

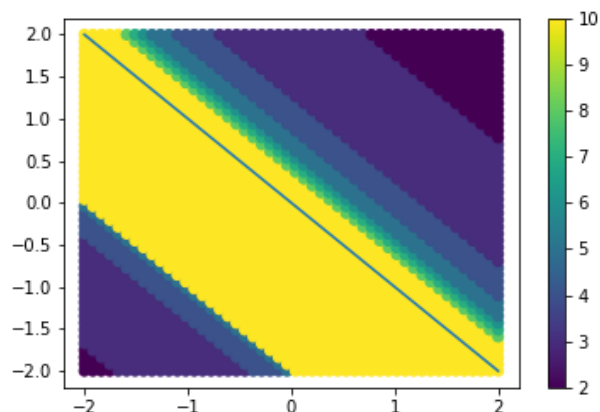
CTA200 Assignment 2

Winter 2020

Jeff Shen

May 7, 2020

Question 1



Methods

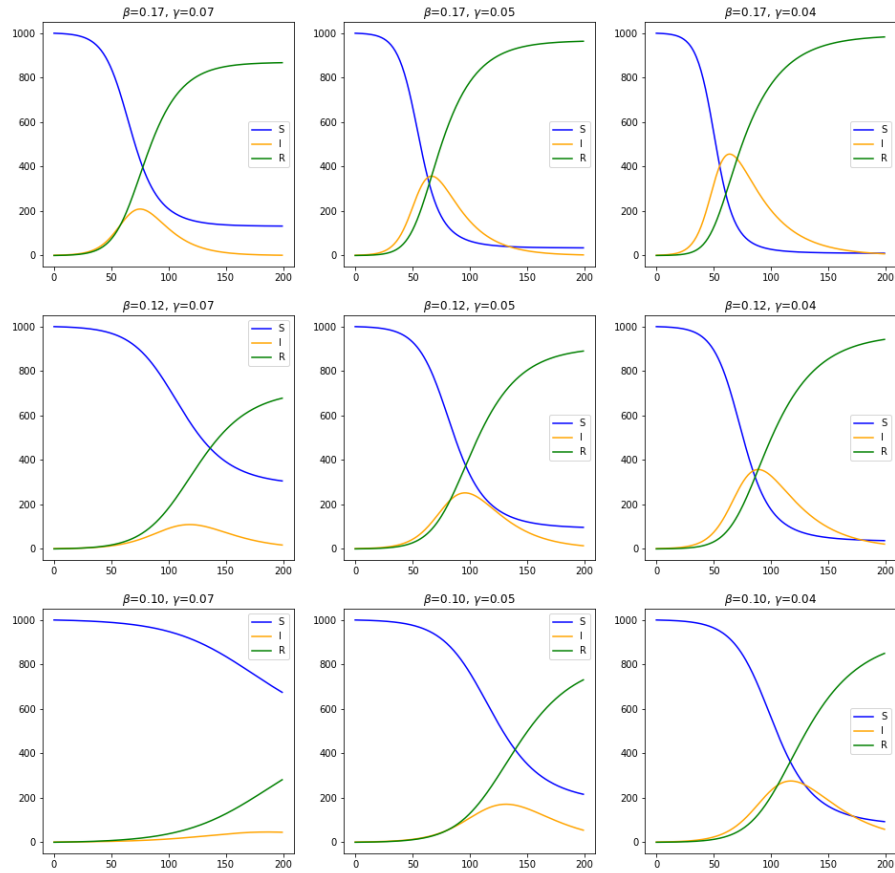
Basic numerical packages were imported in Python, which were used for this and the next question. Two 1D arrays, each sampling 50 points from -2 to 2, were created. One was multiplied by $1j$ in order to make it an array of complex values. Using these two arrays, `np.meshgrid` was used to create a grid of (50, 50) points in the appropriate range. The array was flattened into a (50x50, 1) shape, which then served as c from the question. 10 iterations was the maximum number that does not cause an overflow, and so that is the number that was selected. Each new column was created with the equation $z_{i+1} = z_i^2 + c$ according to the instructions, where each z and each c is a column of 50x50 values. The iteration number at which a given point diverged was calculated by setting a divergence threshold of where the 2D z array was greater than 10. This didn't

seem to differ much with a threshold of 1000. These numbers seem reasonable since they are greater than $|z|^2$ would be for convergent values. Then, a mask was created with `np.where`, which indicated the iteration number for when some row (representing a single c value) crossed the threshold. The original (50, 50) grid was then plotted using matplotlib, with the iteration number set as the colour.

Analysis

There is a distinct pattern to the iteration number at which a value diverges. Note that values coloured with a divergence iteration number of 10 (in yellow) just didn't diverge after 10 iterations. Points along a fixed diagonal $y = -x + c$ seem to have the same behaviour (either all converge, or all diverge at the same iteration number). Moving in a direction orthogonal to this diagonal, the further out the points are from the origin, the fewer iterations it takes for the value to cross the divergence threshold.

Question 2



Methods

`odeint` was imported from `scipy`. A derivatives function was created: it takes parameters y (a tuple of S, I, and R), t (array for integration time), β , γ , and N , and returns the derivatives dS/dt , dI/dt , and dR/dt according to the appropriate ODEs. The initial conditions were set according to the instructions. $1/\beta$ is the expected number of days it takes for an infected person to infect one other person, and the values $1/6$, $1/8$, and $1/10$ were selected for β (ie.

6 days to infect someone else, etc.). $1/\gamma$ is the number of days it takes for an infected person to recover, and the values of $1/14$, $1/21$, and $1/28$ were selected for γ (ie. it takes 2 weeks to recover, etc.). I'm not an epidemiologist and have no experience with disease modelling, but these numbers don't seem wildly unreasonable. For each combination of parameters, the ODE integrator integrated all the way to $t = 200$, and the S, I, and R lines were plotted. In the resulting 3x3 grid of plots, each row has the same β value, and each column has the same γ value.

Analysis

For a fixed row, moving from left to right shows the impact of a decreasing γ value. It is apparent in all three throws that such a decrease causes the I line to peak more dramatically, and to begin this peaking at an earlier time t . For a fixed γ (fixed column), decreasing the β value (going down a column) makes the I curve flatter, in the sense that the "bump" is more spread out. Given the definitions of β and γ from the Methods section, neither result is surprising.