CTA200H 2021 assignment 2

Rosalind Liang

May 2021

1 Numerical approximation of a derivative

1.1 Methods

To numerically estimate the derivative at a given point x0, I defined a derivation function taking the the original function (sin(x)), x0 (.1), and stepsize (h), and outputting the value of the taylor expansion with these parameters. Next I created an array of h values spanning from .01 to 1 and passed each value in the array through the derivation function. For each resulting numerical derivative, I used the formula for error to compare against the analytical derivative (cos(.1)). Finally, I used matplotlib to plot the values of h against error (figure 1).

1.2 Results

As seen in Figure 1, we observe that as h increases, the numerical estimation of the derivative strays further and further from the analytical solution in an exponential like curve. This makes sense since larger values of h mean straying further and further from the local region around x0.

Figure 1: Caption

2 Divergence of functions in the complex plane

2.1 Methods

Part 1: To locate the coordinates for which the sequence diverges, I used a for loop to pass through 100 iterations of the sequence and collected points of divergence with np.nan(z), which recognizes when either the real or imaginary component of z is no longer a recognizable number (suggesting divergence to infinity). Matplotlib was then used to plot these coordinates (figure 2).

Part 2: A similar for loop was used with np.isfinite(z) to count the number of iterations each coordinate took before diverging. For cases where divergence did not occur within 100 iterations, a special value of -20 was assigned to create contrast in the colourmap, before plotting with matplotlib again (figure 3).

2.2 Results

The resulting plot of part 1 is shown in Figure 2, where all points of divergence are in yellow and points that remain bounded are in purple. The plot for part 2 is shown in Figure 3, where all bounded points are in purple and the rest of the plot is a colourmap of each point's divergence iteration. We observe a fractal shaped zone where the sequence stays bounded, ranging from -1 to 1 in the complex axis and -2 to .5 in the real axis. In figure 2, we see that iterations of divergence are significantly higher around the contours of this zone, and drop to approach zero moving away from it.

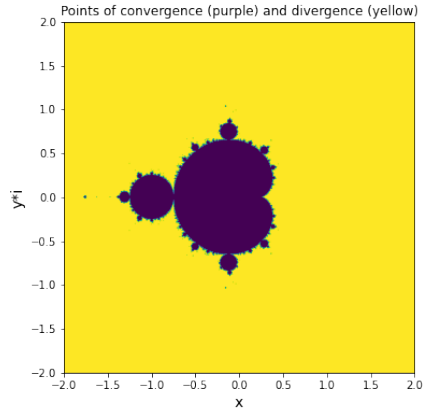

Figure 2: Convergent vs divergent zones

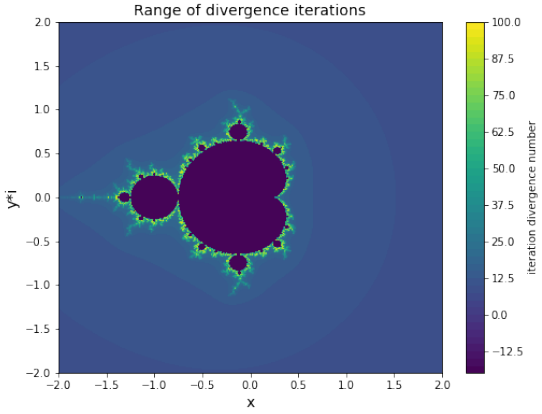

Figure 3: Divergence iteration number

3 The SIR model

3.1 Methods

To integrate each of the ODE's, I defined the model as a function and used odeint from scipy to perform the integration. I then played around with different values of β and γ before selecting a few which generated interesting plots.

3.2 Results

An important observation is that β is the rate of infection and γ is the rate of recovery. In real life pandemic applications, β is almost always greater than γ. As shown in figures 4-6, the ratio between β and γ determines the total proportion of the population infected, as well as the time it takes for the recovered population to plateau. Figure 7 shows a case where γ is 0, meaning the entire population is quickly infected with no one ever recovering.

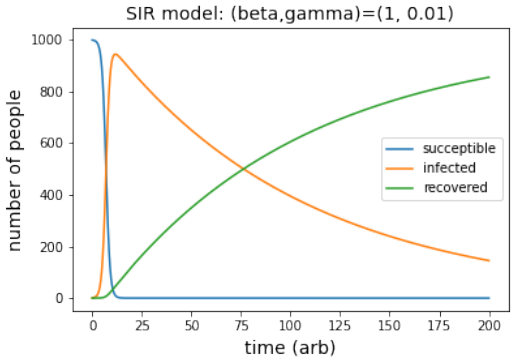

Figure 4: infection:recovery = 1:.001

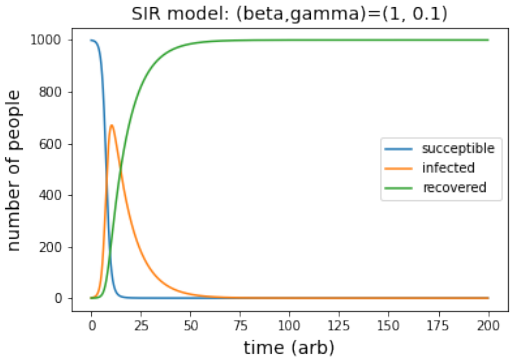

Figure 5: infection:recovery = 1:.01

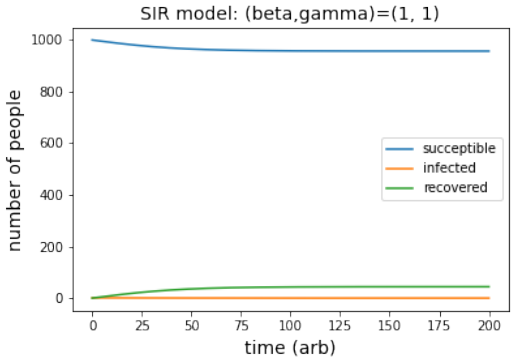

Figure 6: infection:recovery = 1:1

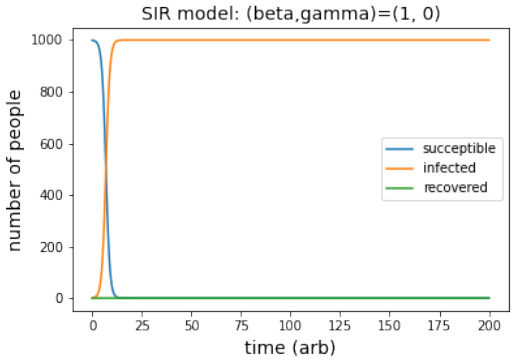

Figure 7: infection:recovery = 1:0