CTA200H Assignment 2

Anatoly Zavyalov

May 7, 2021

Question 1

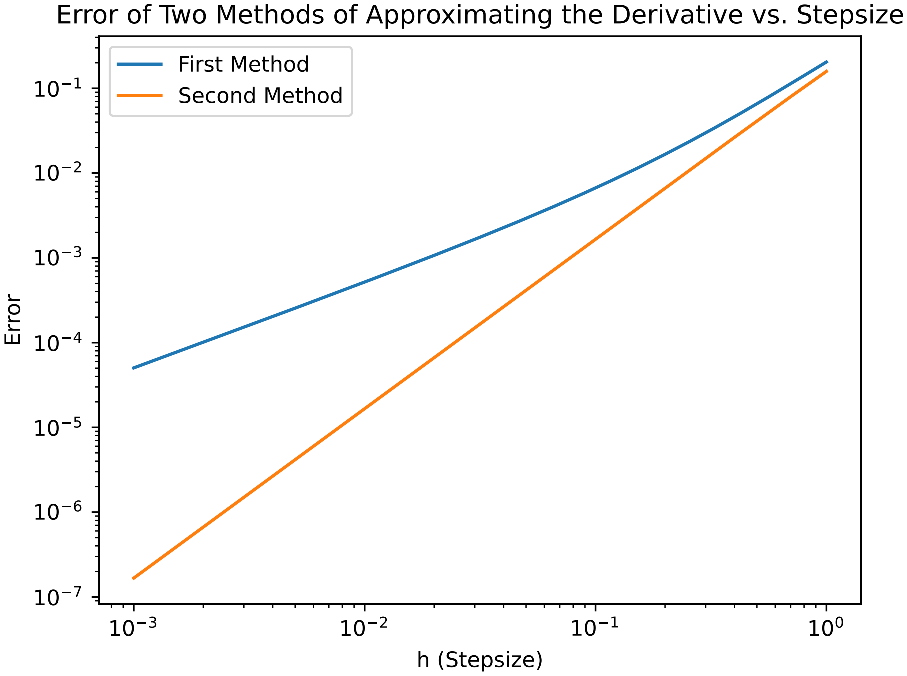

Figure 1: Comparison of errors of two ways of estimating derivative on a loglog plot, based on varying stepsize.

Methods

We first wrote Python functions to represent the two ways of approximating the derivative, and then created a function to determine the error between the numerical and analytical ways of finding the derivative. We then iterated through different values of stepsize h (with a certain step between the values of the stepsize), and plotted the values on a loglog plot using the `matplotlib` Python module.

Analysis

We see that the absolute error of the two methods of approximations have the same approximate innacuracy when the stepsize is high, but the second method of approximation is more accurate by orders of magnitude when the stepsize is decreased. The slope of the error vs. stepsize plot represents how rapidly the method becomes more innacurate as the stepsize is increased.

Question 2

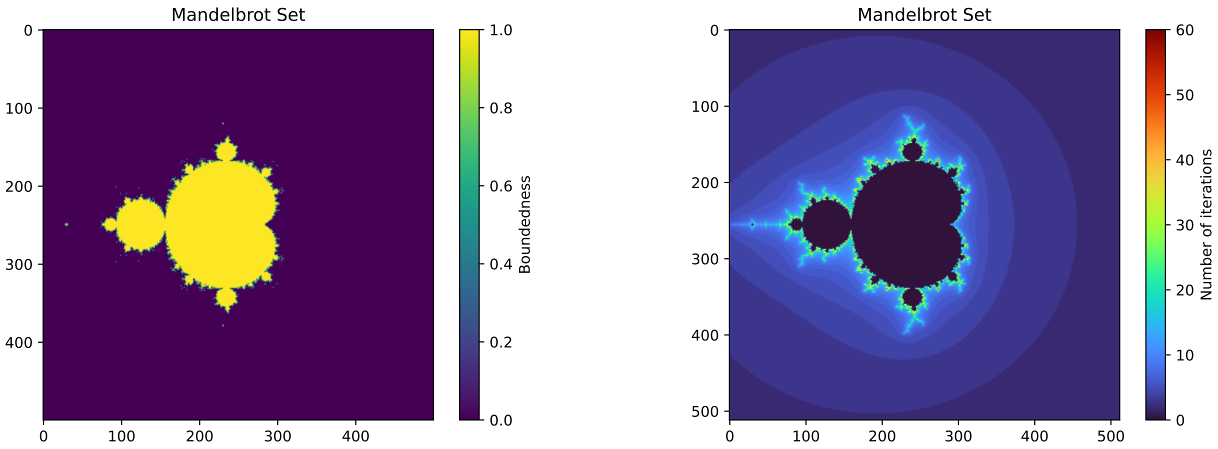

Figure 2: Left: Mandelbrot set with two colors; Right: Mandelbrot set with coloring based on number of iterations before diverging.

