Práctica 6. Extra: Atractor de Lorenz

Guillermo Martín Sánchez

13/05/20

1 Introducción

En esta práctica vamos a estudiar el espacio fásico del Atractor de Lorenz, expresado por el sistema diferencial:

$$\begin{aligned} \frac{\partial q_1}{\partial t} &= 10(q_2 - q_1) \\ \frac{\partial q_2}{\partial t} &= 28q_1 - q_2 - q_1 q_3 \\ \frac{\partial q_3}{\partial t} &= q_1 q_2 - \frac{8}{3} q_3 \end{aligned} \tag{1}$$

2 Material usado

2.1 Ejercicio 1

Discretiza el sistema de ecuaciones diferenciales. ¿Cuántas condiciones iniciales se necesitan para obtener las órbitas?

Para discretizar el sistema hemos usado la fórmula de integración de Euler. Usando una granularidad temporal $t = \delta n$ con $\delta \in \mathbb{R}_{>0}$ y $n \in \mathbb{N}_{>0}$:

$$\dot{q}(q_n) \approx \frac{q_{n+1} - q_n}{\delta} \tag{2}$$

2.2 Ejercicio 2

Encuentra la órbita final de $q = (q1, q2, q3)$ y $p = (p1, p2, p3)$ para $q(0) \in D_0 = \{(1, 1, 1)\}$, usando para ello la representación 3D de q y p por separado (dos gráficas)

Hemos usado las ecuaciones sacadas en el apartado anterior para calcular puntos hasta $n = 30000$ con $\delta = 10^{-3}$ y con condiciones iniciales $(q_0^1, q_0^2, q_0^3) = (1, 1, 1)$.

2.3 Ejercicio 3

Representa gráficamente el espacio fásico de los pares $\{(q_i, p_i)\}_{i=1}^3$ *dado por el conjunto de condiciones iniciales* $q(0) \in D_0 = [-1, 1]^3$ *(por ejemplo en tres gráficas)*

Hemos hecho una malla de condiciones iniciales. Para ello hemos cogido 5 puntos equiespaciados en $[-1, 1]$, llamemos a este conjunto C. Hemos calculado y representado todas las órbitas hasta $n = 30000$ (con $\delta = 10^{-3}$) de condiciones iniciales $(q_0^1, q_0^2, q_0^3) \in C \times C \times C$.

2.4 Ejercicio 4

De acuerdo con la definición de medida de Hausdorff, estima el $d - volumen$ *del espacio fásico de Lorenz con condiciones iniciales* $q \in D_0 = [-1, 1]^3$

Para calcular esto hemos intentado buscar la dimensión de Minkowski–Bouligand que se trata de un límite superior de la dimensión de Hausdorff. Para ello, usamos los puntos calculados en el apartado anterior (llamaremos a este conjunto E) y los normalizamos para que se encuentren entre $[-1, 1]$ (para disminuir la magnitud de algunos cálculos; esto no afecta a la dimensión del conjunto). Después, dividimos el vector $[-1, 1]$ en n intervalos (nótese que aquí usamos la letra n para otra cosa) de igual longitud, dividiendo así el espacio $[-1, 1]^6$ (que es donde se encuentran los puntos del espacio fásico $(q^1(t), q^2(t), q^3(t), p^1(t), p^2(t), p^3(t))$) en 6-cubos con la misma longitud lateral. Esta longitud es $\epsilon = \frac{2}{n}$. Ahora, calculamos cuantos de estos 6-cubos contienen puntos del conjunto y llamamos a esta cifra $N(\epsilon)$. Pues bien, la definición de dimensión de Minkowski–Bouligand es:

$$dim_{box} = lim_{\epsilon \to 0} dim_{box}(\epsilon) = lim_{\epsilon \to 0} \frac{log(N(\epsilon))}{log(1/\epsilon)} \tag{3}$$

Como el número de 6-cubos crece de forma n^6, por límite de memoria, sólo hemos podido calcular $dim_{box}(\epsilon)$ para $n = 5, 6, ..., 28$.

Por otro lado, hemos usado el método de la Práctica 1 - Extra para calcular, dado d, como evoluciona $H_\epsilon^d(E)$. Recordemos que esto consistía en aproximar (nótese que aquí usamos la letra δ para otra cosa):

$$H_\delta^d(E) := \inf_k \left\{ \sum_i ||U_i||^d \right\}_k \quad 0 < ||U_i|| < \delta \tag{4}$$

por

$$H_\epsilon^d(E) := \sum_{i=1}^k ||C_i||^d \tag{5}$$

done $\{C_i\}_{i=1}^k$ son los k 6-cubos que contienen puntos de E y $|| \cdot ||$ es el diámetro. Como $||C_i|| = diam \; \forall i$ podemos escribir esto como $H_\epsilon^d(E) = k(diam)^d$

donde $diam = \frac{||(1,1,..,1)-(-1,-1,...,-1)||_2}{n}$. Así viendo como evoluciona $lim_{\epsilon \to 0} H^d_\epsilon(E)$ aproximamos el verdadero d-volumen de Hausdorff para un valor d dado que es $H^d(E) = lim_{\delta \to 0} H^d_\delta(E)$.

