**Project #4 - Numerical Methods - Spring ‘18**

*Created By: Zheng Zong & Clarisa Leu-Rodriguez*

*Group #5 - June 7th, 2018*

*Dr. Willett - Tacoma Community College*

**Purpose of Project:**

The purpose of this project is to analyze the *SIR* model for a closed population considering two different numerical methods for solving differential equations discussed in class. We will solve the system of interest using the Fourth Order Runge-Kutta Method and the adaptive *ode45()* method built into MATLAB, which is a versatile ODE solver used to solve non-stiff differential equations. We will determine the most efficient method of the two by comparing the runtimes and ease of setup of each.

Additionally, we will analyze how varying the initial conditions and given parameters affects the maximum number of infected people at any given time and the overall trend/behavior of the model.

**Background Information & Description:**

Computational epidemiology is the application of numerical methods, mathematics, and computer science to model the spread of a disease. One of the standard models is called the *SIR* model. In this model, the assumed closed population is divided into three classes and is based on the proportion of the individuals of interest compared to the total population. So, we’ll take *S* to be the proportion of the population that is susceptible to the disease, *I* to be the proportion of the population infected by the disease, and *R* to be the proportion of the population recovering from the disease. In which case, this implies: *S + I + R =1 (*assuming a closed population). In this project, to simplify the math, in like matter we’ll work with the simplified version of the model, in which the birth and death rates of the population are the same and the population remains constant over time.

**The SIR Model:**

We’ll be analyzing the *SIR* model including a seasonally dependent infection rate. The model of interest is:

The parameter is the birth rate, which is assumed to be equal to the death rate for the population. The parameter is the recovery rate of the population. The function:

is the seasonally dependent infection rate. Here is the initial infection rate and *T* is the length of the infection season in months.

We’ll assume the initial conditions for the population and the following set parameters as our baseline model:

* Baseline Initial Conditions:
* Baseline Parameters:

To analyze the model of the population over time, we will create a plot of *S, I,* and *R* over the first 48 months.

**Methodology & Run Times:**

The Runge-Kutta methods are a family of implicit and explicit iterative methods, which include the well-known Euler’s method, used in temporal discretization (integration over time) for the approximate solutions of ordinary differential equations. The Runge-Kutta methods use a trial step at the midpoints of an interval to cancel out lower-order error terms. For this project, we used the classical fourth order Runge-Kutta to get values of Susceptible people, values of infected people, and values of recovered people. This method was lengthy to implement and had average run time of 0.007766126524889 seconds over 10 trials on Clarisa’s computer and 0.02172 seconds over 10 trails on Zheng’s computer.

*ode45()* is a built-in non stiff ODE solver in MATLAB. This method was easy to implement but did require us to implement the SIR model in a separate *.m* file. It had an average run time of 0.009262620943258 seconds over 10 trials on Clarisa’s computer and 0.02242 second over 10 trials on Zheng’s computer. After determining that Runge-Kutta methodwas the faster of the two methods over several trials but, *ode45 ()* was the easiest to implement, we decided to use that method primarily when varying the initial conditions and parameters to compare to our baseline model. Also, for better accuracy, we set the relative tolerance about .

**Baseline Graphs**

As it can be seen, both methods produce the same graphs. If we increase *h* (the step size), the graph of the *SIR* model using the Fourth Order Runge-Kutta method starts to look smoother and there is less spacing between each point, as there are more points to graph. However, the Runge-Kutta still missing some points on the graph, it implies the Kunge-Kutta methods has less stiffness compared to the ode45() method.

