

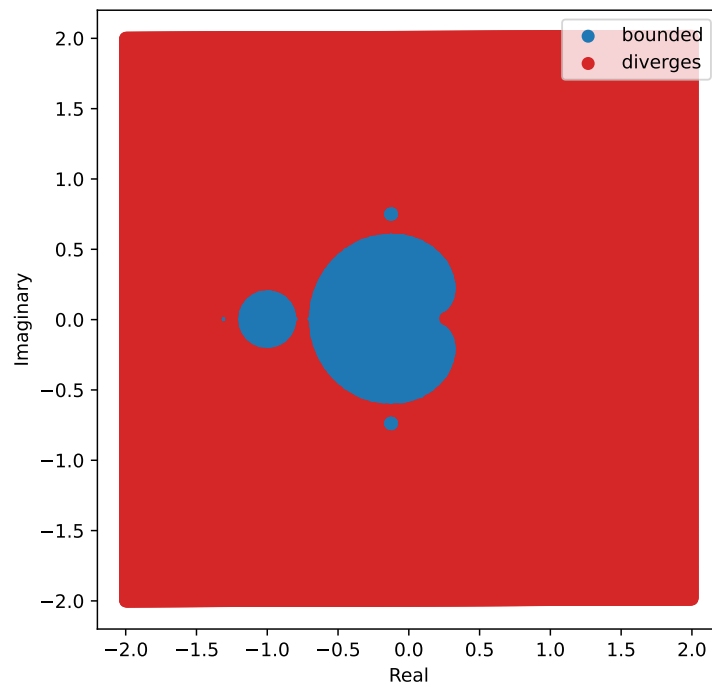
Assignment 3 Writeup

Aeysha Munawwarah

May 14 2024

1 Question 1

Figure 1: *An image of points in the complex plane $c = x + iy$ such that $-2 < x < 2$ and $-2 < y < 2$. The points are colour-coded, with bounded points in blue and divergent points in red.*

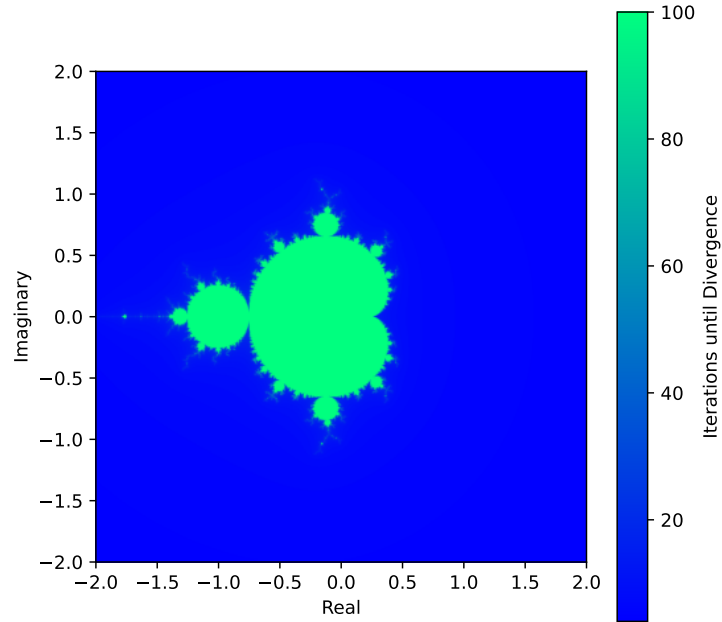


For this question, I first wrote the function `mandelbrot`, which would determine whether a given complex number, c , was contained in the Mandelbrot set, i.e. whether or not the equation $z_{i+1} = z_i^2 + c$ diverged when iterated from $z_0 = 0$.

Then, I created a list containing all of the points in the complex plane $c = x + iy$ with $-2 < x < 2$ and $-2 < y < 2$. Afterward, I looped through each of these points, and depending on the return value of `mandelbrot(c)`, I added the point to a list of all the bounded points (`mandelbrot(c) = max_iter = 100`) or a list of all the divergent points (`mandelbrot(c) < max_iter = 100`).

Then, I plotted an image (see Figure 1) of all the points in the complex plane, where all of the bounded points were coloured blue and all of the divergent points were coloured red.

Figure 2: An image of points in the complex plane $c = x + iy$ such that $-2 < x < 2$ and $-2 < y < 2$. The points are coloured by a colourscale that indicates the iteration number at which the given point diverged.



Finally, I plotted an image in a colourscale (see Figure 2) where the colour of each point indicates the iteration number at which that point diverged. For bounded points, the iteration number would be 100, as the loop would have finished without returning early.

