Stan Fang

113782776

AMS 325

HW4 Report

Introduction

The purpose of the project is to form the Mandelbrot Sets in a giving range and construct a Markov Chain by random variables. We use the methods in "`numpy`" and "`matplotlib.pyplot`" package to complete these task.

Methods

Mandelbrot Sets

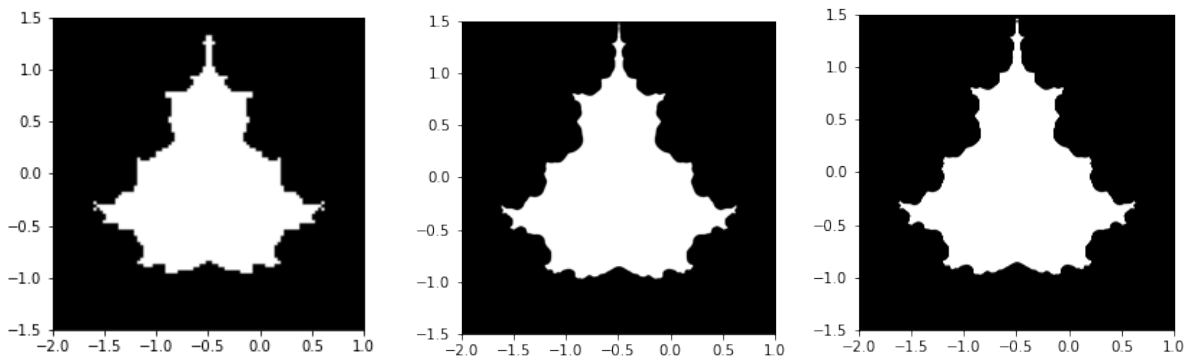
In task 1, we first use "`linespace`" method in "`numpy`" to generate the nth x value and nth y value in the range [-2,1] and [-1.5,1.5]. Construct it into an "`nxn`" matrix by the method "`np.meshgrid(x,y)`". Corresponding the complex values $c = x + yi$. Afterward, we calculate the z value with iteration. We construct a "`100x100`" all-false array named "mask" by method "`np.zeros((100,100), dtype=bool)`". If the absolute value of z is less than the threshold, we mark its location in the all-false array as "True". Finally, we plot the graph by "`imshow`" method.

Markov Chain

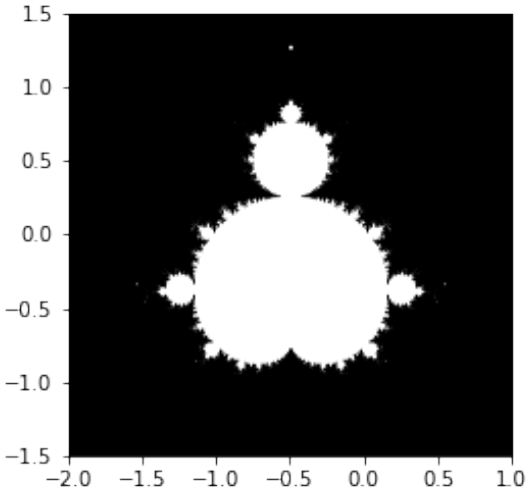
In task 2, we first use "`np.random.dirichlet(np.ones(n), size=1)`" method to construct an array that has a row sum equal to 1 and change the array into matrix format by "`np.matrix()`" method. We can repeat this process by "`for`" loops and achieve the "`nxn`" matrix in the requirement. Using the dot operation "`p=p*P`" by 50 times to get the probability initial with "`p`". Using the "`np.linalg.eig(P.T)`" method to find the eigenvector of P.T and named it "`p_stationary`". After that, we calculate the difference between "`p`" and "`p_stationary`" which in each time dot operation.

Result

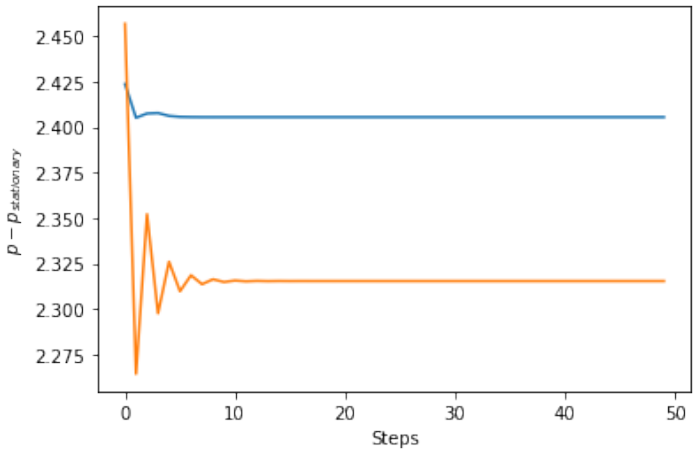
In task 1, I plot the graph with parameter: $N_{max} = 10, threshold = 50$ and the values of n from left to right are $n = 100$, $n = 1000$, and $n = 10000$. When n becomes larger, the edge of the graph become smoother.



After trying to change the n value, I decide to change the N_{max} value. The following picture is plot by $n = 10000, N_{max} = 50,\ threshold = 50.$ Its shape become more concise, but the resolution of the output is enough to show its details.



