

# CTA200 Assignment 2

Hayley Agler

May 8, 2021

## Question 1

### Methods

The goal of this question is to compute the derivative of the function  $f(x) = \sin(x)$  using two slightly different numerical approximation methods. I defined functions for both the method for infinitesimal stepsize  $h$ , which I called method 1:

$$d_x f|_{x_0} \approx \frac{f_{x_0+h} - f_{x_0}}{h} \quad (1)$$

and the method for finite stepsize  $h$ :

$$d_x f|_{x_0} \approx \frac{f_{x_0+h} - f_{x_0-h}}{2h} \quad (2)$$

as well as the function  $f(x)$ .

I also defined a function for the analytic derivative of  $f(x)$  and a function to compute the error compared to this analytic derivative by calculating  $\text{abs}(d_{\text{numerical}} - d_{\text{analytic}})/d_{\text{analytic}}$ . Then I created an array of  $h$  values, two empty arrays for the error values to be stored in, and wrote a for loop to evaluate the error of the methods for each of the  $h$  values. Finally, I plotted the error of each method as a function of the stepsize  $h$  on a loglog plot with matplotlib.

### Results

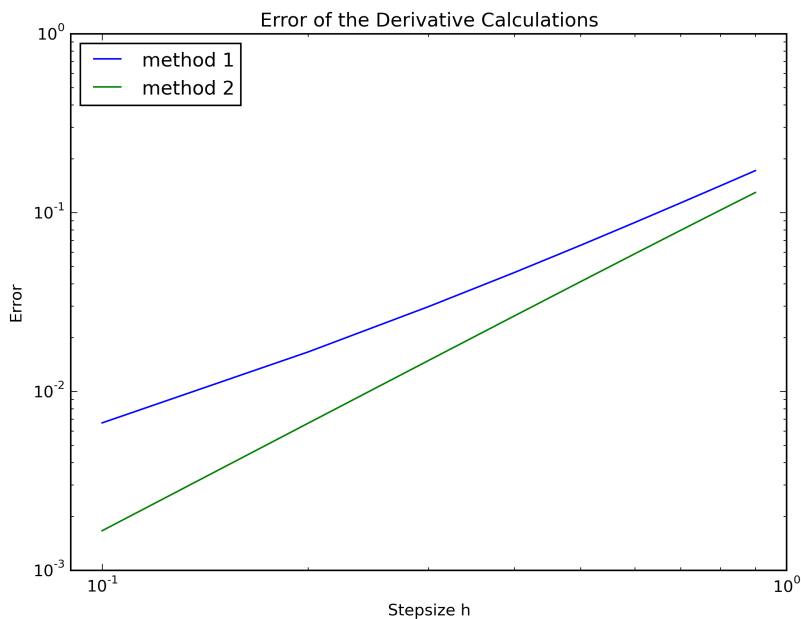


Figure 1: Double log plot of the error of the derivative approximation methods.

## Analysis

From the plot above, we can see that the error in both methods increases with increasing stepsize  $h$ . Since the curves are linear in this loglog plot, the error increases exponentially in both methods. The error for the first method has a smaller slope on the loglog plot, which corresponds to a greater slope of the exponential function. The slope represents the power of the exponential, which determines the rate of increase of the error.

