Charlie Colley

Numerical Analysis Final Project

Xiaozhe Hu

12/12/14

Abstract

In this paper, different methods of finding solutions to ordinary differential equations were tested on models of epidemiology. The models are proved to be Lipschitz continuous to guarantee that solutions would exist. In MATLAB a program was written to compute the solutions with the Forward Euler, Backward Euler, and Crank Nicolson method of numerically solving o.d.e.’s. Each method is discussed with facts about the methods, and some implementation issues that were had in this project. Graphs are produced for different types of simulations to provide a general sense of the behavior of the program with different arguments which would represent different epidemics. Finally the paper discusses what kinds of epidemics these simulations can model, what kinds that can’t be modeled, and what kind of corrections can be made to the model to encompass a wider variety of simulations.

Statement of Problem

Mathematical modeling of epidemiology has been around for a long time. Starting in the 1900’s mathematicians have endeavored to find a model for how infectious diseases spread. Building models enable testing of different solutions to manage or to eliminate epidemics that plague mankind. Because of the complexity involved in infectious diseases it’s much easier to make assumptions and build a model revolving these assumptions, this narrows the scope of the model to a smaller selection of diseases. However these models are no less useful. This paper will focus on modeling a specific branch called SIR models without vital dynamics. SIR models model three groups of populations Susceptible, Infectious, and Recovered. Ordinary differential equations are then used to track how the size of the groups change, and any equilibrium they may reach. Graphs will produced of different types of populations to see how the dynamics of a class of an epidemic behaves.

Assumptions of the Model

SIR models assume a number of things to narrow down to a select group of diseases

* Constant population, no births, deaths, immigration, emigration
* Virus has a constant rate of transmission
* Rate of recovery is constant
* Population is homogeneous, everyone interacts with everyone
* Once an individual recovers, they’re immune to the disease
* Time can be discretized, virus cannot spread in between time steps

Model Description

By the nature of how we solve this problem, the simulation of the epidemic can be broken into a time interval D, where D = [t0,tk] for k individual time steps. This forms a uniform grid for solving our ode’s.

For a population of N people, at any given time the entire population can be broken into three groups

S(t) ≡ the susceptible population R(t) ≡ the recovered populations I(t) ≡ the Infectious population

Note that from our assumption

There are three important constants for SIR models. The rate of transmission of the virus, the recovery rate of the virus, and the basic reproduction number. The rate of transmission models the likely hood that if an individual interacts with a member of the infectious population, what is the chance that they will catch the virus. The rate of recovery is a fixed constant that models the probability that a sick individual will recover. The basic Reproduction number is the ratio of expected new cases when a sick individual is exposed to all susceptible individuals, this number is a great way to characterize epidemics.

Rate of Transmission ≡ Rate of Recovery ≡ γ Basic Reproduction Number ≡

To build the model of how the population changes over time, we can put together the terms that have just been defined to create an intuitive model.

To model how the susceptible population changes, we can take the number of infected individuals and I(t) the number of susceptible individuals S(t) and think about the amount of interactions that a sick person may have. Let’s say that the average amount of people that a sick person may interact in a unit of time is **β**. Then the probability that one of the individuals a sick person meets is susceptible to the disease would be so the number of susceptible people getting sick per unit of time would be

Note that the derivative is negative because the number of susceptible members are always declining, because once an individual has recovered, they are immune to the disease.

The rate of which sick people recover is a simple relationship. Initially we defined the rate of recovery for a disease as a fixed constant γ, so the number of individuals that recover at any time is simply

Finally to calculate the rate of change for the infected individuals, recall the from the initial conditions

so differentiate in respect to t, to get rearrange,

So now we have all of the equations that are necessary to track all of the populations. For each one of these there will be an initial condition that is given when defining the system that the equations will be modeling.

**Proving Lipschitz Continuity**

The metric space being used is the Real number with the standard metric

Definition: a function f is Lipschitz continuous if a real constant such that Domain of f .

In this model the two functions that need to be proved as Lipschitz continuous are

for we rearrange the inequality to get substitute the full equation in and we get

because at all time steps each population is bounded by the total population N.

for we rearrange the inequality to get substitute the full equation to get

we can substitute the total population for the same reason as above.

Algorithms Used

For each step of the simulation, there are three differential equations that need to be solved. Each of the o.d.e.’s depend on different portions of the population, i.e. how many infectious people there are. In order to make sure that the population stays at a constant level throughout the entire simulation, the population sizes used for computing all of the derivatives at time step i are from (i-1)th step of the simulation. In theory, at each step of the simulation, the infectious, susceptible, and recovered populations would all change at the same time, but on a computer each ode needs to be solved one after the other. Three different methods were used to solve the odes forward Euler, backward Euler, and Crank Nicolson. All of these methods are discretized methods of solving o.d.e.’s . They solve the equation over a domain that is cut into segments called a grid. All of these methods are solving the o.d.e.’s on uniform grids, meaning that all the segments of the grid are of equal length.

Forward Euler:

This is a first order differential equation solver. This method takes the derivative of an equation and the initial value and takes constant approximations of an integral. The constant that is used to approximate the integral for the ith point is the value of the derivative at the (i-1)th point. This is an explicit method, meaning the solution at every point can be calculated . The solution of the o.d.e. is a series of points that are connected by tangent lines evaluated at each point. This method is zero stable, but not absolute stable. However this method lead to the best results for solving the SIR model.

Backward Euler:

This is a first order differential equation solver which is similar to the Forward Euler method of solving o.d.e.’s. The difference between the two methods, is that the constant value used to approximate the integral for the ith point, is the value of the derivative at the ith point. Meaning that in order to get the value of the solution at the ith point, you must solve the equation for that value. This makes the Backward Euler method an implicit method, which means that another function was needed to solve for the value at each step. For my implementation I used Newton’s method of solving non-linear equations, which lead to some issues with precision at each of the steps. Backward Euler is absolute stable, which means for long simulations with a lot of steps this is a better method.

Crank Nicolson:

This is a second order differential equation solver. The difference from the different Euler methods is that instead of a constant approximation of the integral used to solve the o.d.e. it used the trapezoid rule. This is an implicit method because the integral that’s being approximated is the integral of the derivative evaluated at the (i-1)th and ith points. In order to solve the equation at each step, Newton’s method was also used here. Crank Nicolson is a numerical stable method.

Plotting the graphs with the implicit method was quite difficult because of the computational cost of using Newton’s method. In order to find a solution, a specific tolerance was chosen for each of the algorithms. Either the tolerance wasn’t low enough, and the models would have floating point error which would make the net population increase (which contradicts a direct assumption about the model). Or in order to calculate the proper precision, the amount of iterations would greatly increase which meant the total run time of the program would take a while for large domains or fine grids.

For the plots below, they were produced with the forward Euler method, because after testing with all of the different methods, with a mesh of h = .01, the plots were all identical to the human eye. Finer meshes were tested, but the data did not change significantly from the h-.01 grid.

Results

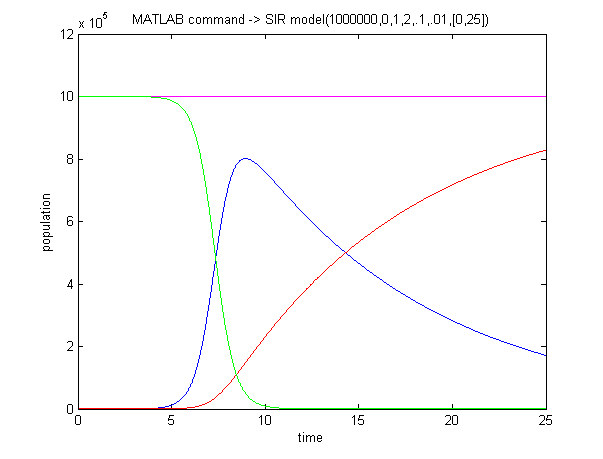
To run the model in MATLAB you must type

