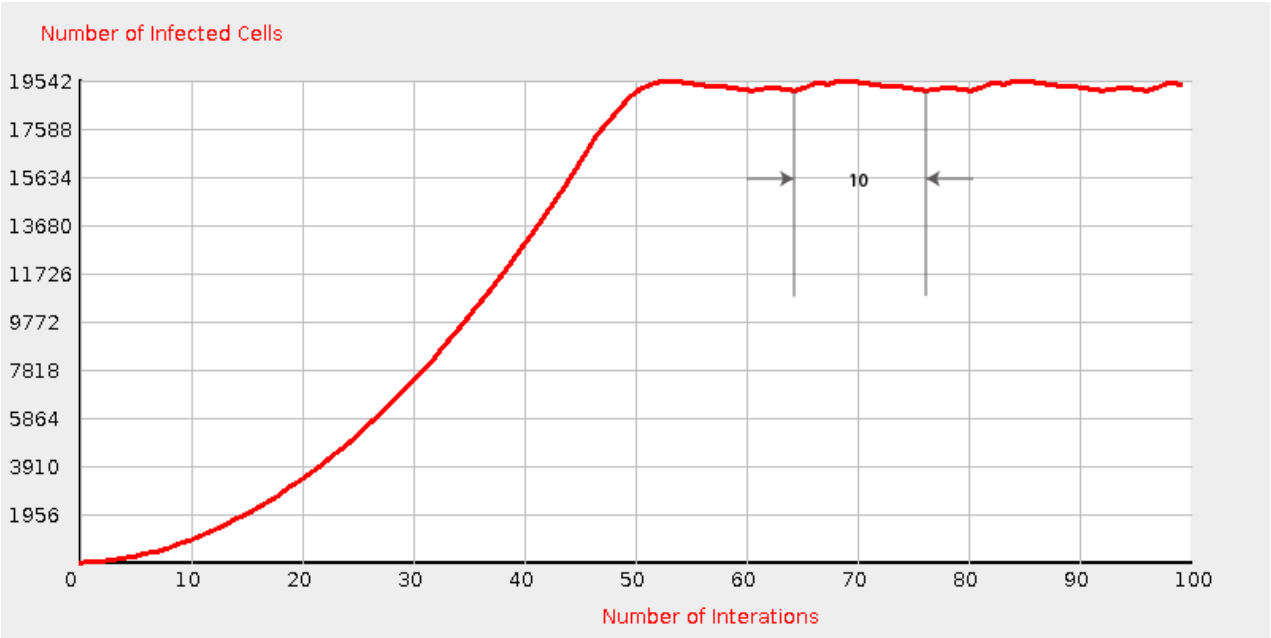
a. PERIOD_OF_INFECTIOUSNESS=2



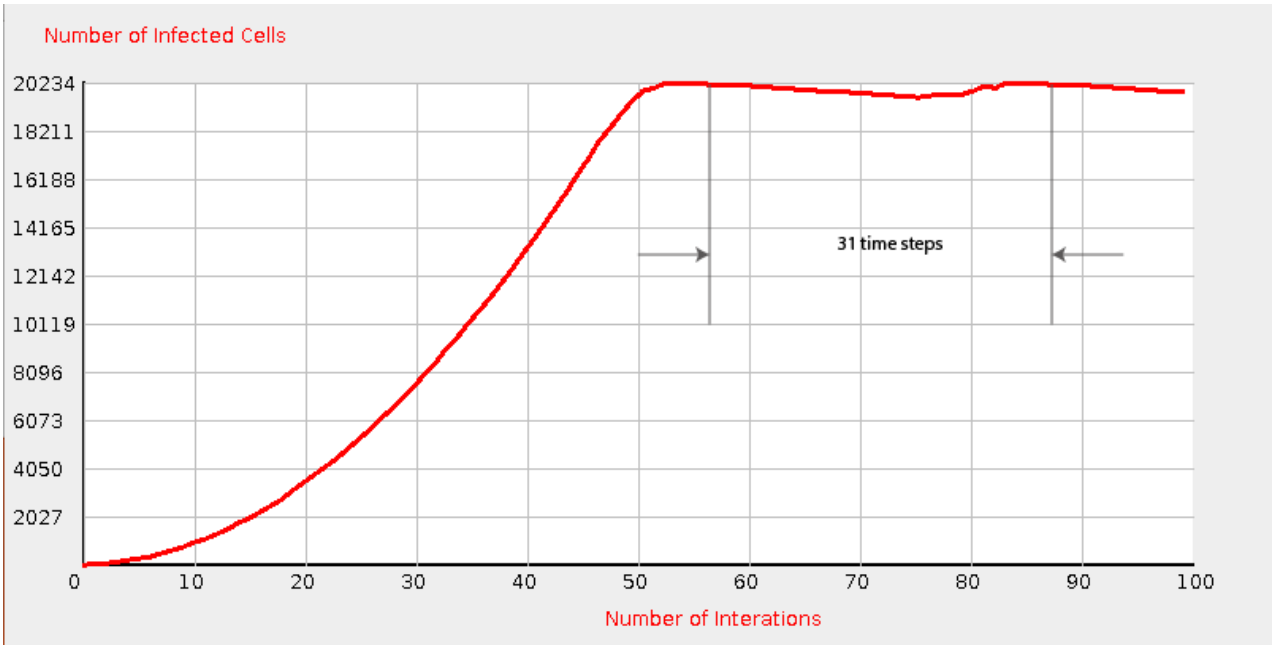
b. PERIOD_OF_INFECTIOUSNESS=5



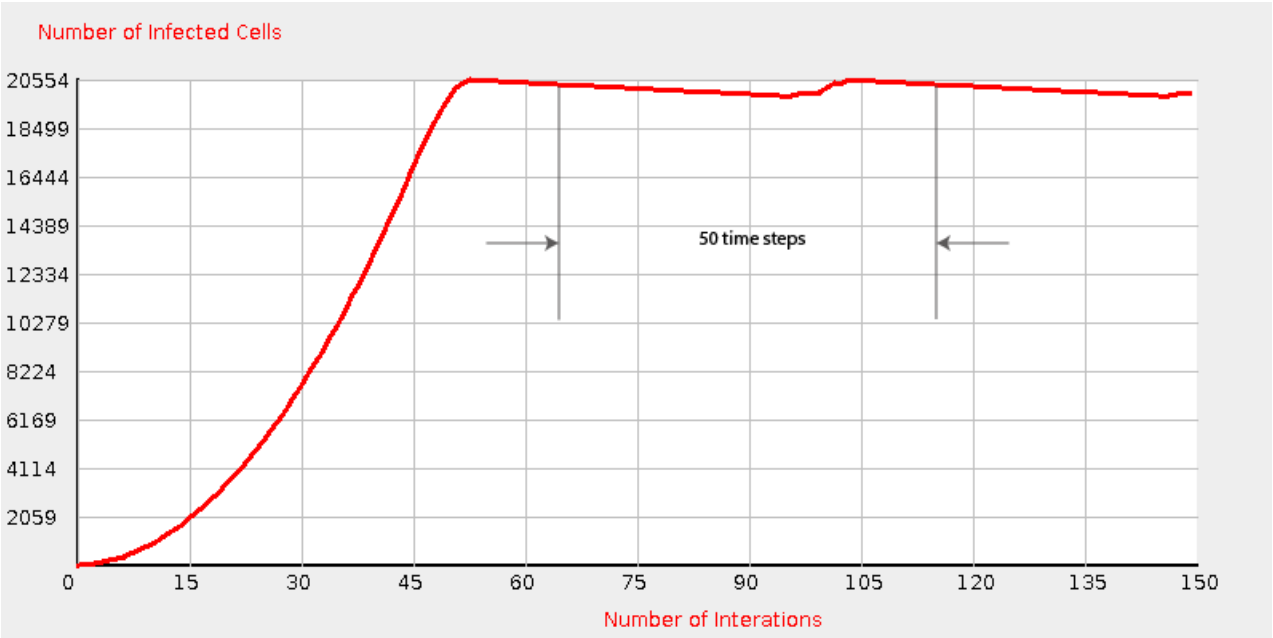
c. PERIOD_OF_INFECTIOUSNESS=15



d. PERIOD_OF_INFECTIOUSNESS=30



e. PERIOD_OF_INFECTIOUSNESS=50



Analysis:

In all cases, the graph grows quadratically from the beginning to time step around 50 and all graphs show a similar curve. This is simply because we use the same display size. In all experiments, each time step, the wave front expands one cell more in all cases.

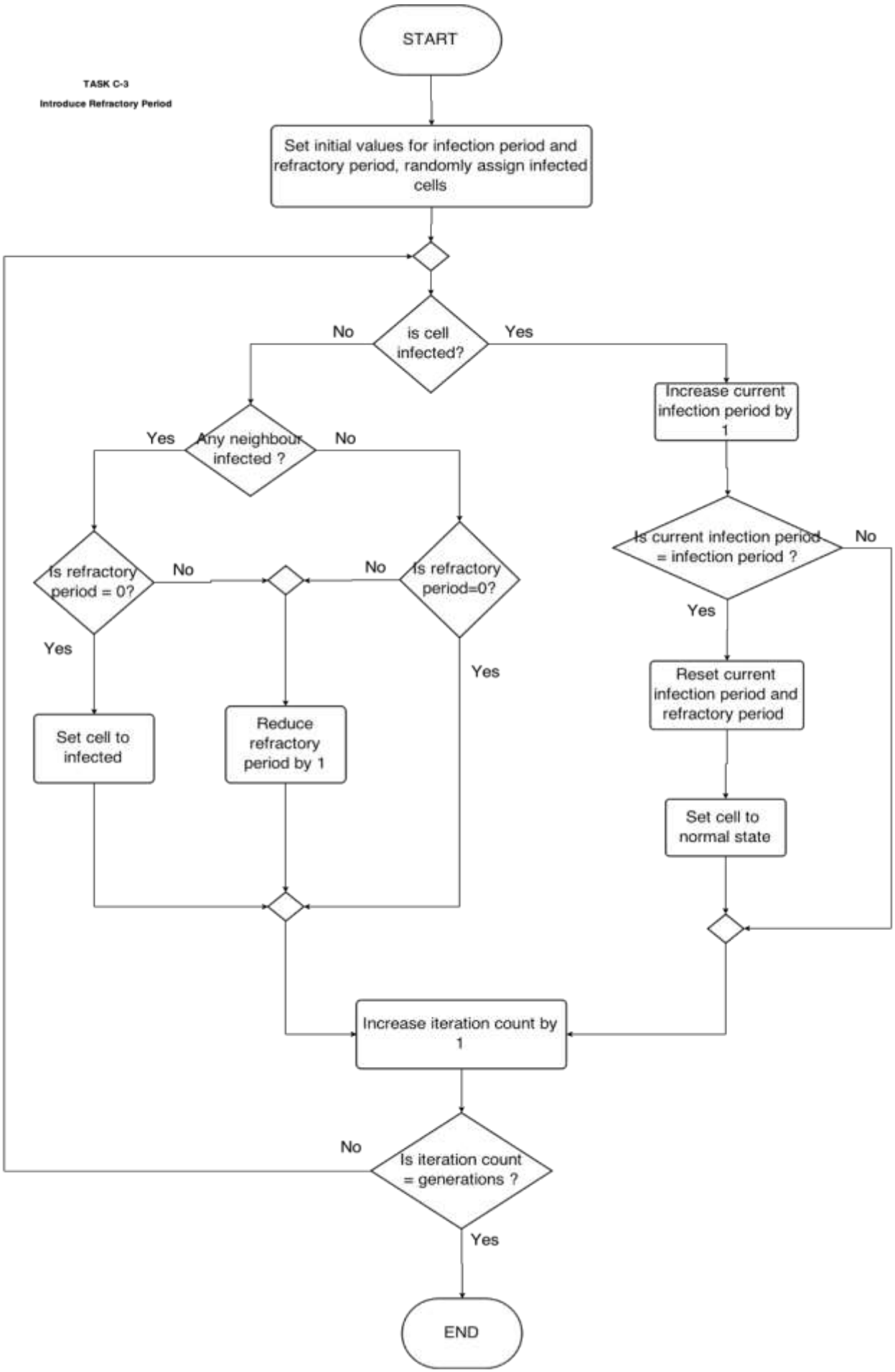
From this point, the number of infected cells periodically fluctuates because of periodic boundary conditions and collisions with other epidemics wave front [C7]. The period of fluctuation increases for larger value of infectiousness period as shown clearly on the graphs above. Changing infection period from 2 to 50 increases the fluctuation period from roughly 3 to 50 time steps.

The current model is homogeneous in which all cells have the same properties [C5], so the infection outbreaks in the form of “circular” front waves, whose ‘thickness’ depends on the infection period. These wave fronts expand, collide and then disappear, and the time they interact corresponds to the fluctuation period. It means the larger the infection period, the longer the fluctuation period.

Task C-3: Refractory Period

A new rule is introduced. After a cell has been infected it will enter a period during which it cannot be infected again. This period is referred to as the ***refractory period***. The value of refractory period for cells which are in refractory state is decremented each iteration.

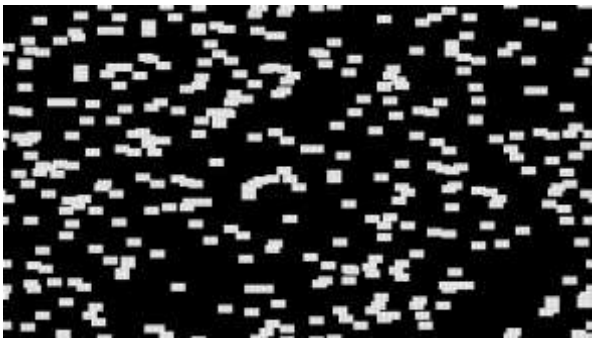
The code is updated and the original flowchart is modified to introduce the process of refractory to the current model. A cell can be infected only if it has at least one infected neighbour and its refractory period is equal to zero.


TASK C-3
Introduce Refractory Period
START
Set initial values for infection period and refractory period, randomly assign infected cells
is cell infected?
No
Yes
Any neighbour infected ?
Yes
No
Is refractory period = 0?
No
Yes
Set cell to infected
Reduce refractory period by 1
No
Is refractory period=0?
Yes
Increase current infection period by 1
Is current infection period = infection period ?
No
Yes
Reset current infection period and refractory period
Set cell to normal state
Increase iteration count by 1
No
Is iteration count = generations ?
Yes
END

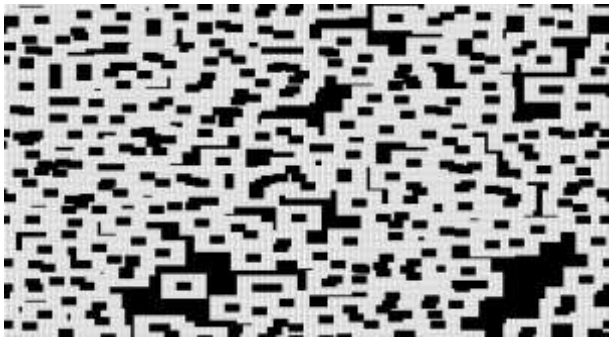
Experiment results:

In this experiment, we set period of infectiousness to 5, and vary the value of refractory period in range 2, 5, 10. We used random initial configuration with randomness level of 1%. The results are shown as following. Period of infectiousness is kept constant.