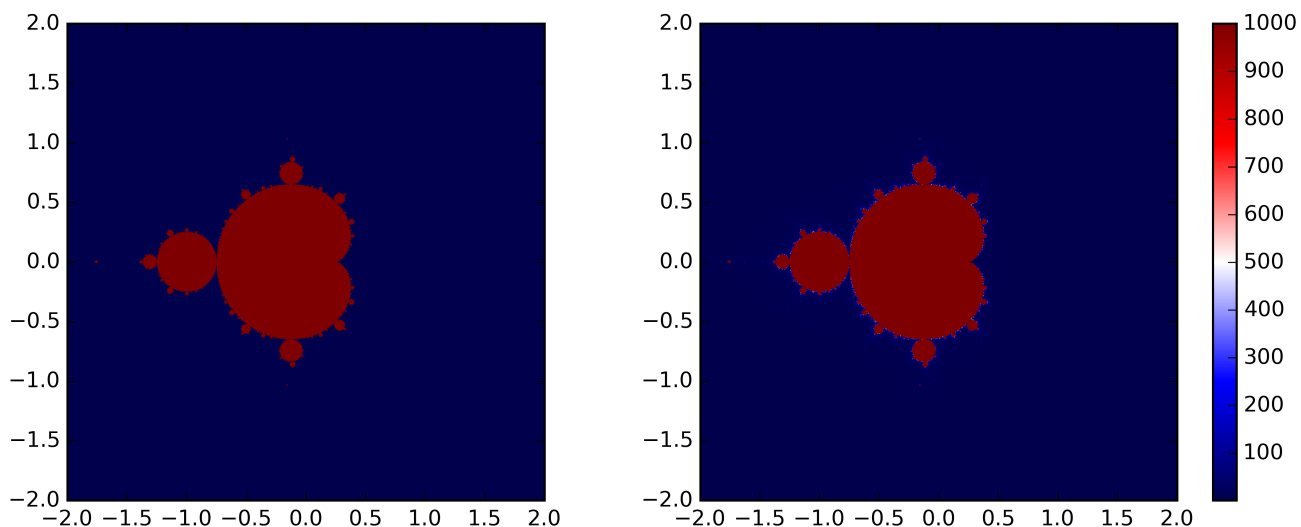
## Question 2

### Methods

The goal for this question is to plot the set of values in the complex plane  $c = x + iy$ ,  $|x| < 2$ ,  $|y| < 2$  for which iterations of the equation  $z_{i+1} = z_i^2 + c$  are bounded or not. First, the question asks to make an image where the points  $c$  that are bounded are given one colour while the unbounded values are given another colour. To do this, I first defined a function 'valu' that determines whether  $|z| < 2$  for the value of  $c$ , within  $N$  iterations. The function returns a 2d array with values of "True" or "False". I then defined arrays of numbers between -2 and 2 which I used to create a complex 2d array  $c$ . The  $c$  array was then input into the function 'valu', and used matplotlib's imshow function to plot the bounded values in red ("True") and unbounded values in navy ("False"). I saved the image as png and pdf files.

Next, the question asks to make an image where the points are coloured by a colourscale that indicates the iteration number at which the point diverged. For this, I defined a function 'count' which counts the number of iterations of  $z_{i+1}$  until it becomes unbounded. I then defined a 2d array of zeros that I filled with the number of iterations for each value of  $c$  using a nested for loop. Finally, I used imshow once again to create an image where the largest number of iterations ( $N$ ) is coloured in red, the least iterations in blue, and a colourbar to indicate the values in between.

### Results



(a) Plot of real  $c$  values, in red are the bounded values and in blue are the unbounded values. (b) Plot of real  $c$  values where the colour is determined by the number of iterations before diverging.

Figure 2

## Analysis

The set of values that remain bounded form an interesting fractal that can be seen in red in the above images. In the second image, we see that the points that diverge with many iterations surround the fractal, with the number of iterations decreasing with increasing distance from the fractal.

## Question 3

### Methods

The goal of this question is to use an ODE integrator to integrate the three differential equations that describe the SIR model of disease spread. I first imported the "odeint" module from scipy.integrate, and defined all of the initial values given in the question. I set up an array of time values  $t$  and lists of various values for parameters  $\beta$  and  $\gamma$  between 0 and 2. I chose these values because  $\beta$  represents the rate of contraction and  $\gamma$  represents the rate of recovery. For 3/4 sets of  $\beta$  and  $\gamma$ , I chose values of  $\beta > \gamma$  because for  $\beta < \gamma$  the S, I, R values stay approximately linear.

Next, I defined a function called 'odes' which contains all three odes, and takes in a dummy variable  $v$ , as well as time  $t$ ,  $N$ ,  $\beta$  and  $\gamma$ . The dummy variable is used to set the initial conditions of S, I, and R in the ode solver. I used 'odeint' to solve the odes for each set of  $\beta$  and  $\gamma$  values. Finally, I created subplots of the S, I, and R curves for different  $\beta$  and  $\gamma$  values.

### Results

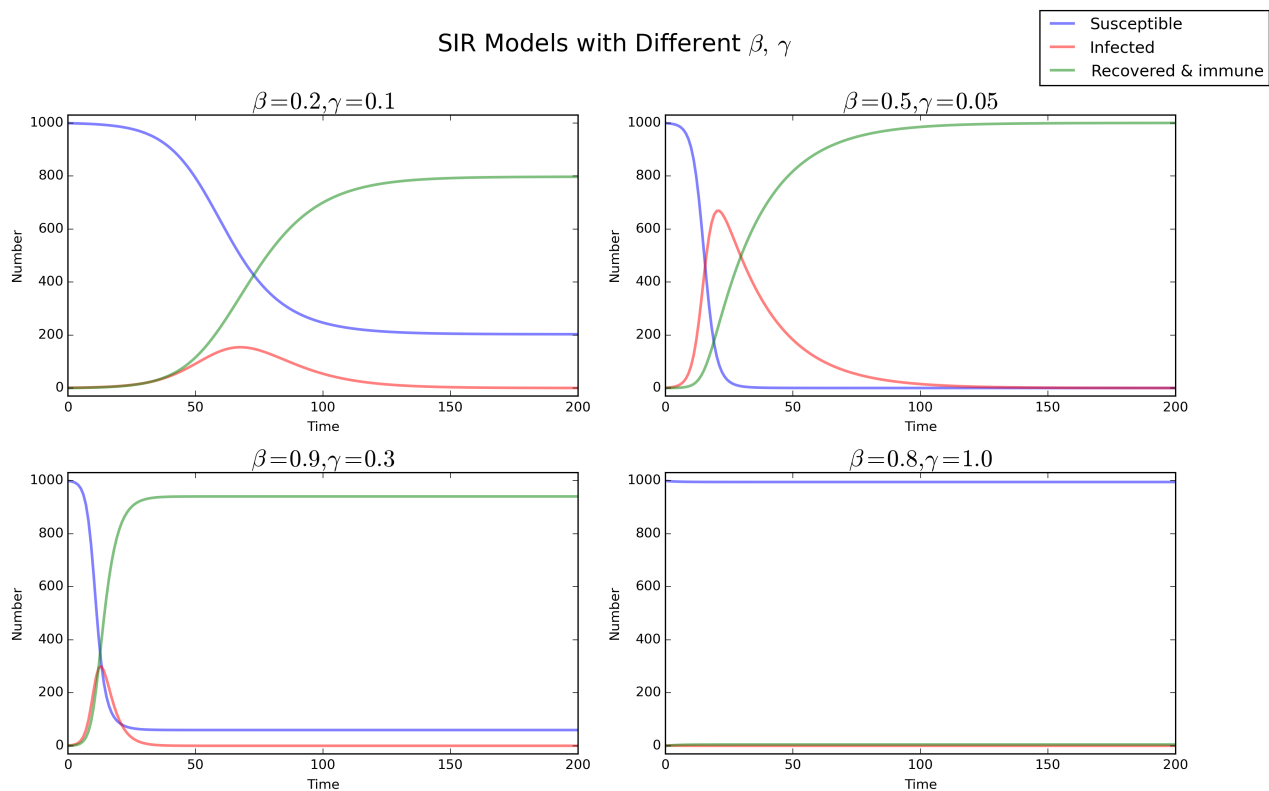


Figure 3: Plots of the SIR model with different values of  $\beta$  and  $\gamma$ .

## Analysis

From the bottom right plot with  $\beta = 0.8$  and  $\gamma = 1.0$ , we see that when  $\gamma$  is larger than  $\beta$ , the number of infected people does not increase. This is because a larger  $\gamma$  means that people are recovering faster than they are becoming infected. Comparing the two plots on the left, we see that for greater values of  $\beta - \gamma$ , the number of infected people peaks earlier on and is larger. This makes sense because if people are infected at a higher rate than they are recovering, we expect to see more infections early on, whereas if people are getting infected and recovering at similar rates, it takes longer for the greater contraction rate to result in a peak of infections. Finally, in the top right plot with  $\beta = 0.5$  and  $\gamma = 0.05$ , we see that the peak in infections tapers off more slowly than it rises, whereas in the other cases the peak was nearly symmetrical. This is because of the slow rate of recovery.