SIR\_model(initial\_S,initial\_R,initial\_I,b,Y,h,Domain)

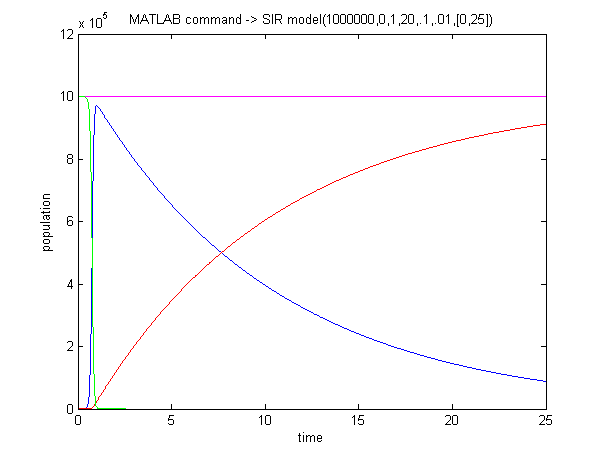
In the documentation of the function there are descriptions of each of the arguments.

For each plot, red lines correspond to the recovered population, blue lines correspond to the infectious population, green lines correspond to the susceptible population, and purple lines correspond to the total population.

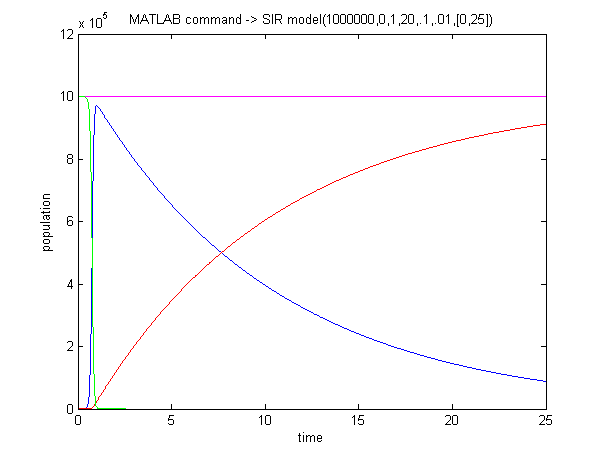
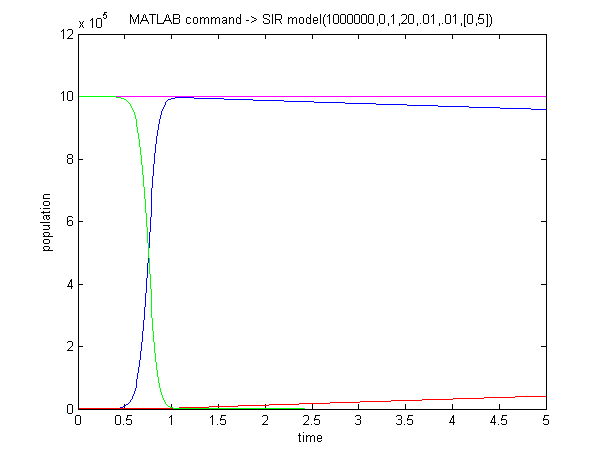
**FIGURE 1**

This model represents a small city of 1,000,000 people with a patient zero. In this model β = 2 meaning infected patients have few interactions

