

atrator_de_lorenz

November 13, 2019

1 EP-05

Atrator de Lorenz

Nome: Eric Satoshi Suzuki Kishimoto

RA: 233974

```
In [1]: # importação de bibliotecas
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D

In [4]: # definição de função para calcular todas as equações do sistema de Lorenz
def lorenz(x, y, z, sigma=10, rho=28, beta=8/3):
    dx_dt = sigma*(y - x)
    dy_dt = rho*x - y - x*z
    dz_dt = x*y - beta*z
    # retorna os resultados das equações do sistema(dx/dt, dy/dt, dz/dt)
    return dx_dt, dy_dt, dz_dt

In [5]: # definição de variáveis
dt = 0.01 # variação de tempo
tf = 100 # tempo inicial
ti = 0.01 # tempo final
n = int((tf/dt)+1) # numero de pontos a serem plotados
# (função int usada porque n será usado em listas que precisam de inteiros em seus ind
xt = np.empty(n+1) # criação de lista para valores de x
yt = np.empty(n+1) # criação de lista para valores de y
zt = np.empty(n+1) # criação de lista para valores de z
xt[0], yt[0], zt[0] = (0, 1, 1) # declaração das condições iniciais

In [6]: # criação das listas contendo os pontos do gráfico
for i in range(n):
    # desempacotamento da função que possui 3 retornos
    # dx/dt, dy/dt, dz/dt
    x, y, z = lorenz(xt[i], yt[i], zt[i])
    xt[i+1] = xt[i] + (x*dt)
    yt[i+1] = yt[i] + (y*dt)
    zt[i+1] = zt[i] + (z*dt)
```

```
In [7]: # plotando o gráfico tridimensional
fig = plt.figure()
ax = fig.gca(projection='3d')

ax.plot(xt, yt, zt, lw=0.5)
ax.set_xlabel("X Axis")
ax.set_ylabel("Y Axis")
ax.set_zlabel("Z Axis")
ax.set_title("Lorenz Attractor")

plt.show()
```