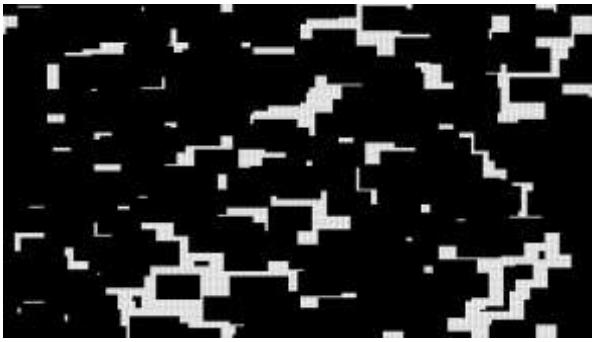
a) Refractory period=2



time=2



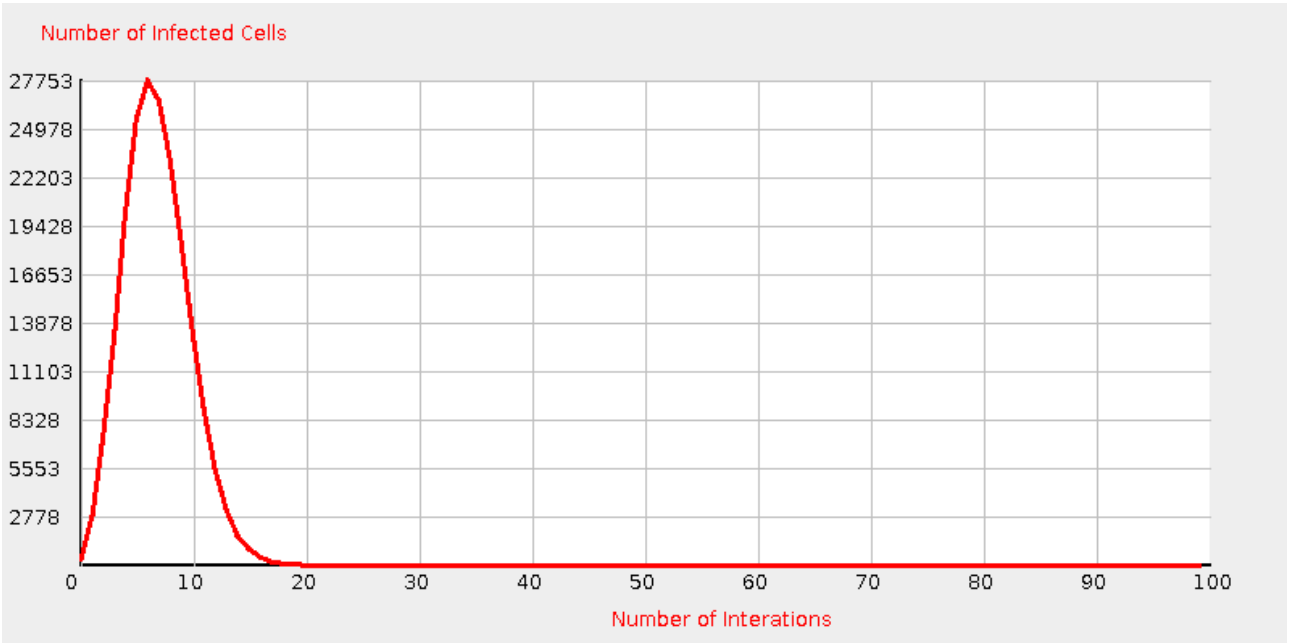
time=7



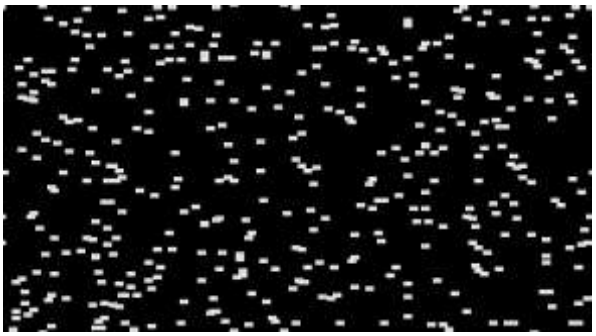
time=12

time=16

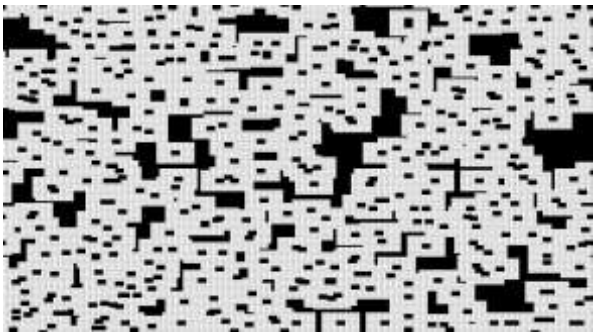
Refractory period=2, 100 generations, 1% random initial configuration



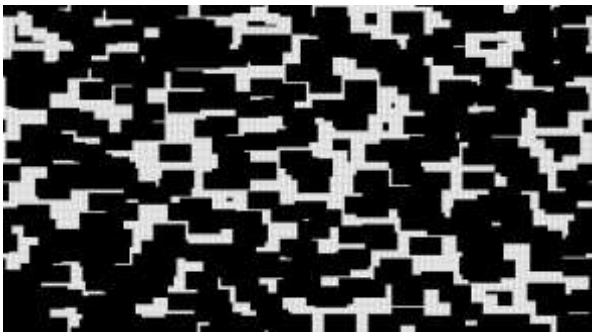
b) Refractory period=5



time=1



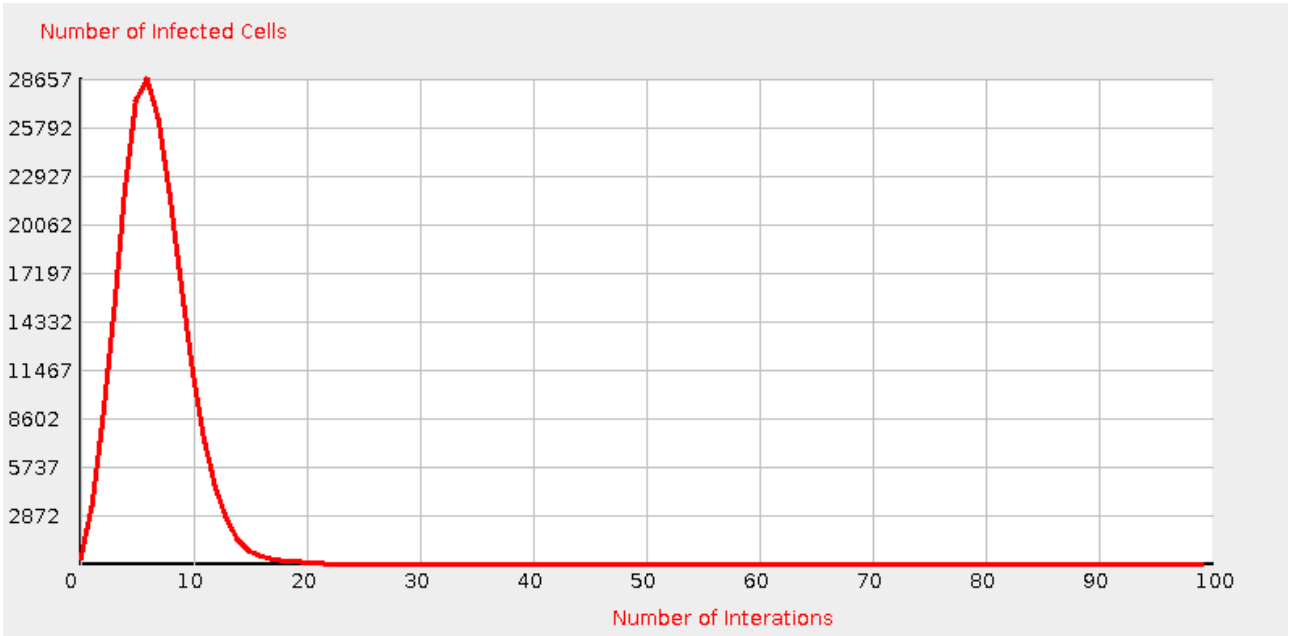
time=6



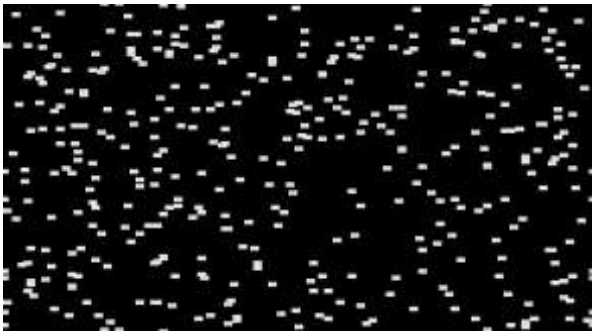
time=10

time=14

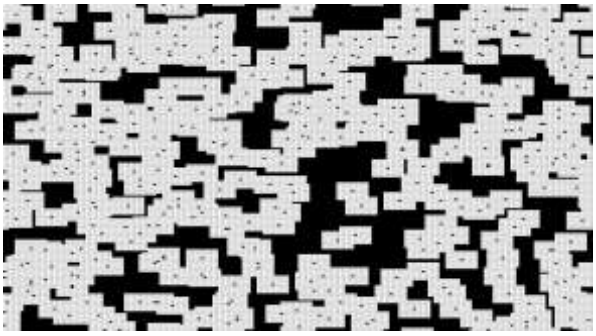
Refractory period=5, 100 generations, 1% random initial configuration



