Spring 5-12-2022

Large Scale Disease Modeling

Walker Mattox

LARGE SCALE DISEASE MODELING

by

Walker Margaret Mattox

A Thesis
Submitted to the Graduate School,
the College of Arts and Sciences
and the School of Mathematics and Natural Sciences
of The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Mathematics

Approved by:

Dr. James Lambers, Committee Chair
Dr. Haiyan Tian
Dr. Huiqing Zhu

May 2022

ABSTRACT

In this we study large scale disease modeling. After understanding the mechanics behind the *SIR* disease model in an ODE sense, we will apply this knowledge to model disease spread in more and more increasing advanced cellular automata. Eventually, some of our cellular automata will include long distance travel. From this discrete data, we can then build an *SIR* model in the *PDE* sense to display large scale disease spread.

ACKNOWLEDGMENTS

This is to thank my research advisor, Dr. James Lambers. I would be nowhere without his assistance, and I am so thankful for how he has helped me grow in this process. I would also like to thank the rest of my committee, Dr. Tian and Dr. Zhu for all they dedicated to listening to my presentations and reading my thesiss as well as for the input on improvements I could make. Finally I would like to thank all of my classmates and other professors who supported me along the way.

TABLE OF CONTENTS

Chapter 1

BACKGROUND

1.1 Inspiration

We all know 2020 was a tough year with the introduction the epidemic of COVID-19 taking over the world. It has kept its grasp on us even up to today, and probably will never fully go away. For me personally, I came into graduate school in peak pandemic mode. Unfortunately, this meant not meeting most of my professors and classmates in person very often. However, I wanted to make the most of my experience at The University of Southern Mississippi.

Thanks to my wonderful advisor, Dr. James Lambers, I have had the opportunity to have to research COVID-19 in a mathematical setting under one of the best mathematicians I know personally. This has been the silver lining in the whole experience. Having the opportunity to study this pandemic from the lens of the subject I love most has helped me develop the skills to be a better mathematician all while understanding large scale pandemics further, such as COVID-19.

1.2 Introduction

In this thesis, the overarching goal is to take SIR models that predict the spread of infectious diseases from ODEs modeling local spread to PDEs modeling the spread over a larger area. Of course, this means we must first understand mathematical modeling for disease spread in time using ODEs.

We begin by introducing the Kermack-McKendrick *SIR* model. It is one of the most simple models to begin understanding disease spread. We note that S stands for susceptible individuals, I stands for infected individuals and R stands for removed individuals. It should be noted that removed individuals in the case of this model could be recovered or deceased. Finally, the combination S, I and R at any given time should add up to the total population, N. Figure 1.1 a visual of an *SIR* model.

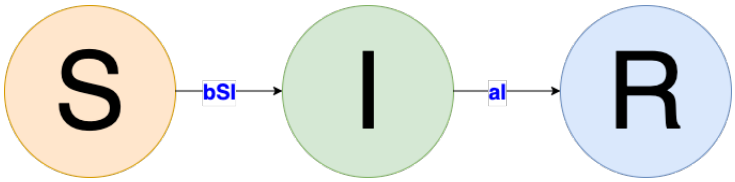


Figure 1.1: SIR Model

Each bubble is just a pool of people in different states of the disease. Between each bubble are what we call transition rates, meaning, people are moving from one state to another. So a susceptible person may be becoming sick or a sick person removed. We will go into more details on the mathematical interpretation of the transition rates in the next chapter. The ODEs that represent our *SIR* model are

$$\begin{aligned} S' &= -bSI \\ I' &= bSI - aI \\ R' &= aI. \end{aligned}$$

Figure 1.2: SIR ODEs

Without even understanding the mathematical interpretation of the transition rates the reader can observe a few things. The first is that adding $S' + I' + R'$ together gives us zero. That is, the population is never changing. We note for this model, once an individual is recovered or deceased, he or she is never susceptible again. Also, S' is negative because as a disease spreads over time, more people get sick, meaning we only lose susceptible people. These once susceptible individuals now pour into the infected category, hence bSI now being positive in the I' equation. But we can see $-aI$ is negative. This is because we are losing infected people due to individuals either recovering or dying. Finally, we have that R' is positive, because it takes in all once infected people into the removed category. We conclude that bSI is some sort of infection rate and aI is some sort of recovery rate.

Now that we know the surface level of what an *SIR* model is, we lay out the course of the thesis from here. In Chapter two, we will go into further depth of the one dimensional *SIR* model. After this, in chapter three, we will show how we can apply this model to build different codes that display disease spread in a discrete setting. Each code is what we call Cellular automaton (CA), which we will explain the meaning of at the introduction of

chapter three.

Our first code (or CA) will model disease spread from on person to another in a small setting (such as a classroom). Eventually, we will increasingly more advanced codes that display disease spread over an entire region (such as a city). Our final code will incorporate long-distance travel. Finally, we will use this discrete data from out last code to build PDEs representing the spread of disease over a large region.

Chapter 2

SIR Models

2.1 One Dimensional SIR Model Terminology

Recall that S is stands for susceptible, I for infected and R for removed people in our model. Also recall $S+I+R=N$ is the total population of the observed area at any given time. Now we dive into the meaning of our transition rates we mentioned earlier in Figure 1.1. Recall, we call bSI and aI the transition rates. Note that b and a are constants representing the infection rate and recovery rate, respectively. The infection rate consists of the transmission probability of the disease being transmitted as well as the contact rate (which comes from having contact with sick individuals). That is, $b=pq$ where p is the contact rate and q is the transmission probability of a disease. The recovery rate is simply the probability of recovery over n days. That is, $a=\frac{p_n}{n}$ where p_n is the probability of recovery in n days.

2.2 One Dimensional SIR Model Examples

To further understand SIR models, we look at two kinds of examples. We will start with a numerical example. Suppose we have some sort of disease spreading such that at the beginning of the disease (so time $t=0$) we have the initial amount of susceptible individuals, $S_0=20,000$, the initial amount of infected individuals, $I_0=1,500$ and the initial amount of recovered or deceased people, $R_0=2,000$. Then we find that the initial transition rates are

$$\begin{aligned} S_0' &= -bS_0I_0 = -3,000 \\ I_0' &= bS_0I_0 - aI_0 \approx 2,785.7 \\ R_0' &= aI_0 \approx 214.3. \end{aligned}$$

Thus after one day, we have

$$\begin{aligned} S_1 &= S_0 + S_0' \approx 17,000 \\ I_1 &= I_0 + I_0' \approx 4,285.7 \\ R_1 &= R_0 + R_0' \approx 2214.4. \end{aligned}$$

Now we can find the transition rates after one day which are

$$\begin{aligned} S_1' &= -bS_1I_1 = -7,285.7 \\ I_1' &= bS_1I_1 - aI_1 \approx 6673.4 \\ R_1' &= aI_1 \approx 612.2. \end{aligned}$$

Hence after two days, we have that,

$$\begin{aligned} S_2 &= S_1 + S_1' \approx 9,714.3 \\ I_2 &= I_1 + I_1' \approx 10,959.1 \\ R_2 &= S_1 + R_1' \approx 2826.6. \end{aligned}$$

Of course, there is no such thing as 'part' of a person, but these results are an approximation of how the disease spreads. We can continue this pattern until we see some sort of convergence, meaning the disease has died out. Now we look into an analytical example. First, we notice that it is possible to write I' in such a manner that it is easy to analyze the behavior of an epidemic. Rewriting yields

$$\begin{aligned} I' &= bSI - aI \\ &= (bS - a)I \\ &= a\left(\frac{b}{a}S - 1\right)I. \end{aligned}$$

It is now clear that if $S > \frac{a}{b}$, then $I' > 0$, meaning the epidemic will grow. Similarly, if $S < \frac{a}{b}$, then $I' < 0$, meaning the epidemic will shrink. We call $\frac{a}{b}$ the threshold.

2.3 Going from a Discrete to a Continuous Model in Time

Recall from earlier that a is the recovery coefficient and $a = \frac{p_n}{n}$, which is the probability of recovering in n days over n days. It is important to note that a is independent of the time unit chosen, but the probability p_n is dependent on the time chosen. Since the rate is independent of the chosen time, we can do some generalizing. If we allow Δt to be a general unit of time and $p_{\Delta t}$ to be the probability of recovering in Δt, then the number of infectives after one unit of time can be given as

$$I(t + \Delta t) = I(t) - p_{\Delta t} I(t).$$

Re-arranging and dividing by Δt on both sides of the equation yields

$$\frac{I(t + \Delta t) - I(t)}{\Delta t} = -aI(t).$$

where $a = \frac{p_{\Delta t}}{\Delta t}$. Since a is constant for all Δt, we can take the limit as $\Delta t \to 0$. Thus

$$\begin{aligned} \lim_{\Delta t \to 0} \frac{I(t+\Delta t) - I(t)}{\Delta t} &= \lim_{\Delta t \to 0} -aI(t) \\ \implies I'(t) &= -aI(t). \end{aligned}$$

To solve for the number of infected individuals at time t using a discrete time model, consider $I(t+\Delta t) = I(t) - p_{\Delta t} I(t)$ with $\Delta t = 1$. Then we have that

$$I_{n+1} = I_n(1 - p_1).$$

Then using a non-recursive formula we get that

$$I_n = I_0(1 - p_1)^n.$$

To solve for the number of infected individuals at time t using a continuous time model, consider $I'(t) = -aI(t)$. Then integrating both sides of the equation with repspect to time yields

$$I(t) = ke^{-at}.$$

With the initial conditions that $k = I(0)$ and $t = 0$, we have that

$$I(t) = I_0 e^{-at}.$$

In essence, a discrete model uses probabilities and results in a difference equation (an actual population size). A continuous model uses a rate and results in a differential equation.

Chapter 3

Cellular Automata

3.1 Using Cellular Automata to Demonstrate Disease Spread

Cellular Automata (or CA) are are very effective at demonstrating disease spread in a discrete sense. We begin this chapter by introducing some key terminology. A cellular automaton (singular) is a collection of “colored" cells on a grid of a specified shape that evolves through a number of discrete time steps according to a set of rules based on the state of the neighboring cells. The “color" (can be a symbol, letter, color, etc...) gives each cell in the CA a distinct state (such as susceptible, infected, or removed). Each cell has its own neighborhood. Two common neighborhoods used in a CA are a Moore Neighborhood and a Von Neumann Neighborhood. A Moore Neighborhood is “square shaped" meaning we consider how the north, south, east, west, northeast, northwest, southeast and southwest neighbors of each cell affect one another with respect to each cell. In a Von Nuemann Neighborhood, you only consider how the north, south, east and west neighbors affect one another with respect to each cell. Finally, a CA has a set of rules, which are just the laws that each cell of a CA abide by. That is, the rules of the CA determine how a cell changes state.

3.2 Application of a Cellular Automaton in MATLAB

3.2.1 When a Matrix Cell is an Individual Person

While a CA can be programmed in whatever language a user is comfortable with, we chose to use MATLAB. Below will be pictured our function code as well as some simulations to show how disease would spread with our CA depending on the parameter values chosen. With each iteration, we will have a change of the transmission probability, number of days, or both. First, here is the code for the function.

```
function A=cellularautomaton(numdays,tp,numrows,numcols,maxinfperiod,...
    immunity,deathprob)
%function that takes user inputs of number of iterations (numdays),
%transmission probability (tp), number of rows(numrows) and number of
%cols(numcols).Also included in input is maxinfection period which would
```

```
%be the time an infected individual should quarantine away from the rest
%of the population. A is an output matrix where 1 is infected,and 0 is
%uninfected. The function itself is a cellular automaton.
B=[1];
%a 1x1 matrix with B(1,1)=1
C=zeros(1,numcols-1);
%a row vector of zeros
D=[B C];
%concatenating B and D together vertically so I can get a row vector of
%zeros other than first entry
E=zeros(numrows-1,numcols);
%a mxn matrix of zeros
A=[ D; E ];
%Concatenating D and E together horizontally so I can get a mxn matrix
%of zeros other than entry A(1,1)=1.  The first entry is the infected
%individual, and the rest of the entries are uninfected to start.
%randprob is a random probability.  Will be between 0 and 1.
A
pause
for k=1:numdays
    %counting through iterations
    A1=A;
    for i=1:numrows
        for j=1:numcols
            %going through all rows and columns of matrix
            neighborcount=0;
            randprob=1*rand();
            if i>1
                neighborcount=neighborcount+(A(i-1,j)>0);
                %do for all other neighborcounts accounts for
                %north neighbor
            end
            if i<numrows
                neighborcount=neighborcount+(A(i+1,j)>0);
                %accounts for south neighbor
            end
```

```
if j>1
    neighborcount=neighborcount+(A(i,j-1)>0);
    %accounts for west neighbor
end
if j<numcols
    neighborcount=neighborcount+(A(i,j+1)>0);
    %accounts for east neighbor
end
if i>1 && j>1
    neighborcount=neighborcount+(A(i-1,j-1)>0);
    %accounts for NW neighbor
end
if i<numrows && j<numcols
    neighborcount=neighborcount+(A(i+1,j+1)>0);
    %accounts for SE neighbor
end
if i>1 && j<numcols
    neighborcount=neighborcount+(A(i-1,j+1)>0);
    %accounts for NE neighbor
end
if i<numrows && j>1
    neighborcount=neighborcount+(A(i+1,j-1)>0);
    %accounts for SW neighbor
end
if A(i,j)==0 && ((tp>=randprob && neighborcount>=1) &&...
        (k<=numdays))
    A1(i,j)=1;
    %infection happens under the condition that person is
    %susceptible and person has at least one sick neighbor.
elseif A(i,j)==maxinfperiod
    A1(i,j)=-1*immunity;
elseif A(i,j)>0 && deathprob>=1*rand()
    %check a random (other) probability to a random number
    %and if prob >  random number
    A1(i,j)=-1/0;
    %this is negative infinity
```

```
            elseif A(i,j)~=0
                A1(i,j)=A1(i,j)+1;
            end
            %the rest of the elseif statement covers infection and
            %recovery.  if someone reaches the max infection period,
            %then the are back to recovered for at least as long as
            %that imunity period is.   So negative numbers mean we are
            %in the immune period.  Otherwise, after someone is
            %infected, we count through the number of days they are
            %infected until they reach that max number of days they
            %can be infected.
        end
    end
    A=A1;
    A
    %     2*double(A>0)-1;
    %     above line will do +
    pause
    %pause to see what happens after each iteration
    %this will display blank space for initially sus, + for infected,
    %- for recovery/temporarily immune/dead,
end

%for loop that takes all the positions of entries in matrix A and
%assigns a value of 0 (uninfected) or 1 (infected) based off of
%comparing the trans probability to a random probability AND based
%off of if neigboring cells are infected.  If the random probability
%generated is greater than the transmission probability AND at least
%one neighbor is infected, then A(i,j) will turn to a 1 and thus be
%an infected entry/cell.
```

Now we will look at three different simulations with transmission probabilities of 10%, 50% and 90% respectively. At each transmission probability, we will see how the disease spreads at day 1, day 7, day 14, and day 31 respectively as well as taking into account convergence (or the day the disease died out).

This MATLAB function models disease spread with the input parameters of: number of

days (31), transmission probability (10%, 50%, 90% respectively), grid dimensions (10x10), maximum infection period (5 days), immunity period (7 days), death probability (1%), and output parameter (matrix) of 100 susceptible, infected, recovered or deceased individuals.

Now we need to note the meaning of the numbers in the output. In our CA, the disease always starts in the top left corner. We denote an infected person with a 1. Once this sick individual enters the area, they can spread the disease only to neighboring individuals (so to the N, S, E, W, NE, NW, SE, and SW cells). If that neighbor gets sick, their cell now turns to a 1. Once sick, each new iteration will count the days infected until we reach the maximum infection period of 5 days. Then a negative count starts at -7 for the immunity period, meaning each recovered individual has a week where they will not get sick again. Finally, "-inf" means an individual is deceased (forever immune since they clearly will not get sick again). Below, we will show images from our simulation.

Transmission Probability: 10%, Day: 1

```
A =

     1     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

Transmission Probability: 10%, Day: 7

```
A =

    -6     4     0     0     0     0     0     0     0     0
     0     3     0     0     0     0     0     0     0     0
     2     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

Transmission Probability: 10%, Day: 14

```
A =

     0    -2     0     3     2     0     0     0     0     0
    -6    -3    -6     0     0     0     0     0     0     0
    -4    -6    -7     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

Transmission Probability: 10%, Day: 31

```
A =

     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

Transmission Probability: 50%, Day: 1

```
A =

     1     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

Transmission Probability: 50%, Day: 7

```
A =

    -6     2     2     4     1     0     0     0     0     0
    -7  -Inf     5     3     1     0     0     0     0     0
     4     3     5     4     2     1     0     0     0     0
     0     0     2     2     2     1     0     0     0     0
     0     1     0     1     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

Transmission Probability: 50%, Day: 14

```
A =

     0    -4    -4    -2    -5     5     2     0     0     0
     0  -Inf    -1    -3    -5    -7    -7     5     4     1
    -2  -Inf    -1    -2    -4    -5    -7    -7     4     0
     5    -7    -4    -4    -4    -5    -6     4     4     1
    -7    -5    -6    -5    -6    -6     5     5     4     1
    -6    -7    -6    -6     5     5    -7     5     3     3
    -7    -7    -7     4    -7     5     0     2     4     1
     5     3     5     5     3     5     1     3     3     3
     2     4     3     4     4     2     4     3     1     0
     0     1     3     2     2     2     3     3     2     1
```

Transmission Probability: 50%, Day: 31

```
A =

     0     0     0     0     0     0     0     0     0     0
     0  -Inf     0     0     0     0     0     0     0     0
     0  -Inf     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0  -Inf
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

Transmission Probability: 90%, Day: 1

```
A =

     1     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

Transmission Probability: 90%, Day: 7

```
A =

    -6    -7     5     4     3     2     1     0     0     0
    -7  -Inf     5     4     3     2     1     0     0     0
     5     5     5  -Inf     3     2     1     0     0     0
     4     4     3     4     3     2     0     0     0     0
     3     3     3     3     3     2     1     0     0     0
     2     2     2     2     2     2     1     0     0     0
     1     1     1     1     1     1     1     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

Transmission Probability: 90%, Day: 14

```
A =

     0     0    -1    -2    -3  -Inf    -5    -6  -Inf     4
     0  -Inf    -1    -2    -3    -4    -5    -6    -7     5
    -1    -1    -1  -Inf    -3    -4    -5    -6    -7     5
    -2    -2    -3    -2    -3    -4    -6    -6    -7     5
    -3    -3    -3    -3    -3    -4    -5    -6    -7     5
    -4    -4    -4    -4    -4    -4    -5    -6  -Inf     5
    -5    -5    -5    -5    -5    -5    -5    -6    -7     5
    -6    -6    -6    -6    -6     5    -6    -6    -7     5
    -7    -7    -7     5  -Inf    -7    -7    -7    -7     5
     5     5     5     5     5     5     5     4     5     5
```

Transmission Probability: 90%, Day: 31

```
A =

     0     0     0     0     0  -Inf     0     0  -Inf     0
     0  -Inf     0     0     0     0     0     0     0     0
     0     0     0  -Inf     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0  -Inf     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0  -Inf     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

Note that the disease converges or dies out at day 25, 29, and 23 for transmission probabilities 10%, 50%, and 90% respectively. One might think that it would make sense for the disease to die out more quickly if the transmission probability is lower, but this is not necessarily the case. Notice that we had the disease die out slower at a transmission probability of 50% versus 90%. This is likely because the disease was so aggressive at transmission probability 90% that one would almost inevitably catch the disease, and to catch it very quickly. Thus at this quick rate of spread, the disease also had to die down more rapidly.

