Mandelbrot Set

Recursive Definition and Reduced Domain

In this section we will proceed to calculate and plot the Mandelbrot set, which is a set of complex numbers which has the following recursive definition.

$$z_{n+1} = z_n^2 + c \tag{1}$$

Where $z_n \subset \mathbb{C}$ and $C = x + iy$. For our work we will be looking at the reduced domain $D = \{(x, y)| - 2 \leq x \leq 2, -2 \leq y \leq 2\}$.

Implementation

To run this iterative computation we initialized 2D array of zeros of dimensions (x, y) where x and y are arrays from -2 to 2 in steps of 0.01. Then we iterated through each cell of our array of zeros and started to populate each cell through the recursive definition in eq.1. Note that for `plt.imshow()` to work properly its input data must be real or at least floating point values, therefore we will be computing the absolute square of z_n through the following expression

$$|z_n|^2 = \mathbb{R}(z_n)^2 + \mathbb{I}(z_n)^2 \tag{2}$$

After we have populated our z-array with the calculate absolute squares we notices that multiple cells had `nan` as their values. This meant that their calculated z_{n+1} diverged, so in order to address this issue we use `np.nan_to_num()` which converted all the diverged cells to a specific floating point value which we we could filter through a binary conditional statement as it can be seen in the left panel of figure 1.

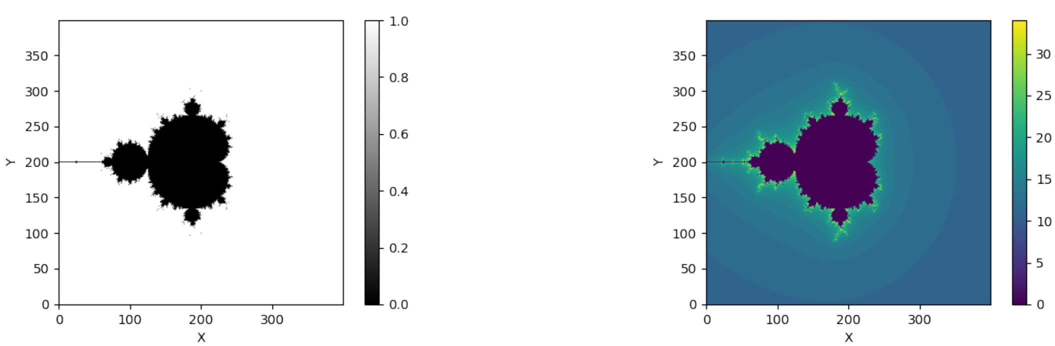

Figure 1: (Left Panel) Binary color coding for points where $|z|^2$ diverges (white) or converges/bounded (black).(Right Panel) Number of iterations before first instance of divergence is encountered. If values never diverged then iteration number was set to 0. Both images where run with a max number of iterations of 35

Following the same iteration algorithm we calculated the number of iterations it takes for a specific z_n to diverge where we would iterate eq. 1 35 at most, and if not divergence was detected then the function would just return zero as the number of iterations. Here the color coding would indicate the number of iterations it took a specific cell to diverge. Note that for all our computation we needed to specify an initial value for z_0 which for all our work was always set to 0.

Lorenz Attractor

In this following section we will be recreating figure 1 and figure 2 from [1], by solving the truncated Lorenz equations through the `scipy.integrate.ode` module. We integrated these equations for a duration of $t = 60$ (dimensionless time units) on a time step of $\Delta t = 0.01$ and we used Lorenz initial conditions $W_0 =$ `[0,1,0]` and dimensionless parameters $\sigma = 10$, $r = 28$, and $b = 8/3$. Note the truncated Lorenz equations are,

$$\begin{aligned} \dot{X} &= -\sigma(X - Y) \\ \dot{Y} &= rX - Y - XY \\ \dot{Z} &= -bZ + XY \end{aligned} \tag{3}$$

After solving the system of equation in 3 we can then plot the solution of Y as a function of iterations which is simply $N = t/\Delta t$, in this case in order to make our plots comparable we split the the axis in steps of 1000 iterations such that we can clearly see the solution evolve through each iteration.

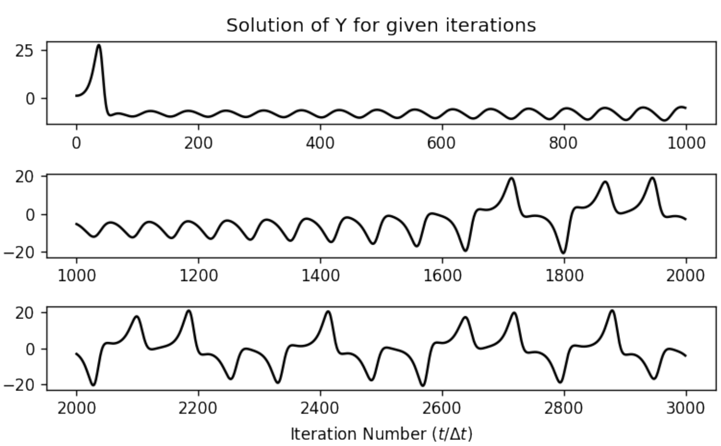

Figure 2: Solution for Y from the system of ODE in 3 as a function of number of iterations