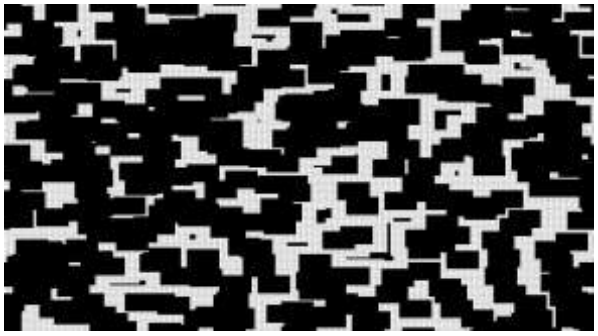
c) Refractory period=10



time=1



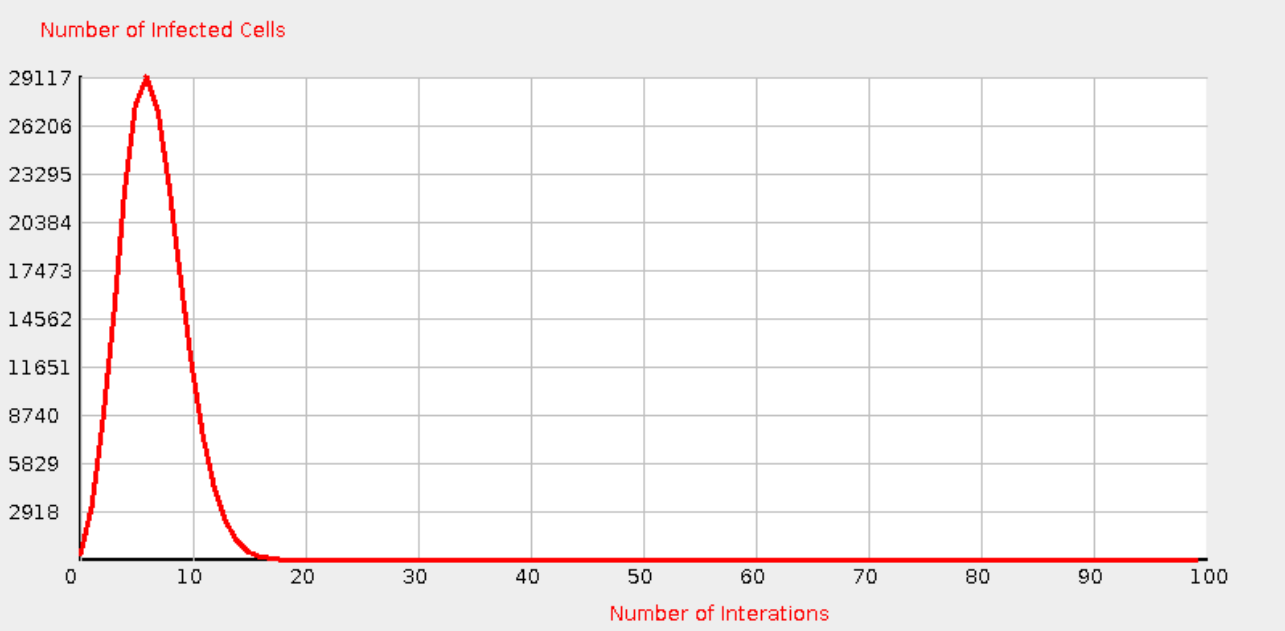
time=5



time=10

time=14

Refractory period=10, 100 generations, 1% random initial configuration



Comments:

In all cases, the model seems to behave in a similar way. The infection outbreaks quickly infecting the entire population, the number of infected cells rockets to reach a peak, and then start dying out. The infection outbreak only lasts about less than 20 time steps. In addition, changing the refractory period, from 2 to 5 to 10, slightly increases the maximum infected cells number, from 27,753 to 28,657 to 29,117, respectively. The findings suggest that changing refractory period may not have any significant effect on the general behaviour of the model.

Each cell in our model immediately enters refractory period when it has been infected. It results in there is only one wave front produced by each cell that is set infected in initial configuration. These font waves will collide and then die out, but the time they interact is only dependent on their 'thickness' which linearly depends on the period of infectiousness, not on the value of refractory period. Therefore, we do not see any difference when varying refractory period in this experiment.

Task C-4: Excitable Medium

Excitable media are systems which propagate waves, sometimes as a form of signal, or as a form of modelling spread of growth, such as in a Mexican wave. The waves are not constant; each 'cell' has to go through a refractory period before a wave can pass over it again. This allows it to be modelled using Cellular Automata, as by the cells turning on or off can allow simulation of a wave front.

In a 2D representation the waves can be observed to be of typically two types, a spreading wave front moving outward from a cell, for example our model shown earlier demonstrates this property. Later a stochastic adoption of the model shows the formation of spiral patterns, another commonly observed pattern for 2D Excitable media waves.

Excitable media can be observed in star formation in galactic dust and in membrane depolarisation in nerve axioms [C17]. A simple example of excitable media is seen in wild fires. The refractory period for a fire is caused by the fact that once an area is burned; fire cannot spread until the plants have regrown. Models allow for diversity among tree types, the layout of the land and weather conditions [C8], and have been improved over time to increased accuracy to allow for more detailed information such as rate of the fire spread [C10].

Partial differential equations have traditionally been the way of modelling excitable media however recently Cellular Automata has been adopted as a way to model these systems. Morphogenetic process, which is the growth and shaping of an organism, can be modelled using Cellular Automata. Originally this was done by having the shapes of the cells being used to generate the shapes of the wave fronts. Later work however found another way to do it, by using different forms of waves to represent different types of tissue and activity, as opposed to just being linked by shape. The complexity of the model can be

increased by adding more parameters, with a minimal model using only three [C9]. These foundations have then been built on to the morphogenesis of specific parts of organisms.

The growth of vegetation can be modelled with 2D CA. Some allow for different species of plants to be modelled and compared [C11], while another model takes into account the concentration of resources [C11], while another model takes into account the concentration of resources, and how it would affect the growth by observing spatial patterns [C12].

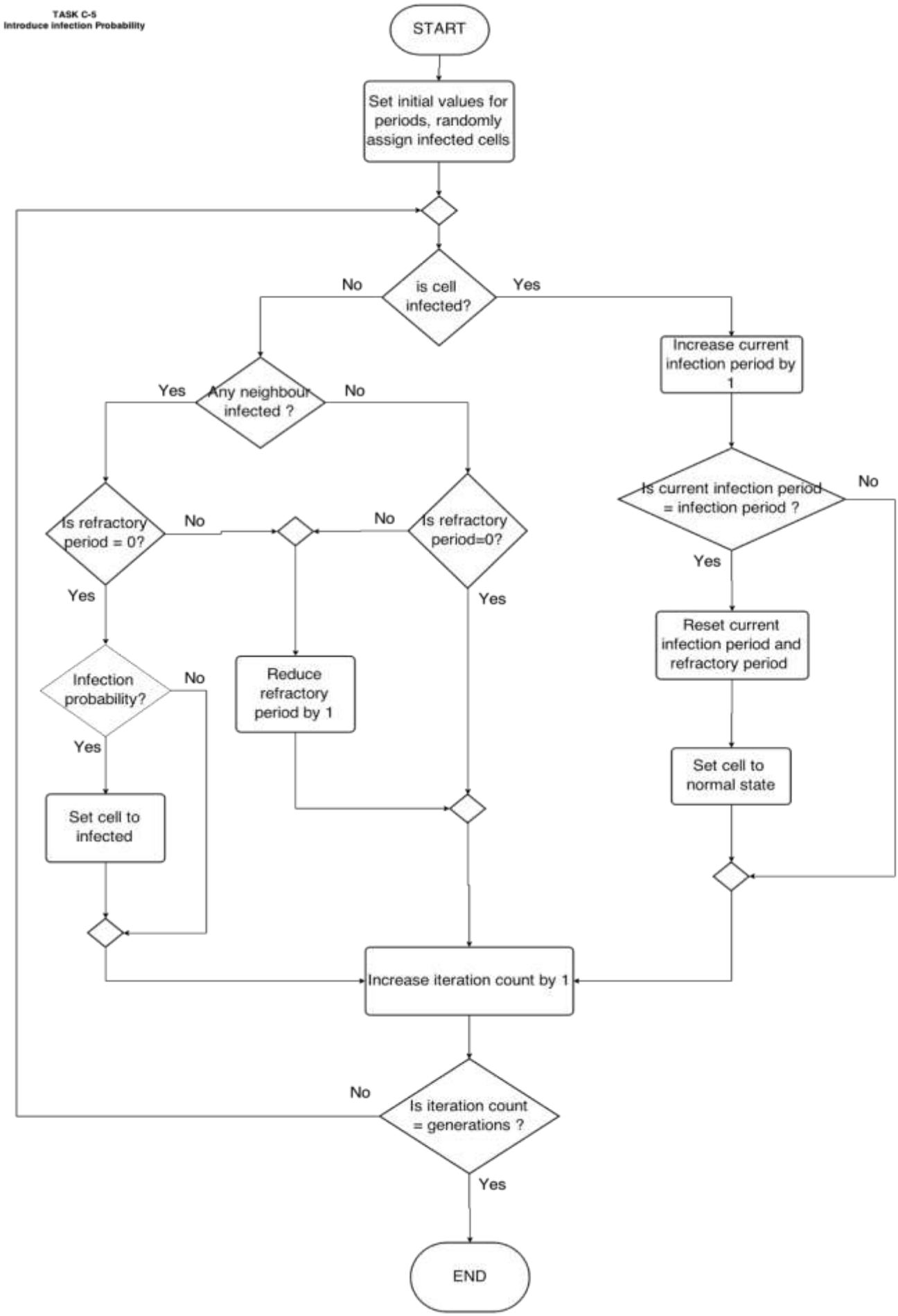
There have been experiments modelling metastability [C13] the displacement of *"an apparent steady state by another, often abruptly and after a prolonged period of time"* [C13], using a model called the Greenberg-Hastings Model (GHM). The GHM which uses three states with the following rule; If a cell is in the excited state then at the next iteration it will be in the refractory state. If a cell is in a refractory state then the next iteration it will enter the resting state, finally if the cell is in a resting state and one or more neighbours is excited then the cell in question will become excited at the next iteration.

Cellular automata are often selected to model excitable media due to the low computational cost involved, the models are simplistic in nature, and they are discrete and states can be calculated by hand with relative ease. Miyamoto and Sasaki have adopted the CA to simulate lava flows by further enhancing the model [C14].The Belousov-Zhabotinskii reaction has been explored using CA models [C15] the CA allows for a simulation of the chemical reaction, showing the use of CA models for studying excitable media in chemical reactions.

In summary, CA has been proven useful in various areas of scientific research where modelling excitable medium systems is required.

Task C-5: Stochastic Model

Transforming the non-stochastic model into a stochastic model allows us to set a probability of infection for each cell. The model flowchart is modified to introduce the concept of infection probability.

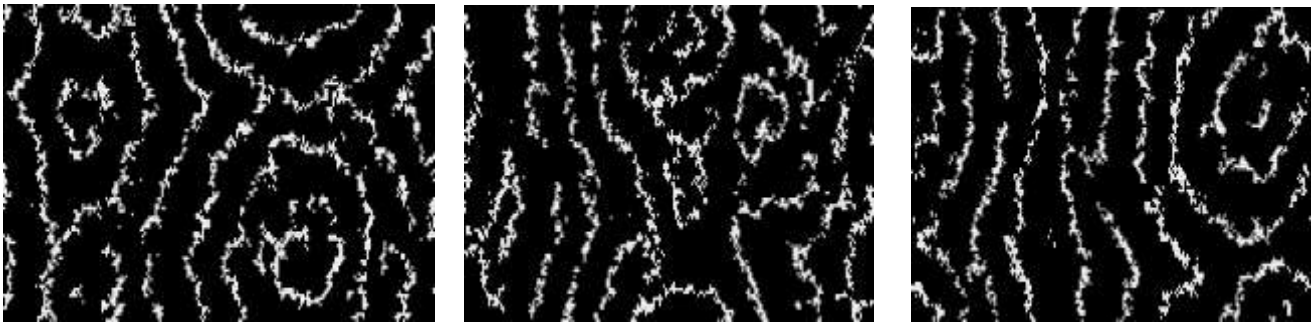

TASK C-5
Introduce infection Probability
START
Set initial values for periods, randomly assign infected cells
is cell infected?
No
Yes
Any neighbour infected ?
Yes
No
Increase current infection period by 1
Is refractory period = 0?
No
No
Is refractory period=0?
Is current infection period = infection period ?
No
Yes
Yes
Yes
Reset current infection period and refractory period
Infection probability?
No
Reduce refractory period by 1
Yes
Set cell to normal state
Set cell to infected
Increase iteration count by 1
No
Is iteration count = generations ?
Yes
END

The code is updated to implement a probability base value, which represents the "contribution" of each neighbour to infection probability and is then multiplied with a pseudo-random number generated using the Java Math class. To investigate this model we vary the probability base.

a) $\boldsymbol{Base = \frac{1}{8}}$

First a base of $\frac{1}{8}$ is selected this means that if all eight neighbours of the current cell using Moore's neighbourhood are infected then the probability of current cell getting infected will be 100%.

Three trials are run using a base of ⅛ during which 500 generations are observed in each trial.



Trail 1 Trail 2 Trail 3

The above are all patterns generated during each individual trial, in all trials after approximately 200 iterations spiral like patterns begin to emerge.

b) $\boldsymbol{Base = \frac{1}{4}}$

Next the base is changed to ¼ to increase the infection probability, three trials are run, the infection vanishes after at most 70 iterations. An example is shown below:

time=26 time=38

time=57

time=63

c) $\boldsymbol{Base = \frac{1}{2}}$

Next ½ chance of infection. Three trials all give similar behaviour shown below; infection dies out after 40 iterations.

time=27

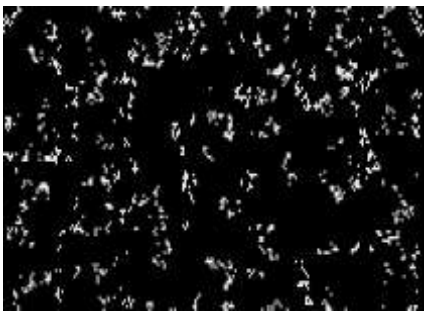
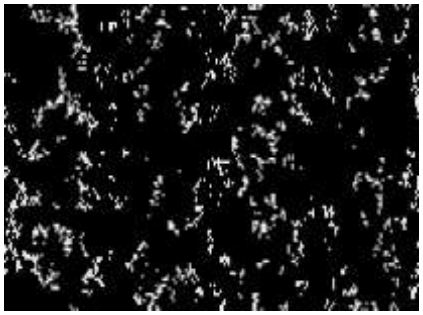
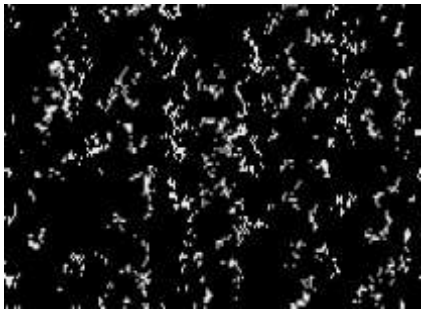
time=38

The behaviour with higher probability seems to match the behaviour of the trials in **Task C3**. When a trial of base 1 is run the same life cycle of the infection is observed as in **Task C3**, a uniform wave of cells being turned ‘on’ and then the death of the infection after x iterations.

d) $\boldsymbol{Base = \frac{1}{12}}$

Base 1/12 is chosen to reduce chance of getting infection. 3 trials are run for 500 iterations.

At around 80 iterations noisy patterns begin to be formed and this behaviour continues for the remainder of the trial. The images below show the final states after each trial.

Trail 1 Trail 2 Trail 3

e) $\boldsymbol{Base = \frac{1}{16}}$

Infection probability is further reduced. Again 3 trials are run for 500 iterations each.



Trail 1 Trail 2 Trail 3

With such a low chance of infection, infection waves from certain seed points can die out permanently which explains how certain regions on the images above show areas where there is no spread of infection.

In conclusion, higher probability creates more defined wave front propagation and lower probabilities tend to show more gradual spreads that don't die out the way we see in **Task C3** as the wave fronts clash.