We also note that at a transmission probability of 10% that the disease did not spread far at all and there were no deaths by the end of the simulation. For 50% and 90% transmission probability we saw three and six deaths respectively. In general, a higher transmission probability yields a higher death count.

3.2.2 Population Dispersal

In the last section, you saw an example of a CA where each matrix cell/grid block represented an individual person. So we had a number representation for each susceptible, infected and recovered person.

Now we want to upgrade to a CA in which each grid block represents a total population in a region. In each region, there will be a mix of susceptible, infected, and recovered people (as in real life). To make the CA as realistic as possible in the future, we first must have a code that implements a population dispersal. Suppose that the center of the matrix represents the downtown/center area of a large city. Then as we get further and further from downtown area, the population gets less and less dense (just like suburbs).

To build this code, we will have to consider different elements of it one piece at a time. First, we will just focus on population density. Here is the code.

```
function [A,w]=popdispersal(totalpop,numrows)
% Here function that disperses population across a n by n city grid.
%The population will be most dense at the city center and as an overall
%trend, it will become more and more sparce as it approaches the
%outskirts/suburbs.
numcols=numrows;
A=1*rand(numrows,numcols);
%generating a matrix of random numbers between 0 and 1.  accounts for
%somewhat random spread of population
c=5/numrows;
%initializing distance scaler, c
center=[(numrows+1)/2,(numcols+1)/2];
%locating center of matrix
for i=1:numrows
    for j=1:numcols
        D(i,j)=norm(center-[i,j]);
        %getting the distance from the city center to each city block
    end
end
w=exp(-1*c*D);
%setting the weight (or scaling factor) for each grid block population
w=w.*A;
%multiplying the weight/scaling factor to each cell in A
w=w/(sum(sum(w)));
%sum(sum(whatever)) sums the rows AND columns.  makes norm of weight 1
A=totalpop*w;
%taking the sum(sum(A))/totalpop should always sum to 1
% surf(A)
% shading flat
% view(0,90)
%plotting figure for pop. dispersal
```

A well known city in the south of the USA is Nashville, TN. It is about 528 square miles with a population of roughly $700,000$ people. In most large cities such as Nashville there will be the largest population at the center of the city and overall the population will be less and less dense as we move to the outskirts of the city (as mentioned earlier). Our code

will take these things into account. Our input parameters consist of the total population and number of rows (number of columns is set to be the same as number of rows). For the sake of matching Nashville's population data as much as possible, we will set the number of rows to 23 (23×23miles=529 miles squared which is close to the actual square mileage of Nashville), and the total population to $700,000$. In Figure 3.1 is is an image of what our population dispersal of a city similar in landmass and population to Nashville would look like according to our code after putting in these input parameters.

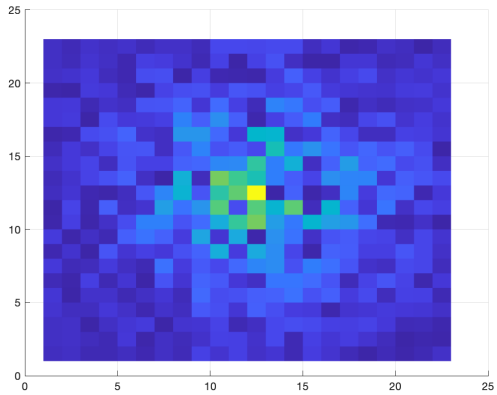


Figure 3.1: City Population Density

Notice that the lighter the color, the more dense the population is. We see an overall trend of a larger population as we are more towards the city center and a smaller population as we get further away from the city center.

3.2.3 Disease Spread with No Long Distance Travel

Now that we have a code for population dispersal, we are ready to create a CA that calls our population dispersal code and then demonstrates disease spread within our population. Remember, in our last CA, every cell in a matrix represented an individual person, so when a matrix cell changed value, it just represented one person changing states (susceptible, infected, recovered or dead).

Now each cell represents a population in a city block. Also, for the time being, we are only assuming that the people in each block are susceptible, infected or recovered, and that these people are only infecting their neighbors in the block. Furthermore, we are assuming that no individual is traveling outside of their city block (matrix cell). It will be later that we implement long distance travel (such as going to work, going across town to eat dinner,

etc...). Below is our newest CA code.

```
function [S,I,R]=diseasespread(I0,totalpop,b,a,timesteps,numrows)
%Function that displays the disease spread for a population grid
%where each cell is a regional population.  Keep in mind, no travel
%from region to region (or new cell to new cell) is being taken
%into account yet for this CA.
A=popdispersal(totalpop,numrows);
%calling pop dispersal
for h=1:numrows
    for k=1:numrows
        I(h,k)=I0*A(h,k);
        S(h,k)=A(h,k)-I(h,k);
        R(h,k)=0;
    end
end
% figure(1)
% surf(S/totalpop)
% caxis([0,1e-2])
% shading flat
% view(0,90)
% title(['i=' num2str(0)])
figure(2)
surf(I/totalpop)
caxis([0,1e-2])
shading flat
view(0,90)
title(['i=' num2str(0)])
% figure(3)
% surf(R/totalpop)
% caxis([0,1e-2])
% shading flat
% view(0,90)
% title(['i=' num2str(0)])
pause
%showing a figure for the intitial S, I and R before running the new
```

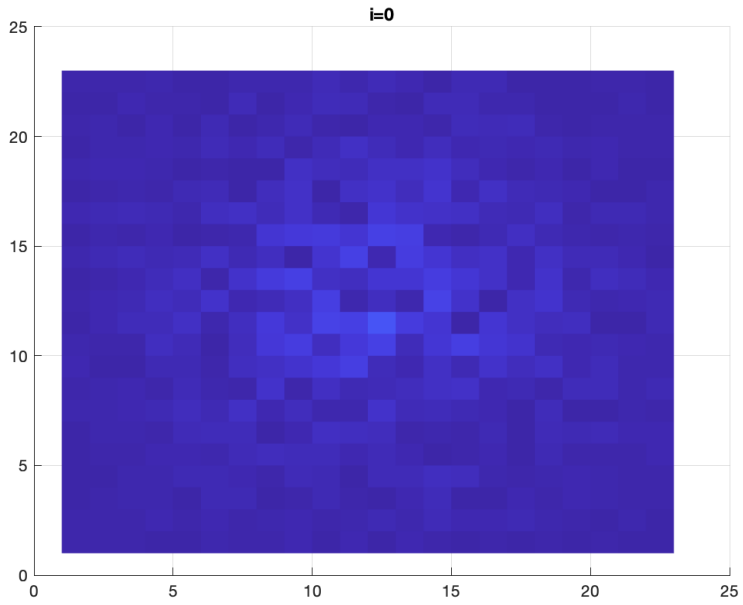
```
%figures.
%initial conditions for S, I and R
for i=2:timesteps+1
    for h=1:numrows
        for k=1:numrows
            Inew(h,k)=I(h,k)+(b*S(h,k)*I(h,k))+(-1*a*I(h,k));
            Snew(h,k)=S(h,k)+(-1*b*S(h,k)*I(h,k));
            Rnew(h,k)=R(h,k)+(a*I(h,k));
        end
    end
    %keeping track of all storage space
    I=Inew;
    S=Snew;
    R=Rnew;
    %     figure(1)
    %     surf(S/totalpop)
    %     caxis([0,1e-2])
    %     shading flat
    %     view(0,90)
    %     title(['i=' num2str(i)])
    figure(2)
    surf(I/totalpop)
    caxis([0,1e-2])
    shading flat
    view(0,90)
    title(['i=' num2str(i)])
    %     figure(3)
    %     surf(R/totalpop)
    %     caxis([0,1e-2])
    %     shading flat
    %     view(0,90)
    %     title(['i=' num2str(i)])
    pause
    %these show the rest of the timesteps for each S, I and R figure.
end
%After getting initial condition matrices for S, I and R, we
```

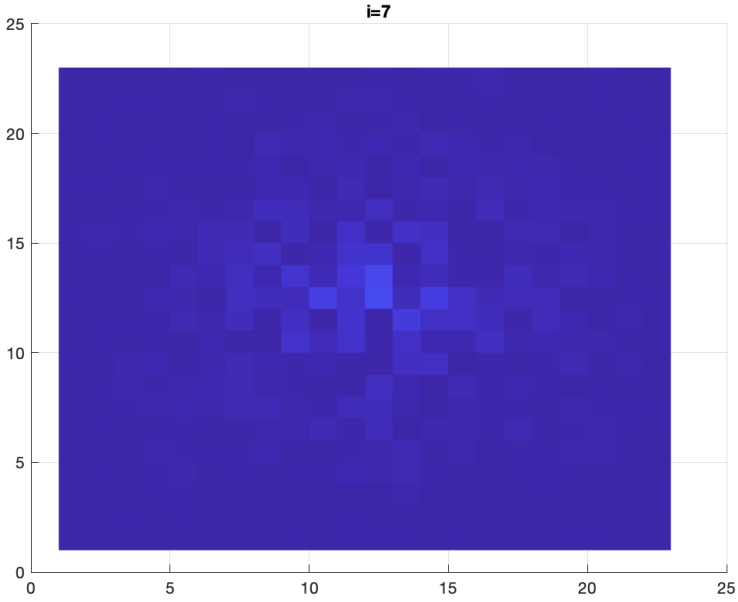
```
%want to update I, S and R after updating each entry within
%the matrix of the the "old" S, I, and  R matrices respectively.
%We do this for as many timesteps as we have.  Then we store these
%values into an updated I, S and R matrix to get our final time
%evolution of the disease spread and display it.
```

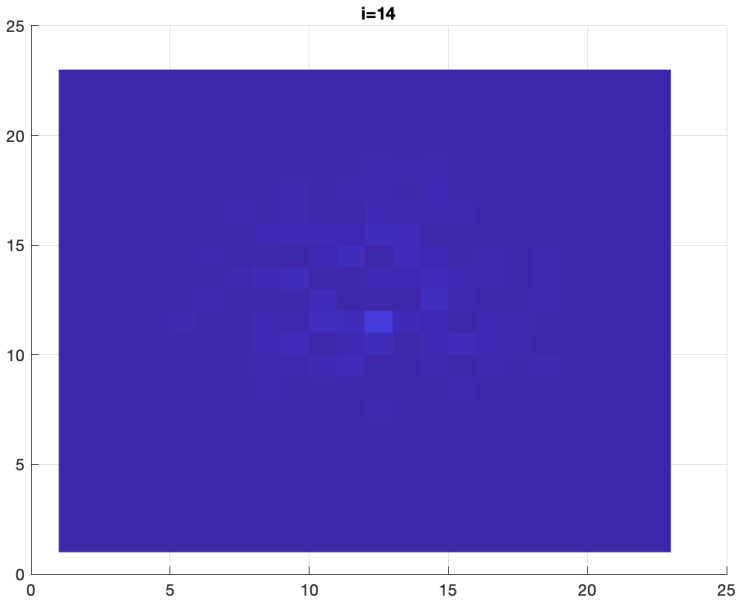
Now suppose we want to model disease spread in what would be considered a fairly large city. We will base our parameters off of Nashville, Tennessee in Figure 3.1. Additional parameters will include the initial infected population I_0, the infection rate b, the recovery rate a, and the the number of time steps (each time step being a day). Figure 3.2 displays our parameter values. Following Figure 3.2 are the color plots of the disease spread (specifically showing infection) at the start of the disease (day 0), one week, one month, two months, a hundred days and three-hundred days respectively.

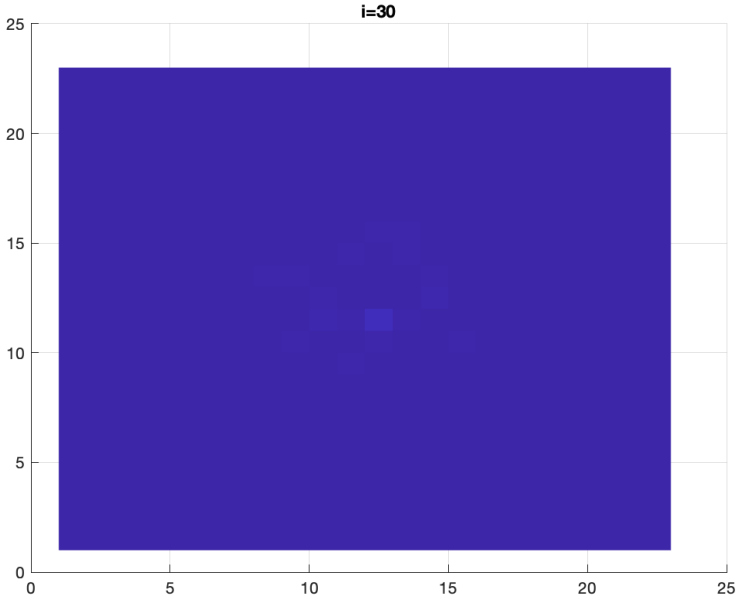
```
I0=.1;
totalpop=700000;
b=1e-5;
a=1/7;
c=0;
timesteps=300;
numrows=23;
```

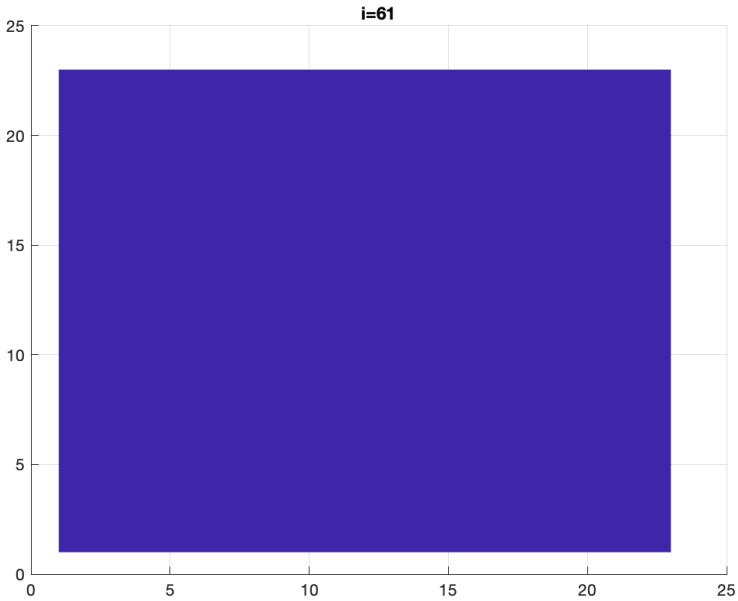
Figure 3.2: Parameters for Disease Spread

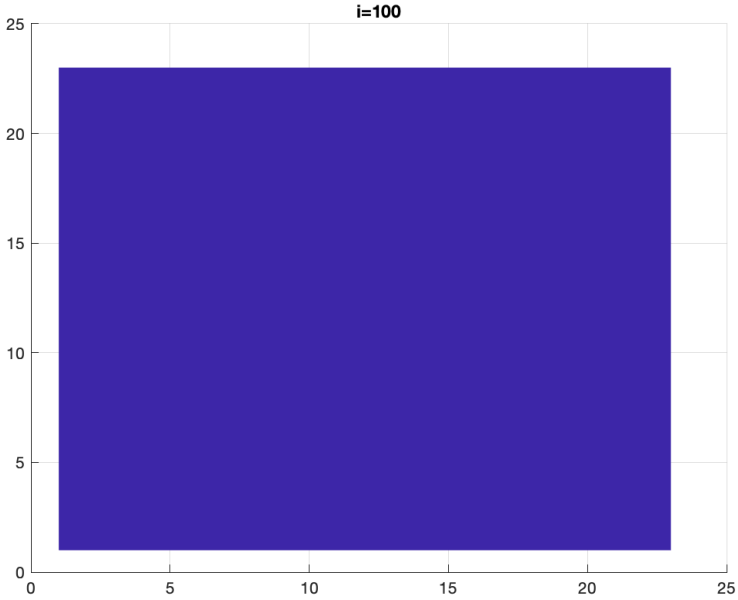


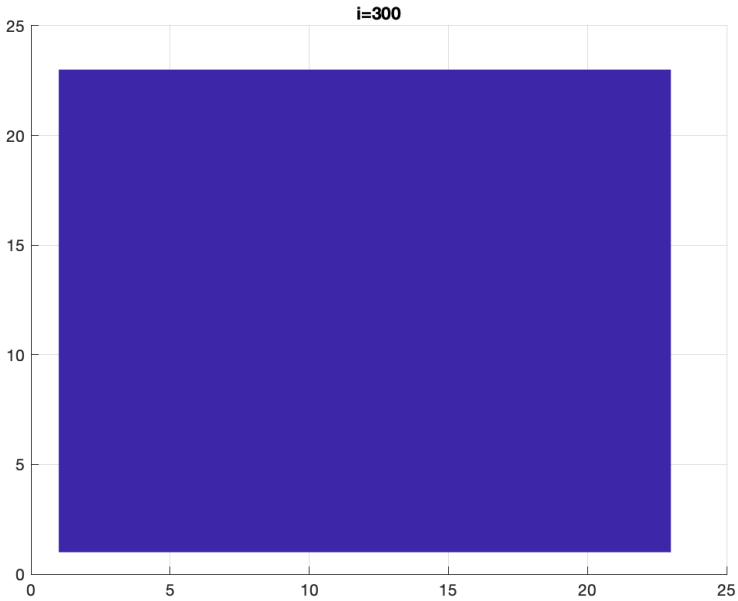

i=7
25
20
15
10
5
0
0
5
10
15
20
25


i=14
25
20
15
10
5
0
0
5
10
15
20
25


i=30
25
20
15
10
5
0
0
5
10
15
20
25


i=61
25
20
15
10
5
0
0
5
10
15
20
25





Notice to begin, we have very little infection throughout the matrix, but the most infection (as demonstrated by the lighter blue colors) towards the center. This is because there is a higher population density at the center by design, meaning it is easier for infection to spread more quickly than as we get further from the center. Then after 1 week, we see that while some infection still lingers around, it is beginning to die out. This is the trend that continues

until we can see it is virtually gone in 2 months, meaning the disease has died out.

