

Deber Seminario

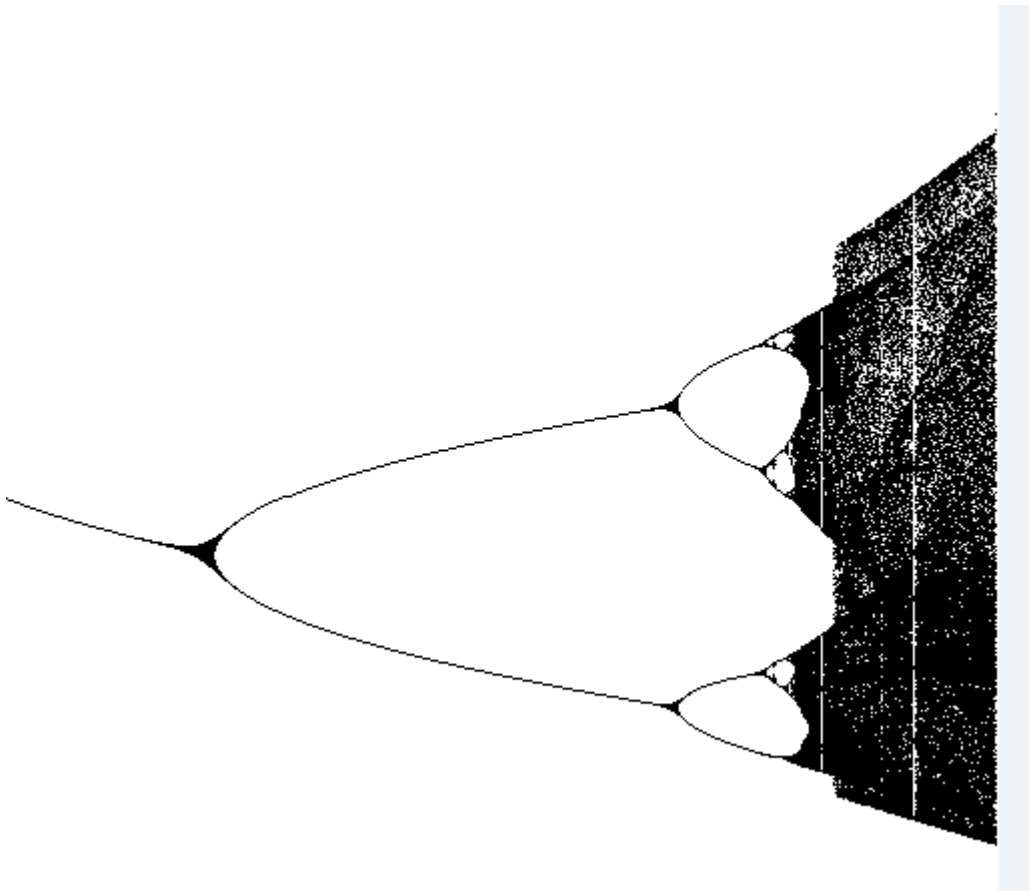
Fausto Fabian Crespo Fernandez

Grafo bifurcacion Henon

En este deber debíamos analizar el mapa de Henon:

$$f(x, y) = (a - x^2 + by, x)$$

Para diferentes valores de los parámetros a y b . En la primera parte se tomó $b=0.4$ y $0 \leq a \leq 1.25$ y se aplicó 1000 veces el mapa, empezando con el punto $(x, y) = (0, 0.5)$. Esto se hizo para valores de a igualmente espaciados en $[0, 1.25]$. En el eje x se graficó los valores de a y en el eje y los valores de x correspondiente a las aplicaciones del mapa. Se obtuvo el siguiente grafico que es similar al de la página 74 del libro "Chaos. An introduction to dynamical systems" Alligood, Yorke and Sauer. Springer 2000



El código usado es una modificación del código en Python propuesto para el grafico de bifurcación del mapa logístico en:

<http://code.activestate.com/recipes/577332-bifurcation-diagram-of-logistic-map/>

En nuestro caso se cambió el valor de la coordenada vertical donde se pone el color negro en la imagen generada, pues a diferencia del mapa logístico donde los valores de la x están entre 0 y 1,

en el mapa de Henon la x y la y que son imágenes del mapa pueden ser positivas o negativas y pueden sobrepasar en valor absoluto a 2.5. Además para darle mayor resolución a la foto se graficó a partir de la iteración 30 en lugar de a partir de la iteración 500 como en el código para el mapa logístico.

Además este código usa la librería grafica de Python PIL.

Código en Python

```
import math
import numpy as np
from PIL import Image
ainic=0
afin=1.25
b=0.4
def HenonMap(x,y):
    return [a-x*x+b*y, x]

widthimage = 500
heightimage = 500
image = Image.new("RGB", (widthimage, heightimage), "white")

arrayparamA=np.array(np.arange(ainic, afin, (afin - ainic) / (widthimage - 1)))
numberiter = 1000
d=-80
for i in range(len(arrayparamA)):
    a=arrayparamA[i]
    x = 0
    y = 0.5
    for j in range(numberiter):
        [x ,y] = HenonMap(x,y)
        if j > 30:
            if x >=0 and x <=2.5:
                d=int(heightimage / 2.0) +int((x / 2.5) * (heightimage / 2.0))
            elif x<0 and x >-2.5:
                d=int(heightimage / 2.0) -int((abs(x) / 2.5) * (heightimage / 2.0))
            if d!= -80:
                image.putpixel((i,d), (0, 0, 0))

image.show()
```