From the solutions of the system of ODE's in 3 we can see at the projection of the solutions

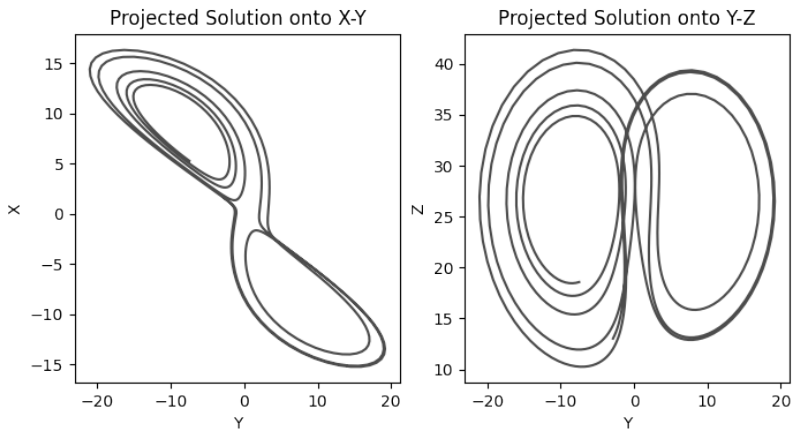

Figure 3: Projected solutions to Lorenz equations (3) in the X-Y plane (left panel) and the Y-Z plane (right panel).

in the X-Y plane and the Y-Z plane and compared them to those in figure 2 in [1]. Note that figure 3 we had to truncate our time domain such that we only see solutions spanning from $t = 14$ to $t = 19$ such that our plotted projections match those in the referenced paper.

Small Perturbation on I.C

We will perturb the initial condition by $[0,1e^{-8},0]$ such that our new initial conditions are $W_0' = [0,1.0000001,0]$. We keep the dimensionless parameters fixed and solve again eq.3 with these new I.C and compare this new solution to our previous one. By comparing we mean we will be computing the euclidean distance between each corresponding point in the solution, as it can be seen in figure 4

References

[1] Edward N. Lorenz. "Deterministic Nonperiodic Flow". In: *Journal of Atmospheric Sciences* 20.2 (1963), pp. 130–141. DOI: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2. URL: https://journals.ametsoc.org/view/journals/atsc/20/2/1520-0469_1963_020_0130_dnf_2_0_co_2.xml.

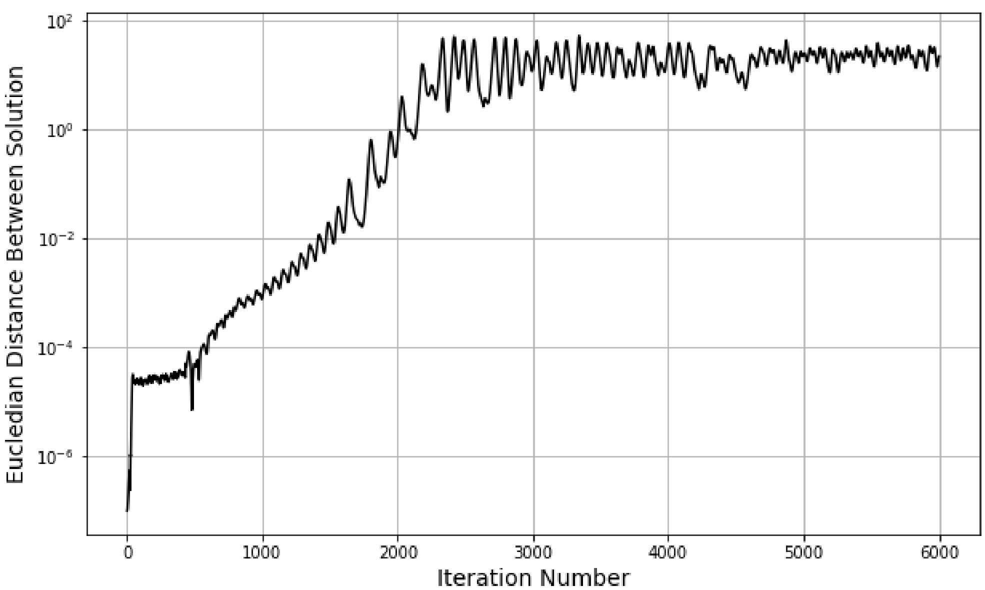

Figure 4: Euclidean distance between solutions to eq.3 where initial conditions we slight perturb by a offset factor of $1e^{-8}$