

Assignment 3 Write-up

Taylor Reeves

May 2025

0.1 Dedications

This report is dedicated to Little Polar Bear and any and all people who have died from suicide and overdose.

1 Introduction

This report details the outcomes of assignment 3 of the CTA200H course offered by the Canadian Institute of Theoretical Astrophysics.

2 Question 1

In Question 1, the problem looks at the Mandelbrot set, which is a data set that emerges as a fractal. The fractal is the set of elements that diverge off to infinity, rather than remaining bounded.

This question also has students creating their own modules for the actual function of the data. The function used was created with the assistance of both Chat GPT, as well as researching the documentation of the numpy module, in order to have the best function for this objective.

The actual code in the Jupyter notebook is rather simple, which mainly is focused on running the function and creating the charts.

Throughout working on this question, the student was able to see the patterns of the fractal that share some common symmetry across the axis. The fractal seems to have a nearly perfect line of symmetry across the $y=0$ point, which represents $0i$ (imaginary unit). This does seem rather relevant, since the function involves squaring the previous term, which would cause the imaginary term to disappear. This also would explain why the terms tend towards the negative side of the real numbers, as the square of an imaginary number is a negative number.

When the colour grid is placed overtop of the fractal, something rather interesting emerges. Based on the colours, it appears that majority of the data that diverges does so quite rapidly, only requiring ≤ 10 iterations before it diverges. The edges of the fractal are where it begins to require more iterations, until it reaches the ends of the fractal, when it has completed the allowed iterations.

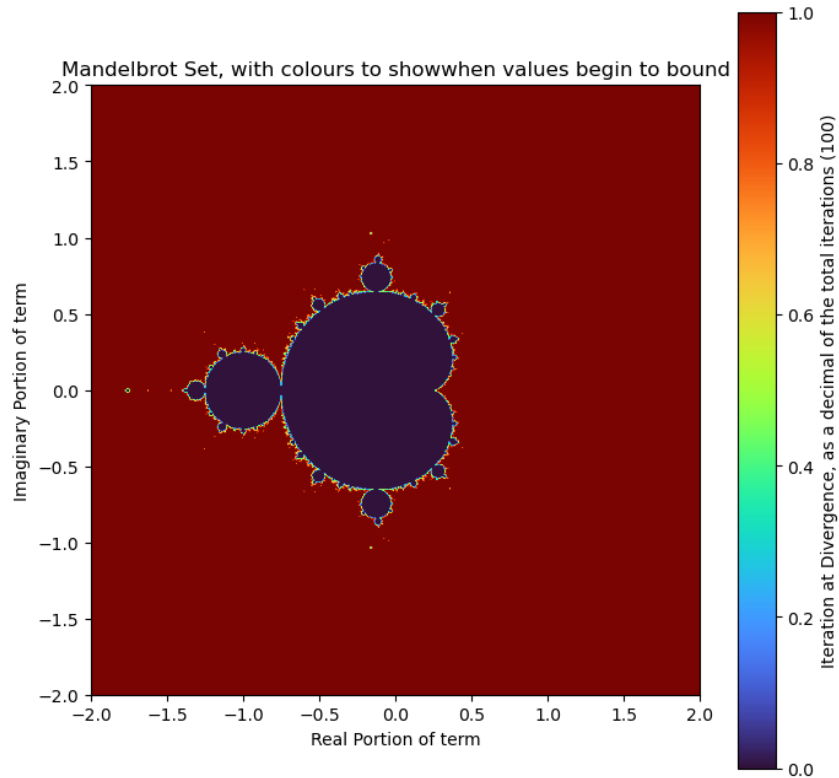


Figure 1: Colour mapped Mandelbrot Set

Attached is the figure from part 2 of Question 1, which looks at the color map of the Mandelbrot set. This figure shows when the fractal diverges, with the red colours representing the spaces when the dataset did not diverge (it remained bounded), and the purples show where the data-set diverges rather rapidly.

The colour bar included uses decimal numbers, rather than whole numbers. This is not impactful to the data, and it rather represents the fraction of the total number of iterations, which in this case is 100, that the system took to diverge, if at all.

3 Question 2

In Question 2, the problem looks at the Lorenz equations, looking at modeling atmospheric data.

The main function of the equations involves creating time derivatives of the X, Y, and Z vectors. This portion of the code was rather simple to create, and works by simply copying the expressions given into a python function.

The equation solver uses the "SolveIVP" element of the Scipy module. This function allows for the solving and graphing of an initial value problem, with set initial parameters, and a function to work off of. This element of the code was derived from work completed in the UTSC course PHYB54, when students looked at modeling oscillations using a python script.

The first graph was the simplest to create, and the author was able to create it without significant issues, as the code is rather similar to generating other plots with matplotlib. Compared to the actual image used in Lorenz's paper, the image generated in Figure 2 appears to be an amalgamation of all the charts, attached altogether into one.