In task 2, I plot the graph with parameter: $n = 3\ and\ n = 5\ with\ N = 50$ for both lines. Since when N becomes larger, the "$p * P$" becomes a constant. It is enough to set it into "50". The blue is the line of $n = 5$, the orange is the line $n = 3$. The difference of the p and $p_{stationary}$ tends to a fixed value when steps become larger.



Conclusion

The code of project 2 can basically complete the requirement. It has many problems when the parameter becomes great, especially in the task 1 "Mandelbrot Sets".

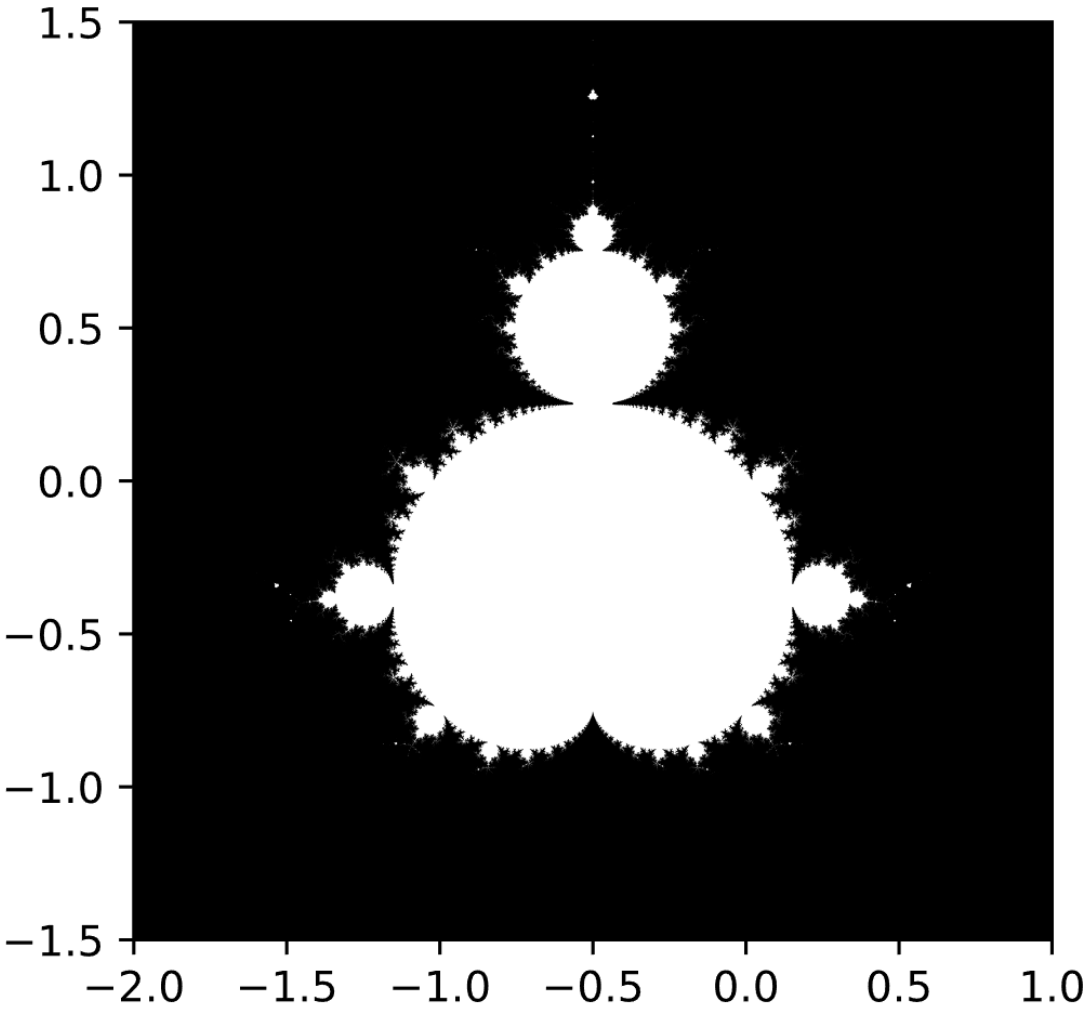
In task 1, when $N_{max} > 10$, python will hand out a "`RuntimeWarning`". It really takes a long time to plot the graph when $n = 10000$ and $N_{max} = 50$. However, the result is not satisfactory enough. Because of the resolution of the output, the details which takes long time to compute does not show. Therefore, change the dpi value in the "`plot`" method, and save the file in "`.svg`" format. In the appendix is the screenshot when "`dpi = 4000`".

In task 2, I did not use the "`np.dot`" method although it got the same value, it seems not so formal. Moreover, although I draw two lines in my graph, I cannot clearly figure which one is the line n = 3 and which one is the line n = 5. I can only distinguish them by the generate sequence in my code. I try to add the label with changing variable, but it seems does not work with "`str.format`".

Appendix:

GitHub links: https://github.com/0xuana/AMS-325-Homework-4

Mandelbrot Sets ($n = 1000, N_{max} = 50, threshold = 50, plot\ dpi = 4000$)



Other outputs of Markov Chain graph: