

CTA200 Assignment 2

Xiaoyi Ma

May 8 2020

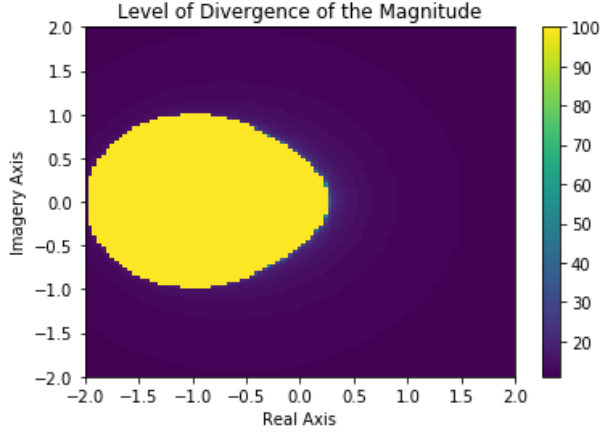
1 Question 1

1.1 Method

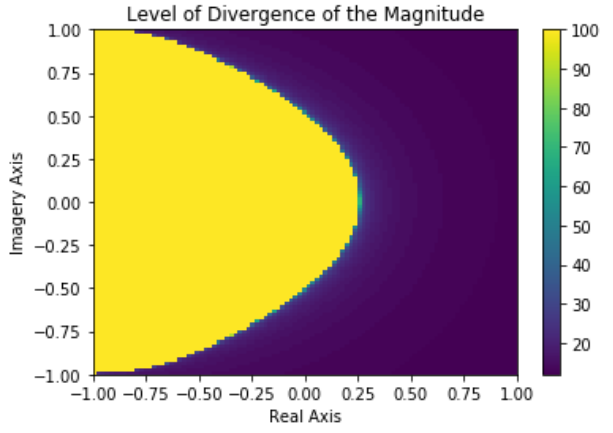
A function called `draw` is created to plot the image with input for the range of x and y . Inside the function, there are 100 values for each x and y in the assigned range. There are 10000 values of c is calculated as $c = x + iy$ for each x and y . For each c value, an array for 100 values of z is calculated as $z_{i+1} = z_i^2 + c$ [2]. The convergence of this sequence and index of first diverged value is determined with `if` statement. The index of diverged value is recorded in a matrix according to the x and y index for c value. For the converged sequence, the index of diverge is assigned to be the length of the sequence as 100. The image of the c values with x -axis as real axis and y -axis as imagery axis and the color density for index of diverge is plotted with `pcolormesh` in `matplotlib` package. The function is used for four different ranges that is shown in Figure 1 below.

1.2 Analysis

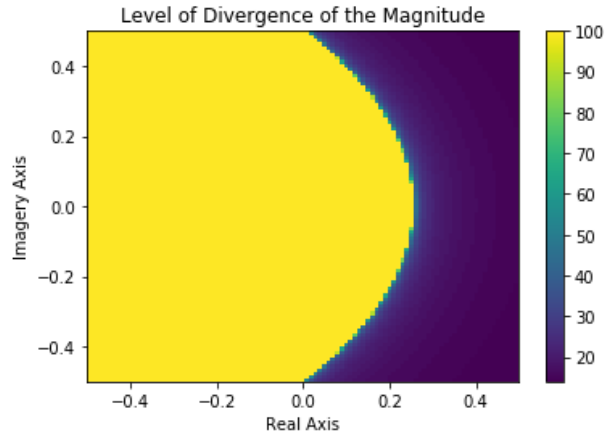
In order to make the image to show more accurate distribution for the convergence, the number of values for x and y is choose to be 100, which results 10000 c values that is large amount of data points for distribution. Since most of the z magnitude sequence diverge fast within 20 values as shown in the diagram, 100 points for z magnitude sequence is enough for show the convergence result for the sequence. From Figure 1, we can know that there are more than half of the point with x and y in the range of -2 and 2 diverge within first 20 values. There are some points in the "egg" shape with x range around -2 to 0.3 and y range around -1 to 1 that are converge. As zooming in x and y scales, from Figure 1a to 1d, a region of the sequence becomes "more" converge from 0.3 to 0.23 is found at the tip of the egg shape in Figure 1d.



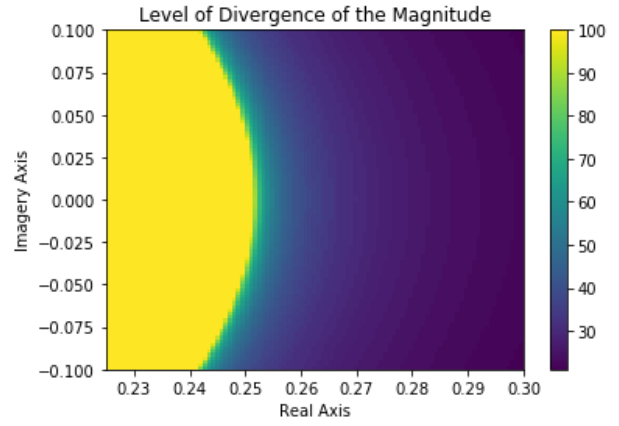
(a) Range of -2 and 2 for both x and y



(b) Range of -2 and 2 for both x and y



(c) Range of -0.5 and 0.5 for both x and y



(d) Range of -0.0225 and 0.3 for x

Figure 1: Level of Convergence of Z Sequence for Each C Points

2 Question 2

2.1 Method

This system of ODE is solved with `scipy.integrate.solve-ivp` function, with the default method explicit Runge-Kutta method of order 5.[1] First, the initial and final value of time is defined and the array of time points with step size 1000 is created. The value of β and γ is defined with different four pairs of number. The initial condition for susceptible (S), infected (I) and recovered (R) function is defined in an array, and N as total number of population is unchanged as 1000.[2] The function called SIR is defined with output of the S,I,R function in an array. [2] The SIR system of function, initial and maximum value of time, initial condition for S,I,R and the time array is input into `solve-ivp` function and the result S,I,R is output as `solve-ivp.y`. The result array for S,I,P is plotted into the same graph with `plot` in matplotlib package, that is shown in Figure 2.

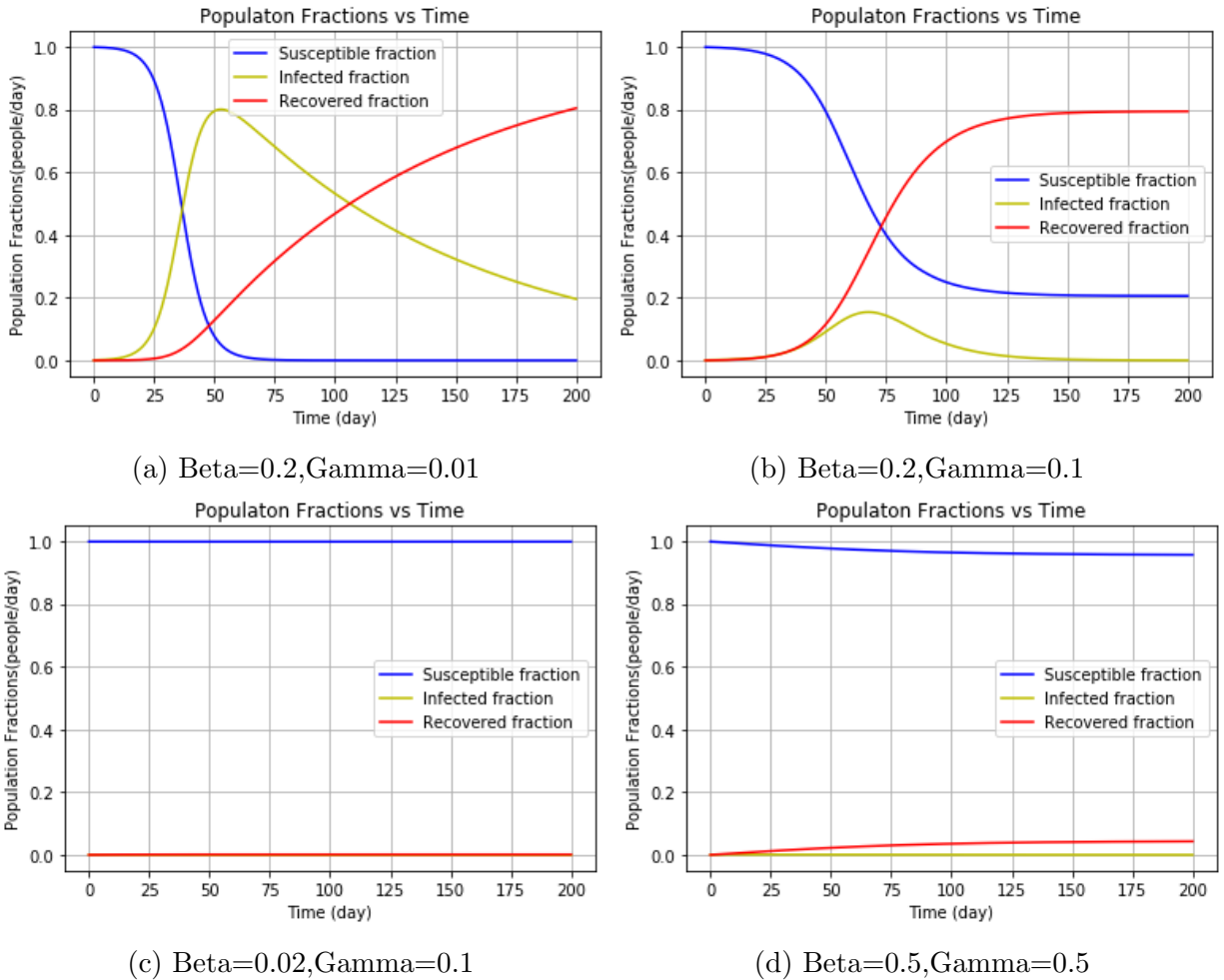


Figure 2: Population Function vs Time for SIR Model

2.2 Analysis

From the functions of S,I,R, we notice that β is infected rate. When $\beta = 0.2$, it means 20 % of people who are susceptible will infected in the unit of time. γ is the recovery rate. When $\gamma = 0.1$, it means 10 % of people who are infected will recover in the unit of time. By comparing Figure 2a and Figure 2b, we know that as the recovery rate γ decreasing, the infected population increase and decrease much slower with respect to time. The recovery fraction is also increase slower with respect to time. From Figure 2c, we know that when the infected rate β is low, the infected fraction is also very low. From Figure 2d, we know that when β and γ are low, the infected fraction is also very low and recovery fraction increase slowly with respect to time.

2.3 Bonus

For the death parameter D, if the death rate(mortality) is denoted as δ , the time derivative of the death parameter is $\delta \cdot I$. The infected parameter is the population fraction that is infected but neither dead or recovered, therefore its time derivative needs to minus the derivative of recovered and death parameter. The susceptible parameter does not affect by the recovered or death parameter. According to the SIR function in the previous question, the new model SIRD with death parameter is the following functions. [3]

$$\begin{aligned}\frac{dS}{dt} &= -\frac{\beta SI}{N} \\ \frac{dI}{dt} &= \frac{\beta SI}{N} - \gamma I - \delta I \\ \frac{dR}{dt} &= \gamma I \\ \frac{dD}{dt} &= \delta I\end{aligned}\tag{1}$$

According the above equation (1), the new SIRD model can be defined in python. By following the same procedure as the SIR model, the plot of SIRD model can be generated which is shown as Figure 3. In this case, β is 0.2, γ is 0.02, and the mortality δ is 0.006.

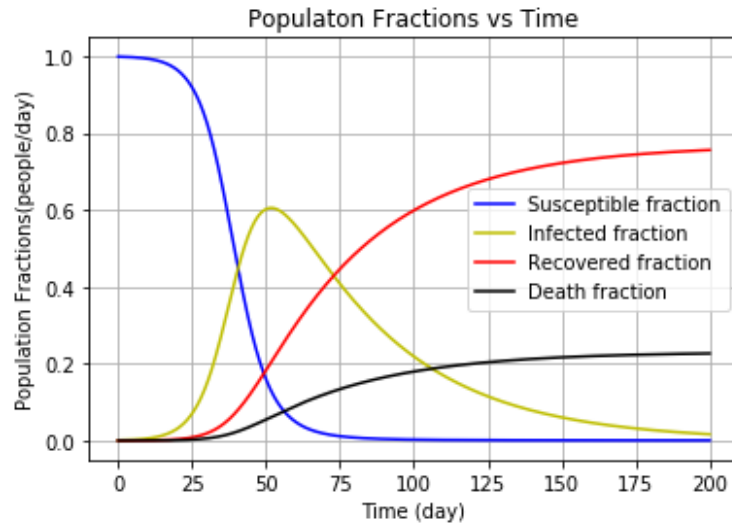


Figure 3: Population Function vs Time for SIRD Model

As we can see from the graph, according to the mortality and above equation (1), the graph of death fraction is increase slower than the recovered fraction with same trend. It means that as the time increase, the recovered population is becoming more greater than the death population.

References

- [1] The SciPy community. `scipy.integrate.solve-ivp`-Scipy Reference Guide. Dec 2019.
- [2] Fergus Horrobin. CTA 200 Assignment 2. May 2020.
- [3] Amenaghawon Osemwinyen and Aboubakary Diakhaby. Mathematical modelling of the transmission dynamics of ebola virus. *Applied and Computational Mathematics (New York)*, 4:313–320, 07 2015.