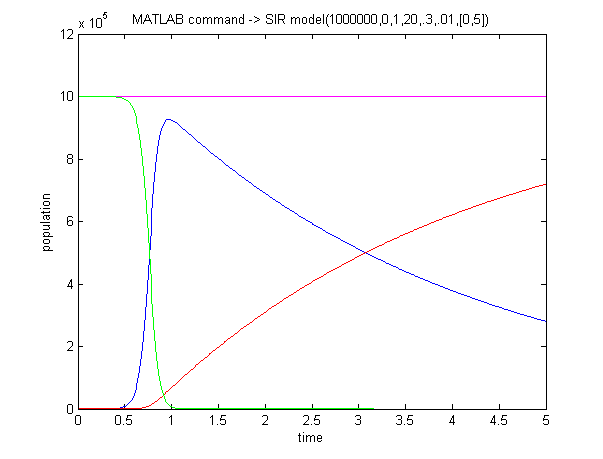
**FIGURE 2**

****This model is the same city but with a β = 20 , meaning that a sick individual would have 10 times as many interactions as FIGURE 1

**FIGURE 3**

This population is similar to FIGURE 2, but the γ = .01 .Everyone gets sick quickly, and it’s a slow recovery

**FIGURE 4**

This figure has a similar population to FIGURE 3, but instead has a γ = .3 , meaning a quick recovery rate

Model Discussion

One thing to note about the simulation is that the behavior of the models are quite similar for scaled populations. So if you scale each initial population, and the b and Y constants, then the graphs will look the same, only the factor of the y-axis will change.

The most important arguments for the graphs are the b(β) and Y(γ)arguments. These encompass how the virus will spread, and how quickly those afflicted can recover. The long time behavior of every simulation run will always be that the susceptible population and infected population will go to zero, and the recovered population will equal the entire population. This is because the recovery rate is a constant value, and those who recover are immune to the virus.

Because this model does not include mortality and births, the kind of epidemics that it can model will be scoped to smaller epidemics over the course of a few months. Colds and flu strains, things that spread easily from person to person but will not hinder a person from continuing meeting people and spreading the virus are better suited to be modeled by these equations. Possibly social trends could be modeled in this fashion. Trends start with a select few individuals, they spread quickly when they catch on, they’re non-lethal (hopefully), and when someone “recovers” from a trend they’re usually not going to pick it back(immunity).

The most important information that these models will tell you, are the peaks of the infection populations. Some epidemics will spread to the entire population if there’s a lot of mixing between people in the society, and if there’s a slow recovery rate then it’s possible that the entire population will be infectious at some point. Other simulations may have low mixing, and a high recovery rate which would mean that a smaller portion may receive the virus. Based off preliminary data, these models can be used to predict how likely the virus is to spread and whether or not action should be taken to introduce vaccination or raise awareness about spreading the disease.

Conclusion

While these graphs make for interesting visuals, they encompass only a small portion of epidemiology and are limited in what they can tell you about an epidemic and what kinds of precautions a society should take. What these models are good for, are a spring board to more complex models. The next steps in development should be to create functions that would determine whether or not a sick individual would be able to catch the virus again (or a similar strain). Having sick individuals becoming immune to the virus after recovering limits the long term behavior of the graphs produced because with a constant recovery rate, everyone will eventually get better and the recovered populations will always tend to the total population, while the infectious and susceptible populations will tend to 0. If recovered individuals can get sick again, this means that different equilibriums can be attained. Other things to consider are to introduce mortality and births into the society, allowing the model to encompass more serious epidemics. Vaccinations can also be added in, which would alter the way the recovered and susceptible populations behave, allowing medical staff to test the effectiveness of different vaccination programs.

References

Forward Euler:

-class slides

-http://en.wikipedia.org/wiki/Euler\_method

Backward Euler:

-class slides

-http://en.wikipedia.org/wiki/Backward\_Euler\_method

Crank Nicolson:

-class slides

-http://en.wikipedia.org/wiki/Crank%E2%80%93Nicolson\_method

SIR models:

-http://www.maa.org/publications/periodicals/loci/joma/the-sir-model-for-spread-of-disease-the-differential-equation-model

-http://mathworld.wolfram.com/SIRModel.html

-http://en.wikipedia.org/wiki/Compartmental\_models\_in\_epidemiology