Task C-6: More concepts

1. Population density

We represent variation in population density by assign different values of infection period into different areas. We do it by dividing our area into four smaller regions, each of which has a different value of infection period, so that we can examine how differently infection outbreaks in these areas. Below is an example of how to divide the area.

inf=p1	**inf**=p2
inf=p4	**inf**=p3

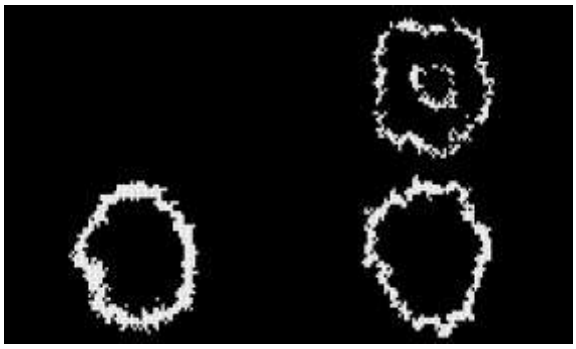
(**inf**=period of infectiousness)

Results:

a. p1=2, p2=4, p3=6, p4=8 and refractory period=15.

time=2

time=8



time=41

time=1000

b. p1=2, p2=5, p3=10, p3=20, refractory period=15

Now we assign only one infected cell in the centre as initial configuration.

time=80

time=383

Comment:

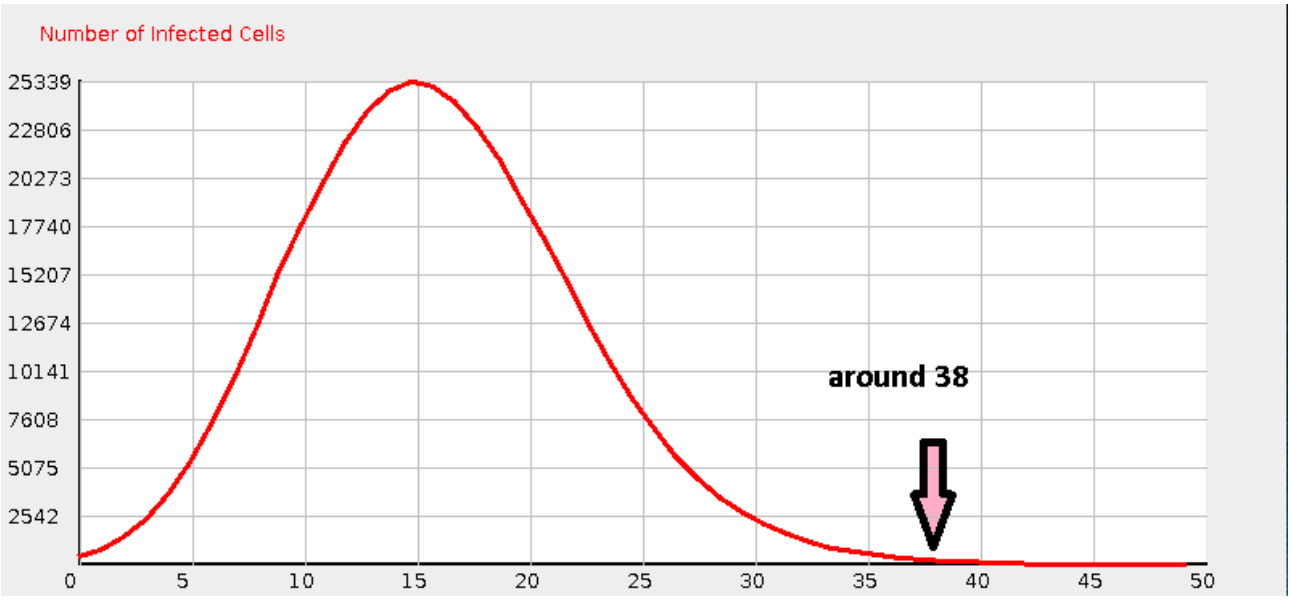
When introducing concept of ***population density*** into our model, which means that if there are many people living in an area, then the infection will stay there longer due to the time it takes to spread within that area. It can be seen clearly from the above image that the infection spreads in different ways in different areas. In the bottom left region, which has largest value of infection period, represents a city, the infection takes more time to spread in an cell as the 'thickness' of the epidemics front is thicker than the others. On the other hand, in the top left area, which has lowest infection period, representing a remote area, there are less people infected as shown in image above with just some 'white' cells.

2. Immunity

There are two kinds of ***immunity*** explored in this model, immunity gained after recovering from an infection and 'natural' immunity which is granted as an innate quality of a given cell. We introduce these concepts into our model by setting two respective values probability values, **IMMUNE_BEFORE** and **IMMUNE_AFTER**.

a. IMMUNE_AFTER

We keep infection period constant (i.e. 15), using 1% random initial configuration, and vary only IMMUNE_AFTER. The results are shown as follows.



IMMUNE_AFTER=5%

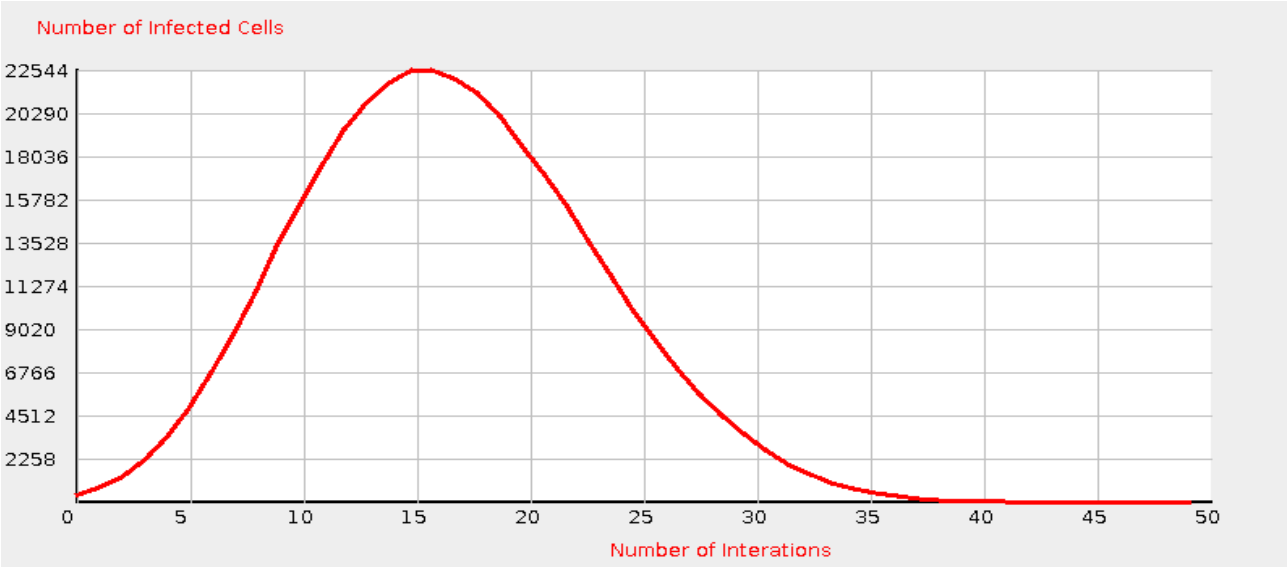
IMMUNE_AFTER=50%

Comments:

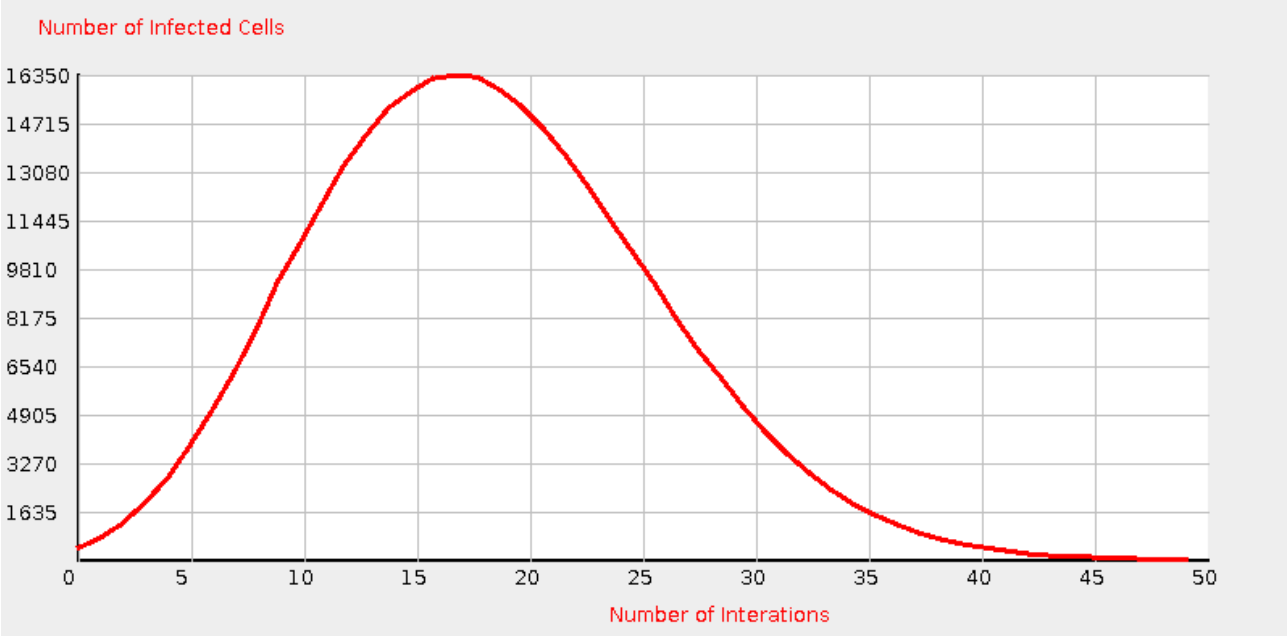
As we can see from the graphs the number of infected cells decreases faster when the population gains immunity afterwards. When increasing IMMUNE_AFTER, we increase the chance of a cell to become immune after recovering from infection, we would expect the disease dies faster.

b. IMMUNE_BEFORE

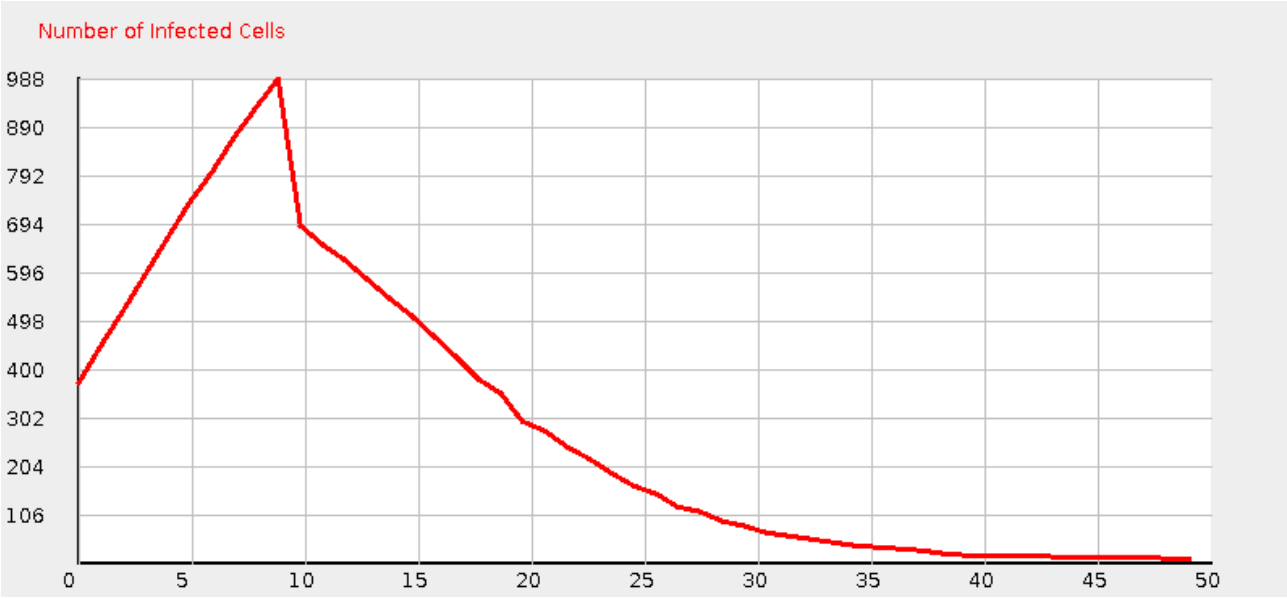
We vary the **IMMUNE_BEFORE** probability in large range while keeping refractory period constant (i.e. 20).



IMMUNE_BEFORE=5%



IMMUNE_BEFORE=20%



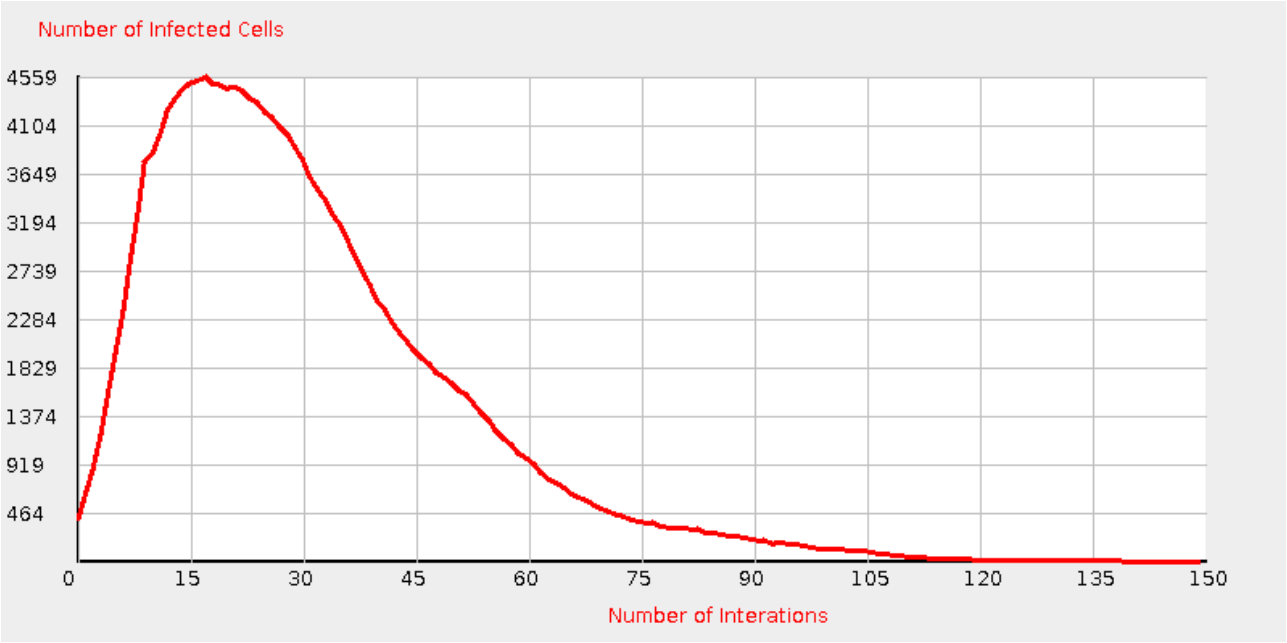
IMMUNE_BEFORE=80%

Comments:

The data shows that when increasing IMMUNE_BEFORE, the maximum number of infected cells decreases, from 22544 down to 16350 down to 988. This is intuitive and expected as with disease if potential hosts are immune they will not be affected by the disease, hindering its infectious spread.