Methods

In both illustrations, for $c \in \mathbb{C}$, we developed a function to check the number of iterations of the function $z(n) = z(n-1)^2 + c$ with $z(0) = 0$ before $z(n)$ was greater than some threshold, up to a maximum number of iterations. This function was vectorized using `numpy`'s `vectorize` method, and was then applied to a `numpy.meshgrid` in order to get a two-dimensional mapping of the function. Then, `matplotlib`'s `imshow` method was used to display the graphics. For the illustration that required two colors, the function just returned `1.0` if the point was bounded, and `0.0` otherwise, instead of returning the number of iterations, thus resulting in a two-colored output.

Analysis

We see that the output is the famous Mandelbrot set! This is a fractal (as can be seen by increasing resolution and zooming in on the `matplotlib` output in the Jupyter notebook), and a very cool one at that. We see that the large middle region is completely bounded upon repeatedly applying the function, and it "branches out" into a fractal when heading toward regions that become unbounded.

Question 3

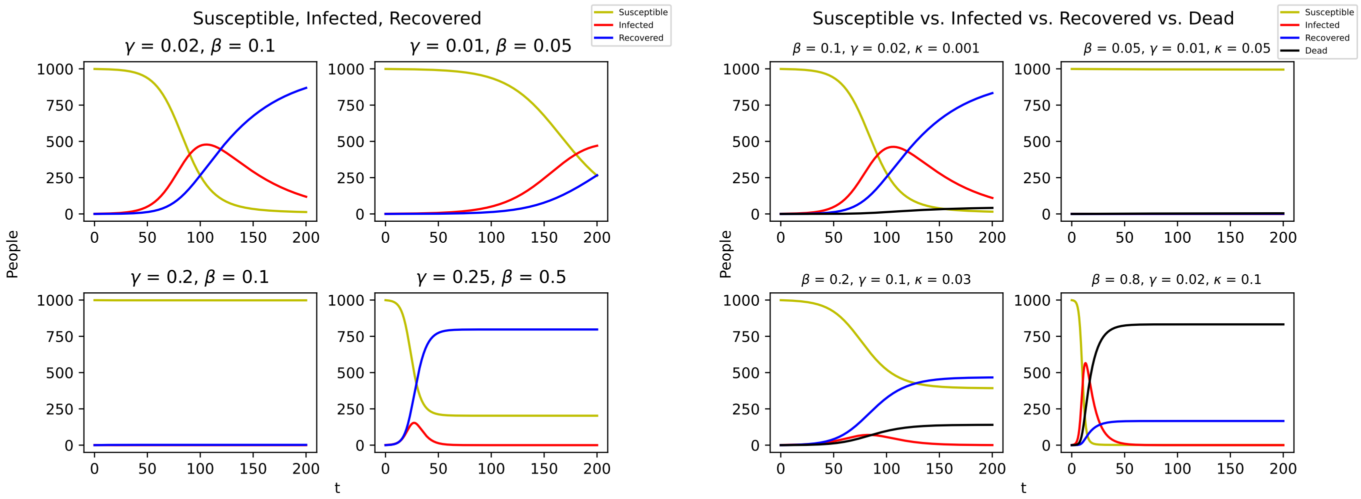

Figure 3: Left: SIR Model with different values of β, γ. Right: SIRD Model with different values of β, γ, κ.

Methods

Similar to lecture, we create a function to return the right hand of the ODEs (that have the derivative in terms of time on the left hand side). We then use `scipy`'s `integrate.ode` function to numerically integrate the ODEs, then convert the data into `numpy` arrays, and finally draw the plots using `matplotlib`.
For the SIRD model, we added a death parameter and death coefficient κ, which represents how much of the infected population dies every unit of time due to the disease. We then modify our code from the SIR model to accept this fourth parameter, and recreate the graphs.

Analysis

We see that, for the SIR model, when $\beta > \gamma$, the size of the infected population rises in a single wave before slowly dropping, with the size of the recovered population increasing shortly after, and with the size of the susceptible population dropping quite rapidly. This makes sense, as if the recovery rate is lower than the infection rate, the disease will have the chance to infect some people before the infected population recovers. However, when $\gamma > \beta$ (such as in the bottom left), the chance of recovery of an infected person is greater than the possibility of a susceptible person becoming infected, hence almost no one gets infected.

For the SIRD model, we see that if $\kappa \geq \beta$ (such as in the top left example), then it is more likely for infected people to die than it is for susceptible people to get infected. Hence, the disease basically does nothing, the small infected population dies, and that's the end of it. We also see that, in the bottom right example, if the infection coefficient is high, recovery coefficient is low, and death coefficient is relatively low (so disease can kill people slowly enough without just completely killing every infected person quickly), we have quite a deadly pandemic on our hands.