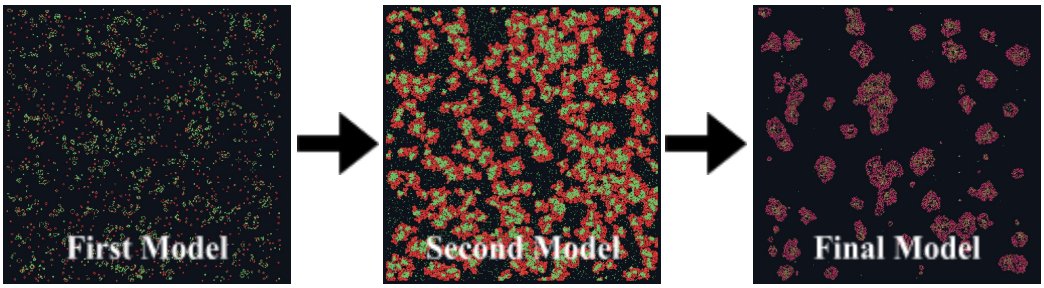
Yeon Jae (Skyler) Cho
Group 7
Randomness and Computation
May 11, 2022

Modeling Disease by Using a Cellular Automaton

In light of the pandemic, our group modeled the spread of disease using a cellular automaton, namely the Game of Life. A cellular automaton is basically a collection of colored cells on a grid that has a random shape that changes and evolves over time according to a set of rules based on the states of neighboring cells. John Conway's Game of Life is a discrete, deterministic game, meaning that if we start the game with the same board, cells on the same spot of the grid, then we should see the exact pattern every time. We attempted to explore a probabilistic cellular automaton model that achieves deterministic patterns from randomness.

To modify the Game of Life, we first set the cells to spawn randomly on the grid, then added another random number of cells, determined by a probability, that is considered "infected." These infected cells have a certain probability of infecting a healthy cell. However, our first working model did not achieve our goal of modeling a real-life pandemic, for following the original rulesets of the Game of Life caused the healthy cells to die off way too quickly. We couldn't observe the cornerstone of infection: interaction among cells. The healthy cells spawned at the beginning of the game were almost all gone by the time the infected cells started to grow, and so the infected cells, following the rules of the Game of Life, couldn't move or change their shape.

To increase the life expectancy of the healthy cells, we modified the original rulesets of the Game of Life. We edited the rules so that the cells don't die, and the infected cells will become healthy cells again after some time (after a few evolutions), which corresponds to people recovering from a disease. To differentiate the cells, we colored the healthy cells green and the infected cells red; being infected means turning from green to red and recovering means turning from red to green. This modification allowed small clusters of cells to grow and cover the entire board, meaning that we could see interactions between cells much more clearly. To add more complexity to the growth pattern, we added a fourth state to the game, which creates cells that lie over the population, like a cloud of infected cells that covers the susceptible population. This fourth state is the "spread" state, meaning that if a cell in the spread state interacts with a healthy cell, it becomes infected. We colored the cells in the spread state yellow. To track the state of each cell over time, we assigned all healthy cells the value "1" and the infected cells the value "0".



After creating a working model, we analyzed the change in the population concernin each cell's state over time using the SIR model. According to mathworld.wolfram.com, an SIR model is an "epidemiological model that computes the theoretical number of people infected with a contagious illness in a closed population over time." As the name suggests, it includes 3 equations that model different things: the number of susceptible people, *S(t)*, the number of people infected, *I(t)*, and the number of people who have recovered, *R(t)*. By plotting graphs out of the population of each state–S, I, and R–we were able to produce graphs that look very similar to the actual SIR model graphs.

$$R_0 = \frac{\beta}{\gamma} \qquad \frac{dI}{dt} = \left(R_0 \frac{S}{N} - 1\right)\gamma I,$$

In the actual model and the equations used by the model, increasing the probability of spawning cells that are in the spread state, **r**, increases **β**, the number of contacts per person per time, thus also increasing the basic reproduction number $\mathbf{R_0}$, which is also the expected value of new infections. This again leads to an increase in the number of the infected population over time, denoted as $\frac{dI}{dt}$. From the SIR models we created, one can see that as r increases, the graph of I, the number of people infected, shifts leftwards and gets steeper because of the increase in $\frac{dI}{dt}$. Thus we successfully created a probabilistic cellular automaton that simulates the spread of disease that also follows the official SIR model.

My role in this project: After Hankai created a working model, I received from him the data that contained the number of population of each state over time and created various SIR graphs. In the first spreadsheet, the graphs I created resemble the SIR model, and from the equations given by the spreadsheet, I could find out the average equation for each variable, S, I, and R, and also the actual data's correlation to the auto-created equation, which is denoted as R^2. In the second spreadsheet, the graphs I created show the effect of increasing r, the probability of spawning cells that are in the spread state, especially on the graph of I, the number of people infected.

Furthermore, although we didn't present it in class, I tried to create a binary version of Hankai's model by using Python, where alive cells are shown as 0s and dead cells are shown as a dot. To do that, I first tried to run the original Game of Life in a loop in the same binary settings and make it count how many generations it takes to reach a stable, non-changing state, but I couldn't figure out how to compare the current state of the cell in each position of the grid to that of the previous generation. Thus, we ended up only using Hankai's model. I'll still attach the code I was working on, which is mostly from Martin Andersson Aaberge, on Gradescope.

Works Cited

https://betterprogramming.pub/how-to-write-conwells-game-of-life-in-python-c6eca19c4676

https://en.wikipedia.org/wiki/Compartmental_models_in_epidemiology

https://mathworld.wolfram.com/SIRModel.html