2023

Percolation Modelling

CITS4403 PROJECT SEMESTER 2 2023

CALLUM BROWN (22985036)

1. Background

1.a. Problem Statement

In principle, percolation is the process in which a liquid propagates through a material with variable porousness. Using a classic coffee percolator as an example, a filter is placed into the top of the machine with a blanket of coffee grounds spread over it. The pot is placed over a heat source until hot water rises from the base of the pot through a vertical tube where it cools again and drops back down over the coffee grounds, gradually 'wetting' the coffee, absorbing it into the hot water. This liquified coffee then begins to travel down through the porous material of the filter through a series of path of microscopic holes in the paper, until it drips back into the hot water below.

This process filters out impurities in the coffee grounds and allows the coffee to be kept at a high temperature without affecting the flavour, as the process continues while the water is hot enough to rise through the vertical tube. The paths that the hot coffee take through the filter material are called 'percolating clusters'.

Percolation theory 'describes the behaviour of clustered components in random networks' [1]. The process can be applied to several real-world situations, both of literal percolation in groundwater recharge and hydrogen gas in micropores, and in theoretical percolation used to model pandemics and forest fires.

1.b. Importance Discussion

Surface water is constantly percolating through earth and rock into groundwater reserves, a process which filters and purifies drinking water naturally. Through our CA model we can simulate how this water travels through the rock based on the porousness of different rock types, allowing us to analyse what materials are best for filtering water quickly and effectively, and allows us to learn which materials to avoid. In a world with an increasing population, maintaining adequate drinking water is becoming more and more of an engineering task. Simulating an efficient method for generating drinking water is of a high priority to many geographical areas around the globe.

Another particularly relevant application is the use of percolation theory to study disease propagation, as was done with COVID-19 [3]. Modelling the spread of infectious disease provides key insights into the factors driving transmission between individuals and can help to forecast how the disease will spread allowing for the implementation of effective preventative measures.

In the context of our model, individuals susceptible to infection are represented as cells with higher porousness, and their social networks are the adjacent cells in all directions representing friends and family, each with their own individual porousness. A central wet cell representing patient zero will begin an outward propagation in all directions, contrary to the coffee percolator model. While our model is more simplistic than a real-world scenario, the modelling and prediction of the spread of infectious disease is critical to public health. Another similar application of this central outward percolation is used in the modelling of forest fires.

1.c. Overview

We will examine two cellular automaton models, top-down percolation, and forest fire percolation, establishing the effect of changing population size and porousness of the automata through grid-like simulations. These simulations will allow us to make non-trivial predictions about the behaviour of complex real-world systems, ground water recharge and forest fire spread, respectively.

From the results of the top-down percolation simulations, we can assume the population size (n) has negligible effect on how likely a percolating cluster (a connected path from top to bottom of the grid) is to occur. We also establish a critical point for the porousness of the graph, the point at which a percolating cluster has a 50/50 chance of occurring. This point, at below approximately 0.59, is an indication of a minimum porousness a simulation can have while still allowing liquid to pass all the way through.

Looking at the results of the forest fire percolation, as exhibited by the top-down percolation, changing population size again has negligible effect on how likely a forest is to be destroyed. From examining a range of ‘forest density’ values, we have obtained a critical point around 0.58, around and above which a fire is likely to leave only 10% of the forest remaining. Our forests should be lower in density than this if we expect them to survive. Below this value, however, a fire is unlikely to spread very far, leaving most of the forest in good health.

2. Model Description

2.a. Model Explanation

We will be exploring a two-dimensional cellular automaton (2-D CA) as a model of percolation theory. A cellular automaton 'is a model of a world with very simple physics' [2]. A 'Cellular' world is split into discrete chunks or 'cells'. An 'automaton' is a computer used to perform simple calculations/simulations on these cells. The simulation begins with a single row of 'dry' cells and appears as a row of white squares. In the context of percolation theory, a dry cell is white, and a 'wet' cell is black. Each of these squares has a probability 'q' of being 'porous'. If a cell is porous, and the cell above it is 'wet' then on the next iteration, liquid will be able to drip down into the cell and 'wet' it, colouring it. Circling back to our initial description of percolation theory using a coffee filter, the simulation represents a blown up side-on view of the filter as coffee begins to pass through it. Paths of black cells will begin to appear through the 'filter' representing percolating clusters, where the coffee is able to pass through and drip back down into the liquid below.

A second simulation looks at a ‘forest fire’ or, for example, the spread of an aggressive infectious disease. Starting in the centre of the grid, the fire begins to spread to ‘forest’ or porous cells adjacent in all directions. Once a cell has been on fire for an iteration, the forest is burnt down and becomes an empty square (or after a person has been infected for a while, they die). The animation continues until the fire ‘goes out’.

2.b. Assumptions and Rules

Using a two-dimensional cellular automaton to model percolation, we have several rules and assumptions:

- each cell is either porous, with a hole in it, or non-porous, without a hole allowing liquid to travel through. Whether a cell is porous or not is calculated by the automaton in each iteration using a parameter 'q', a value between 0 and 1 describing the probability a cell is porous or not. a high 'q' value will result in a more porous overall material with more percolating clusters.
- each cell has a binary state 'dry' (0) or 'wet' (1). When the simulation starts, all cells are dry as liquid has not entered the 'filter'.
- Each iteration, if a dry cell has a wet 'neighbour' or adjacent porous cell, it will become wet.
- The simulation terminates when it reaches a state where no more cells are changing between wet and dry states.

A similar set of assumptions are used for the forest fire model:

- a cell with a forest begins to burn if any of its neighbours are on fire.
- a cell on fire becomes an empty cell in the next step.

2.c. Initial Configurations and Parameter Choices

The base model takes two parameters during input,

n – the number of cells in each row, or the width of the 'filter', a positive integer value.

q – the probability any cell in the model is porous, a floating-point value between 0 and 1.

The simulation starts by building a two-dimensional array of size n x n, where each cell has a chance q of being porous or not. q directly correlates with how porous a material is in the real world, a value closer to one yielding a very high percentage of wet cells on completion of the simulation. We will initially select a value of 10 for n and a q value of 0.65 for demonstration. In the basic model, the top row of the grid is 'wet' to begin.

For the forest model, we start with a slightly larger population n=35, with the same q value, which now represents 'forest density'. To start, the centre of the grid is set on fire. Green cells represent 'forest', orange cells represent 'fire' and white cells are empty.

3. Results

3.a. Visualisation and Discussion

We will refer to bolded section numbers corresponding to code in the jupyter notebook file. These figures will also be provided in the appendix with consistent numbering.

See **section 1.0** in the notebook. The code generates a 10x10 grid of square cells with porous ones in light blue, and non-porous cells in white. The grid begins to populate with wet cells from the top down, quickly finding a path to the lowest point, while filling neighbouring dry cells with liquid as it goes. Wet cells bordered by non-porous cells appear to reach a dead end, as they can't propagate through them.

See **section 1.1.** A larger configuration is now generated showing the emergence of patterns in the automaton. Our variable 'n' is increased from 10 to 50, keeping q constant at 0.65. We can see the emergence of more complex patterns and 'fractals' now, appearing as small groups of dry cells (many of which are porous) as they have been surrounded by non-porous cells.

See **section 2.0.** The textbook code for the top-down percolation has been rewritten to simulate a forest fire. Instead of propagating from top to bottom, the reaction starts in the centre of the grid and spreads in all directions, with the same principles as in the first model. See **section 2.1.** We generate a model to simulate a forest with a fire starting in the middle. The result is a very empty looking grid as most of the forest has been killed by the fire. Generating a larger model in **section 2.2** with the same q value yields a similar result, with more complex remaining forest configurations than in 2.1. Finally, we examine three simulations in **section 2.3**, each with a different q value, at three different stages. In the q=0.5 simulation, the fire goes out almost straight away. In the q=0.58 simulation, the fire spreads chaotic/unpredictably and has gone out by step 100. In the q=0.7 simulation, the fire spreads in an expanding circular shape, and has burnt through most of the forest by step 300.

