

# CTA200 Assignment 2

Sinthushan Sivakumar

May 8<sup>th</sup>, 2021

## 1 Question 1

The numerical approximation of the derivative of a mathematical function  $f$  evaluated as  $x_0$  can be written as:

$$d_x f|_{x_0} \approx \frac{f_{x_0+h} - f_{x_0}}{h}$$

Where  $h$  is some small "step". This is found by looking at the Taylor series of  $f(x)$  and taking  $h \rightarrow 0$ . A better approximation, when  $h$  is finite (rather than infinitesimal) is:

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

Make a python function that has the form `'def deriv(f, x0, h)'` that takes in a python function, and returns the approximation of the function  $f(x) = \sin(x)$  for  $x_0 = 0.1$  using a variety of values of  $h < 1$ . Plot the error compared to the analytic derivative (ie  $\text{abs}(d_{\text{numerical}} - d_{\text{analytic}})/d_{\text{analytic}}$ ) as a function of  $h$  for each method on the same loglog plot. What do you notice? What does the slope represent?

### 1.1 Method

Created two functions in Jupyter Notebook that take the derivative of an arbitrary function using the infinitesimal way and the finite way. Both functions take in a python function,  $f$ , a point,  $x_0$ , and step size,  $h$ , and returns the derivative of the function  $f$  at the point  $x_0$ . Next, the error between each of the numerical derivatives and the analytical derivative was calculated using the numpy module for values of  $h$  running from 0.001 to 0.9999 in 1000 steps, for the function  $\sin(x)$  at the point 0.1. Finally, using the matplotlib.pyplot, the error in each method of determining the derivative was graphed as function  $h$  on the same plot.

### 1.2 Analysis

Looking at the graph, one can see that for small values of  $h$  around  $10^{-4}$ , deriv2 has errors 3 orders of magnitude smaller than deriv1. However, for values of  $h$  near 1, both deriv2 and deriv1 have the same magnitude of error (of about  $10^{-1}$ ). The slopes of the lines for each of the methods represents the rate at which the error grows. Using np.plotfit, the slope for deriv1 was determined to be 0.7317 while the slope for deriv2 was determined to be 0.5027. Therefore, deriv2 is more accurate in determining the derivative at a point.

## 2 Question 2

For each point in the complex plane  $c = x + iy$ , with  $-2 < x < 2$  and  $-2 < y < 2$ , set  $z_0 = 0$  and iterate the equation  $z_{i+1} = z_i^2 + c$ . Note what happens to the  $z_i$ 's: some points will remain bounded in absolute value  $|z|^2 = \Re(z)^2 + \Im(z)^2$ , while others will run off to infinity. Make an image in which your points  $c$  that diverge are given one color and those that stay bounded are given another. Make a second image where the points are coloured by a colour scale that indicates the iteration number at which the given point diverged.

### 2.1 Method

### 2.2 Analysis

## 3 Question 3

The SIR model is a simple mathematical model of disease spread in a population. The model divides a fixed population of size  $N$  into three groups, which vary as a function of time,  $t$ :

- $S(t)$  is those that are susceptible but not yet infected
- $I(t)$  is the number of infected individuals
- $R(t)$  is those individuals that have recovered and are now immune

The model can be described by a set of 3 first order differential equations for each of the variables as

$$\frac{dS}{dt} = -\frac{\beta SI}{N} \quad (1)$$

$$\frac{dI}{dt} = \frac{\beta SI}{N} - \gamma I \quad (2)$$

$$\frac{dR}{dt} = \gamma I \quad (3)$$

Using the ODE integrator of your choice (must be callable in Python, we recommend using Scipy as will be covered in lecture on Friday), integrate the equations with  $N = 1000$  from  $t = 0$  to  $t = 200$  for various values of  $\gamma$  and  $\beta$  (at least 3-4 values, justify your choices physically). Use the initial condition  $I(0) = 1$ ,  $S(0) = 999$ ,  $R(0) = 0$  (you can also experiment with other initial conditions if you wish). Plot the curves for  $S$ ,  $I$ ,  $R$  on the same figure with a legend. make separate plots for each choice of the parameters (hint: use the `subplots()` command). Bonus (+0.5 mark): Add a 4th parameter  $D$  for deaths and justify the addition of a 4th differential equation as well as any RHS terms that must be changed on the initial 3 equations. Integrate the new set of equations for some choice of parameters and comment on the results compared to the SIR model.

### 3.1 Method

Created a function for the  $SIR$  model. Then from `scipy.integrate` used `ode` to solve the system of differential equations, for the initial condition of 999 susceptible individuals, 1 infected individual, and 0 recovered individuals at time  $t = 0$ . Finally, the results of the integration was plotted using `matplotlib.pyplot`. However, trying to run the code produced an error, "'float' object is not iterable". This error was not solved in time for submissions.

## 3.2 Analysis