3 Resultados y discusión

3.1 Ejercicio 1

Obtenemos las siguientes ecuaciones de evolución:

$$\begin{aligned} q^1_{n+1} &= q^1_n + 10\delta(q^2_n - q^1_n) \\ q^2_{n+1} &= q^2_n + \delta(28q^1_n - q^2_n - q^1_n q^3_n) \\ q^3_{n+1} &= q^3_n + \delta(q^1_n q^2_n - \frac{8}{3} q^3_n) \end{aligned} \tag{6}$$

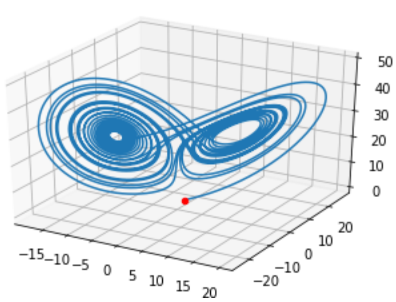
Teniendo en cuenta que $p = \dot{q}/2$ también obtenemos:

$$\begin{aligned} p^1_n &= \frac{q^1_{n+1} - q^1_n}{2\delta} \\ p^2_n &= \frac{q^2_{n+1} - q^2_n}{2\delta} \\ p^3_n &= \frac{q^3_{n+1} - q^3_n}{2\delta} \end{aligned} \tag{7}$$

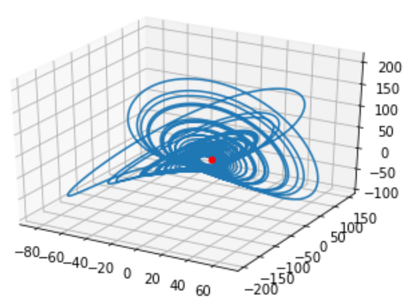
Como se puede ver sólo necesitamos (q^1_0, q^2_0, q^3_0) como condiciones iniciales ya que a partir de ellos podemos sacar (q^1_1, q^2_1, q^3_1) con Eq 6 y a partir de ambas, sacar (p^1_0, p^2_0, p^3_0) con Eq 7.

3.2 Ejercicio 2

En la siguiente figura encontramos la representación gráfica obtenida (en rojo el punto inicial):



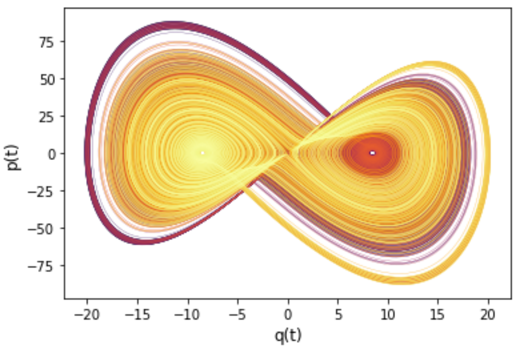
(q^1, q^2, q^3)



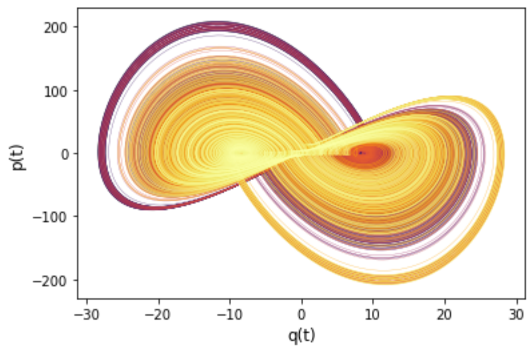
(p^1, p^2, p^3)

3.3 Ejercicio 3

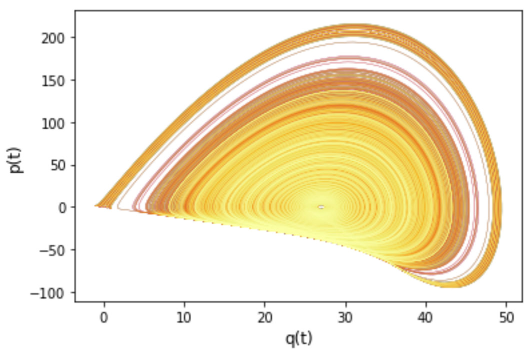
En la siguientes figuras encontramos la representación gráfica obtenida:



(q^1, p^1)



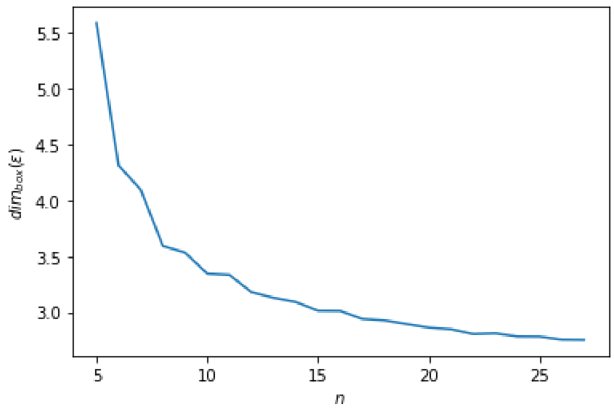
(q^2, p^2)



(q^3, p^3)

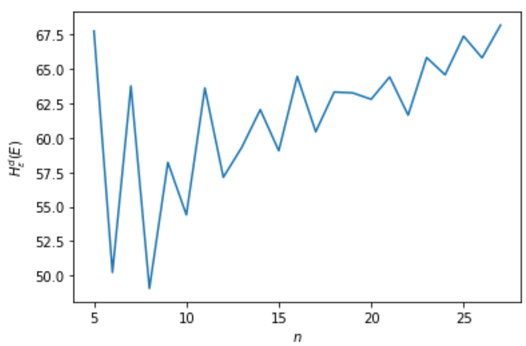
3.4 Ejercicio 4

Como se puede ver en la siguiente figura tenemos que $dim_{box} = lim_{\epsilon \to 0} dim_{box}(\epsilon) = lim_{n \to \infty} dim_{box}(2/n) \approx 2.75$. Esta es la dimensión de Minkowski–Bouligand y por tanto un límite superior de la dimensión de Hausdorff.

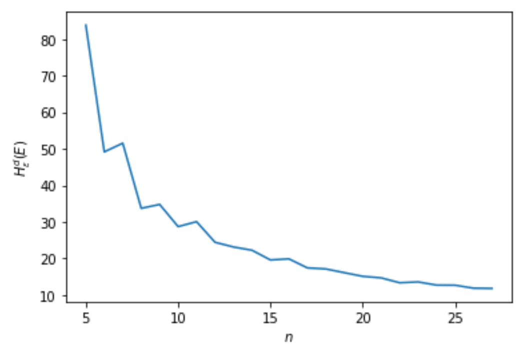


Gráfica que nos relaciona n con $dim_{box}(\epsilon = 2/n)$

Como pequeña comprobación vemos cómo se comportan $H^d_\epsilon(E)$ para $d = 1.7$ y $d = 2.75$. Efectivamente vemos que el límite con $d = 2.75$ no tiende a infinito por lo que la dimensión de Hausdorff tiene que ser menor o igual a esta. Por otro lado, el límite con $d = 1.7$ sí tiende a infinito y por lo tanto, la dimensión debe ser mayor.



Gráfica que nos relaciona n con $H^{d=1.7}_{\epsilon=2/n}(E)$



Gráfica que nos relaciona n con $H^{d=2.75}_{\epsilon=2/n}(E)$

4 Conclusión

Hemos estudiado el espacio fásico de un sistema caótico. Hemos visto diferentes opciones para representarlo ya que siendo 6-dimensional no se podía hacer de forma completa en una sola gráfica. Por otro lado, hemos visto cómo la dimensión de Minkowski–Bouligand se relaciona con la dimensión de Hausdorff y cómo de costoso es de calcular. Hemos podido acotar tanto superiormente como inferiormente la dimensión de Hausdorff aunque con un intervalo bastante grande.