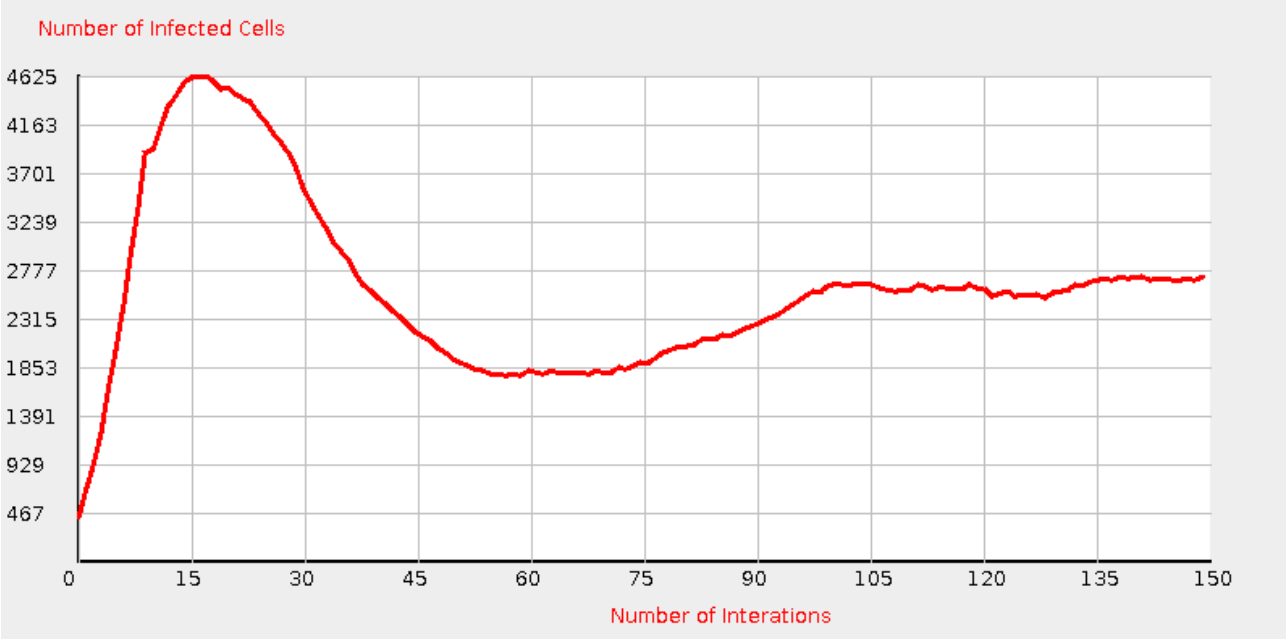
c. Both concepts

We introduce both concepts to the model. Here we use refractory period=20, both IMMUNE_BEFORE and IMMUNE_AFTER equal 50%.



Comments:

The model again behaves in the way we would expect. Below is the graph when introducing only IMMUNE_BEFORE to the model, which has value of 50%. The infection dies faster with both immunities active.



Comparing these two graphs, it can be seen that the maximum number of infected cells goes down from 4625 to 4559. The infection comes to an end after about 120 time steps but it does not end after 150 time steps when introducing only IMMUNE_BEFORE.

Task C-7: Comparison

The model developed by Hoya White et al [C16] bears similarities to the one developed for this report, the CA has been adopted to represent space and population; the configuration of cell states represent whether a cell is healthy or infected over time.

Hoya White experiments with both Von Neumann and Moore Neighbourhood (we have only used Moore Neighbourhood) and they show properties seen in our model, particularly the expansion of an infectious wave front from an infected point centred in space. Also similar is the way infected and recovered cells vary in quantity over time, especially how the infected cells number shows a rise during the simulation and then simply dies out as time progresses. The overall behaviour of both models are closely related, differences in results are often determined by different approaches such as the complex interaction factor between cells.

Both models suffer the same limitations Hoya White refers to the ‘small world’ effect for their model and likewise in our trials it can be seen that we are limited in modelling an infectious disease when dealing with a finite grid that does not consider the world’s geographical context.

The model we designed in this experiment is referred as so-called SIR model, which is possibly the framework for all such models [C3]. Besides, there are other examples of epidemics model in which people often simulate the infection outbreak using ***Differential Equation,*** which is a very common method used in this field. However, this model is not suitable when the boundary and initial condition are complicated [C5] because it assumes space is homogeneous (i.e. well-mixed stable population). This assumption seems to be adequate for large scale space but perhaps results in unrealistic outcome for small one [C6]. In particular, it neglects individual communication and changing behaviours which are two of the most significant factors affecting infection spreading [C6].

Some of the problems mentioned above have been solved using another model called ***Global Stochastic Field Simulation*** (GSFS) which considers “heterogeneous population, demographics, constraints, contact structure and disease dynamics” [C1]. Each cell in the model has its own set of three states (S,I,R) and “specific demographic character” [C1] on which individual contacts and spread of infection depend. In particular, the simulator models only interactions that can lead to a successful disease transmission [C1] instead of considering all possible interactions.

References

Task A

[A1] Wolfram, S. (1984) Universality and complexity in cellular automata. *Physica D*, 10, pp. 1-35.

[A2] Langton, C. (1990) Computation at the Edge of Chaos: Phase Transitions and Emergent Computation. *Physica D*, 42, pp. 12-37.

[A3] Berto, F., Tagliabue, J. (2012) *Cellular Automata* [online]. Summer 2012 Edition. Available from: http://plato.stanford.edu/archives/sum2012/entries/cellular-automata [Accessed 9th March 2015].

Task B

[B1] Weisstein, E. W. *Game of Life* [online]. Available from: http://mathworld.wolfram.com/GameofLife.html [Accessed 9th March 2015].

[B2] Floreano, D. (2008) *Bio-inspired Artificial Intelligence*. Cambridge, MA, MIT Press.

[B3] Weisstein, E. W. *Eric Weisstein's Encyclopedia of the Game of Life* [online]. Available from: http://www.ericweisstein.com/encyclopedias/life [Accessed 9th March 2015].

Task C

[C1] Mikler, A. R., Venkatachalam, S., Abbas, K. (2005) Modeling infectious diseases using global stochastic cellular automata. *Journal of Biological Systems*, 13(4), pp. 421-439.

[C2] Beauchemin, C., Samuel, J., Tuszynski, J. (2005) A simple cellular automaton model for influenza A viral infections. *Journal of Theoretical Biology*, 232(2), pp. 223-234.

[C3] Johansen, A. (1996) A Simple Model of Recurrent Epidemics. *Journal of Theoretical Biology*, 178(1), pp. 45-51.

[C4] Ahmed, E., Agiza, H. N. (1998) On modeling epidemics. Including latency, incubation and variable susceptibility. *Physica A: Statistical Mechanics and its Applications*, 253(1-4), pp. 347-352.

[C5] Sirakoulis, G. C., Karafyllidis, I., Thanailakis, A. (2000) A cellular automaton model for the effects of population movement and vaccination on epidemic propagation. *Ecological Modelling*, 133(3), pp. 209-223.

[C6] Di Stefano, B., Fuks, H., Lawniczak, A. T. (2000) Object-Oriented. Implementation of CA/LGCA Modelling Applied To The Spread Of Epidemics. *Canadian Conference on Electrical and Computer Engineering*, 1, pp. 26-31.

[C7] Rousseau, G., Chaté, B., Rousseau, R., Giorgini, H., Livi, H., Livi, H. (1997) Dynamical phases in a cellular automaton model for epidemic propagation. *Physica D: Nonlinear Phenomena*, 103(1-4), pp. 554-563.

[C8] Karafyllidis, I., Thanailakis, A. (1997) A model for preventing forest fire spreading using cellular automata. *Ecological Modelling*, 99(1), pp. 87–97.

[C9] Markus, M. (1990) Modelling morphogenetic processes in excitable media using novel cellular automata. *Biomedica Biochimica Acta*, 49(8-9), pp. 681-696.

[C10] Encinas, H. A., Encinas, H. L., White, H. S., Martin del Rey, A., Sanchez, R. G. (2007) Simulation of forest fire fronts using cellular automata. *Advances in Engineering Software*, 38(6), pp. 372-378.

[C11] Balzter, H., Braun, P. W., Kohler, W. (1998) Cellular automata models for vegetation dynamics. *Ecological Modelling,* 107(2–3), pp. 113–125.

[C12] Colasanti, R., Grime, J, P. (1993) Resource dynamics and vegetation process: a deterministic model using two-dimensional cellular automata. *Functional Ecology*, 7(2), pp. 169-176.

[C13] Fisch, R., Gravner, J., Griffeath, D. (1993) Metastability in the Greenberg-Hastings model. *The Annals of Applied Probability*, pp. 935-967.

[C14] Miyamoto, H., Sasaki, S. (1997) Simulating lava flows by an improved cellular automata method. *Computers & Geosciences*, 23(3), pp. 283-292.

[C15] Gerhardt, M., Schuster, H., Tyson, J. J. (1990) A cellular automaton model of excitable media: III. Fitting the Belousov-Zhabotinskii reaction. *Physica D: Nonlinear Phenomena,* 46(3), pp. 416-426.

[C16] White, H. S., Martin del Rey, A., Sanchez, R. G. (2007) Modelling epidemics using cellular automata. *Applied Mathematics and Computation*, 186(1), pp. 193-202.

[C17] Weimar, J. R., John J. T., Layne T. W. (1992) Third generation cellular automaton for modeling excitable media. *Physica D: Nonlinear Phenomena*, 55(3), pp. 328-339.