

# CTA200 Assignment 2 Question 3

Caleb Lammers

May 11, 2020

## Question 1.

### Methods

To create the graphs shown below, this program iterates through points  $c$ , determining for each whether  $|z|$  diverges and plotting the point accordingly. In more detail, for  $c = x + iy$ , the program starts with  $x = -2 + \text{incr}$  (for some small  $\text{incr}$ ) and checks the divergence of  $|z|$  for  $y = -2 + \text{incr}$ ,  $y = -2 + 2 \cdot \text{incr}$ , ...,  $y = -2 + n \cdot \text{incr}$  until  $y = -2 + n \cdot \text{incr} \geq 2$ . Then, the program repeats this whole process for  $x = -2 + 2 \cdot \text{incr}$  and so on up until  $x = -2 + n \cdot \text{incr}$  (note that it would be more efficient to use `np.meshgrid`, however, this is more intuitive). To check the divergence of  $|z|$  for a given point  $c$ , the program recursively calculates  $z$  by  $z_{i+1} = z_i^2 + c$  with  $z_0 = 0$ . If at some point in the first 50 iterations  $|z| > 10$  we conclude that  $|z|$  will diverge. In fact, this is actually overkill as it is well known (and can be shown relatively easily) that if  $|z| > 2$ ,  $|z|$  certainly diverges.<sup>1</sup> The choice of 50 iterations is somewhat arbitrary but for high numbers of iterations there are few difference in the image, due to the limited resolution (there are few noticeable differences with 100 or even 1000 iterations). For higher resolution images, a smaller  $\text{incr}$  value is required, however, this increases the amount of computation. Using `matplotlib.pyplot`, in graph 1, the points for which  $|z|$  diverged were plotted in white and those which did not were coloured black. In the second graph, points for which  $|z|$  did not diverge were again plotted in black but those that did were coloured using a `colourmap` according to the number of iterations until  $|z| > 10$ . The third graph was obtained by zooming into graph 2.

1 - <http://mathforum.org/library/drmath/view/68518.html>

## Analysis

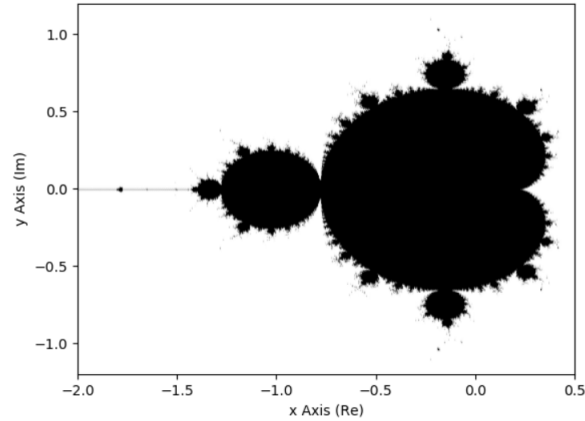


Figure 1: Black and white graph (incr = 0.002)

In the black and white graph, we can see that the points for which  $|z|$  did not diverge are around the origin, with most points having a negative real part. With the recursive equation in mind, this makes sense as we are repeatedly adding  $c$ . That is, if  $c$  itself has a very large magnitude, it is clear that the recursion equation will diverge. As we are squaring  $|z|$ , if  $c$  has a negative real part, adding  $c$  will decrease the magnitude of  $z$  (as long as  $z$  does not have too large of an imaginary part). Furthermore, even if you do not recognize this as the Mandelbrot set, it is visually clear that the graph has additional interesting properties. This is especially noticeable after colouring the diverging points.

