

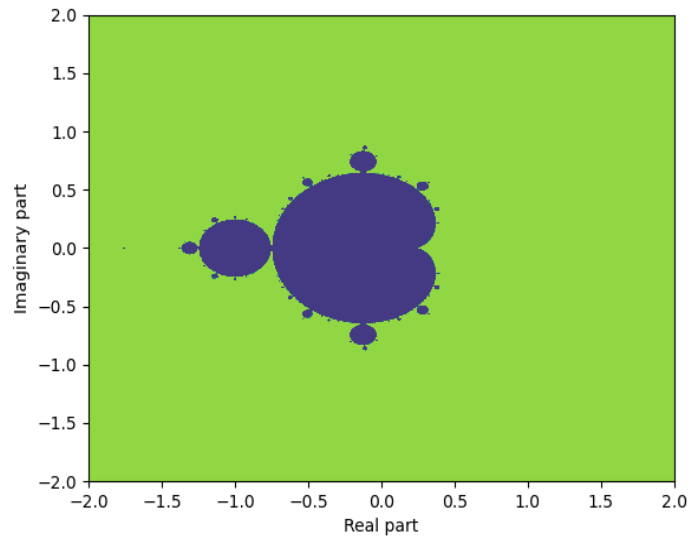
Assignment 3 write-up

Stephanie Lee

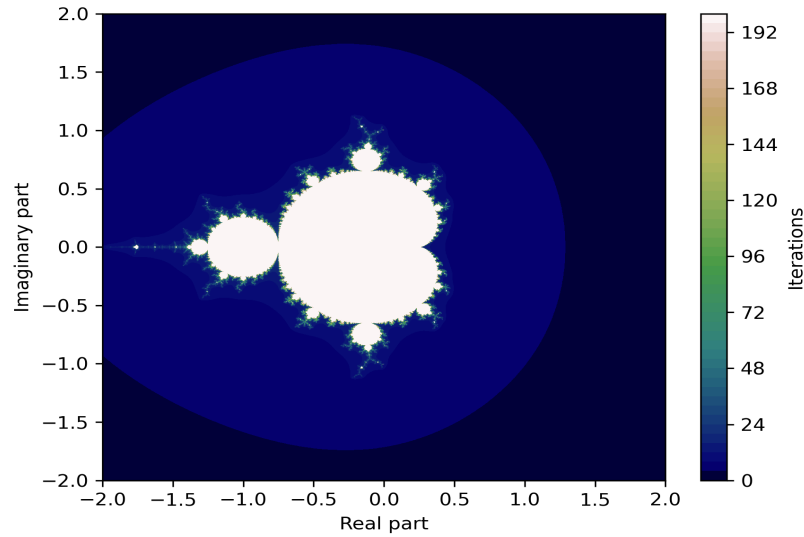
May 8, 2023

1 Question 1

To plot the Mandelbrot set in Python using `matplotlib.pyplot`, I first utilized NumPy's `meshgrid` to create two matrices of the same dimension. For one the matrices, I kept it as a matrix of complex numbers $c = x + iy$ where $-2 < x < 2$ and $-2 < y < 2$ were generated using `linspace`, while the other one I updated with information on whether the complex number at that entry diverged and after how many iterations it does so. To obtain information on divergence, I wrote a double for loop to iterate through the entries of the complex number matrix and used a while loop to make sure my z quantity stayed bounded and the number of iterations was below the constant I set. The iteration formula utilized was $z_{i+1} = z_i^2 + c$ with $z_0 = 0$ and c being a complex number from our matrix. The blue region below encloses points in our subset of the Argand plane that do not diverge while the green region is where points do.



The following figure encodes information on how many iterations it takes each point to diverge. While the white region appears to "diverge" after 200 iterations according the color bar, 200 is just the number of maximum iterations I ran the program with so these are the points that do not diverge. The outer region with the darkest blue contain points that diverge the fastest, which is expected as they are farthest from the origin.



2 Question 2

For Part 1, I began by defining the function dW as a vector-valued function where the entries \dot{X} , \dot{Y} , and \dot{Z} are given by Lorenz's equations 25, 26, and 27. To solve the differential equation, I used SciPy's solve ivp with dW as the function, $[0, 60]$ as the time span, $[0, 1, 1]$ as the initial conditions, $[\sigma, r, b] = [10, 28, \frac{8}{3}]$ as the dimensionless parameters, and 0.001 as the maximum step size. With the solution from solve ivp, I plotted time against the amplitude of Y for Part 3. Reproducing the Lorenz attractor diagrams required plotting Y against Z and Y against X , where the X axis is flipped. Below are the attractor diagrams for Part 4.

Finally, I used solve ivp and $[0, 1.00000001, 1]$ as the initial conditions with all else held the same as before to compare how it would affect W . To calculate the distance between the outputs, I simply used the distance formula on the solution arrays. The time versus logarithm of distance plot is as shown.

