

# Model of a Virus Outbreak on a Contained Population

CJ Cahill

COMP SYS LAB

2008-2009

## Abstract

With the rapidly increasing population of the world, it is getting more and more likely that a virus or bacterium will cause an epidemic. This is a major problem because if an epidemic were to occur, then possibly the majority of the world's population could be looking at death or life-altering events. Similar things have happened in the past, such as the Spanish Flu and the Bubonic Plague. That is where this is where the simulation comes in. It will be able to predict how a virus will spread through a population and will also be able to predict the number of deaths due to the virus. From this data we be able to correctly counteract the outbreak to the best of our ability.

## Method

The first major part of the project is determining the variable values for the simulation. These values form a unique set for each virus. It is practically impossible to determine these values without any data from either a previous outbreak or just random infections. In order to be able to determine the values without an outbreak one would need specific information concerning how the virus infects the body and from that create a separate simulation made just to extrapolate these values.

After determining these values the next step is just to input all of them into the program and run it. After the program concludes all the data is outputted to a text file which is then read by a Matlab script that produces an easy-to-read graph. This graph shows the total healthy, infected, immune, and dead people in the simulation. However, there are a few factors involved that are unrelated to the virus itself. The population has a maximum capacity, which is settable by a variable at the start of the program. This population also has an age limit, and when an agent reaches this age they immediately die. There's also a birth rate, and that goes through all the empty spaces (created by deaths) and decides whether to put in a new baby based on a random number generator.

## RESULTS

The results from the most recent versions of the program show that it doesn't model a real world scenario very well. The only way to make the program correctly show the endgame of the scenario is to artificially MAKE it do so, and that's pretty much cheating. Another way to make the program more accurate would be to add in all the real life variables that this program does not have the capacity to add. So because of that the program was unable to accurately simulate the distribution and result of an outbreak of a virus on a real world population. It does work for this simplified world without all the actions people take such as going to hospitals and going to schools and any government reactions to the outbreak, but that is far from what the real world is like, and so that makes this program rather unnecessary in determining what actions to take against a virus.

## Evolution

It started out as just an array of cells and infected cells have a chance of infecting their neighbors. The second version of the program incorporated immunity and rebirth. It was still primitive in the sense that it still just used one array to represent a group of people. The change from cells to people didn't change any part of the program besides the name of the variable, but it did make it seem more true. Cells cannot recover from a virus infection because they explode when the virus is done with them. People, however, can recover from a virus. Eventually the data structure of the program wasn't able to cope with the size of the population and took way to long to finish simulation than would be desired. Because of this a third generation program was constructed from scratch using ArrayLists to improve efficiency and variability. This new program was able to take on larger populations without slowing too much, and also opened the door for additional variables. The birth function was corrected, and now the system reaches a state of equilibrium.

