Simulation of the Spread of a Virus Using Agent Based Modeling

Matt Wade

June 10, 2008

Abstract

My goal is to make an agent based modeling simulation that shows the spread of a cold through a school. It will start with an amount of infected students and healthy students received as inputs and will show how much the virus spreads or possibly recedes over time.

1 Introduction

My program will answer the question as to how quickly and fully different types of sicknesses will be able to spread through the population of a school once introduced. This will show how likely it is for a disease to be spread by a set amount of sick people coming to school with the sickness.

I expect my program to show how many people are infected with a virus after any amount of time given an input of how many people there are and how many are infected at the start. The model will display the number of infected people and the period that the agents are in at the moment in the display window plus possibly some other information such as the ratio of infected to healthy people, the total number of people that have been infected (including those that have gotten healthy after being infected), and other similar results.

The finished the program will display values for the number of agents are sick, healthy, total agents, total infections, and total recoveries. The simulation will also have a display showing the locations of all of the agents and their status (healthy or sick). Eventually it will be able to have different diseases which have different recovery times and infection rates. I will use research data for all of the stats relating to the different diseases such as the recovery time and the infection rates based on real values for different diseases.

2 Background

I have been researching different examples of agent based modeling using MASON through the MASON website and the book Growing Artificial Societies which specifically discusses the Sugarscape model. A model similar to mine is part of the demo models for MASON which describes the spread of a virus through a workplace. In this example the agents become immune to the virus once they become healthy, they stay sick for a set amount of time, and move to work and come back home without moving to other places. They are infected through close contact with sick agents just like in my program and this program only has one type of disease unlike mine will when finished. I will attempt to make my model more realistic in most of these areas by using actual data. There are also many papers documenting the usefulness of computer models and simulations in science such as Tutorial on Agent-Based Modeling and Simulation by Cayzer and Sullivan, and Modelling Danger and Anergy in Artificial Immune Systems by Macal and North, which is another paper documenting the use of a computer model in analyzing the human immune system. **References:** Macal, C., and North, M. (2005). Tutorial on Agent-Based Modeling and Simulation; Cayzer, S., and Sullivan, J. Modelling Danger and Anergy in Artificial Immune Systems.; Macal, C., and North, M. (2005). Tutorial on Agent-Based Modeling and Simulation Part 2: How to Model with Agents.; Epstein, J. (1996). Growing Artificial Societies: Social Science from the Bottom Up. Brookings Institution Press.

3 Development

My program has three main classes. An Agent class which defines what values an Agent will store and how to construct it. A class that creates the GUI. And the most complicated of the three, the Model class which is where all of the calculations occur. In the Model class there is an arraylist containing all of the Agents in the simulation. With these Agents the class has to define a step() function which moves the simulation forward. This function has to update the locations of all of the agents, check to see if any of them get infected, and check to see if any recover from being sick. First it goes through the list of Agents and moves them all to the next location in their schedule. In order to check for any new infections it goes through the entire list of Agents finding each sick Agent. Whenever it finds a sick Agent it finds any healthy Agents in the same location and checks a randomly generated number against the sick Agent's infectiousness value. If the random number is lower then the healthy Agent is switched to sick and the method continues on through the rest of the list. It also does one additional check of a random number against the infection chance in order to compensate for the possibility of getting sick through outside influences when traveling from one location to the next. To check if any Agents recover from sickness it goes through the list checking the recoverytime value and if it equals zero the Agent is switched to healthy.

The GUI class contains another important function, the ability to graph data gathered through the simulation. In the Model class I added new arraylists to retain information over time for all of the main variables (number of sick agents, number of healthy agents, infections per step, etc.). Whenever the graphing method is called the Model class gets information from two drop down menus as to which arraylists are going to be the x and y axis variables and sends them to the graph class. The graph class then takes these variables and goes through the arraylist graphing a scatterplot of the data. It scales the graph by finding the maximum value in the arraylist of data for each axis and divides 700 for the x axis or 500 for the y axis by that max. This spreads the data points out along the entire axis rather than having it take up a very small part of the graph, or not fitting in the graph. There is still a problem which I have not been able to fix in the scaling of the y axis as it often scales the graph correctly, but uses unequal intervals when labeling the graph. This makes it more difficult to read the graph, because the labels are usually grouped in a small area of the middle of the axis so you have to estimate values near the top or bottom of the axis.To obtain a conclusion from the program, I ran the simulation through multiple trials each with the same inputs. I made there be 1999 healthy agents and 1 sick agent, spread across 80 classes. This meant there was an average of 25 agents in each class which is normal for a school.