We note that there was no disease spread at all towards the outer parts of the matrix. This is because there is no individual traveling outside of their own city block in this simulation. The reader may be wondering why we did not see a spike in infection. In Figure 3.3 we will show an analysis of what is happening with the susceptible, infected and recovered populations in the most centered cell (if there is not one, we will choose the most centered by taking the ceiling of the row and column center averages) with the same parameters as shown in Figure 3.2.

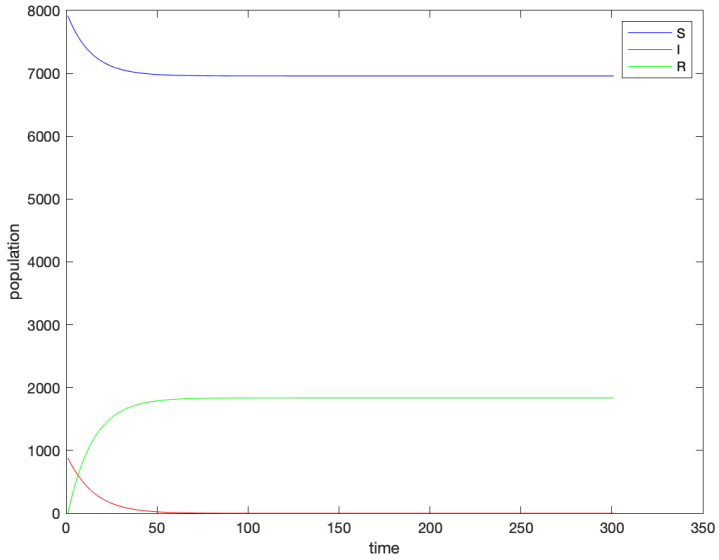


Figure 3.3: S, I and R when $S < a/b$

First, we see there are roughly $9,000$ individuals in this city block. The blue curve represents the susceptible population, the red curve represents the infected population and the green curve the recovered population. In this block, we see the infection drops until it is gone at about day 50. As the infection dies, we see the recovered population rise at a similar rate. The recovered and susceptible populations do converge to a nonzero number because we have not introduced re-infection yet.

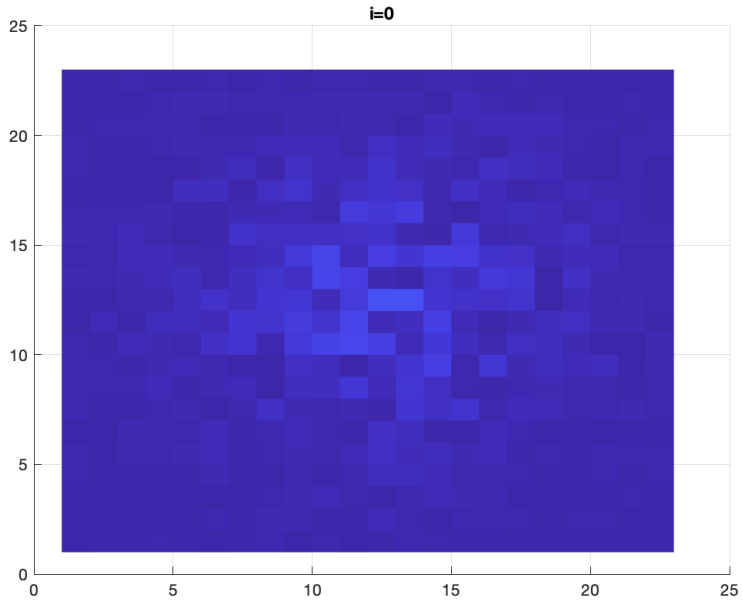
So let's bring back the question at hand: why is there no spike in infection? Recall from our analytical example of a one-dimensional *SIR* model (bottom of page 2) that $S = \frac{a}{b}$ is a fixed point. That is, the epidemic sticks around, but it will not grow nor shrink if $S = \frac{a}{b}$. If

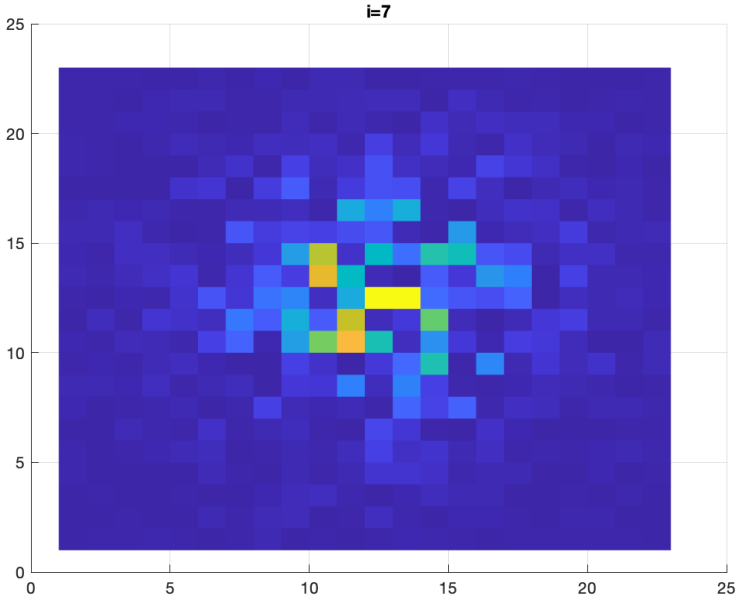
$S < \frac{a}{b}$ from the start, the disease dies out. Since $\frac{a}{b}$ is a constant rate, if $S < \frac{a}{b}$ in one of the most populated cells in the matrix (so the center), then certainly this remains true for the rest of the cells. Given our parameters, Figure 3.4 displays our initial S compared to our fixed point $\frac{a}{b}$.

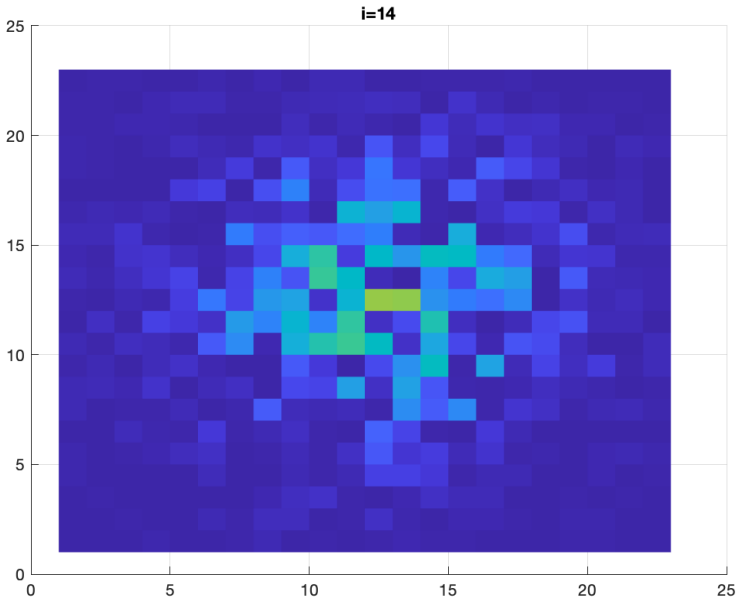
```
>> S(1)

ans =

   7.9158e+03

>> a/b

ans =

   1.4286e+04
```

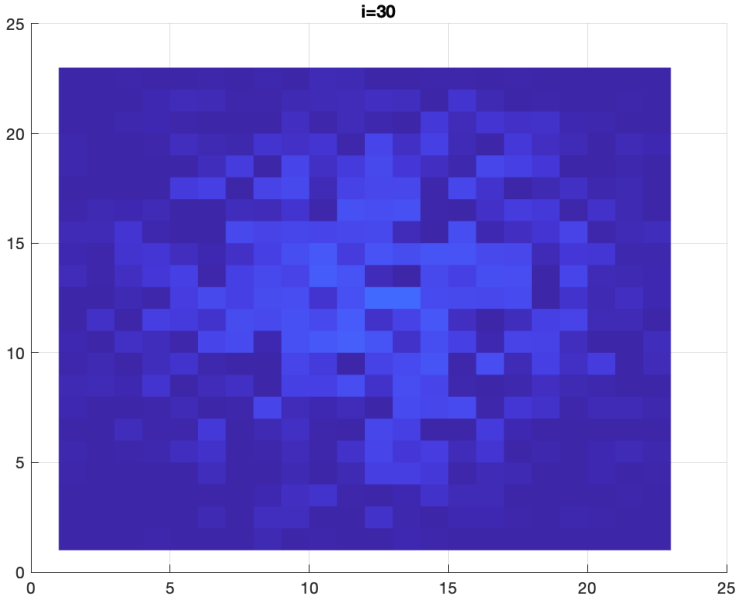
Figure 3.4:

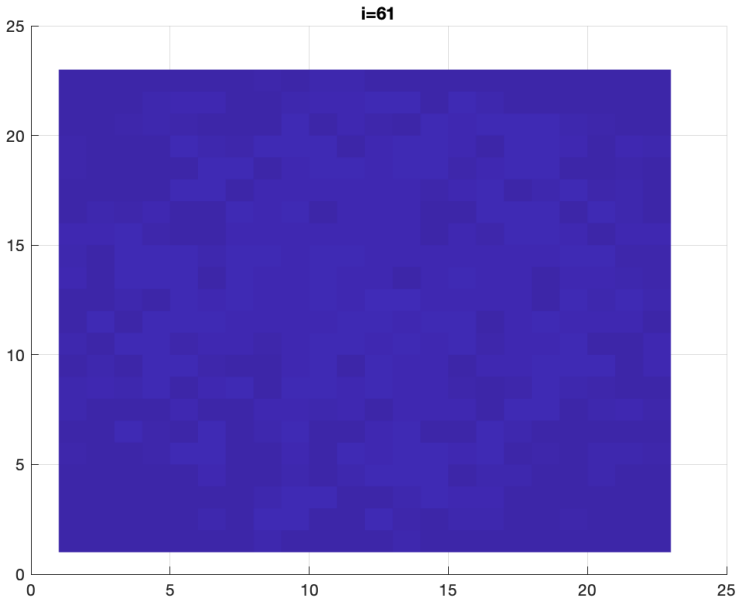
Now, we can clearly see that our initial S is less than our fixed point and thus our disease dies out as we want. Below are the surface plots showing what happens to our disease spread matrix when $S > \frac{a}{b}$. We keep the same parameters from Figure 3.2 other than $b = 0.0001$ and $a = \frac{1}{14}$. The images, like earlier, demonstrate the infection from the *SIR* model.

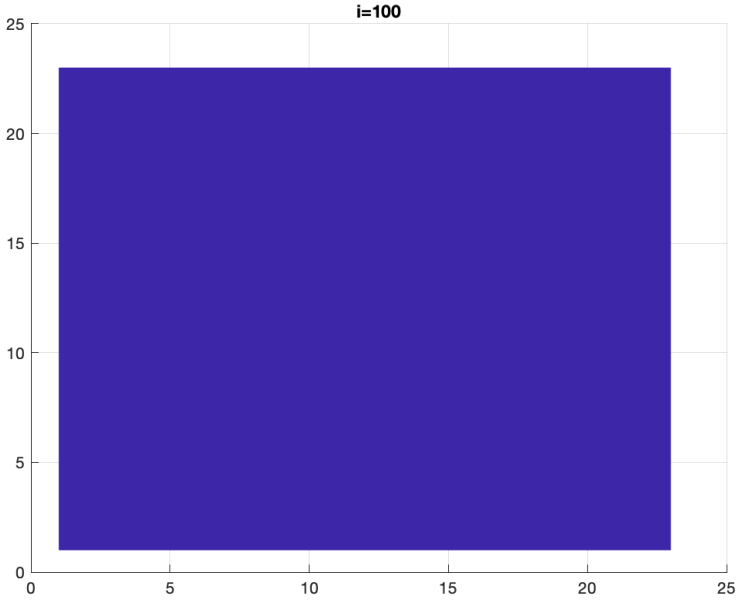


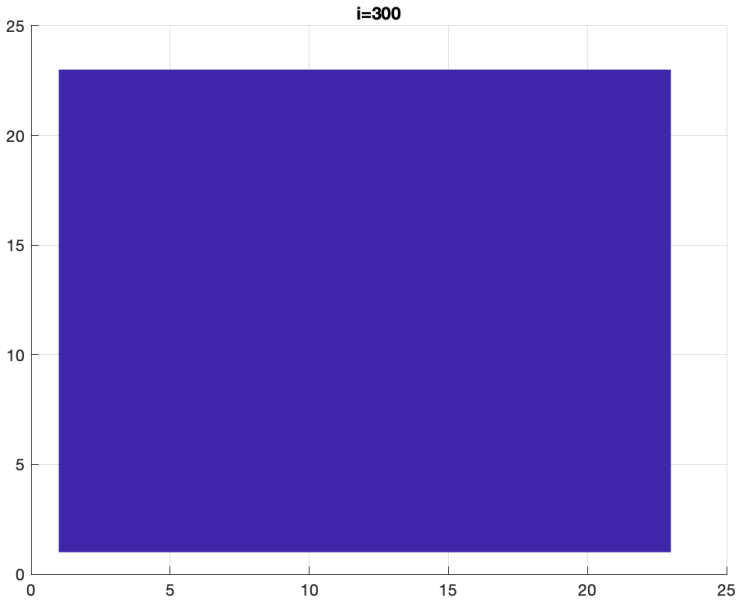

i=7
25
20
15
10
5
0
0
5
10
15
20
25


i=14
25
20
15
10
5
0
0
5
10
15
20
25


i=30
25
20
15
10
5
0
0
5
10
15
20
25


i=61
25
20
15
10
5
0
0
5
10
15
20
25





We see now a very quick spike in infection by a week of the disease running its course. By two weeks the disease spikes in some areas and dies down in others. The overall trend by a month is the disease is dying down. There is still infection lingering all over the matrix by two months in, but it is nearly gone in each cell. By 100 days the disease appears to be gone. Since re-infection is not a part of this simulation, the disease is still gone by day 300. Now

we observe the comparison of S, I and R in a centered cell with the exact same parameters. Figure 3.5 displays this and is followed by the comparison of S with $\frac{a}{b}$.

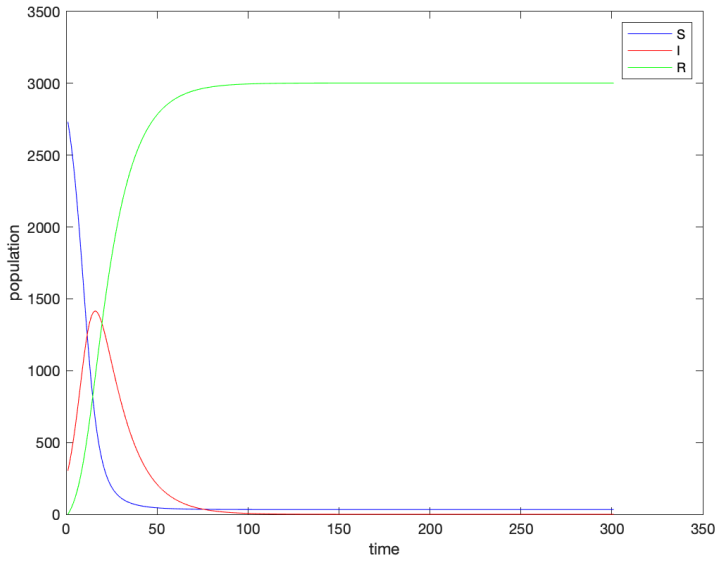


Figure 3.5: S, I and R when $S > a/b$

```
>> S(1)

ans =

   2.7320e+03

>> a/b

ans =

  714.2857
```

Here, we see S is quite a bit larger than our threshold, $\frac{a}{b}$. Thus it makes sense that there would be a very quick spike in infection, which is what we saw in our disease spread matrix earlier. Since there no individual is leaving their city block and there is no reinfection, after a large spike in infection, the disease will also die down very quickly. Eventually anyone who was infected recovers.

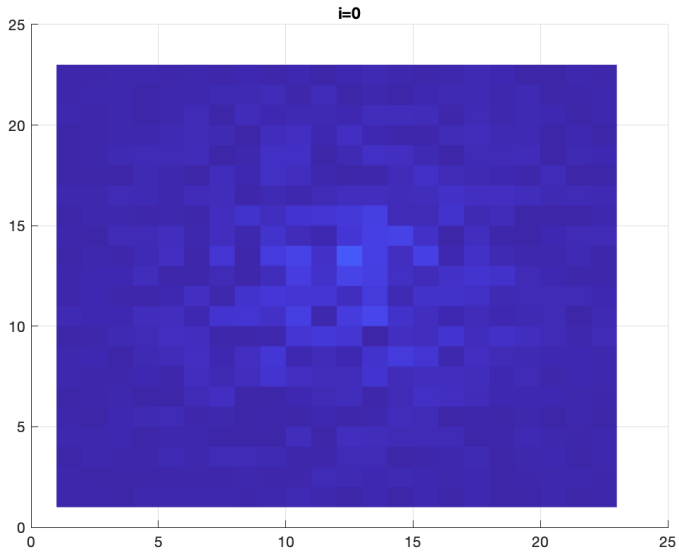
As you can imagine, we have a lot more scenarios that could happen by changing up parameters. However, it is of more interest that we see what happens when we introduce reinfection. This will be covered in the next section.

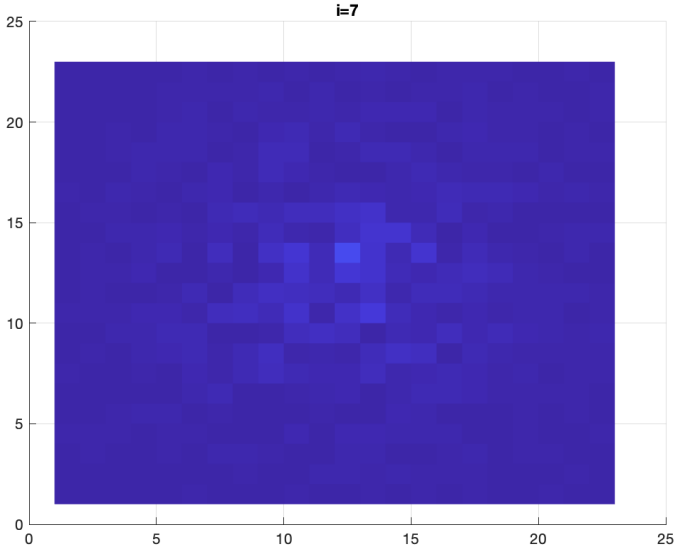
3.2.4 Disease Spread with Reinfection but No Long Distance Travel, but Reinfection

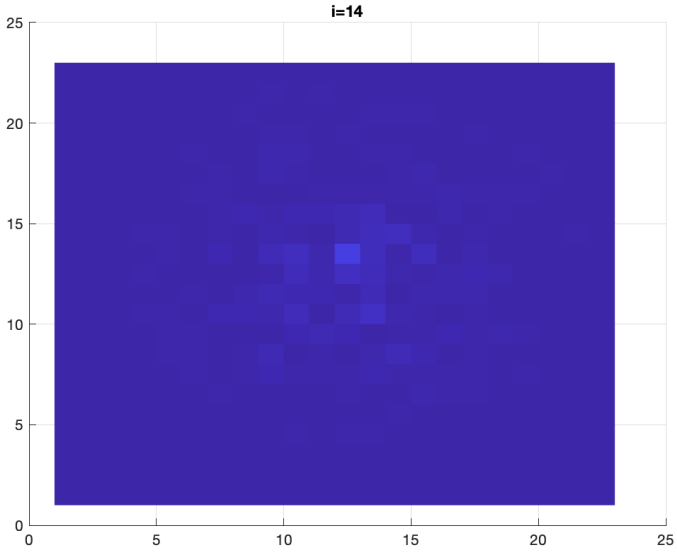
Notice in the last section, we had no coefficient that accounted for re-infection/temporary immunity. Now we will consider what happens when recovered people can only have a temporary immunity period. However, we are still assuming nobody is traveling outside of their city blocks. We will call this recovery coefficient c. Below is a snippet of the code showing how this new coefficient affects our discrete transition S and R equations.