3.b. Parameter Comparison and Analysis

For the base model, we can say the percolation is 'successful' if it has at least one path from top to bottom or 'percolating cluster'. We test a simulation for this with a function as in **section 3.0**, test_perc(). It returns true if there is a path from top to bottom, and false otherwise. It works by continuing a loop until no changes are made to any cell states. The following function find_critical() does a 'random walk' starting at an initial q value. A percolation object is constructed and checked using test_perc(). If it does, q is too high, and is reduced by 0.005. if it doesn't, q is too low, so we increase it by 0.005. Storing the q values as it goes, the result is a list of values that when averaged indicate an accurate estimate of the critical point for q, or the probability at which there is a 50 percent chance of the automaton having a percolating cluster. The obtained value for q is approximately 0.59. This indicates that if a filter is more porous than 0.59, it will almost always have a percolating cluster and if it is less, it will most likely not have a percolating cluster.

We then take this critical point value for 'q' and test it against a range of population (n) values in **section 3.1**. From the results, we can see there is no obvious correlation between population size and whether the automaton connects, so we can say the critical point 'q' appears to be consistent across all population sizes.

Now, looking at **section 4.0**, we examine the effect of the initial forest density 'q' on the density of surviving forest after the fire. The results are quite interesting, remaining approximately constant and healthy until q=0.58, after which density after fire drops drastically. From this we can determine there is a point where forest density becomes dangerously high. We can apply this to a real-world scenario, literally in the sense that we shouldn't let our forests grow over this density, and in a more abstract example, where the fire represents a pandemic, and the population represents human population density in a geographical area. We should either prevent living this densely or add extra protection for those who live in these highly populated areas to prevent deaths in the result of an aggressive epidemic.

Finally, in **section 4.1** we test the effect of varying population size (with constant density) on surviving forest density. We can see that there is no change for the linear values either side of the critical point q (around 0.58) for changing n values, and for the q=0.58 values, they are too chaotic to establish any real correlation. As a result, we can assume that the initial population's effect on the final population is independent on the size of the population.

4. Conclusion

After simulation and analysis of two types of percolation models, we have successfully simulated top-down percolation, giving us a model that can predict, for example, how groundwater will flow through rock and earth into aquifers. We have also simulated forest fire percolation, yielding a model that can be used to predict the spread of forest fires or infectious disease.

While both models are designed to represent their literal real-world applications, of course they lack complexities that exist in the real-world scenarios and are just approximations. We could work to improve the models by adding more variables, for example, in the context of the top-down percolation, having different porousness values in different areas of the grid would closer represent the different types of rock the water would reach as it continues its downward journey to an aquifer. In the context of the forest fire percolation, we could add a variable for likelihood of fire spreading to a particular forest cell, product of the type of forest, the type of weather and topography of the area. This would allow us to more accurately explore how the fire would travel, as it is more likely spread downhill, through drier forest, and spreads faster on hotter, windier days.

We have established a critical point for the 'porousness' in each of the models, indicating there is a particular density of porous cells that after which are highly likely to reach their respective forms of 'completion'. Designing a coffee filter, we would want the average porousness of the paper to be just above 0.59, as we want the coffee to be adequately filtered, but still able to pass through the filter comfortably. We would also want to prevent a forest growing above a density of 0.57, as the simulation has shown above this density, a fire begins to have catastrophic effects when it occurs. Overall, these percolation simulations are feasible applications of computational modelling, and have a range of applications in predicting and simulating real world complex systems.

5. References

[1] Beffara, V., Sidoravicius, V. (2006) *Percolation Theory*. Elsevier Ltd, 1(1)

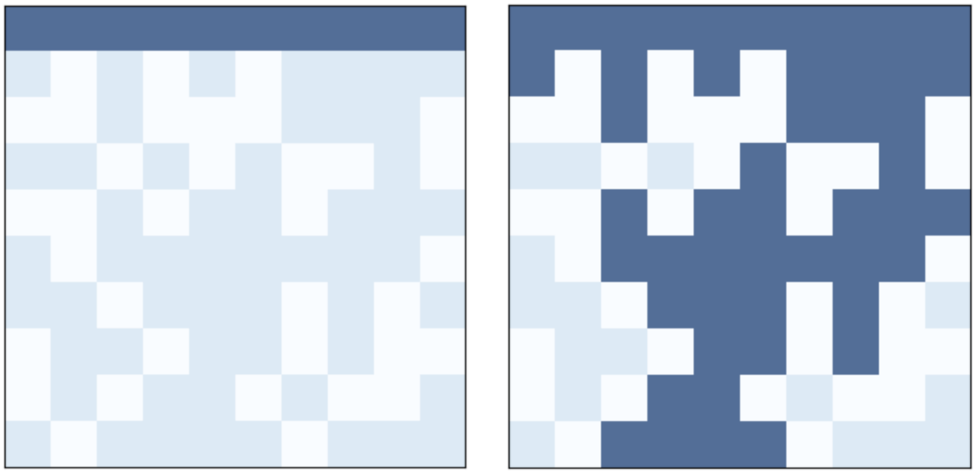
[2] Downey, A. (2016) *Think Complexity* (V2.6.3). Green Tea Press

[3] Browne, C. A., Amchin, D. B., Schneider, J., & Datta, S. S. (2021). *Infection Percolation: A Dynamic Network Model of Disease Spreading*. Frontiers in Physics, 1(1), Page numbers: 1-9

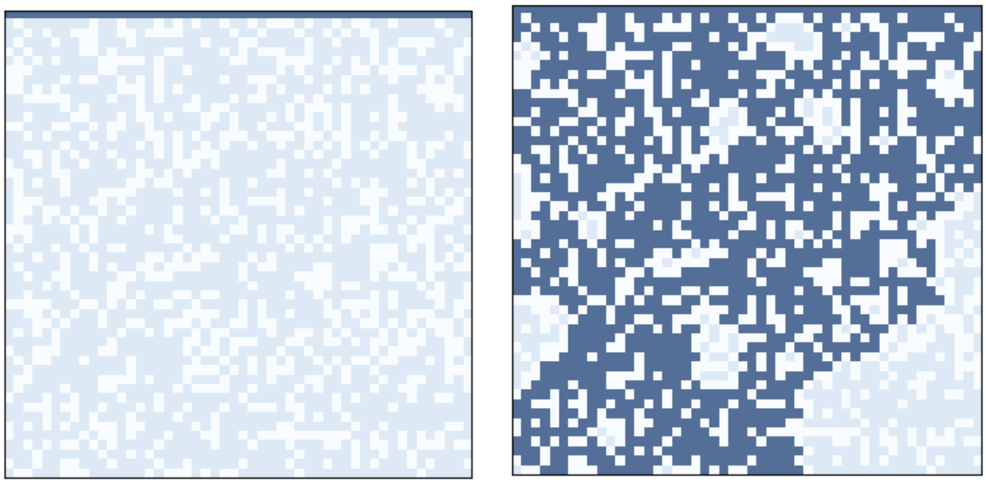
[4] Downey, A. (2021, Oct 6). *chap07.ipynb*. GitHub Repository. https://github.com/AllenDowney/ThinkComplexity2/blob/master/notebooks/chap07.ipynb