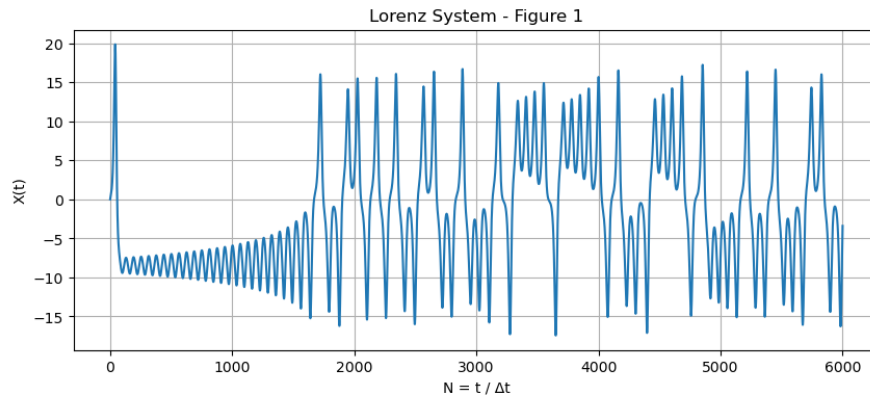


Figure 2: Plot of Part 1, Question 2

The second part of the question looked at replicating Lorenz's second figure, which represent convection patterns in the atmosphere. Similar to in part 1, the plot generated looks somewhat similar to the full plot, however it looks to be incomplete. There is a potential that more permutations could improve the plot, allowing it to be closer to the actual plots provided, however this has not been tested.

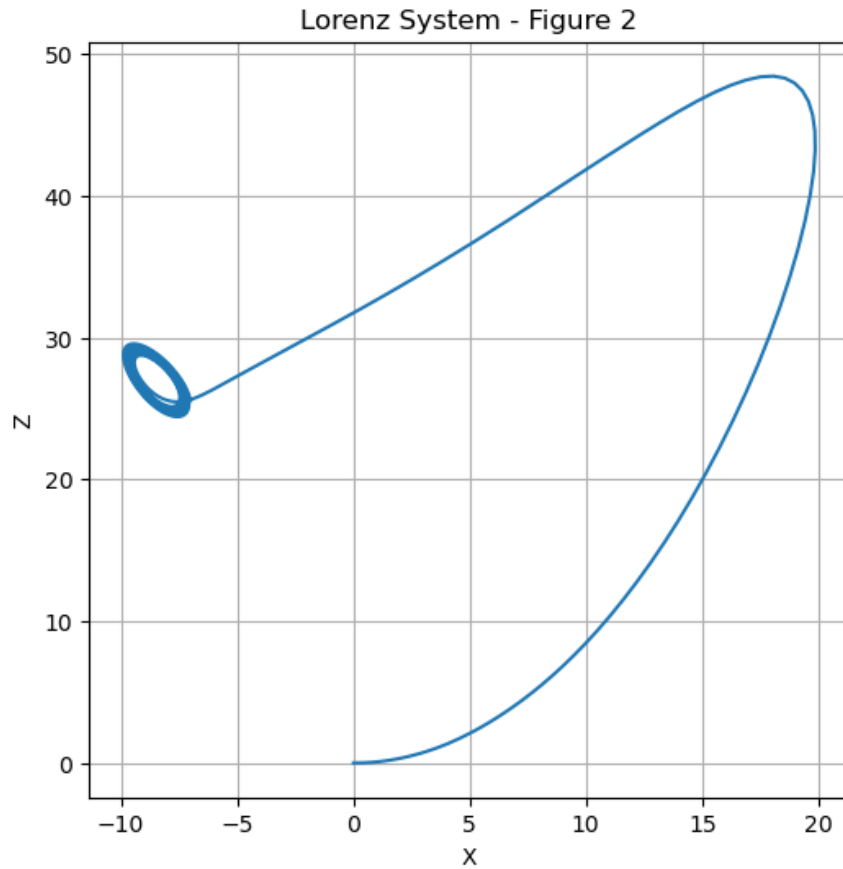


Figure 3: Plot of Part 2, Question 2

Part 3 of the question proved to be the most difficult to solve. This part involved creating a method to add 2 lists together, in order to create a new starting vector that is ever so slightly different. This proved to be somewhat more complicated than desired, and thus the addition was completed manually.

After this issue, the remainder of the code proved rather trivial, relying somewhat on existing code that the student had, and some code generated by Chat GPT.

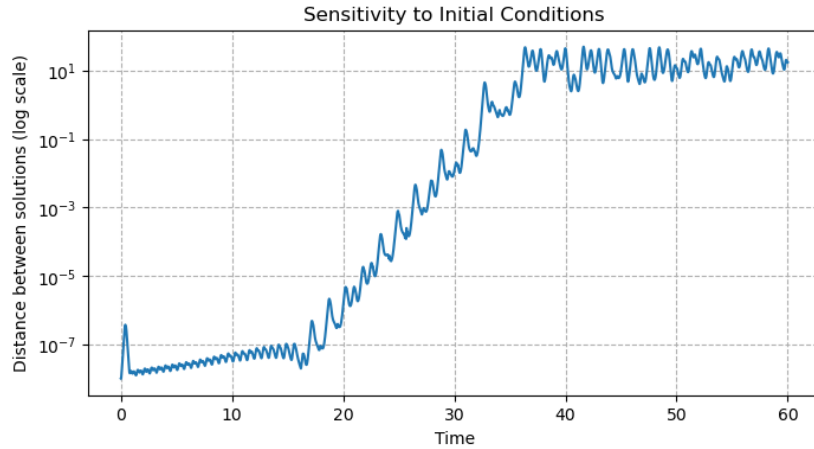


Figure 4: Plot of Part 3, Question 2

Upon looking at the data, it is not a straight line, as Lorenz had achieved. The plot generated grows and grows as time goes on, indicating that future data is more and more off between the two sets of initial conditions.

4 How to use

All of the code attached in this folder is designed to run in a Jupyter notebook. The module file (.py) is directly in the same folder as the 2 Jupyter notebooks, which allows the modules to be imported with minimal extra installation. Upon restarting the kernel, the main notebooks should be able to pull their respective functions out of the module. For any questions or troubleshooting, please contact the author for assistance.