5 Código

```
# -*- coding: utf-8 -*-
"""
Coursework 6 Extra: Phase space of Lorenz Attractor
"""
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Compute the derivative dq from an array of consecutive points q with initial
# derivative dq0 with timestep d
def deriv(q,d=0.001):
   dq = (q[1:len(q),:]-q[0:(len(q)-1),:])/d
   return dq

# Differential equation of Lorenz Attractor
def F(q):
    ddq1 = 10*(q[1]-q[0])
    ddq2 = 28*q[0] - q[1] -q[0]*q[2]
    ddq3 = q[0]*q[1] - (8/3)*q[2]
    return np.array([ddq1, ddq2, ddq3])

# Compute the orbit of the differential equation F with initial values q0 and dq0,
# n points and timestep d
def orb(n,q0, F, d=0.001):
    q = np.empty([n+1,3])
    q[0,:] = q0
    for i in np.arange(1,n+1):
        q[i,:] = q[i-1,:] + d*F(q[i-1,:])
    return q

# Plot an orbit of the differential equation F with initial values q0 and dq0,
# n points and timestep d with color col and linestyle marker. Return orbits computed
def simplectica(q0,F,ax1,ax2,ax3,col = 0, d = 10**(-4),n = int(16/10**(-4)),marker='-'):
    q = orb(n,q0=q0,F=F,d=d)
    dq = deriv(q,d=d)
    q = q[:-1,:]
    p = dq/2
    c=plt.get_cmap("inferno",125)(col)
    ax1.plot(q[:,0], p[:,0], linewidth = 0.25,  c=c )
    ax2.plot(q[:,1], p[:,1], linewidth = 0.25, c=c)
    ax3.plot(q[:,2], p[:,2], linewidth = 0.25, c=c)
    return q,p

# Compute for some n the estimation of the box-counting dimension for
```

```python
# a cover of epsilon = 2/(n-1) 6-squares
def box_count(points, n):
    lim = np.linspace(-1,1,n)
    H,_ = np.histogramdd(points, bins = (lim,lim,lim,lim,lim,lim))
    count = np.sum(H > 0)
    return count


#%%
"""
Exercise 2: Plot (q1,q2,q3) and (p1,p2,p3)
"""

# Compute orbit
d = 1e-3
q = orb(30000,[1,1,1],F)
dq = deriv(q)
p = dq/2

# Plot (q1,q2,q3)
fig = plt.figure()
ax = plt.axes(projection = '3d')
ax.scatter3D(1,1,1,c='r')
ax.plot3D(q[:,0],q[:,1],q[:,2])

# Plot (p1,p2,p3)
fig = plt.figure()
ax = plt.axes(projection = '3d')
ax.scatter3D(p[0,0],p[1,1],p[2,2],c='r')
ax.plot3D(p[:,0],p[:,1],p[:,2])

#%%

"""
Exercise 3: Plot (q1,p1), (q2,p2) and (q3,p3)
"""

# Compute ticks of interval: C
seq_q0 = np.linspace(-1,1.,num=5)

# Initialise figures
col = np.linspace(0,1,125)
c = 0
fig = plt.figure()
ax1 = fig.add_subplot(1,1, 1)
fig = plt.figure()
```

```python
ax2 = fig.add_subplot(1,1, 1)
fig = plt.figure()
ax3 = fig.add_subplot(1,1, 1)

# Store data for next exercise
allq = np.zeros((1,3))
allp = np.zeros((1,3))

# Plot orbit for initial conditions in C x C x C
for i in range(len(seq_q0)):
    for j in range(len(seq_q0)):
        for k in range(len(seq_q0)):
            q0 = [seq_q0[i], seq_q0[j], seq_q0[k]]
            q,p = simplectica(q0=q0,F=F,ax1=ax1,ax2=ax2,ax3=ax3,col=c,marker='ro',
                              d= 10**(-3),n=int(30/d))
            c = c + 1

            # Store data for next exercise
            allq = np.concatenate((allq,q),axis =0)
            allp = np.concatenate((allp,p),axis =0)

ax1.set_xlabel("q(t)", fontsize=12)
ax1.set_ylabel("p(t)", fontsize=12)
ax2.set_xlabel("q(t)", fontsize=12)
ax2.set_ylabel("p(t)", fontsize=12)
ax3.set_xlabel("q(t)", fontsize=12)
ax3.set_ylabel("p(t)", fontsize=12)
plt.show()

#%%
"""
Exercise 4: Hausdorff dimension
"""
narr = range(5,28)
points = np.array(list(zip(allq[:,0],allq[:,1],allq[:,2],allp[:,0],allp[:,1],allp[:,2])))

# Normalize
m = np.max(np.abs(points))
points = points/m

# Test dimension d
d = 2.75

Hd = []
He = []
for n in narr:
```

```
    # count boxes with side epsilon=2/(n-1)
    count = box_count(points,n)

    # diameter if each box
    diam = np.sqrt(24)/(n-1)
    Hd.append(count*(diam**d))

    eps = 2/(n-1)
    He.append(np.log(count)/np.log(1/eps))

# Plot evolution of H^d_{\epsilon}
plt.figure()
plt.plot(narr,Hd)
plt.xlabel('$n$')
plt.ylabel('$H^{d}_{\epsilon}(E)$')

# Plot evolution of dim_{box}(\epsilon)
plt.figure()
plt.plot(narr,He)
plt.xlabel('$n$')
plt.ylabel('$dim_{box}(\epsilon)$')
```