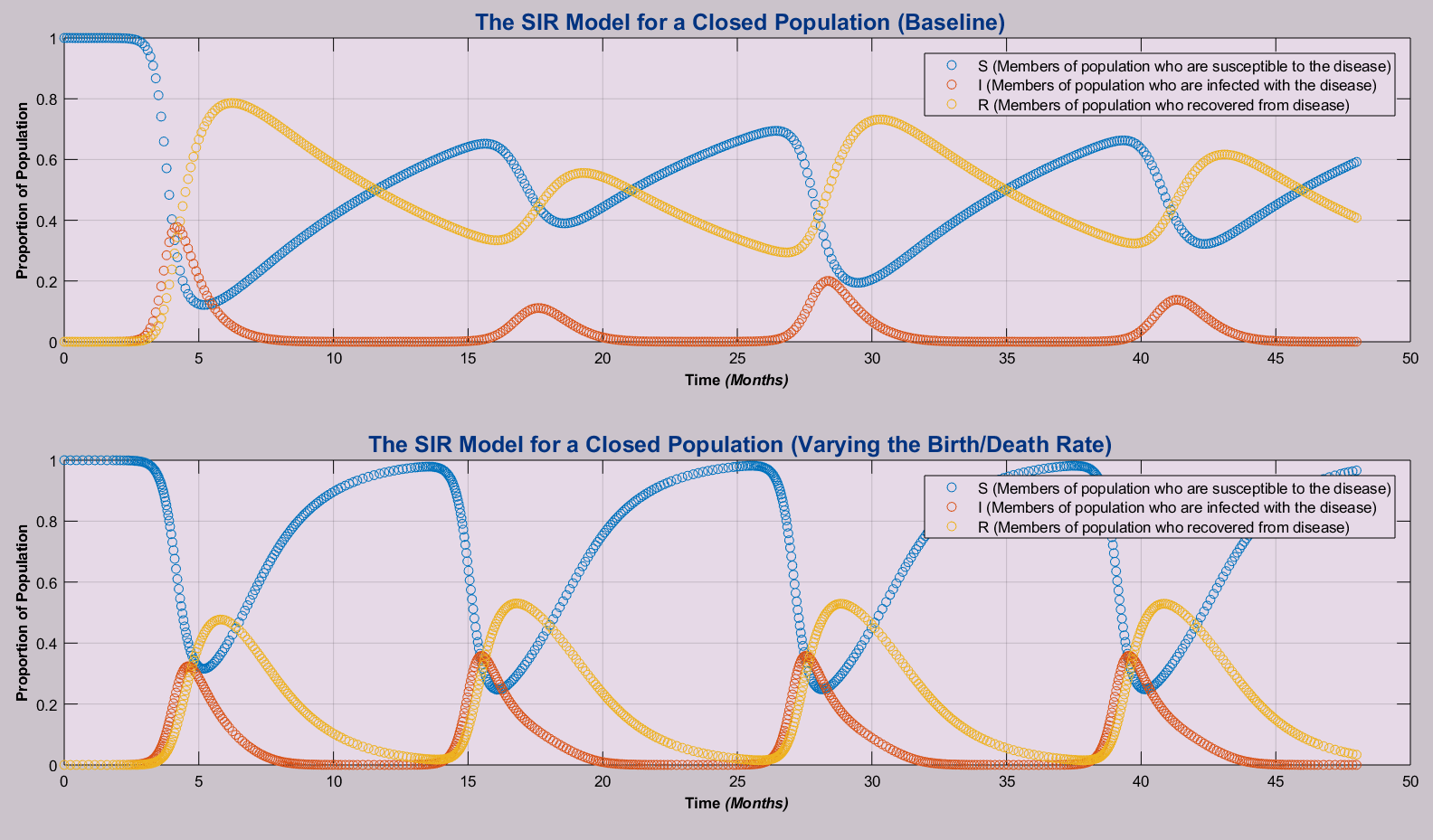
**Varying the Initial Conditions S (0) and I (0):**

To analyze how sensitive the model is to changes in the initial conditions, we adjusted

*S (0)* and *I (0)* accordingly while holding everything else constant with respect to the given baseline values. We set *S (0) =0.199999* and *I (0) =0.800001* accordingly as *S+I+R=1* where *R=0*. *S (0)* is the initial proportion of individuals susceptible to the disease and *I (0)* is the initial proportion of individuals infected with the disease.

Analyzing how the change of *S (0)* and *I (0)* effects the overall behavior of the model, it seems that the overall trend still behaves the same as our baseline model with few variances over the initial time intervals. Upon analysis, decreasing *S(0)* whilst increasing *I(0)* slightly increases *I(17.5)* and *I(42.5)* and decreases *I(28)* which are approximate points of the maximums of *I* value.

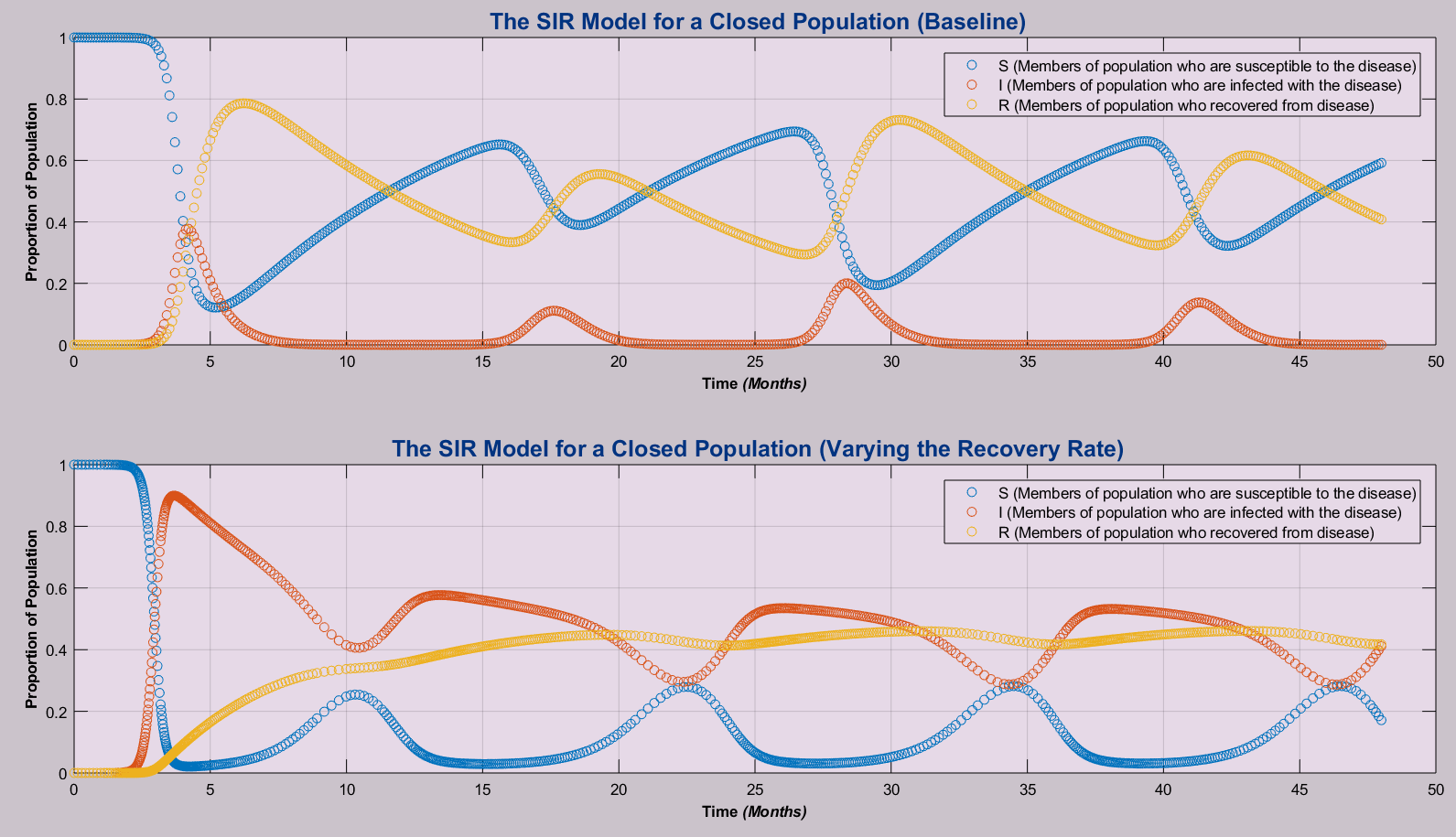
**Varying the Birth/Death Rate of the Population:**

To analyze how sensitive the model is to changes, we also varied the birth/death rate of the population while holding everything else constant with respect to the given baseline values. We set accordingly.

Analyzing how changing affects the overall behavior of the model, it can clearly be seen that if than the maximum of *S, I*, and *R* will always be held at a constant value for each maximum found over the interested time intervals. The maximums of *I* also increase in relation to increasing the birth/death rate of the population.

**Varying the Recovery Rate of the Population:**

While analyzing how sensitive the model is to change, we also varied the recovery rate of the population while holding everything else constant with respect to the given baseline values. We set accordingly.



Analyzing how decreasing affects the overall behavior of the model over the interested time interval and the maximum values of *I*; decreasing the recovery rate increases the maximum proportion of infected individuals at any given time. It can also be seen from the above graph that *S* and *R* decrease in relation to this as expected.

**Conclusion:**

In conclusion, after analyzing the runtime of each numerical method used, the fastest method was Runge-Kutta method*.* The easiest method to implement was also *ode45()* in our opinion. After analyzing how the model behaves when varying *S(0)* and *I(0)*, decreasing *S(0)* increases *I(0)* and therefore increases the maximum proportion of members infected with the disease in the population. Setting the birth/death rate to 0.5 also has an interesting effect on the model, as the maximums of *S, I,* and *R* are held at a constant value over time. Also, increasing the birth/death rate increases the maximums of *I.* Decreasing the population’s recovery rate as increases the maximums of *I.* Our favorite part of this project was learning more about computation epidemiology and learning more about how to model systems in MATLAB.