

12-08_Lorenz_Attractor_Simulation

December 8, 2021

```
[1]: import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
[2]: sigma = 10
B = 8 / 3
rho = 28
```

```
[7]: def f(X, t):
    x = X[0]
    y = X[1]
    z = X[2]
    return np.array([
        sigma * (y - x),
        x * (rho - z) - y,
        x * y - B * z,
    ])
```

```
[4]: X0 = 10 * np.ones(3)
X0
```

```
[4]: array([10., 10., 10.]
```

```
[5]: from scipy.integrate import odeint
```

```
[29]: t = np.arange(0, 50, 0.02)
X = odeint(f, X0, t)
```

```
[30]: X.shape
```

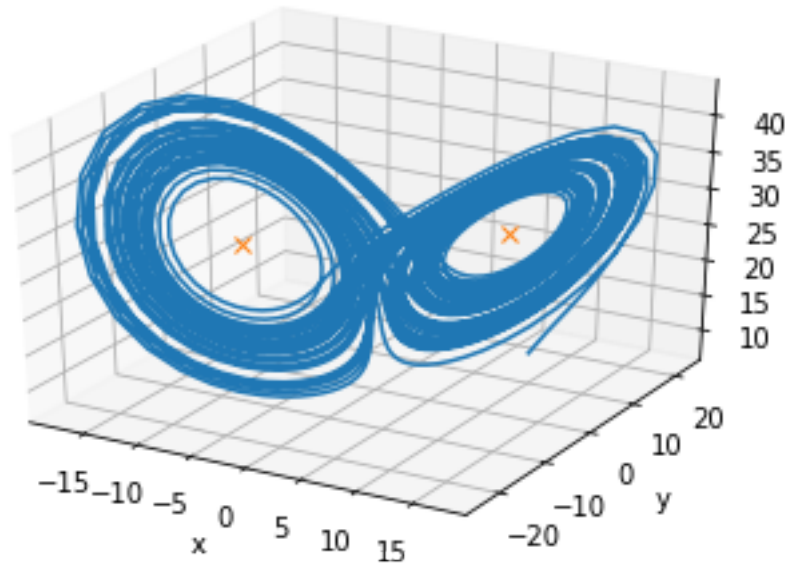
```
[30]: (2500, 3)
```

```
[31]: from mpl_toolkits import mplot3d
```

```
[32]: ax = plt.axes(projection='3d')
ax.plot3D(X[:,0], X[:,1], X[:,2])
ax.set_xlabel('x')
ax.set_ylabel('y')
```

```
ax.set_zlabel('z')
ax.plot3D([-8.5, 8.5], [-8.5, 8.5], [27, 27], 'x')
```

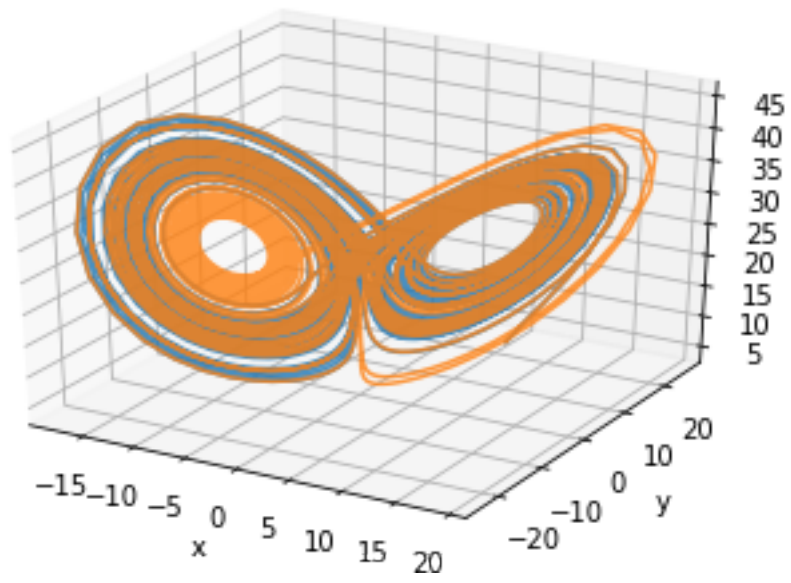
[32]: [



```
[33]: X0_2 = np.array(X0)
X0_2[0] += 1e-8
X2 = odeint(f, X0_2, t)
```

```
[35]: ax = plt.axes(projection='3d')
ax.plot3D(X[:,0], X[:,1], X[:,2], alpha=0.8)
ax.plot3D(X2[:,0], X2[:,1], X2[:,2], alpha=0.8)
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')
```

[35]: Text(0.5, 0, 'z')



[]: