# The Lorenz Attractor

Chris Hayward

November 16, 2017

The Lorenz System is a system of differential equations used for modeling atmospheric convection. It was first studied by Edward Lorenz, who developed it in 1963.[1] The derived equations are used to model flow patterns in a fluid when it is heated from below while simultaneously being cooled from above. A subset of results from these equations are notable for being chaotic and producing images of butterflies or figure eights when plotted as shown in Figure 1 and Figure 2. This specific set is known as the Lorenz attractor.

These are the three equations Lorenz derived.

$$\frac{dx}{dt} = \sigma(y-x), \tag{1}$$

$$\frac{dy}{dt} = x(\rho-z)-y, \tag{2}$$

$$\frac{dz}{dt} = xy-\beta z. \tag{3}$$

The chaotic behavior of the Lorenz attractor arises when the values are equal or close to $\sigma = 10$, $\beta = \frac{8}{3}$, and $\rho = 28$. Figure 1 shows a plot of the $x$ and $y$ values of Lorenz function plotted over 1000 iterations using those variables and with starting values of $x = 1$, $y = 1$, and $z = 1$. Figure 2 does the same, though with the starting values set at $x = 13$, $y = 21$, and $z = 47$ instead. As expected, both of these produce different plots, that are identifiable as figure eights. These figures were generated using the Python code shown on the last page.

The code begins with a function aptly named 'lorenz' with variables $x$, $y$, $z$, $a$, $b$, $c$, and $dt$. $A$, $b$, $c$, and $dt$ all have default values corresponding to the values mentioned previously, plus an added timestep of 0.01 for $dt$. The function uses the values and the equations written above to return the coordinates of the next point in 3d space. Lines 6 and 7 of the code are simply imports necessary for functions used later. Lines 8 through 21 are devoted to the function 'lorenzArray', which is the main section of code. It takes three variables as input, $a$, $b$, and $c$ which have a default value of 1. The function stores these variables in a numpy

[1]`https://en.wikipedia.org/wiki/Lorenz_system`

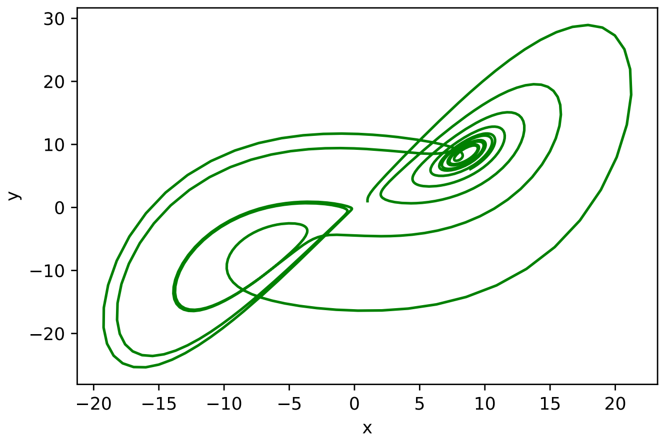


Figure 1: Lorenz function with starting values $x = 1$, $y = 1$, and $z = 1$

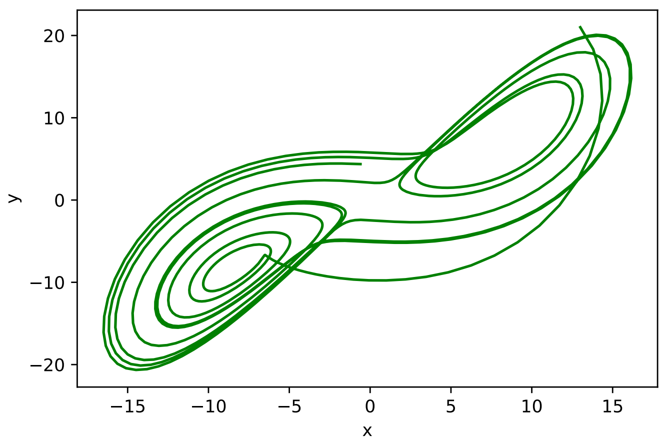


Figure 2: Lorenz function with starting values $x = 13$, $y = 21$, and $z = 47$

array named *seq* so that we can track $a$, $b$, and $c$ as they change over time. It then uses the previously mentioned 'lorenz' equation to calculate new values for $a$, $b$, and $c$ and add them to *seq*. It does the previous step 1000 times, each time using the most current version of $a$, $b$, and $c$. After it has finished 1000 iterations the function uses pyplot to begin creation of the plots shown in Figures 1 and 2, and exports the plot as a pdf named foo.

```
def lorenz(x, y, z, a=10, b=28, c=(8/3), dt=(0.01)):
    dx = a*(y-x)
    dy = x*(b-z)-y
    dz = (x*y)-(c*z)
    return(x+dt*dx, y+dt*dy, z+dt*dz)
import numpy as np
import matplotlib.pyplot as plt
def lorenzArray(a=1,b=1,c=1):
    seq = np.array((a,b,c))
    seq = seq[None,:]
    i = 0
    while (i<1000):
        temp = lorenz(seq[i,0],seq[i,1],seq[i,2])
        seq = np.vstack([seq, [temp[0],temp[1],temp[2]]])
        i += 1
    plt.plot(seq[0:,0], seq[0:,1], 'g-')
    plt.xlabel('x')
    plt.ylabel('y')
    fig = plt.gcf()
    fig.savefig('foo.pdf', dpi=100, format='pdf')
    plt.show()
```