Matthew Abruzzese Ott

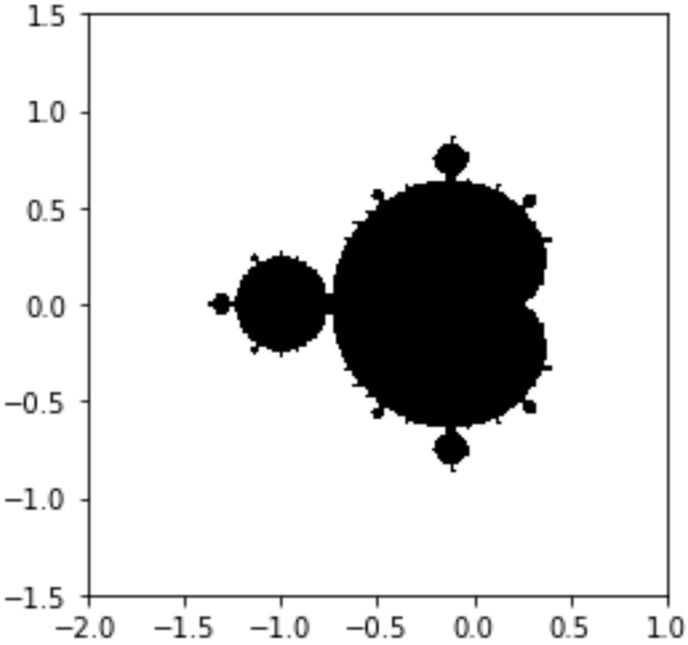
111520701

Python Project 2 Report

AMS 595

Question 1

This program uses a for loop and a mask to generate the Mandelbrot set and plot the Mandelbrot fractal given a threshold (number to approximate infinity) and a number n_max (number to approximate infinity) which determines how many times the for loop iterates. This program uses numpy to generate a 2-D array for the x and y values to compute c as a grid. The value z is initialized to c (an array), and the for loop iterates n_max times. It goes through all values in the z array, and if the absolute values of z[i] are greater than or equal to the number specified as the threshold, they become replaced with infinity within the array. Once this is done, z becomes (z^2 + c), and the loop iterates again. Once the for loop is done iterating, a mask is used to plot the Mandelbrot fractal. This mask is used to mask all values of the absolute values of z[i] that are equal to infinity. Therefore, the Mandelbrot fractal displays only the points which are less than the given threshold as an output.



Question 2

This program is broken up into four parts. For the first part, the program takes a number of states (num_states) as an input, and generates a random (num_states) x (num_states) matrix, i.e., matrix P. It then normalizes each row of matrix P as an output. For the second part, the program takes num_states as an input again and generates a random vector of size num_states, i.e., vector p. This vector is then normalized as well. A for loop is then used to multiply the transpose of matrix P by vector p fifty times. This is known as applying the transition rule fifty times. The new vector p_50 is given as the output. For the third part, the program takes the transpose of matrix P as an input and finds the eigenvector v of the transpose of P that corresponds to the eigenvalue closest to one. This eigenvector v is then scaled so that it becomes the stationary distribution (p_stationary), i.e., it is normalized, and is given as an output. For the fourth and final part, the program takes the vectors p_50 and p_stationary as inputs and checks if the absolute values of the differences between all of the components of p_50 and p_stationary are within a given bound, 1e-5. If so, the program prints that they "match with each other within 1e-5" as an output.