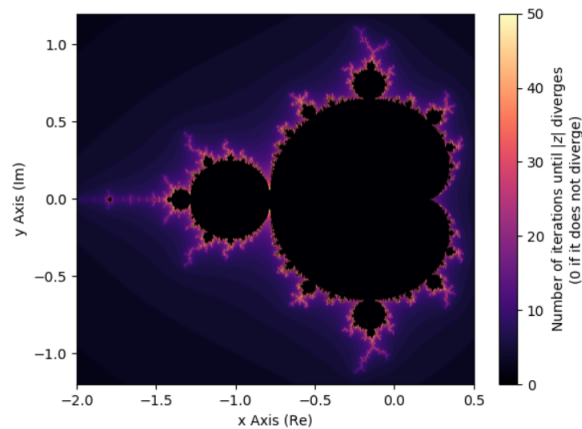


Figure 2: Coloured graph (incr = 0.002)

We see that a higher number of iterations is required to determine whether  $|z|$  diverges for points near the perimeter of the figure and far fewer iterations are required for points far from the perimeter, which is to be expected. Additionally, the coloured graph is quite aesthetically pleasing. One important observation to note is that it appears as though there are smaller copies of itself branching from the original shape.

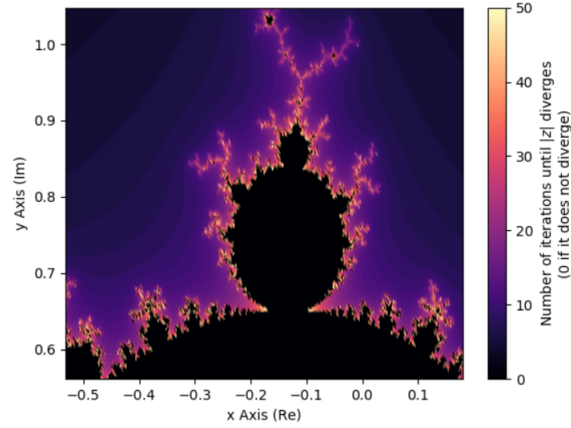


Figure 3: Zoom in on Figure 2 (incr = 0.002)

In the zoomed image, we can especially see the so-called "self-similarity" of the Mandelbrot set that makes it a fractal. In fact, in this image we can even see smaller similar shapes branching from itself again. This actually continues indefinitely and there are many videos on the internet zooming into the Mandelbrot set at a specific point to very high level of magnification. Indeed, the Mandelbrot set has become a staple of mathematical beauty.

## Question 2.

### Methods

Using the Scipy ODE solver and motivated choices of parameters, graphs representing the idealized spreading of disease can be created. Using the SIR model ODEs, the derivatives of  $S$ ,  $I$ , and  $R$  can each be calculated at time  $t$ . By specifying  $t_0$ ,  $\beta$ ,  $\gamma$ ,  $t_{end}$ ,  $dt$  and initial values for  $S$ ,  $I$ , and  $R$ , Scipy's ODE solver is used to numerically solve the functions. Then, graphs are created using matplotlib.pyplot. From equation (1), we see that  $\beta$  is controlling the rate at which the number of susceptible members of the population is decreasing. That is, roughly,  $\beta$  is controlling how infectious the disease is. In equation (2), we see that  $\gamma$  corresponds to the rate at which people people recover from the disease. With these interpretations in mind, we will be able to choose reasonable values

for  $\beta$  and  $\gamma$  depending on the situation. To incorporate deaths ( $D$ ) into the model, notice that in this model recovered and dead members of the population are analogous. So, the new equation is:

$$\frac{dD}{dt} = \lambda I$$

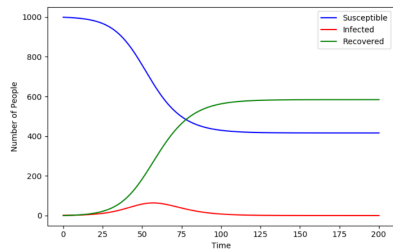
Of course, the addition of this equation does not affect equation (3) or equation (1) directly (which is clear by the analogy to equation (3)). However, now we have

$$\begin{aligned} 1000 &= S + I + R + D \\ \Rightarrow 0 &= \frac{dS}{dt} + \frac{dI}{dt} + \frac{dR}{dt} + \frac{dD}{dt} \\ \Rightarrow \frac{dI}{dt} &= \frac{\beta SI}{N} - \gamma I - \lambda I \end{aligned}$$