2 Question 2

For this question, I first wrote the function `lorenz.equations`, which defined the three Lorenz equations given by:

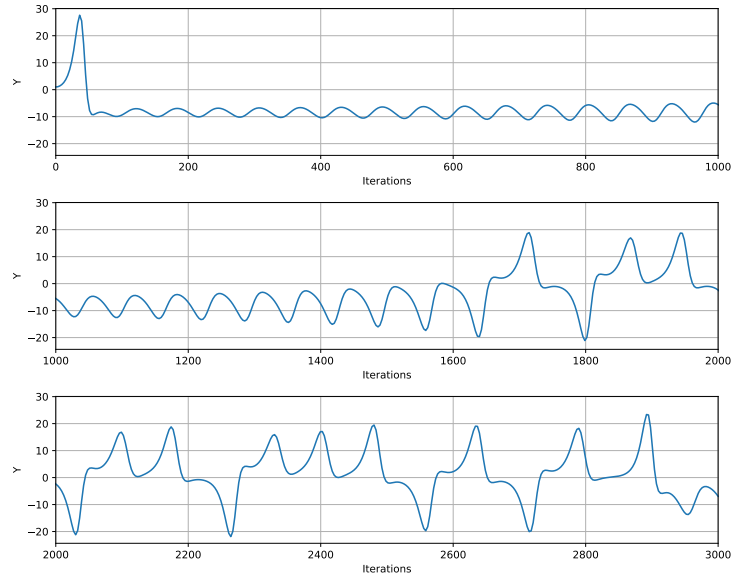
$$\dot{X} = -\sigma(X - Y) \quad (1)$$

$$\dot{Y} = rX - Y - XZ \quad (2)$$

$$\dot{Z} = -bZ + XY \quad (3)$$

Then, I used scipy's `solve_ivp` to integrate the equations from $t = 0$ to $t = 60$. I used Lorenz' initial conditions $W_0 = [0.0, 1.0, 0.0]$ as well as his parameter values $[\sigma, r, b] = [10.0, 28.0, 8.0/3.0]$. I then stored the solution at $t = 60$ in the variable `w_t_60`.

Figure 3: A reproduction of Figure 1 from Lorenz' paper: a plot of Y as a function of time for the first 1000 iterations (uppermost graph), the second 1000 iterations (middle graph), and third 1000 iterations (lowermost graph).



For the third part, I reproduced Figure 1 from Lorenz' paper on deterministic nonperiodic flow (see Figure 3). I integrated the equations from $t = 0$ to $t = 30$, and used the equation $N = t/\Delta t$ to get the number of iterations (N) associated with each t during this time interval. Here, $\Delta t = 0.01$. Then I plotted Y as a function of time, splitting the graph into three intervals of 1000 iterations.

For the fourth part, I reproduced Figure 2 from Lorenz' paper (see Figure 4). I integrated the equations from $t = 0$ to $t = 60$ and then extracted the segment of the trajectory from iteration 1400 to 1900 before plotting the projections onto the XY and YZ planes.

Finally, for the fifth part, I found the solution to the system of equations by using the same values of (σ, r, b) , but this time with initial conditions very slightly different than W_0 : $W'_0 = W_0 + [0., 1.e - 8, 0] = [0.0, 1.000000001, 0.0]$. I integrated the equations twice from $t = 0$ to $t = 60$, first using W_0 and then using W'_0 . I then calculated the distance between W' and W as a function of time, and plotted the result on a semilog plot (linear time, log distance; see Figure 5).

Figure 4: A reproduction of Figure 2 from Lorenz' paper: a plot of projections onto the XY (left graph) and YZ (right graph) planes in phase space of the segment of the trajectory extending from iteration 1400 to iteration 1900.

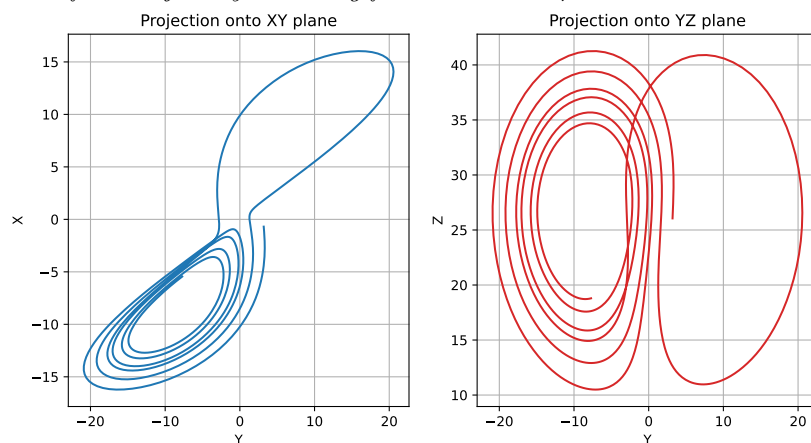


Figure 5: A semilog plot of the distance between W' and W as a function of time.