Appendix

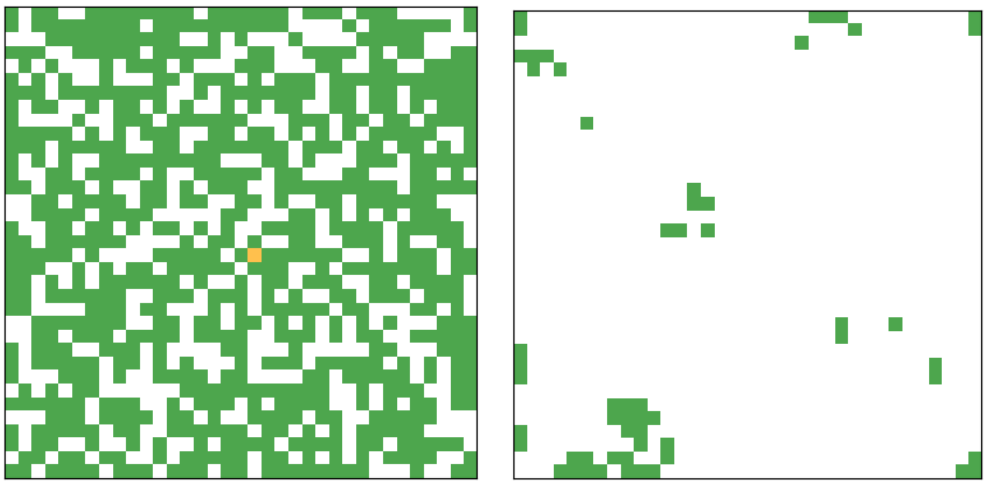
1.0 Basic Top-Down Before and After Simulation



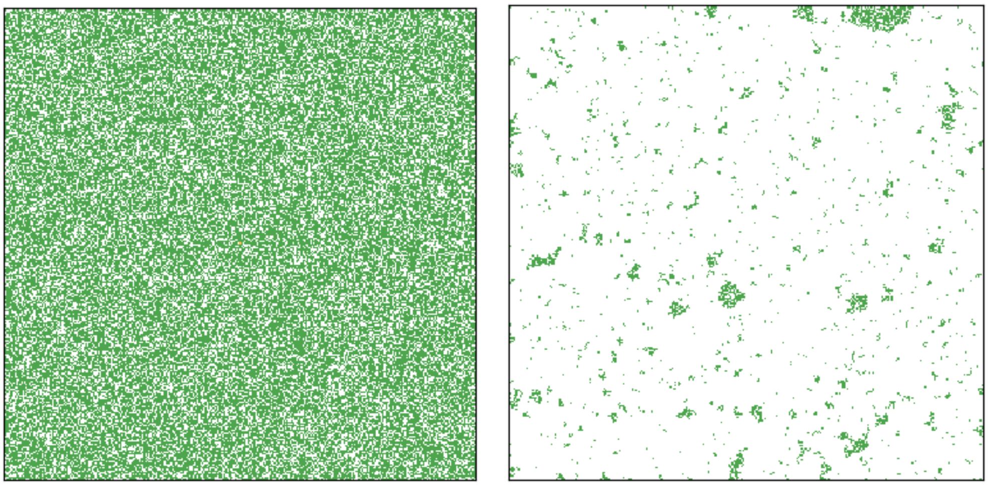
1.1 A larger configuration of the same simulation (increased n)



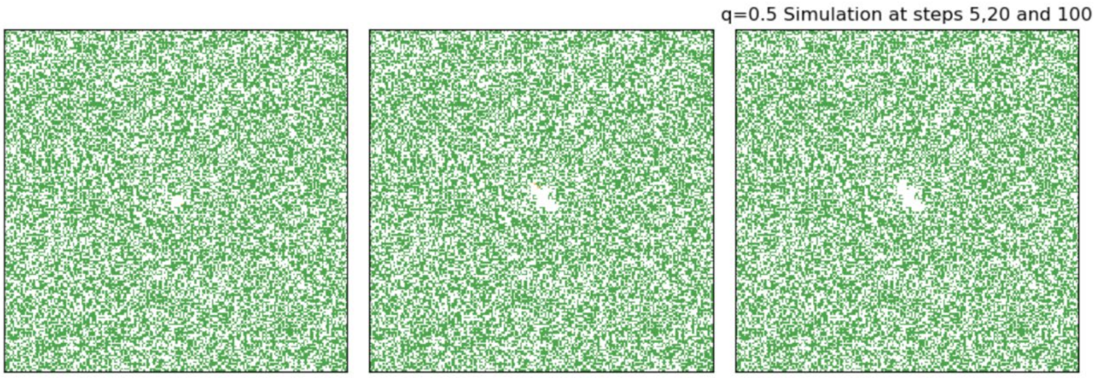
2.0 Basic Forest Fire Before and After Simulation

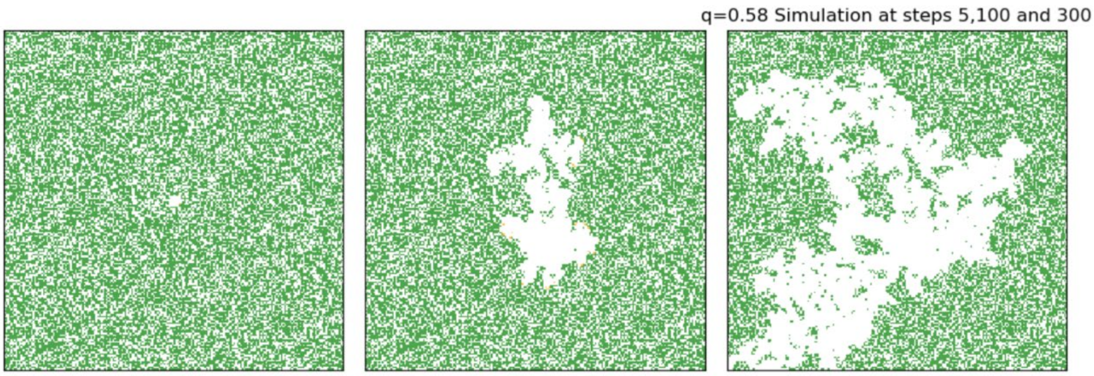


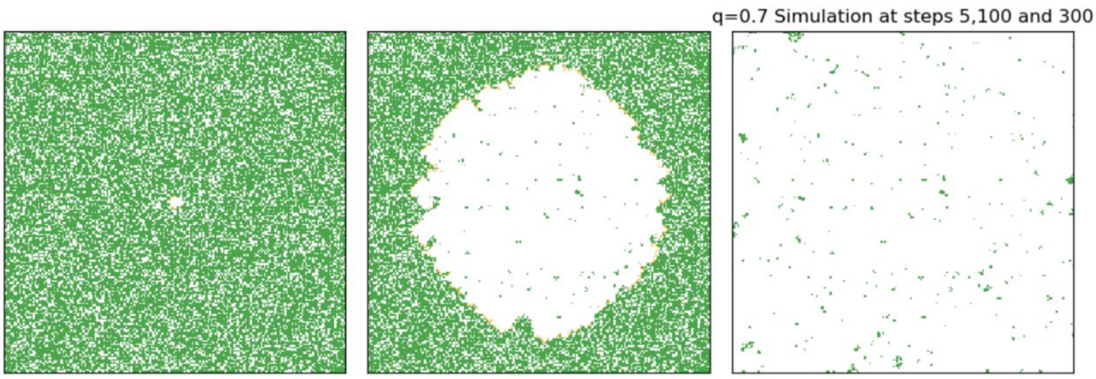
2.1 A Larger configuration of the same simulation (increased n)



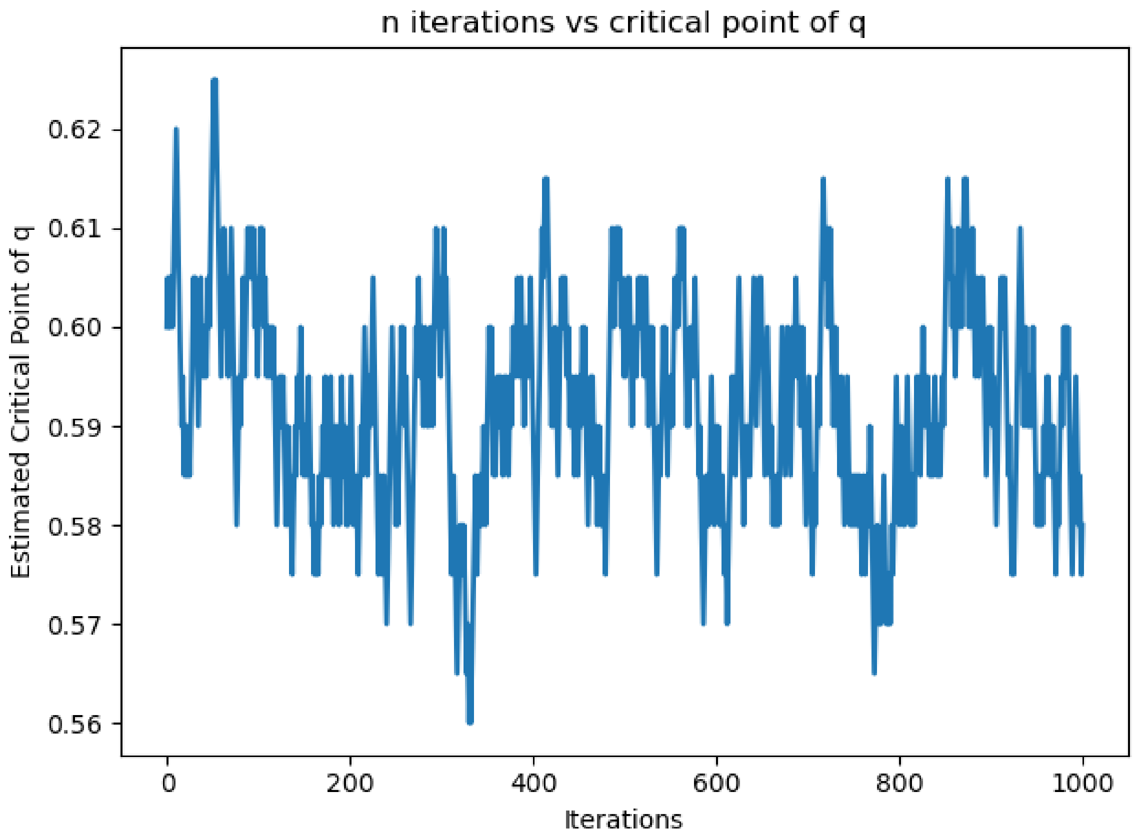
2.3 Comparing different q value simulations.