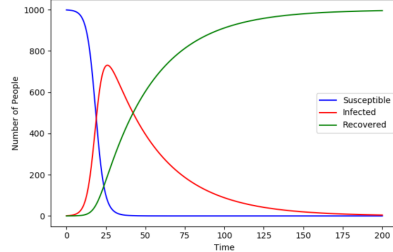
This makes sense, as people who die from the disease can no longer spread the disease, which should decrease  $I$ .

## Analysis

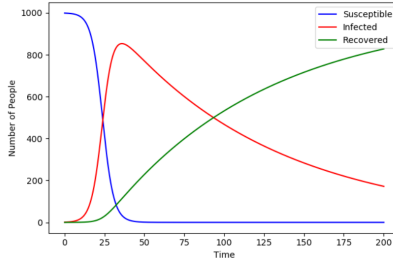
Depending on the values of  $\beta$  and  $\gamma$ , over the course of  $t = 0$  to  $t = 200$ , varying proportions of the population will be infected and recover/not recover.



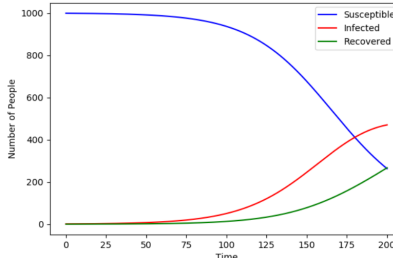
(a)  $\beta = 0.3, \gamma = 0.2$



(b)  $\beta = 0.4, \gamma = 0.03$



(c)  $\beta = 0.3, \gamma = 0.01$



(d)  $\beta = 0.05, \gamma = 0.01$

Figure 4: Plots of  $S$ ,  $I$ , and  $R$  for different motivated  $\beta$  and  $\gamma$

Different physical situations require varying values of  $\beta$  and  $\gamma$ . Figure 4. a) depicts a disease that has a high recovery rate and a somewhat low infection rate. This is modelled by a medium  $\beta$  and a relatively high  $\gamma$ . Figure 4. b) shows a disease that has a high infection rate and a recovery rate that is high enough for everyone to recover but low enough that everyone is infected. This type of disease requires a relatively high  $\beta$  and a medium  $\gamma$ . Next, Figure 4. c) represents a disease with a somewhat high infection rate but a low recovery rate, modelled with a medium  $\beta$  and a small  $\gamma$ . Lastly, Figure 4. d) shows a disease with a low infection rate and a low recovery rate, which requires a low  $\beta$  and  $\gamma$ . Of course, there are many other examples of physically possible situations that each require their own modifications to  $\beta$  and  $\gamma$ .

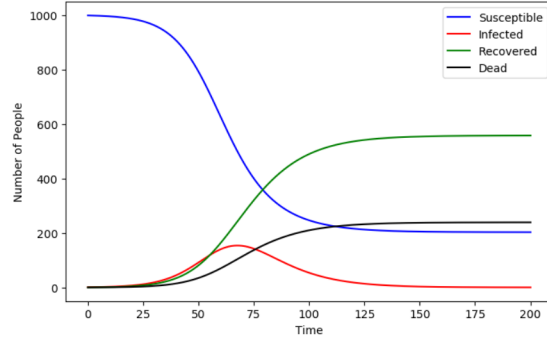


Figure 5: Plot including deaths with  $\beta = 0.2, \gamma = 0.07\lambda = 0.03$

When also considering deaths, there even more possible circumstances. Above is the graph of a disease with a relatively low infection rate, high recovery rate, and medium mortality rate (the parameters required are as you would expect). Despite starting with idealized equations, it is interesting to see the utility they provide in modelling something as complex as the spread of a disease.