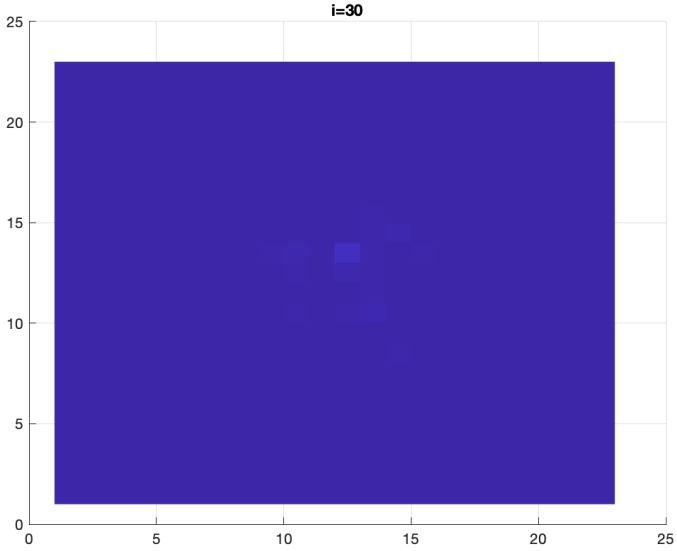
Again, we will show what happens in the case of $S < \frac{a}{b}$ and $S > \frac{a}{b}$ respectively. Below is the change in code where we put our new reinfection parameter value, c, into the S and R equations. Also included are images of how the reinfection affects our matrix over time for the case that $S < \frac{a}{b}$. We will assume the same parameters as Figure 3.2 with the addition of a new temporary immunity parameter c. Assuming an individual has a temporary immunity of a month, we will set $c = \frac{1}{30}$.

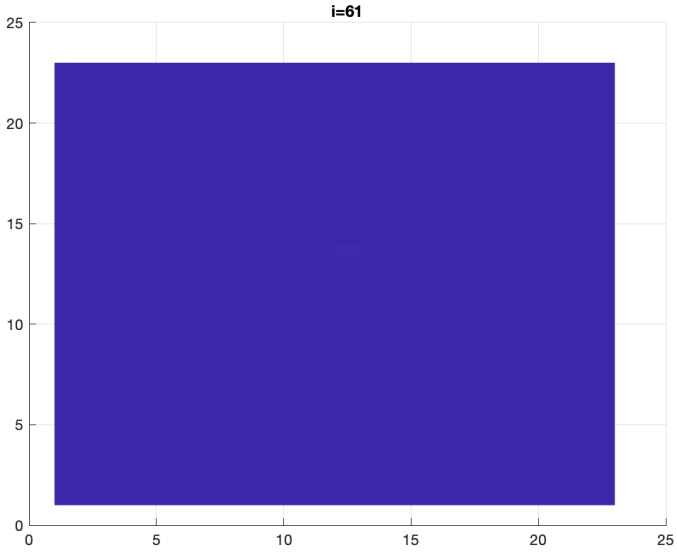
```
for h=1:numrows
        for k=1:numrows
            Inew(h,k)=I(h,k)+(b*S(h,k)*I(h,k))+(-1*a*I(h,k));
            Snew(h,k)=S(h,k)+(-1*b*S(h,k)*I(h,k))+c*R(h,k);
            Rnew(h,k)=R(h,k)+(a*I(h,k))-c*R(h,k);
        end
end
```

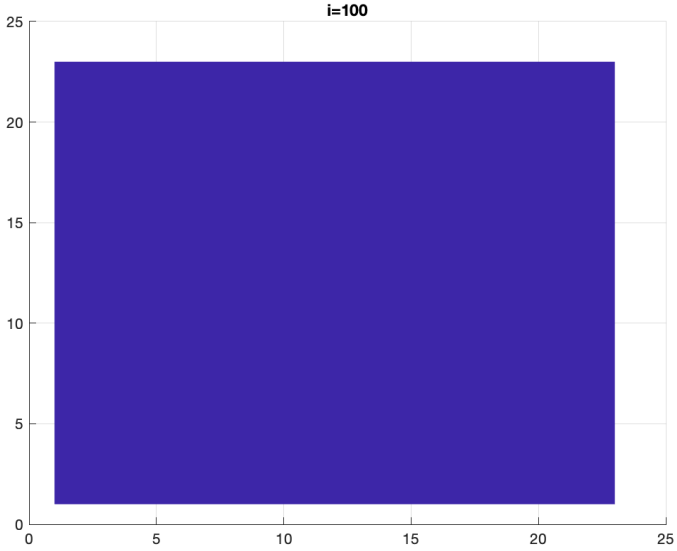


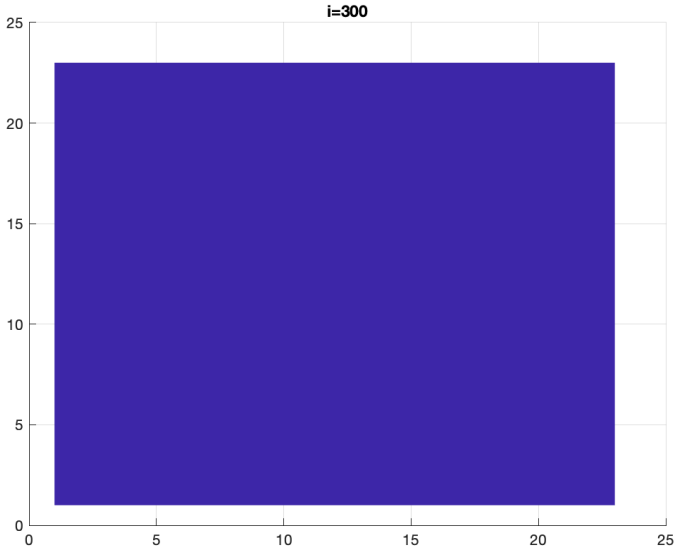

i=7
25
20
15
10
5
0
0
5
10
15
20
25


i=14
25
20
15
10
5
0
0
5
10
15
20
25


i=30
25
20
15
10
5
0
0
5
10
15
20
25


i=61
25
20
15
10
5
0
0
5
10
15
20
25





Again, we see an overall trend of the infection going away over time. In fact, we can see it does not even spike up before going down since we contained the ratio of a to b. Figure 3.6 shows the center cell analysis comparing S, I and R and is followed by the comparison of S with $\frac{a}{b}$.

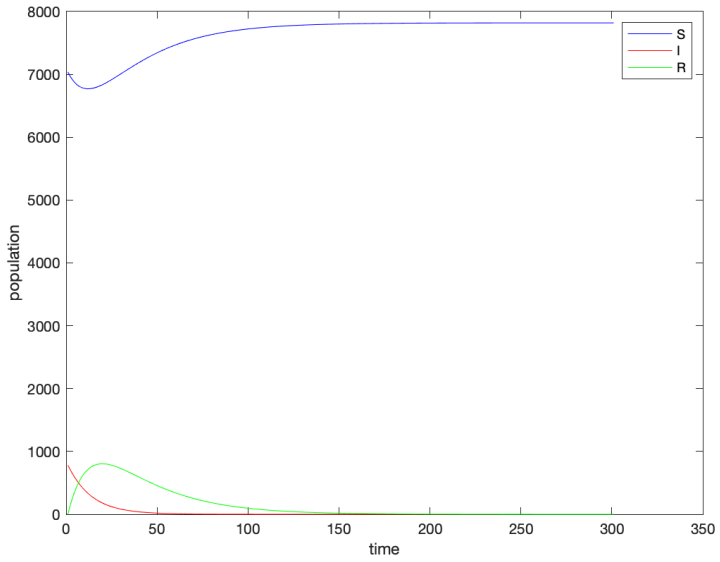
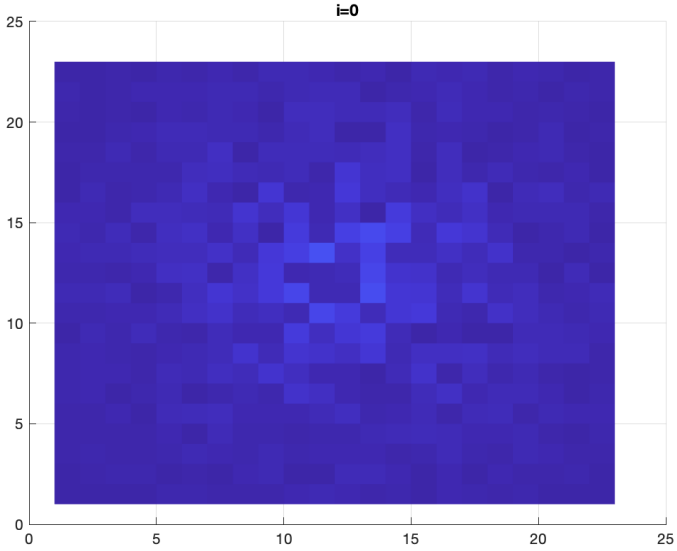


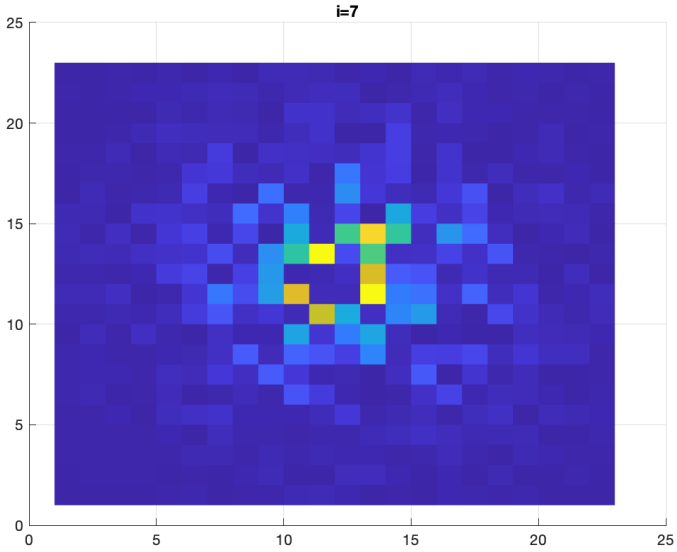
Figure 3.6: S, I and R when $S < a/b$

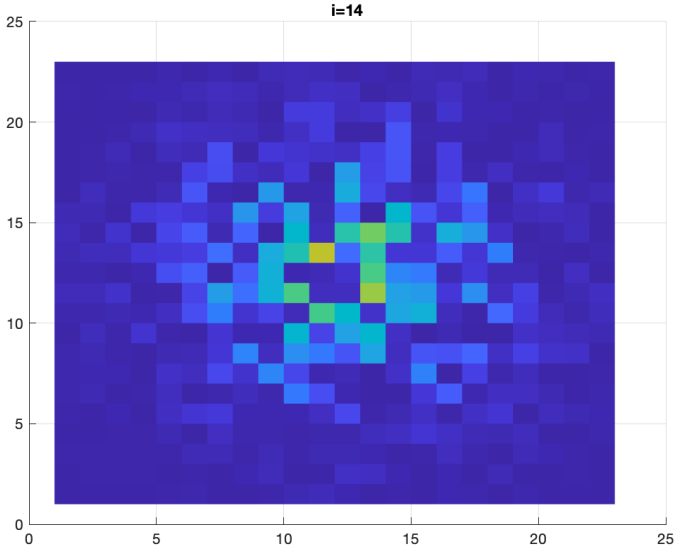
```
>> S(1)

ans =

   7.0377e+03

>> a/b

ans =

   1.4286e+04
```

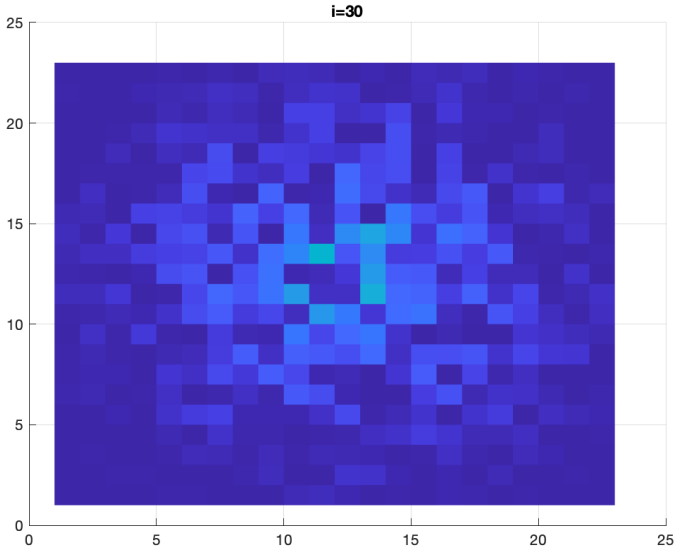
Notice that the key difference between Figure 3.6 and 3.3 is that since we have a temporary immunity coefficient, c, we now see that eventually, all the recovered population goes back into the susceptible population.

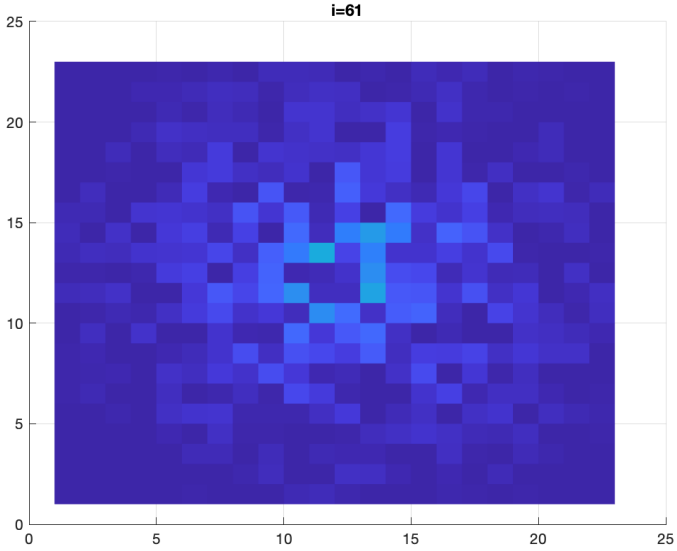
Now let's look into the case where $S > \frac{a}{b}$. To get a good picture of what happens, we will keep all of the same parameters as Figure 3.2 other than this time $a = \frac{1}{14}$ and $b = 0.0001$. Below are images of what happens to our matrix over time with these parameters.

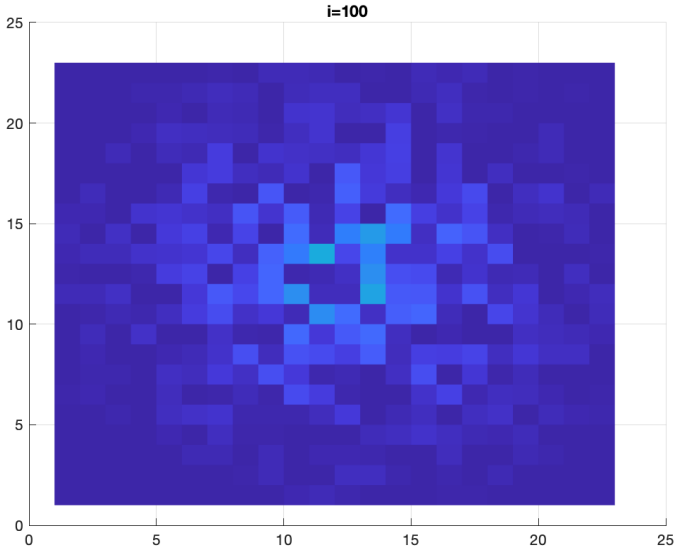

i=0
25
20
15
10
5
0
0
5
10
15
20
25

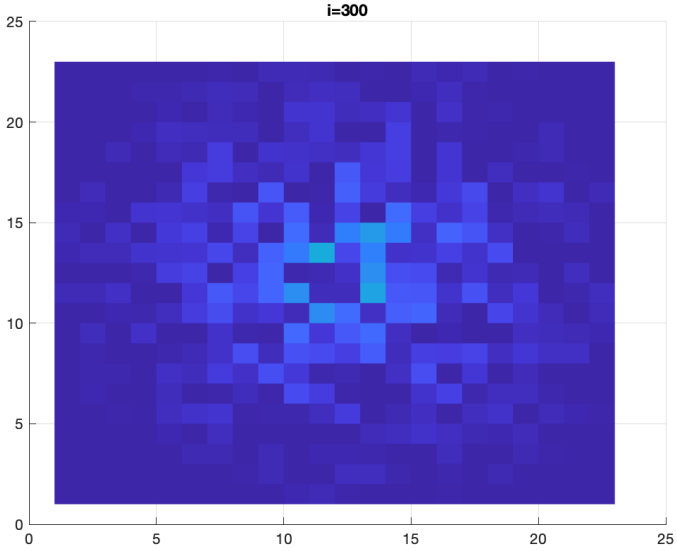

i=7
25
20
15
10
5
0
0
5
10
15
20
25


i=14
25
20
15
10
5
0
0
5
10
15
20
25


i=30
25
20
15
10
5
0
0
5
10
15
20
25


i=61
25
20
15
10
5
0
0
5
10
15
20
25


i=100
25
20
15
10
5
0
0
5
10
15
20
25



It is clear to see the infection never dies out. It eventually converges because even once people recover, they are eventually susceptible again putting them back into the same cycle. The disease just lingers around. Figure 3.7 displays S, I and R in the center cell of the matrix followed by an image of our initial S compared to $\frac{a}{b}$.