4 Results

Using the infection percentage value found from research, the simulation ran with zero other agents becoming infected each time. This seems odd, because if none of the many people that the sick agent came in contact with became sick in a any trials, the illness would not ever spread to anyone and would die out. This situation has not happened in real life because the flu is still a concern. I think this is due to having such a small number of agents. The data I used was for the percentage of the population of the world that becomes sick with the flu each year. This value includes all the people who do not come into contact with an infected person, which is likely a much larger amount of people than the percentage of agents in my simulation that are in contact with the sick agent. Since the agents are in a more concentrated area than the people in the research data, this results in the infection chance being lower than it actually should be.

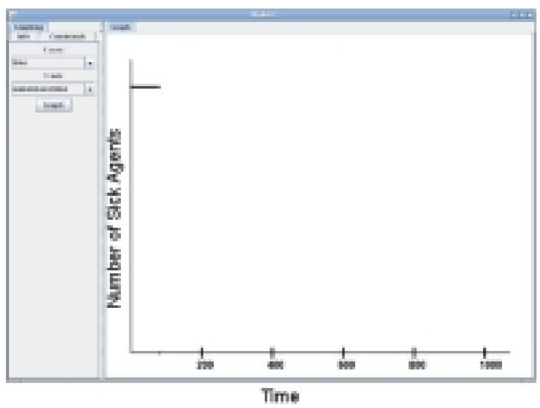

Figure 1: Results using research value.

To counteract the above effect I tried using other values for the infection chance by multiplying the value by two, five, and ten, but each time this resulted in too many agents being infected. Multiplying by five and ten resulted in unrealistically large numbers of infections with the simulation balancing out at more than 50 percent of agents being infected. Multiplying by two was slightly better usually resulting in around 100 sick agents. Going strictly by the simulation with the research data, I can conclude that if you are sick with the flu there is no reason that you should avoid going to school or work unless you are physically unable to, since it is unlikely that you will spread the flu around.

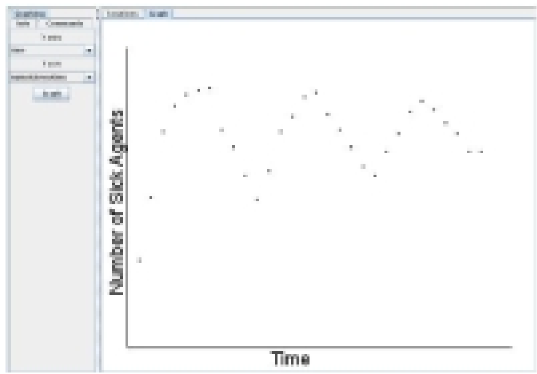

Figure 2: Results when value is multiplied by two.

If further work were to be done on this or a similar project the most important addition to make would be to find research data that better fits the simulation. The project could also be furthered by optimizing the simulation so that it can run quickly with a greater number of agents, because at the moment even simulating 2000 agents for 1000 steps takes around a minute to complete. This would help to confirm or refute the results found using this simulation, and would tell if using data from such a large sample group gives realistic results in a simulation of a much smaller group.

References

[1] Macal, C., and North, M. (2005). Tutorial on Agent-Based Modeling and Simulation

[2] Cayzer, S., and Sullivan, J. Modelling Danger and Anergy in Artificial Immune Systems.

[3] Macal, C., and North, M. (2005). Tutorial on Agent-Based Modeling and Simulation Part 2: How to Model with Agents.

[4] Epstein, J. (1996). Growing Artificial Societies: Social Science from the Bottom Up. Brookings Institution Press.

[5] Leblebicioglu H. (2007). Influenza. Emedicine.

[6] Mather D., and Crofts N. (1998). A Computer Model of the Spread of Hepatitis C Virus Among Injecting Drug Users. Kluwer Academic Publishers.