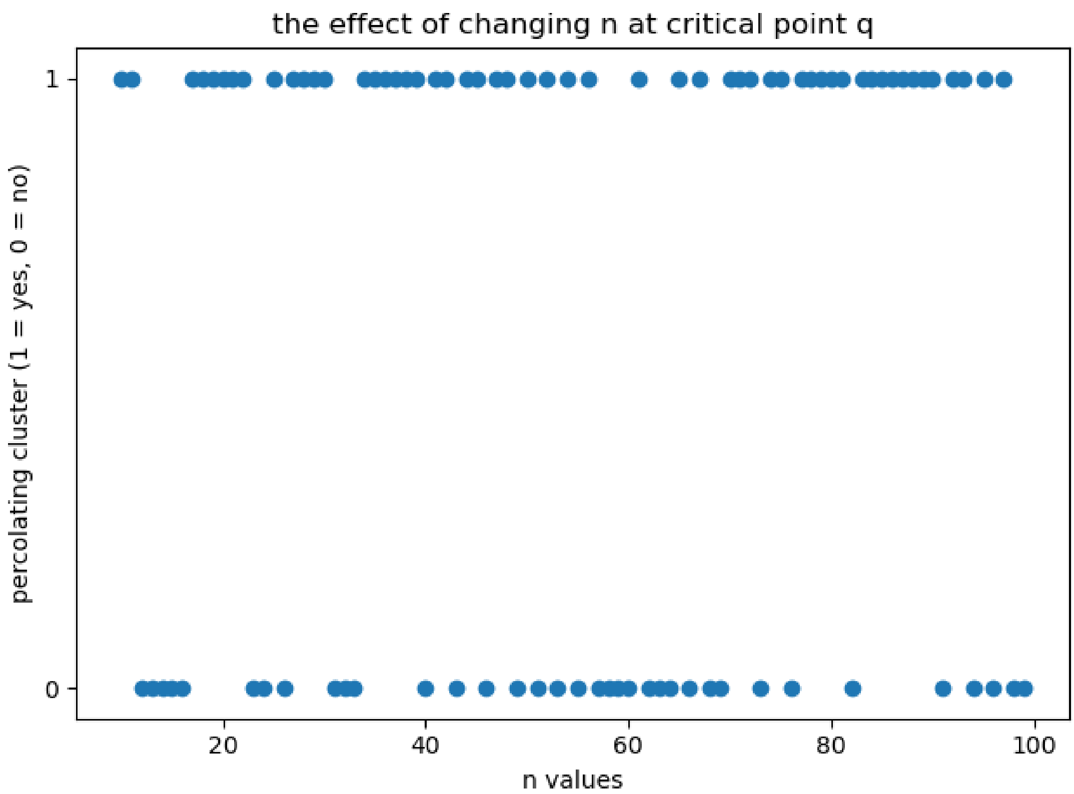




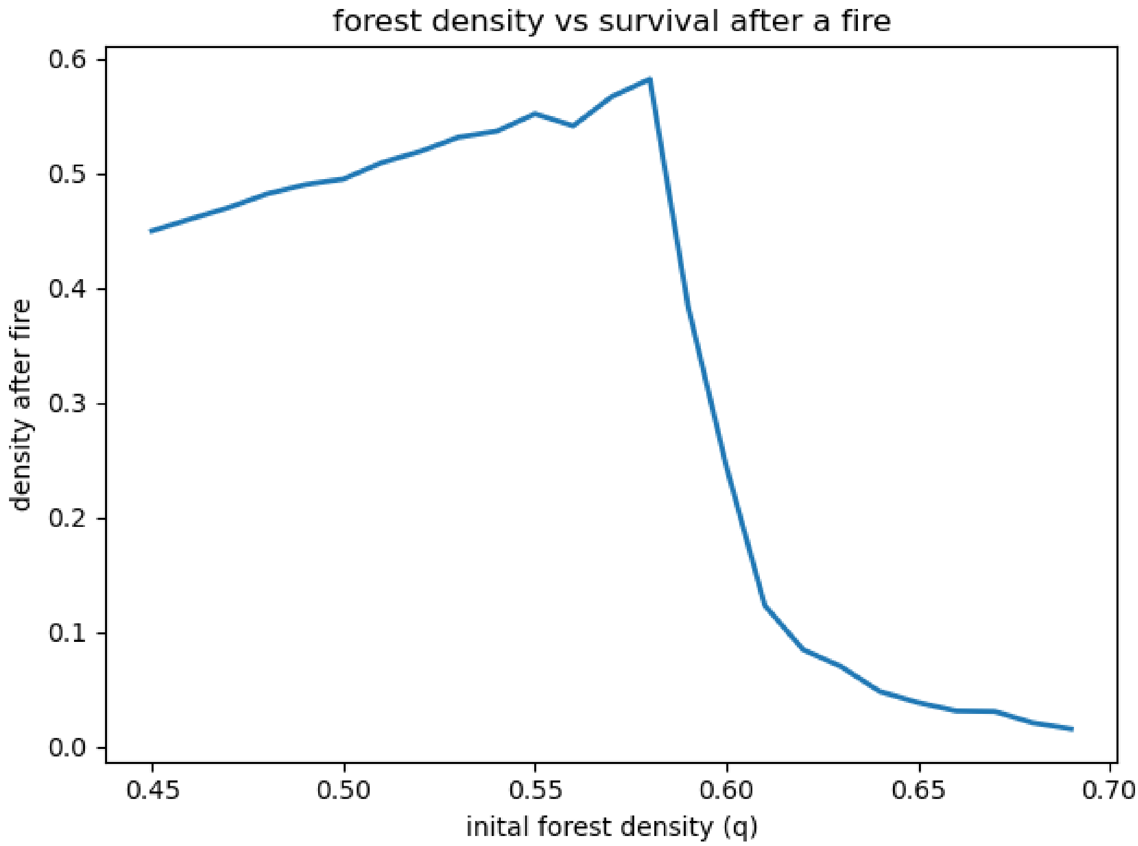
3.0 Trends and Patterns of the top-down model



3.1 Critical Point against changing population size



4.0 Trends and Patterns of the Forest Fire model



4.1 Critical Point against changing population size