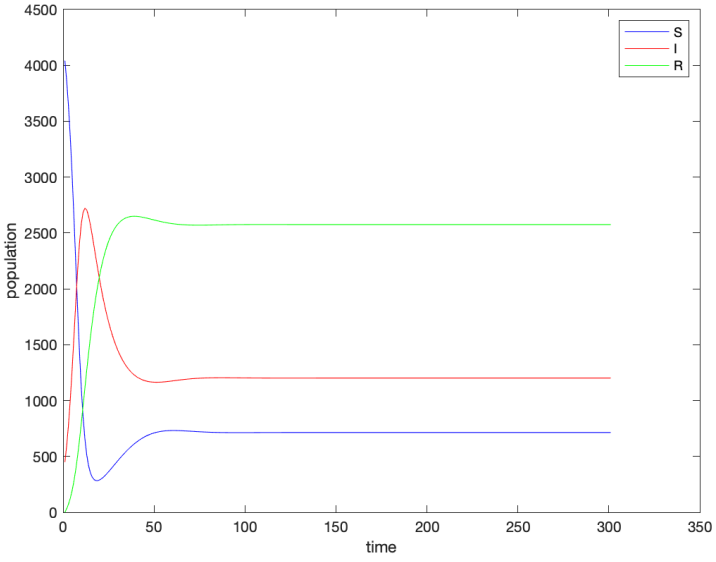


Figure 3.7: S, I and R when $S < a/b$

```
>> S(1)

ans =

   4.0424e+03

>> a/b

ans =

  714.2857
```

We can see the spike in infection, the decline, and then that the infection never goes away.

3.2.5 Disease Spread with Reinfection and Long Distance Travel

We finally are approaching the main goal of this project. The natural next advancement to our CA is to add in long distance travel. This means that now, individuals can leave their city blocks freely to go to work, to eat, etc... Since each non-boundary cell has eight neighbors (Moore Neighborhood), we use a nine-point stencil for the Laplacian operator multiplied by a travel coefficient added onto the previous S, I and R iterates to update our equations in each time-step. Below is a code of our next CA.

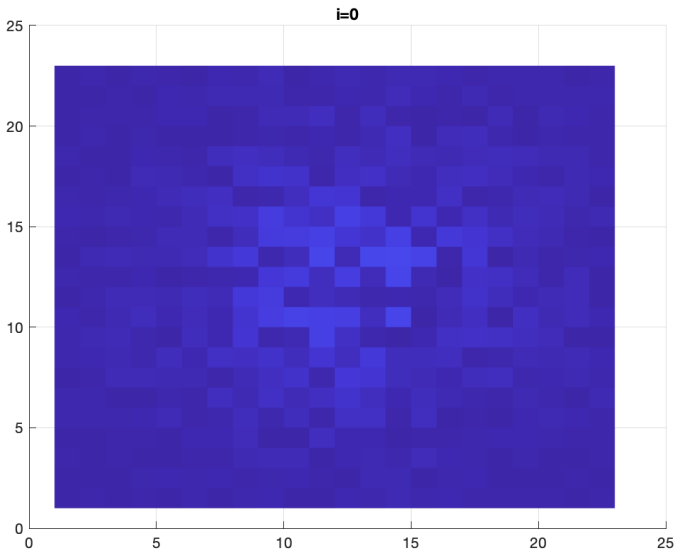
```
function [S,I,R]=diseasespreadwithtravel(I0,totalpop,b,a,c,timesteps,...
    numrows)
%Function that displays the disease spread for a population grid where
%each cell is a regional population.  Keep in mind, no travel from
%region to region (or new cell to new cell) is being taken into account
%yet for this CA.
[A,w]=popdispersal(totalpop,numrows);
T=10; %defining scaling coefficient for travel
t=T*(w-(1/numrows^2));
%calling pop dispersal
for h=1:numrows
    for k=1:numrows
        I(h,k)=I0*A(h,k);
        S(h,k)=A(h,k)-I(h,k);
        R(h,k)=0;
    end
end
Isuburb1(1)=I(2,2);
Isuburb2(1)=I(22,2);
```

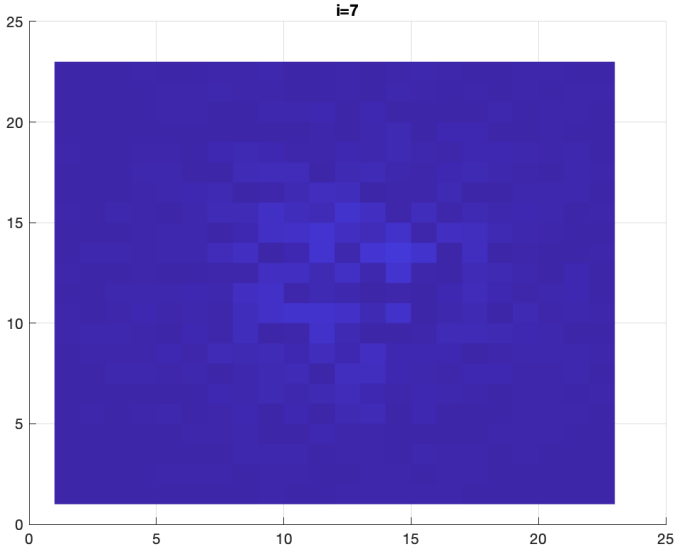
```
% figure(1)
% surf(S/totalpop)
% caxis([0,1e-2])
% shading flat
% view(0,90)
% title(['i=' num2str(0)])
figure(2)
surf(I/totalpop)
caxis([0,1e-2])
shading flat
view(0,90)
title(['i=' num2str(0)])
% figure(3)
% surf(R/totalpop)
% caxis([0,1e-2])
% shading flat
% view(0,90)
% title(['i=' num2str(0)])
pause
%showing a figure for the intitial S, I and R before running the new
%figures.
%initial conditions for S, I and R
for i=2:timesteps+1
    %negating timesteps happens here
    for h=1:numrows
        for k=1:numrows
            Inew(h,k)=I(h,k)+(b*S(h,k)*I(h,k))+(-1*a*I(h,k));
            Snew(h,k)=S(h,k)+(-1*b*S(h,k)*I(h,k))+c*R(h,k);
            Rnew(h,k)=R(h,k)+(a*I(h,k))-c*R(h,k);
            %add on travel t(h,k) to all terms multiplied by 5
            %point stencil for lapacian to all terms in an if statement
            %inside the loops to cover all non-edge cells.
            if (h > 1 && k > 1) && (h < numrows && k < numrows)
                Inew(h,k)=Inew(h,k)+t(h,k)*(I(h-1,k-1)+I(h-1,k+1)
                +I(h+1,k-1)+I(h+1,k+1)-20*I(h,k)+(4*(I(h+1,k)+I(h-1,k)
                +I(h,k+1)+I(h,k-1))))/6;
```

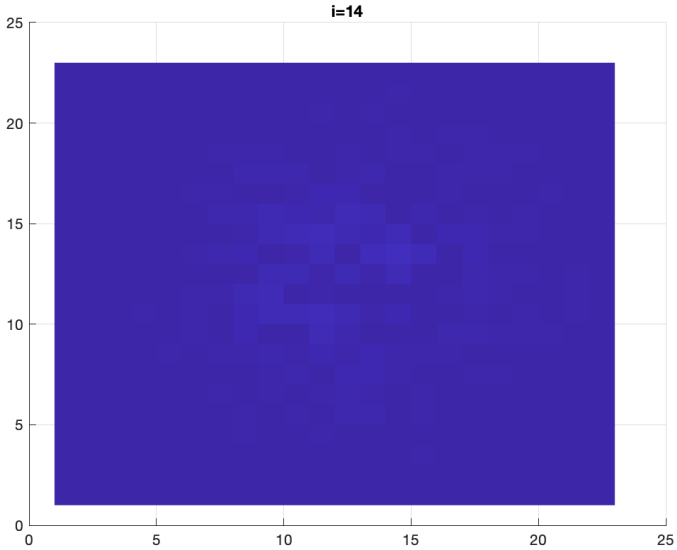
```
            Snew(h,k)=Snew(h,k)+t(h,k)*(S(h-1,k-1)+S(h-1,k+1)
            +S(h+1,k-1)+S(h+1,k+1)-20*S(h,k)+(4*(S(h+1,k)+S(h-1,k)
            +S(h,k+1)+S(h,k-1))))/6;
            Rnew(h,k)=Rnew(h,k)+t(h,k)*(R(h-1,k-1)+R(h-1,k+1)
            +R(h+1,k-1)+R(h+1,k+1)-20*R(h,k)+(4*(R(h+1,k)+R(h-1,k)
            +R(h,k+1)+R(h,k-1))))/6;
        end
    end
end
%keeping track of all storage space
t=-t;
I=Inew;
S=Snew;
R=Rnew;
Isuburb1(i)=I(2,2);
Isuburb2(i)=I(22,2);
%     figure(1)
%     surf(S/totalpop)
%     caxis([0,1e-2])
%     shading flat
%     view(0,90)
%     title(['i=' num2str(i)])
figure(2)
surf(I/totalpop)
caxis([0,1e-2])
shading flat
view(0,90)
title(['i=' num2str(i)])
%     figure(3)
%     surf(R/totalpop)
%     caxis([0,1e-2])
%     shading flat
%     view(0,90)
%     title(['i=' num2str(i)])
pause
%these show the rest of the timesteps for each S, I and R figure.
```

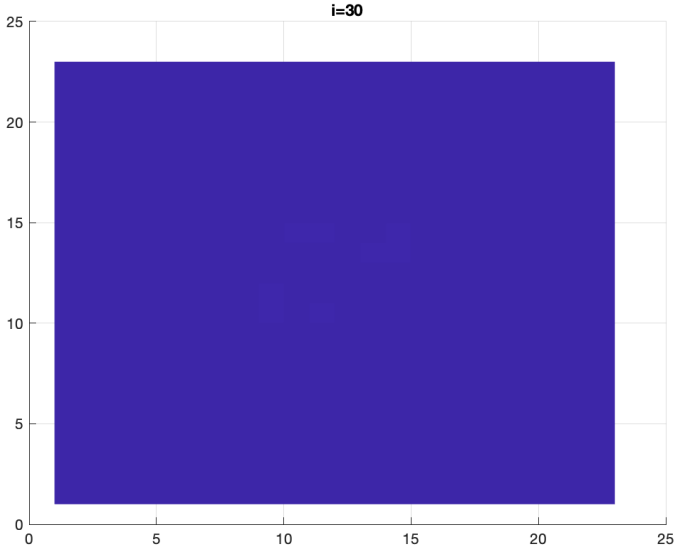
```
end
figure(4);
plot(Isuburb1,'b');
hold on
plot(Isuburb2,'r');
hold off
```

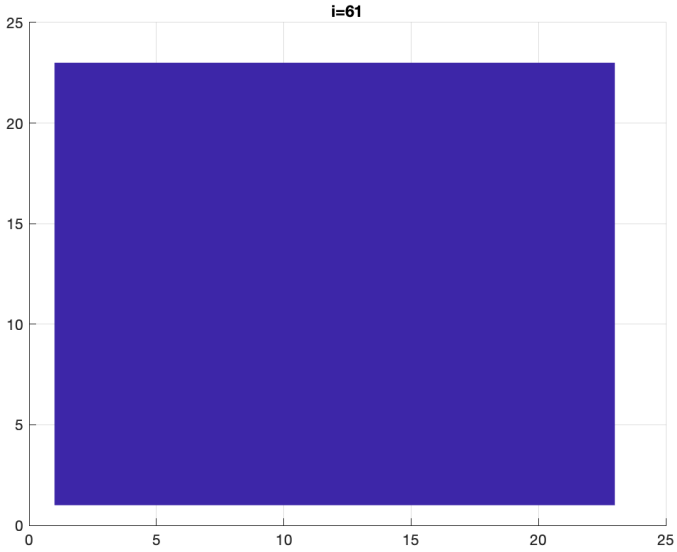
Again, we will have the same parameters as Figure 3.2. We will also set $c = \frac{1}{30}$. We also note that we choose our travel coefficient to be 10. It is also important to note that we now count a time-step as half a day. Odd time-steps account for individuals traveling towards the city center, and even time-steps account for those individuals coming back home. Since traveling is now a part of the CA, it would be interesting to compare infection in two city blocks near the outskirts of a city which was also implemented in the previous code. Here are the surface plots for S, I, and R at time 0, 7, 14, 30, 61, 100, and 300 respectively, followed by Figure 3.8 showing curve plots of the infection in cells $I(2,2)$ and $I(22,2)$.

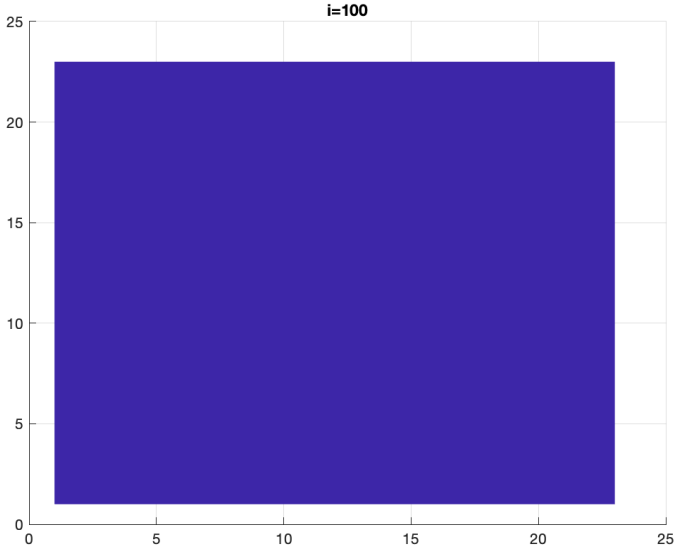


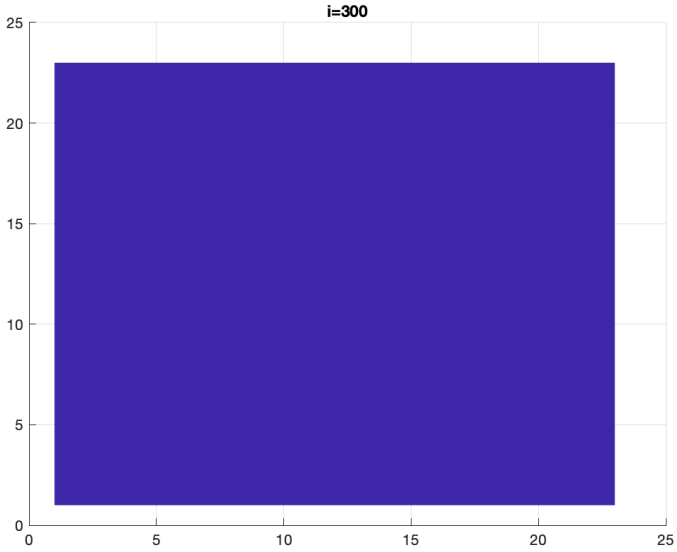

i=7
25
20
15
10
5
0
0
5
10
15
20
25


i=14
25
20
15
10
5
0
0
5
10
15
20
25


i=30
25
20
15
10
5
0
0
5
10
15
20
25


i=61
25
20
15
10
5
0
0
5
10
15
20
25


i=100
25
20
15
10
5
0
0
5
10
15
20
25


i=300
25
20
15
10
5
0
0
5
10
15
20
25

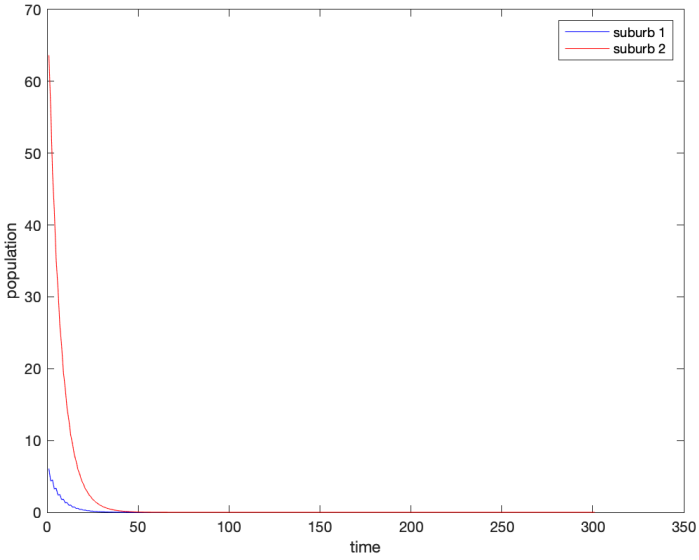


Figure 3.8: Suburb1 1 vs. Suburb 2 when $S < a/b$

Although travel is now allowed outside of one's city block, we do not see a wide spread of infection due to the initial susceptible population being smaller than the fixed point $\frac{a}{b}$. However, we can now see there is at least a little infection in these city blocks far from the center of town. It would give us a little more insight into what is happening to see a curve plot of what is happening in this scenario. Recall earlier that we showed curve plots just for the center cell of S, I and R. Since travel is allowed now, it would be more interesting to see these curve plots for the total populations of S, I, and R over all time-steps. Below is the code that can show these these results. Following the code will be images of the initial susceptible population vs. $\frac{a}{b}$ and the curve plot just mentioned (Figure 3.9).

```
function [S,I,R]=SIRplots(I0,totalpop,b,a,c,timesteps,numrows)
%Function that displays the disease spread for a population grid
%where eachbcell is a regional population.  Keep in mind, no travel
%from region to region (or new cell to new cell) is being taken into
%account yet for this CA.
[A,w]=popdispersal(totalpop,numrows);
T=1; %defining scaling coefficient for travel
t=T*(w-(1/numrows^2));
Itotal=[];
Stotal=[];
Rtotal=[];
```

```
%calling pop dispersal
for h=1:numrows
    for k=1:numrows
        I(h,k)=I0*A(h,k);
        S(h,k)=A(h,k)-I(h,k);
        R(h,k)=0;
    end
end
Stotal(1)=sum(sum(S))
aoverb=a/b
Rtotal(1)=sum(sum(R));
Itotal(1)=sum(sum(I));
% Itotal(1)=I(12,12);
% Stotal(1)=S(12,12);
% Rtotal(1)=R(12,12);
%showing a figure for the intitial S, I and R before running the new
%figures.
%initial conditions for S, I and R
for i=2:timesteps+1
    %negating timesteps happens here
    for h=1:numrows
        for k=1:numrows
            Inew(h,k)=I(h,k)+(b*S(h,k)*I(h,k))+(-1*a*I(h,k));
            Snew(h,k)=S(h,k)+(-1*b*S(h,k)*I(h,k))+c*R(h,k);
            Rnew(h,k)=R(h,k)+(a*I(h,k))-c*R(h,k);
            %add on travel t(h,k) to all terms multiplied by 5
            %point stencil for lapacian to all terms in an if statement
            %inside the loops to cover all non-edge cells.
            if (h > 1 && k > 1) && (h < numrows && k < numrows)
                Inew(h,k)=Inew(h,k)+t(h,k)*(I(h-1,k-1)+I(h-1,k+1)
                +I(h+1,k-1)+I(h+1,k+1)-20*I(h,k)+(4*(I(h+1,k)
                +I(h-1,k)+I(h,k+1)+I(h,k-1))))/6;
                Snew(h,k)=Snew(h,k)+t(h,k)*(S(h-1,k-1)+S(h-1,k+1)
                +S(h+1,k-1)+S(h+1,k+1)-20*S(h,k)+(4*(S(h+1,k)
                +S(h-1,k)+S(h,k+1)+S(h,k-1))))/6;
                Rnew(h,k)=Rnew(h,k)+t(h,k)*(R(h-1,k-1)+R(h-1,k+1)
```

```
                +R(h+1,k-1)+R(h+1,k+1)-20*R(h,k)+(4*(R(h+1,k)
                +R(h-1,k)+R(h,k+1)+R(h,k-1))))/6;
            end
        end
    end
    %keeping track of all storage space
    t=-t;
    I=Inew;
    S=Snew;
    R=Rnew;
    Itotal(i)=sum(sum(I));
    Stotal(i)=sum(sum(S));
    Rtotal(i)=sum(sum(R));
    % Itotal(i)=I(12,12);
    % Stotal(i)=S(12,12);
    % Rtotal(i)=R(12,12);
end
time=1:timesteps+1;
%for plotting time
figure(4);
plot(time,Stotal,'b');
hold on
%need to plot I and R on top of S
plot(time,Itotal,'r');
plot(time,Rtotal,'g');
hold off
```

```
Stotal =

   6.3000e+05

aoverb =

   1.4286e+04
```

