

Math 99 Summary

Our project focuses on the variation of Conway's Game Of Life (hereinafter to be referred as Game). It is a cellular automation that can be a simulation of a certain community or ecosystem. Specifically, we tried to use the Game model to simulate the predator-prey model with two or three species.

First, we implemented the basic Game with only one species. Under the direction of Anton Bobkov, we implemented the Game on C++ and visualized it by using the SDL2.0 Library. We use a two-dimensional array to store all the information of the cells on the plane, including its current living state, the position of its neighbors and so on. We constructed an interface with a rectangle board representing the ecosystem, and several functional buttons such as pause, resume, restart, clear the board, and quit (and these could also be done by the pressing on the keyboard). The Game can run automatically because we use the SDL Timer; when the user pauses the game, he can change the living state of an individual cell by mouse click. We also construct other windows to update the time-population graph and phase graph (for multiple species). To accomplish all the functions mentioned above, we learned the SDL Poll Event function and function pointers.

In addition, we implemented some variations of the Game. The most common one is named S23/B3 (Survive only if 2 or 3 neighbors, born only if 3 neighbors). We tried S23/B36 (highlife) and S1357/B1357 and S2468/B2468 (replicators).

For the predator-prey model with two species, we tried many different life rules. The basic logic behind the scene is that a predator will die if there are not enough preys around it; a prey will die if it is alone (since herbivores usually live in group) or there is any predator around it; the preys are very productive while the predators will be able to give birth only if there is one additional unit of food around them. When we plot the time-population graph of both species, it turns out that if both of them could survive, the populations of both of them will be oscillating within a certain range with the population of predators trailing that of prey by 90° in the cycle.

This reminds us of the Lotka-Volterra equations ($dx/dt = \alpha x - \beta xy$, $dy/dt = \delta xy - \gamma y$). We guess that some relations between our simulation and the Lotka-Volterra equations could be found by figuring out the parameters. We took Anton Bobkov's

Guoan Wang 004268976

Xi Lin 404326451

suggestion that once we save all the population data in the simulation, we can do linear regression to get the value of these parameters. After we do linear regressions on x , xy , and y , the parameters turn out to be statistically significant (with p value less than 0.001). However, these parameters will vary with different initial configuration; in other words, each time we start a new simulation, the parameters will also be different. Furthermore, the adjusted R-square value is only about 0.6. In fact, if we do linear regression on more variables such as x^2 , y^2 , x^2y , xy^2 , the adjusted R-square will be much larger. Therefore, although the population of both species can explain our model in some extent, the influence from the actual configuration is not negligible.

Furthermore, we expand our model to the three-species model. We have two variations of that. The first one is the “rock-paper-scissors” one, in which $A \rightarrow B \rightarrow C \rightarrow A$ (the arrow represents energy flow, e.g. $x \rightarrow y$ means x is eaten by y). This model also turns out to be stable: after a period of time, the population of the three species will become periodic. From the ecosystem board we can observe a infinite spiral loop of three colors. The second one is an extension of the two-species model, in which $C \rightarrow B \rightarrow A$ (i.e. A is the dominant species, B is the middle one, and C is the bottom one). We modified the life rules by adding a factor that represents an individual’s tolerance to danger and starvation. This time, we plot the time graph a little differently: we plot A and B the same as before, but we reflect the population of C over the x -axis (as if we are plotting the negative value of the population C). The three lines look pretty much alike in shape, and one is imitating the behavior of another.