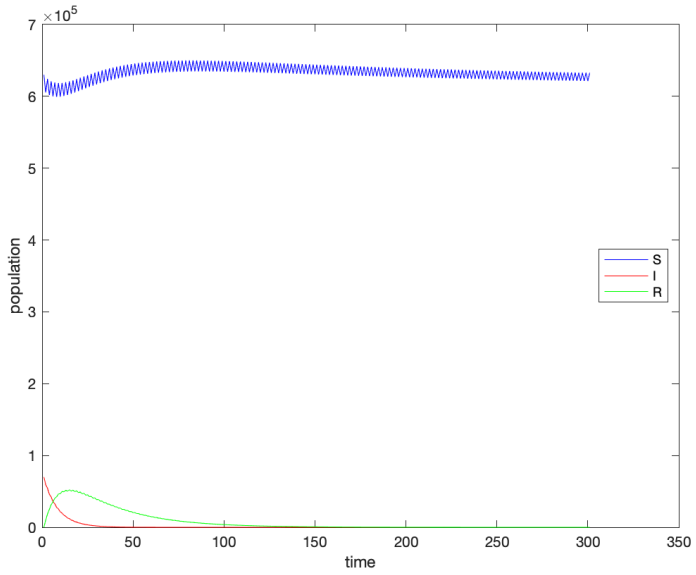


Figure 3.9: $S < a/b$ and $T = 10$

First, we note our travel coefficient, t, has the following properties of: its average over the spatial domain is zero, it's periodic in time with the period equal to a day, and the overall population does not change due to travel. Notice the jagged effect that we can see takes place with our blue susceptible curve. This has to do with our scaling travel coefficient T. We see the jagged effect because people are coming in and out of town throughout the day. The larger the travel coefficient, the steeper the jag is (until a certain point). Lets take a look at the curve plot with $T = 1, 20, 30$ and 35.

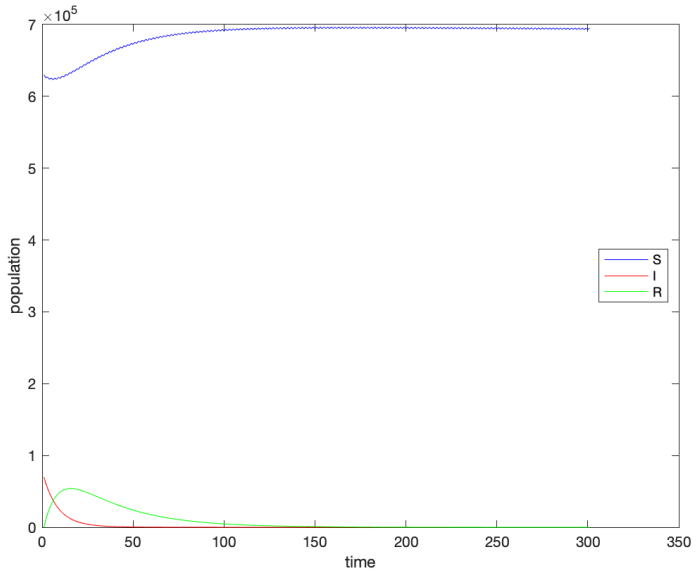


Figure 3.10: $S < a/b$ and $T = 1$

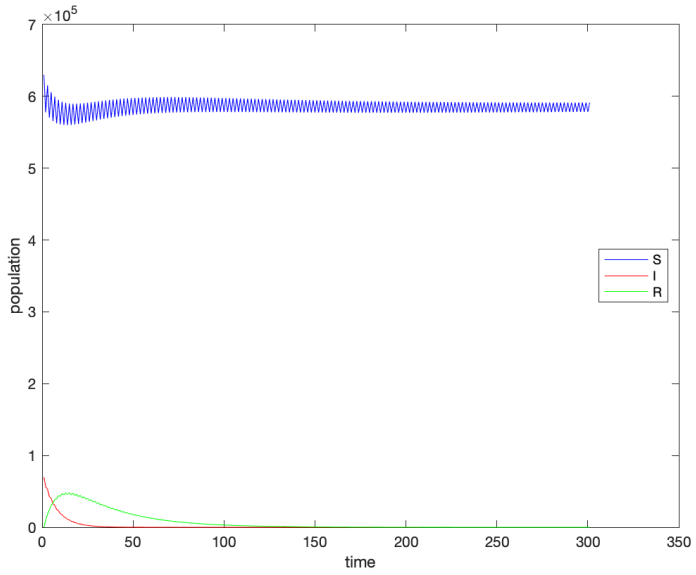


Figure 3.11: $S < a/b$ and $T = 20$

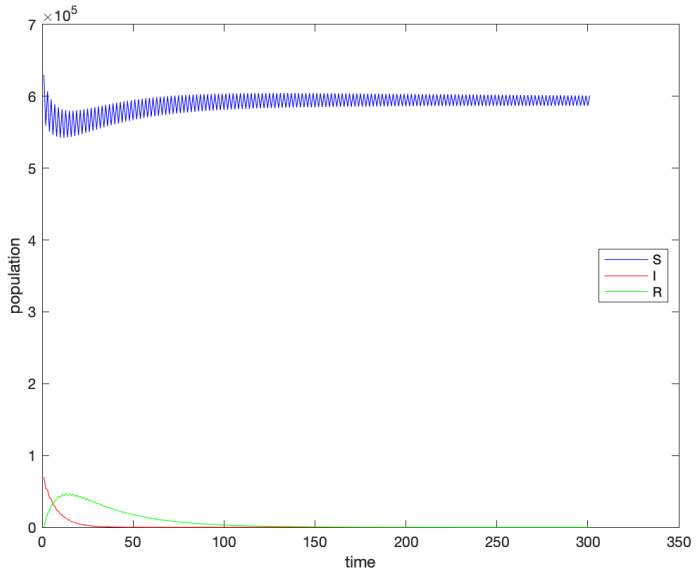


Figure 3.12: $S < a/b$ and $T = 30$

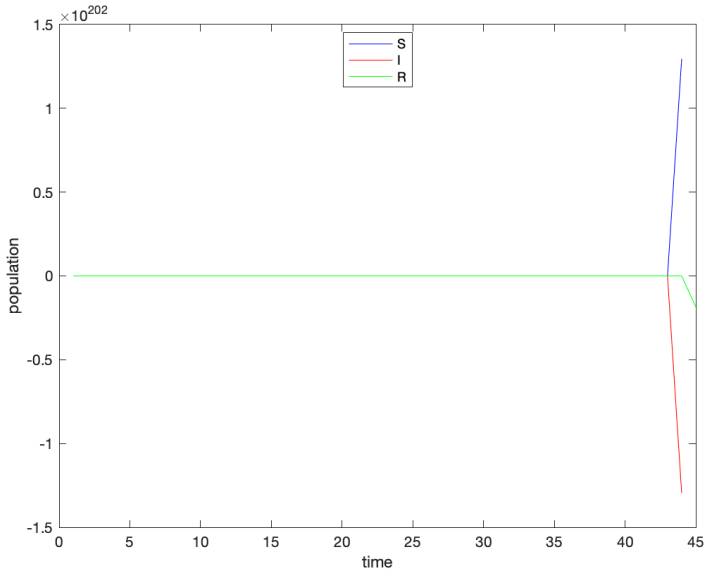
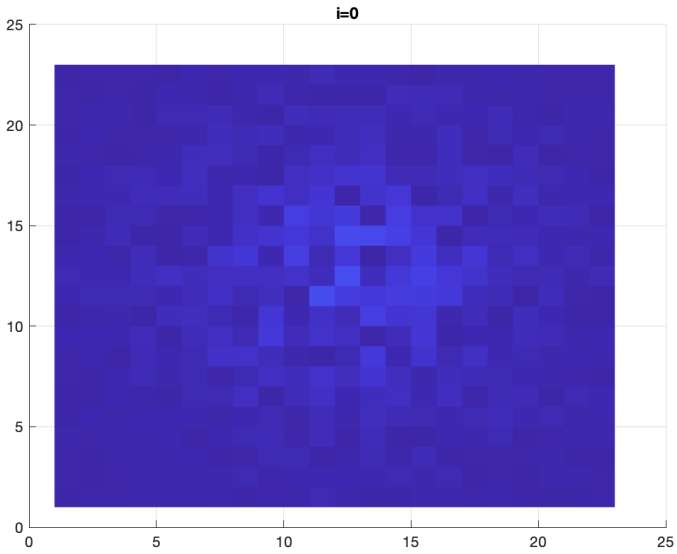
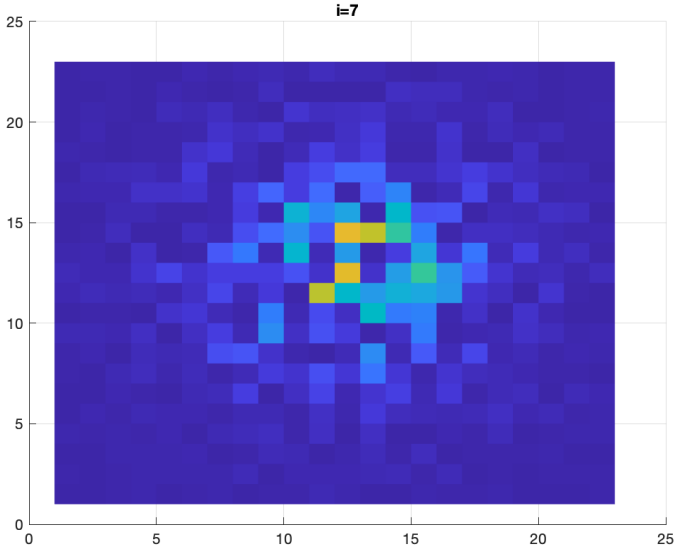


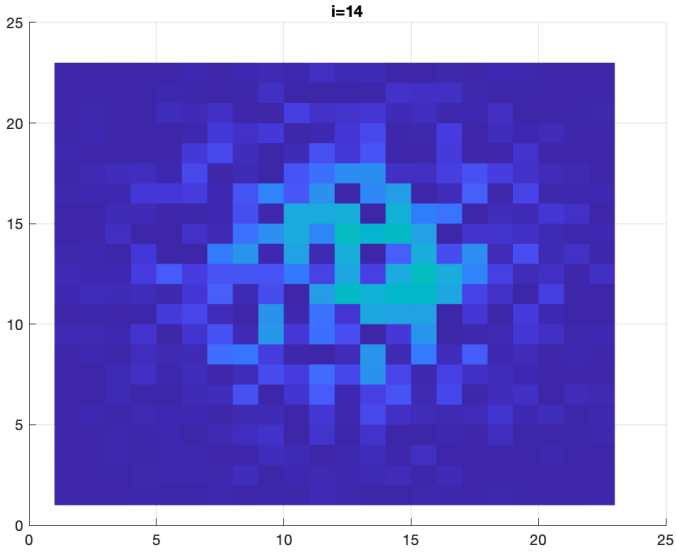
Figure 3.13: $S < a/b$ and $T = 35$

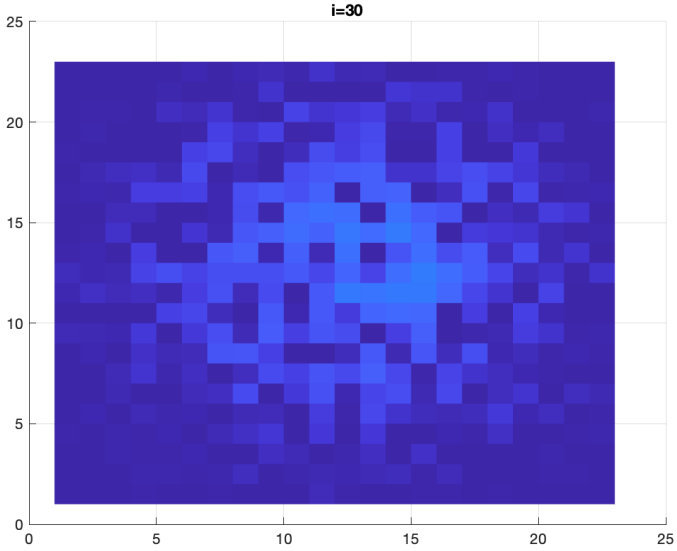
First, we note that there is a little bit of jag effect we can see in I and R for values of T greater than 10. We believe this has to do with these populations traveling at a more aggressive rate. We were expecting is to see is that changing the scaling factor T would overall change the dynamics between S, I and R. However, this does not appear to be the case. In the future, we hope to understand more of why these dynamics do not change with a higher traveling scaling factor, or if there is something more we can do to show the dynamics between these populations changing as we expected. I am not sure what to make of the curve plot for $T = 35$, but it is at least clear that it's not stable due to the coefficient being too large relative to the time step..

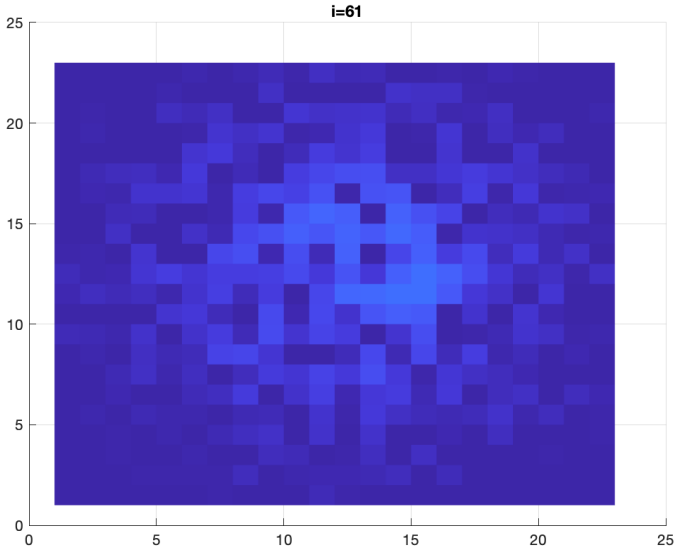
Let us look at our surface plots of infection followed a curve plot comparing $I(2,2)$ and $I(22,2)$ as well as a curve plot comparing S, I and R for the case that $S > \frac{a}{b}$ (Figures 3.14 and 3.15 respectively). We will follow the same parameters as figure 3.2 with the exception of $a = \frac{1}{14}$ and $b = 0.0001$. Also, $c = \frac{1}{30}$. Note we set $T = 10$ back in the CA code.

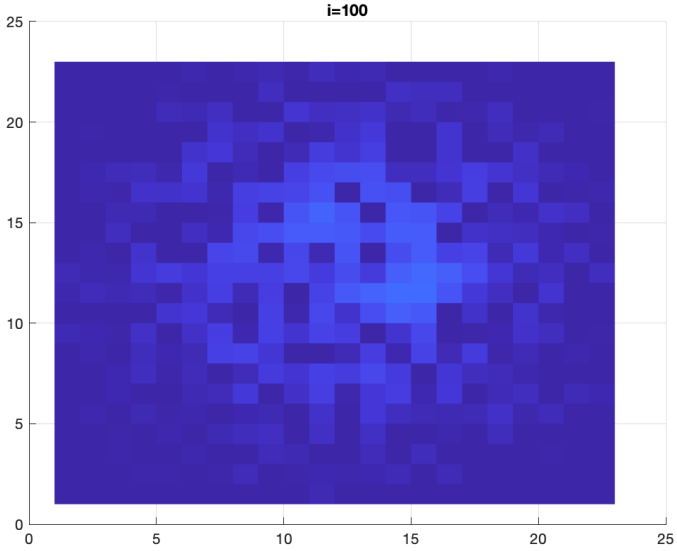


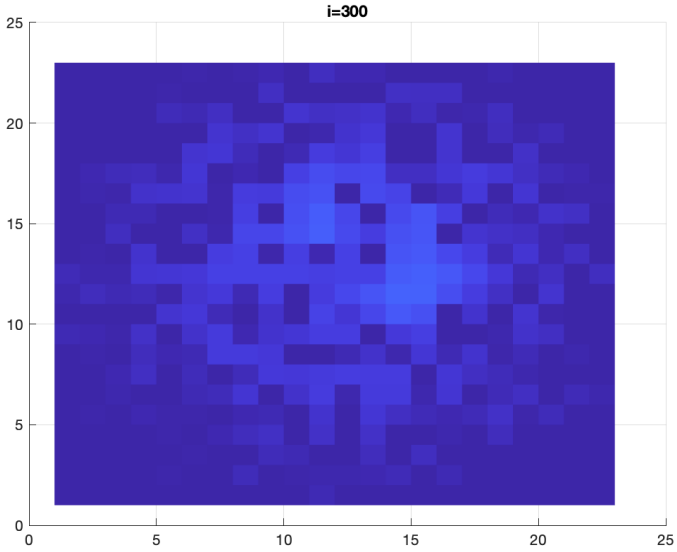

i=7
25
20
15
10
5
0
0
5
10
15
20
25


i=14
25
20
15
10
5
0
0
5
10
15
20
25


i=30
25
20
15
10
5
0
0
5
10
15
20
25


i=61
25
20
15
10
5
0
0
5
10
15
20
25


i=100
25
20
15
10
5
0
0
5
10
15
20
25


i=300
25
20
15
10
5
0
0
5
10
15
20
25

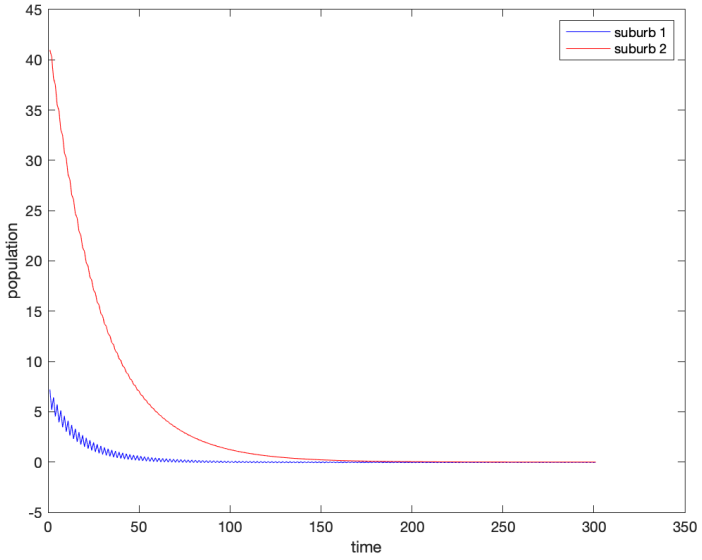


Figure 3.14: Suburb 1 vs. Suburb 2 when $S > a/b$

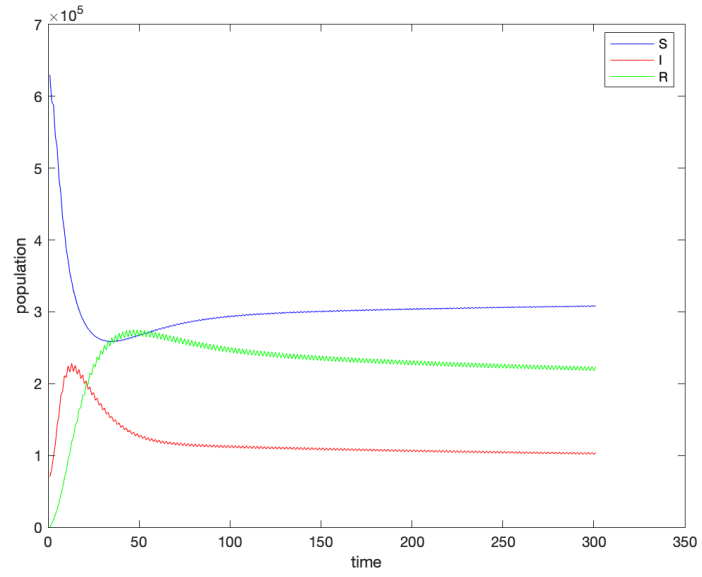


Figure 3.15: S, I and R with $S > a/b$ and travel

```
Stotal =

   6.3000e+05


aoverb =

  714.2857
```

We see with both the surface plots for infection and curve plots from Figure 3.15 that there is a spike infection and that the infection never goes away. Also none of the classes of populations ever die out since S is greater than our fixed point.

It may seem strange to the reader that we have a similar pattern compared to the $S < a/b$ infection case in suburbs in Figure 3.14, but keep in mind, these areas are sparsely populated. Thus Figure 3.16 takes a look at what happens in a third city block, that being the city center $I(12,12)$.

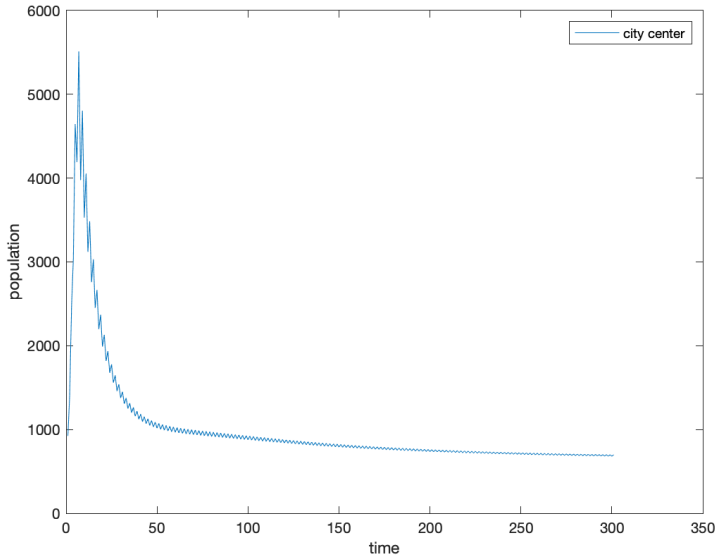


Figure 3.16: City center with $S > a/b$

Here we see what we would expect. That is, there is a large spike in infection before the disease dies back down. City blocks near the city center would resemble this sort of pattern. Similarly, as we get nearer to the outskirts, city blocks would resemble those of suburbs displayed in Figure 3.14. Now let's look at the comparison of S, I and R with $T = 1, 20, 30$ and 35 for comparison.

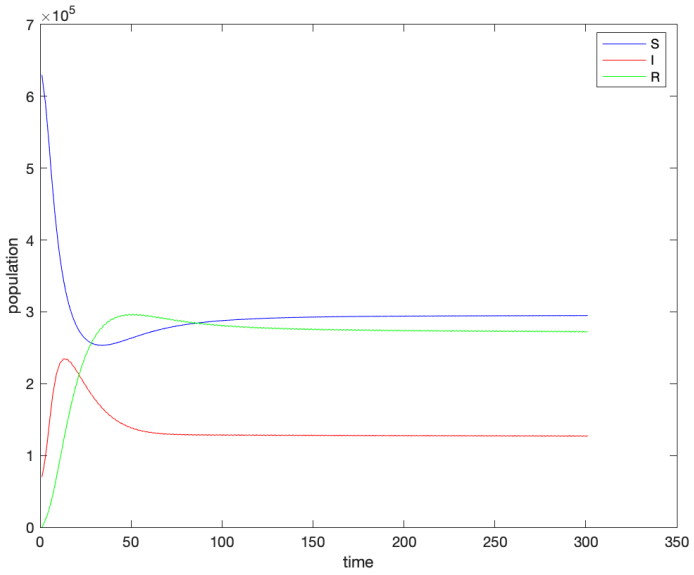


Figure 3.17: $S > a/b$ and $T = 1$

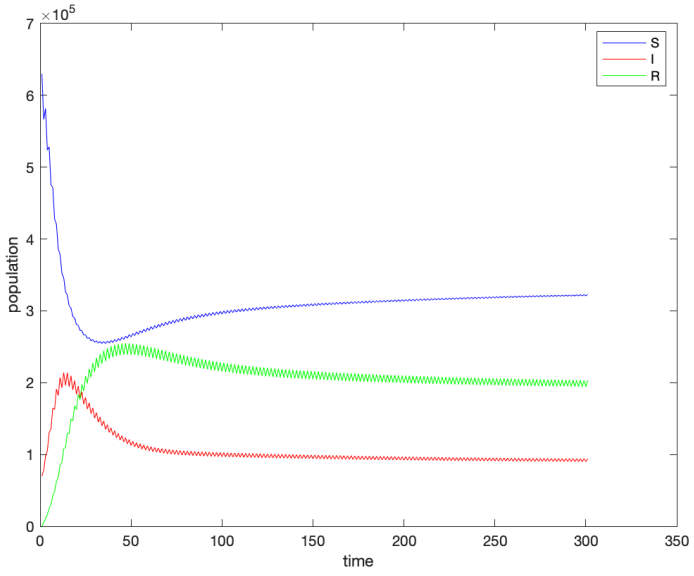


Figure 3.18: $S > a/b$ and $T = 20$

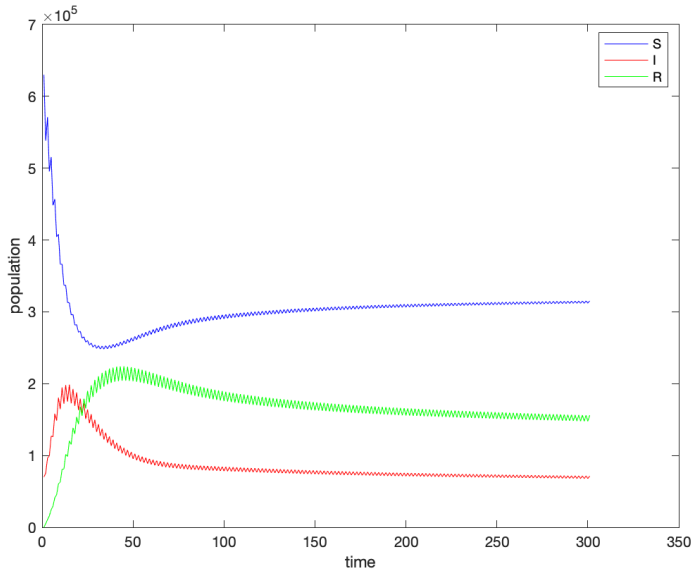


Figure 3.19: $S > a/b$ and $T = 30$

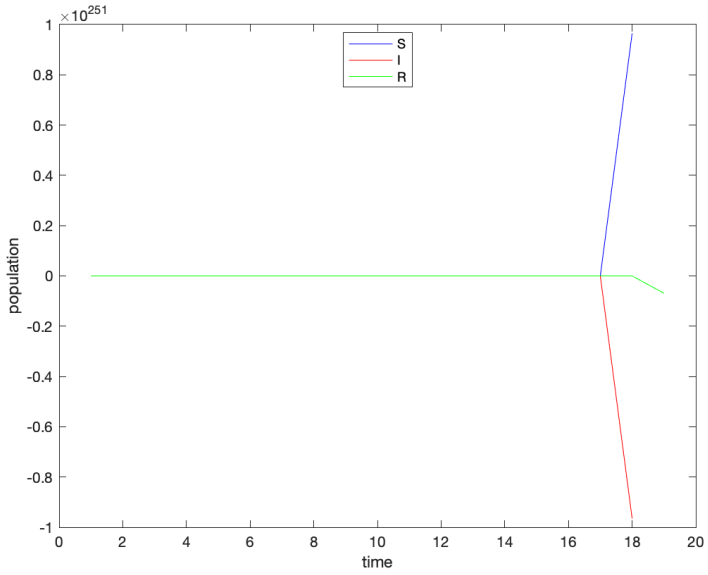
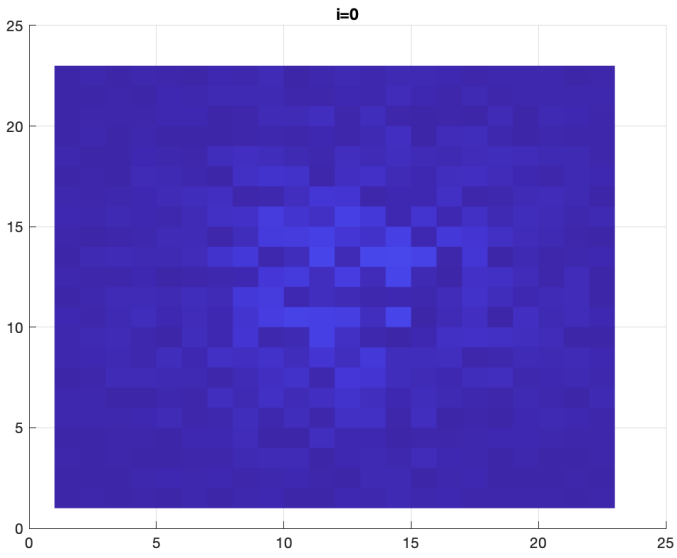
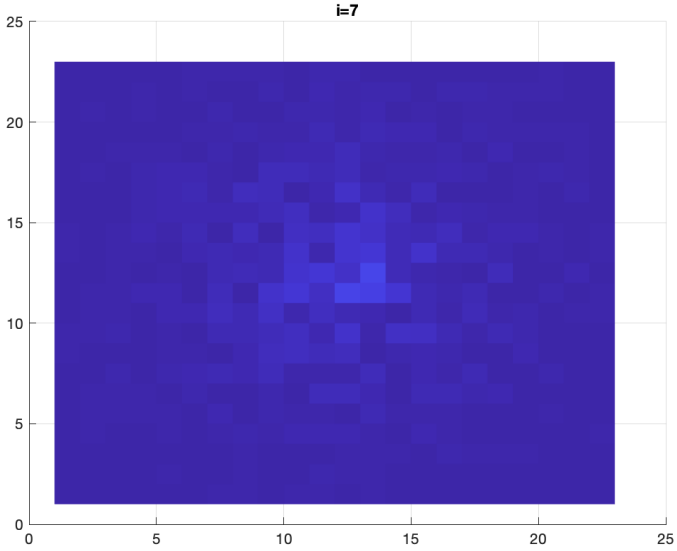


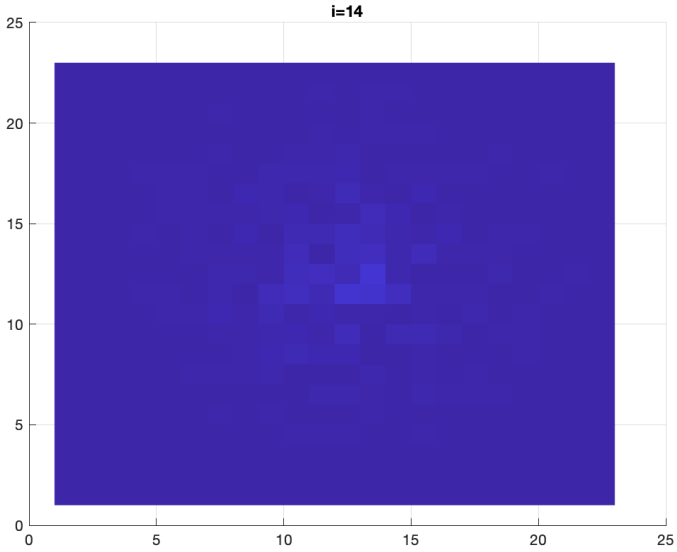
Figure 3.20: $S > a/b$ and $T = 35$

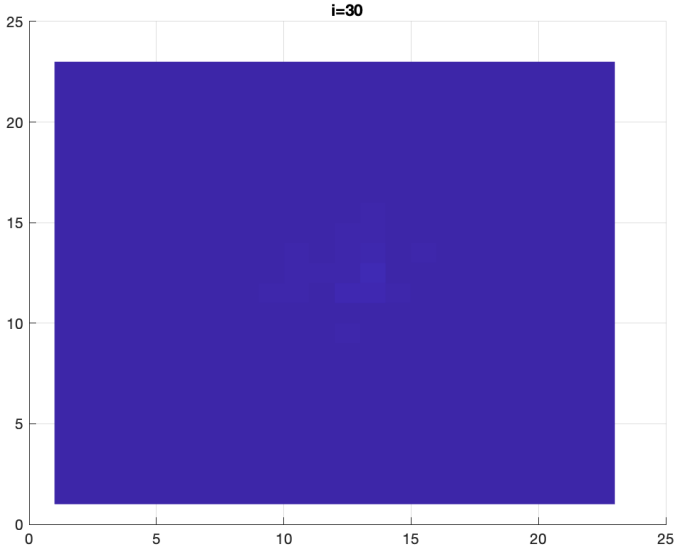
The analysis of these four plots is pretty similar to that of the $S < a/b$ case in that the dynamics of S, I, and R do not change with T changing. However for T greater than 10, we do see more of the jag effect in I and R, but less of it in S. This makes sense because infection is spreading more quickly, and thus we will have more infected and recovered individuals traveling around. Finally, we notice that the infected and recovered populations do not converge to zero. This is due to S being larger than the fixed point.

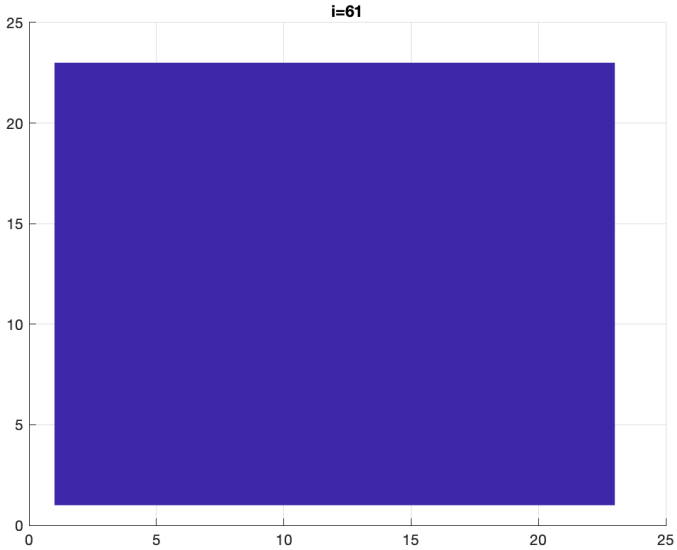
One more interesting simulation to consider is one where infected individuals do not travel. All we have to do to make this happen is to comment out the code for that updates the infection each time-step in our most recent CA. First we cover the case where the initial susceptible population is less than $\frac{a}{b}$. Hence we will keep the same parameters that went into the creation of Figure 3.9 as well as the surface plots proceeding it, and compare the difference. Below are the new surface plots followed by the curve plot with $T = 10$ (Figure 3.21).

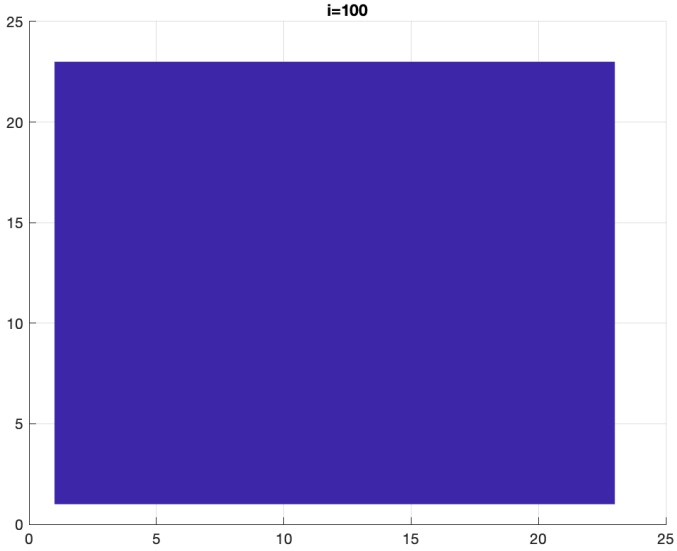


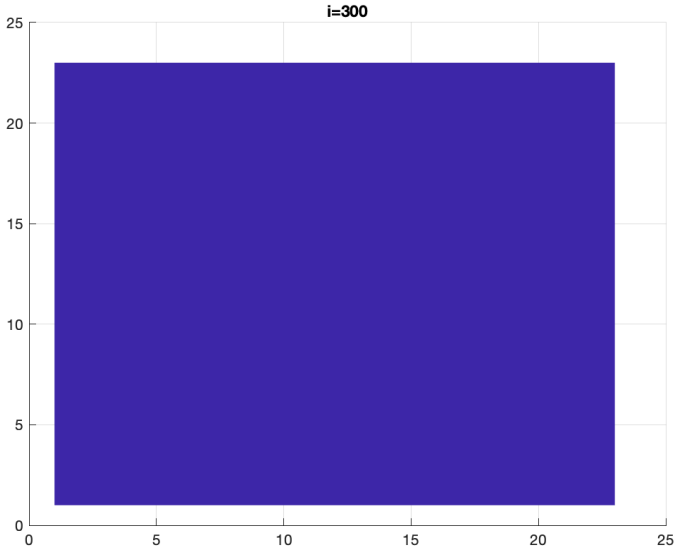

i=7
25
20
15
10
5
0
0
5
10
15
20
25


i=14
25
20
15
10
5
0
0
5
10
15
20
25


i=30
25
20
15
10
5
0
0
5
10
15
20
25


i=61
25
20
15
10
5
0
0
5
10
15
20
25


i=100
25
20
15
10
5
0
0
5
10
15
20
25


i=300
25
20
15
10
5
0
0
5
10
15
20
25

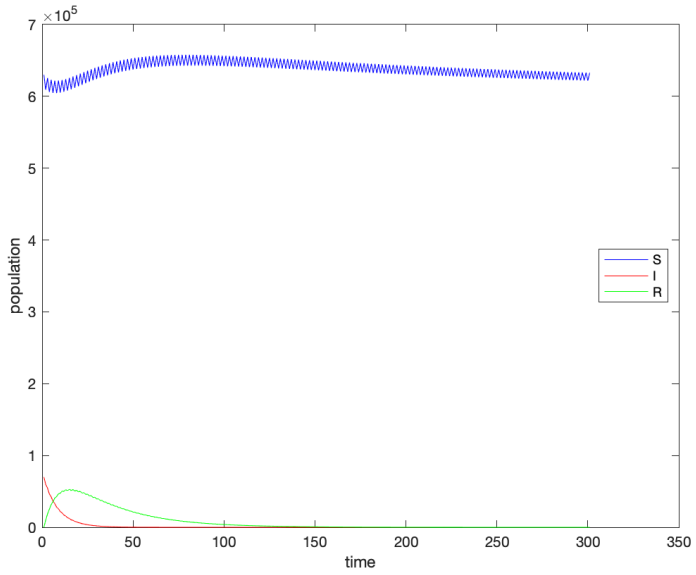
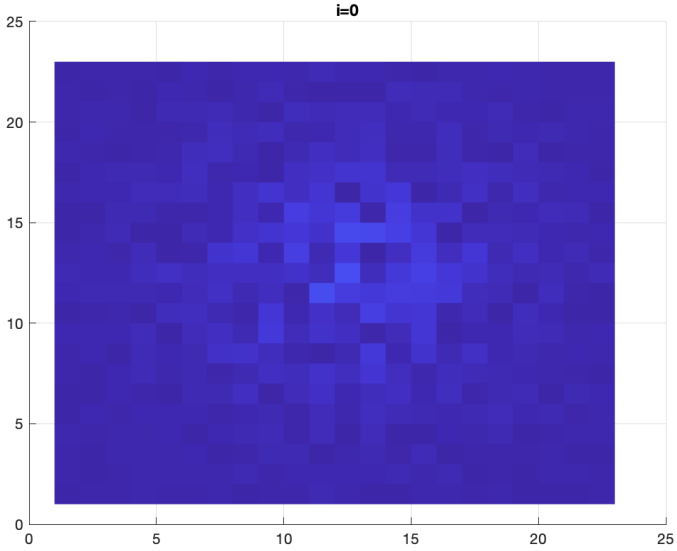
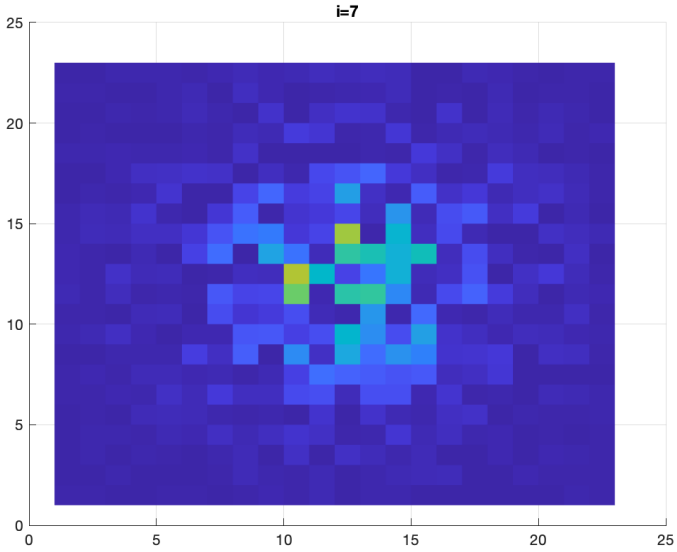


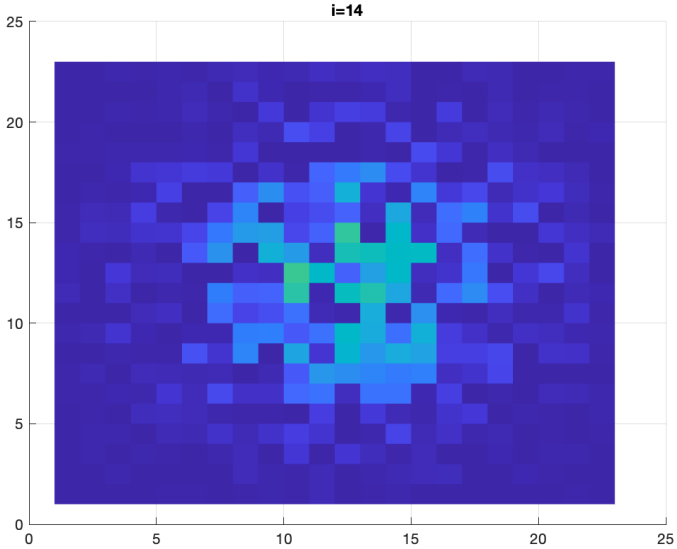
Figure 3.21: S, I and R with $S < a/b$ no infected travel

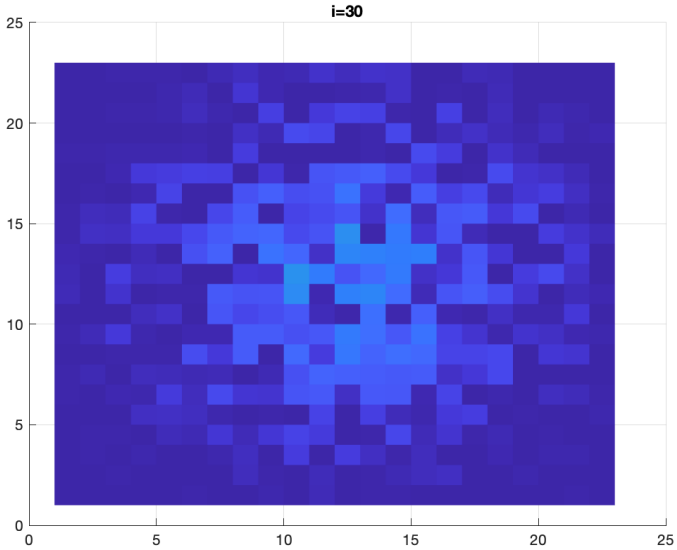
We notice that there is no spike in infection since the initial susceptible population is smaller than the fixed point $\frac{a}{b}$. Also, the infection curve is smooth since there is no traveling back and forth to the city center. I was expecting at least a somewhat different dynamic between the susceptible, infected and recovered populations, but will assume for now that there is not much change due to the fact that infection will not spike due to our parameters anyways.

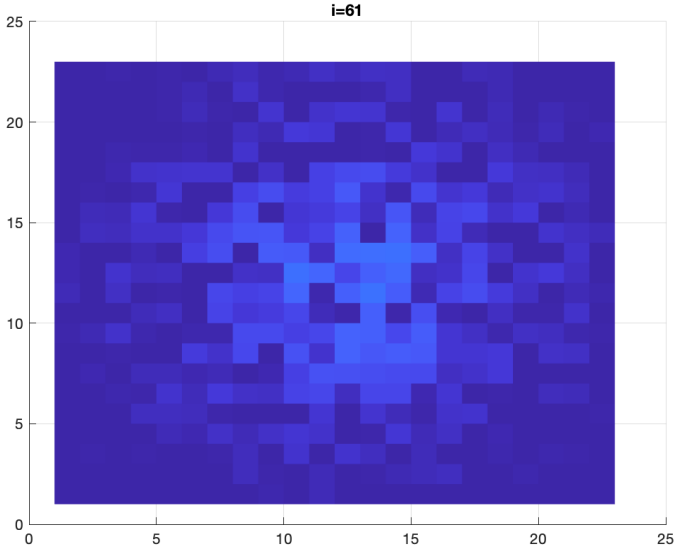
Now we cover the case where the initial susceptible population is more than $\frac{a}{b}$. Hence we will keep the same parameters that went into making the creation of Figure 2.15 as well as the surface plots proceeding it, and compare the difference. Below are the new surface plots followed by the curve plot with $T = 10$ (Figure 3.22).

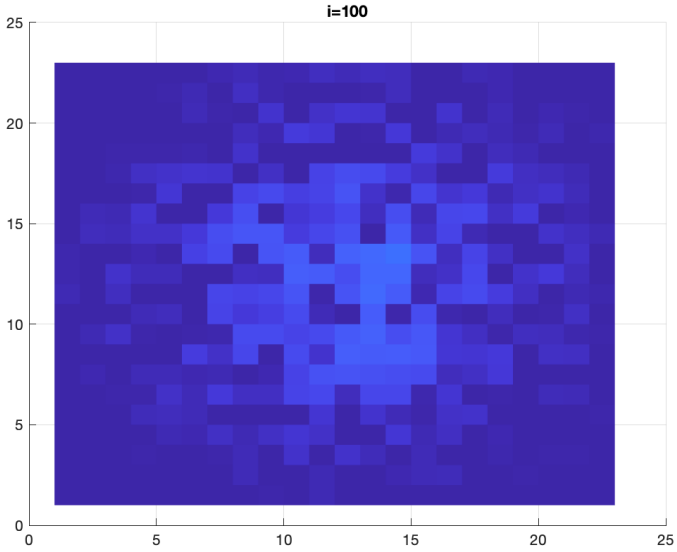

i=0
25
20
15
10
5
0
0
5
10
15
20
25


i=7
25
20
15
10
5
0
0
5
10
15
20
25


i=14
25
20
15
10
5
0
0
5
10
15
20
25


i=30
25
20
15
10
5
0
0
5
10
15
20
25


i=61
25
20
15
10
5
0
0
5
10
15
20
25


i=100
25
20
15
10
5
0
0
5
10
15
20
25

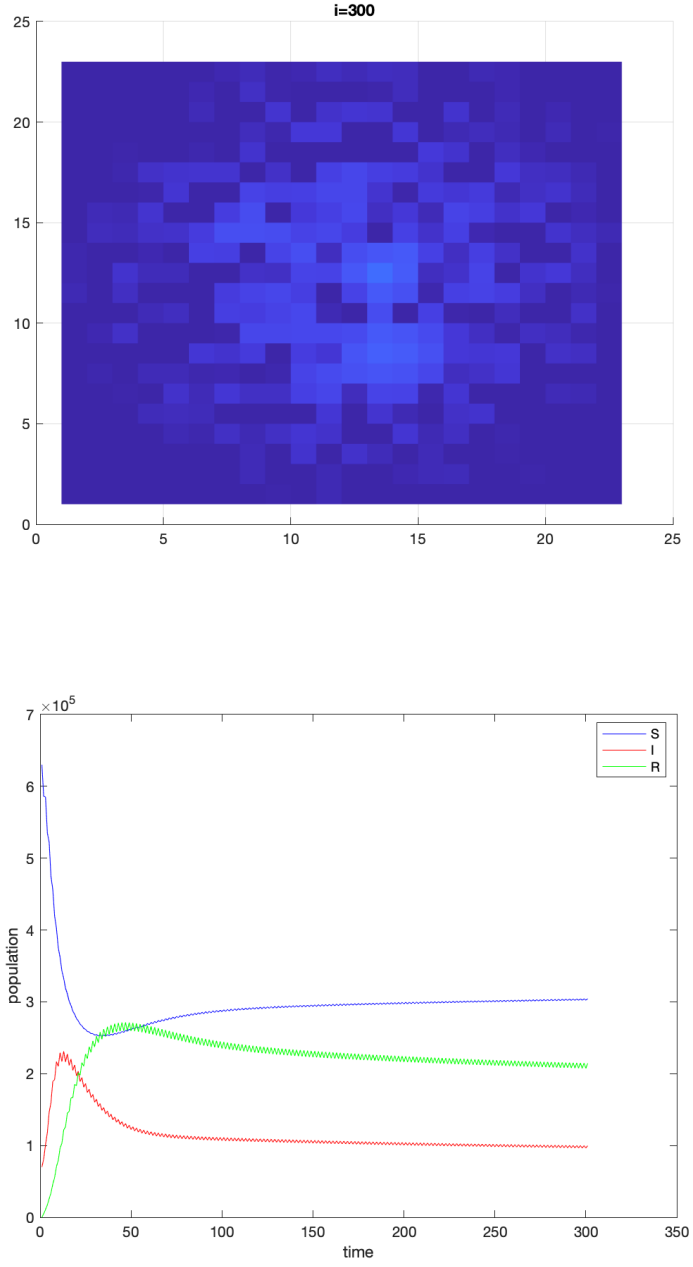


Figure 3.22: S, I and R with $S > a/b$ no infected travel

We notice that there a spike in infection since the initial susceptible population is larger than the threshold $\frac{a}{b}$. Also, the infection curve is smooth since there is no traveling back and forth to the city center. Again, as far as they dynamics between the susceptible, infected and recovered populations, I was surprised they do not seem to change from figure 2.15.

In the near future, this is what we want to investigate. If the dynamics are supposed to

stay similar from Figure 3.15 and 3.22, we need to study more into why this is. If they are supposed to differ, we need to figure out what to change in our code. We would need to do the same for comparing the dynamics between Figures 3.9 and 3.21.

3.3 SIR Model Represented as PDEs

Finally, we reach our end goal, which is to demonstrate the *SIR* model in the sense of PDEs. That is, we want our derivatives to represent large scale disease spread in a continuous setting. Note that now we will call our travel coefficient l instead of t to minimize confusion. Based off of our final CA on pages 50 and 51, we can represent our PDEs as

$$\begin{aligned}
\frac{\partial S(t,x,y)}{\partial t} &= -bS(t,x,y)I(t,x,y) + cR(t,x,y) + l\frac{\partial^2 S(t,x,y)}{\partial t} \\
\frac{\partial I(t,x,y)}{\partial t} &= bS(t,x,y)I(t,x,y) - aI(t,x,y) + l\frac{\partial^2 I(t,x,y)}{\partial t} \\
\frac{\partial R(t,x,y)}{\partial t} &= aI(t,x,y) - cR(t,x,y) + l\frac{\partial^2 R(t,x,y)}{\partial t}.
\end{aligned}$$

Chapter 4

Conclusion

4.1 Summary

To conclude, this thesis used some powerful tools to model disease spread. First we discussed a classic *SIR* model in an ODE sense. After understanding the parameters that make this model, we did a numerical and analytical example, followed by discussing how to go from a discrete to continuous case in time.

Later, we introduced Cellular Automata and how they can be practical in modeling disease spread in space and time. We built increasing more advanced CA in each stage of getting to our main goal of representing disease spread for the *SIR* model in a PDE sense. In our first CA, each cell of a matrix was an individual, and disease spread across a small setting such as a classroom. In our next CA, each matrix cell was a city block with different populations. In general, the closer to the city center, the more populated a block was. Disease only spread within each block. Our next CA introduced reinfection. Finally, our last CA implemented long term travel, meaning people would travel outside their city block and back home.

4.2 Future Considerations

There are plenty of interesting avenues that could be explored in the future. We could consider how disease spreads in a Hexagonal Neighborhood instead of a Moore Neighborhood. That is, each cell in our CA has six sides instead of four.

We could also consider what happens when we introduce new parameters in our CA that change the dynamics of our resulting PDEs. For instance, we could add in parameters for births/deaths, vaccinations, age, gender, etc...

Additionally, we could consider other variations of *SIR* models in our Cellular Automata. That is, we could consider a model such as the *SEIRS* (E would be a new category of exposed people) where now we have an additional PDE for E.

These are just a few suggestions of where to take this thesis from what we have already done. In essence, we always have the opportunity to improve mathematical models by adding more parameters as well as taking away more assumptions.

BIBLIOGRAPHY

[1] Duane Nykamp. “Math Insight.” “A discrete SIR infectious disease model - Math Insight". https://mathinsight.org/discretesirinfectiousdiseasemodel.

[2] Gerda de Vries, Thomas Hillen, Mark Lewis, Johannes Muller, and Birgitt Schonfisch. *A Course in Mathematical Biology: Quantitative Modeling with Mathematical and Computational Methods*. SIAM, 2006.

[3] Sheng Bin, Sun Gengxin, and Chih-Cheng Chen. “Spread of Infectious Disease Modeling and Analysis of Different Factors on Spread of Infectious Disease Based on Cellular Automata.” *International journal of environmental research and public health. U.S. National Library of Medicine*. https://pubmed.ncbi.nlm.nih.gov/31775236/.

[4] Systems Innovations. “Cellular Automaton - Youtube.” https://www.youtube.com/watch?v=eUcviWn9ACc.

[5] Lorenzo Sadun. “Running the Sir Model - YouTube.” https://www.youtube.com/watch?v=-eigCrevl0.

[6] Bernhard Pfeifer, Karl Kugler, Maria M Tejada, Christian Baumgartner, et al. “A Cellular Automaton Framework for Infectious Disease Spread Simulation.” The open medical informatics journal. Bentham Open, 2008. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2666960/.

[7] Rachel Thomas and Marianne Freiberger“Maths in a Minute: ‘R Nought’ and Herd Immunity.” *plus.maths.org*. (January 15, 2021). https://plus.maths.org/content/maths-minute-r0-and-herd-immunity.

[8] Tom Rocks Math. “Oxford Mathematician Explains Sir Travelling Wave Solution" https://www.youtube.com/watch?v=uSLFudKBnBI.

[9] Bentout Soufiane, Abdennasser Chekroun, and Toshikazu Kuniya. “Parameter Estimation and Prediction for Coronavirus Disease Outbreak 2019 (Covid-19) in Algeria.” *AIMS public health. AIMS Press*. (May 22, 2020). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7327392/.

[10] quantpie. “Calculating R0 from NGM - YouTube.” https://www.youtube.com/watch?v=xJ5mizenTII.

[11] Baojun Song. “Compute R_0 Using Next Generation Operator.” https://qrlssp.asu.edu/sites/default/files/brnmtbi2016.pdf

[12] James V. Lambers and Amber C. Sumner. *Explorations in Numerical Analysis*. World Scientific, 2018.

[13] Bjørnstad, O.N., Shea, K., Krzywinski, M. "The SEIRS model for infectious disease dynamics." *Nature Methods*. https://www.nature.com/articles/s41592-020-0856-2.