

El orden a partir del caos

Moreno Hernández Ana Isabel
Universidad de Guanajuato
 División de Ciencias e Ingenierías
 Loma del Bosque 103, Lomas del Campestre
 C. P. 37150
 León, Guanajuato
 México

(e-mail: morenoha2016@licifug.ugto.mx)

Resumen

El caos es una complejidad generada por la relación entre dos o más factores. Puede predecirse por la forma que tienen las ecuaciones diferenciales de un sistema, pero no siempre es así, se necesitan otras técnicas para concluir que se está ante la presencia de caos. En este documento se presentan diferentes secciones en las que se trata la teoría del caos, cada una de ellas con definiciones importantes, ejemplificadas con algunos modelos caóticos y diagramas que permiten un mejor entendimiento. La sección de bifurcaciones es una sección extra sobre la teoría del caos y es de carácter teórico. Después de esta sección, se analiza un sistema mecánico caótico, el cual es una partícula amortiguada en un doble pozo de potencial. Al final se encuentran varios apéndices de los códigos que se utilizaron para los diagramas, los cuales fueron programados en Jupyter.

Debemos intentar comprender el comienzo del universo a partir de bases científicas. Puede que sea una tarea más allá de nuestras capacidades, pero al menos deberíamos intentarlo.

-Stephen Hawking

Las ecuaciones diferenciales han sido objeto de estudio por un largo periodo, desde que Isaac Newton y Gottfried Leibniz elaboraron el cálculo diferencial e integral hasta ahora. En la época de Newton, era y todavía es impresionante la manera en la que con las leyes de Newton se puede describir la dinámica de un sistema, ya que se obtienen ecuaciones diferenciales que nos permiten encontrar las soluciones para el movimiento.

Después surgieron nuevos problemas, ya que no todas las ecuaciones diferenciales que se obtenían al hacer un análisis de fuerzas eran lineales, por lo tanto, era difícil resolverlas analíticamente. Para estos casos, se intentaba reducir grados de libertad del sistema, despreciar algunas variables e incluso se expandían funciones con series de Taylor para poder resolverlas, pero esto no siempre funcionaba y tampoco generaba un entendimiento total sobre la dinámica de un sistema. Afortunadamente, en los años de 1880 a 1910 [6], Poincaré introdujo nuevas técnicas para encontrar las soluciones de las ecuaciones diferenciales que describen a un sistema, una de ellas fueron las conocidas secciones de Poincaré;

y no sólo eso, también notó que el movimiento de algunos sistemas dinámicos era caótico.

Por un periodo de tiempo, se analizaron las ecuaciones de movimiento de un oscilador simple, amortiguado o amortiguado y forzado, con el objetivo de comprender la diferencia en el movimiento resultante de estos tres casos. Después, se empezó a estudiar sobre los puntos en los que un sistema físico tiende a moverse, preguntándose si se trataba de un punto fijo estable o inestable. Esta estabilidad fue estudiada por Lyapunov, quien elaboró a los conocidos exponentes de Lyapunov que pueden determinar si un sistema es caótico o no. Luego se descubrió la existencia de oscilaciones periódicas entre dos puntos fijos, los cuales generaban una órbita periódica, como es el caso de algunos mapeos.

Los puntos fijos estables de un sistema físico son atractores del movimiento del sistema, lo cual puede observarse al graficar el espacio fase, ya que el movimiento se concentra alrededor de ese punto. Para aquellos sistemas en donde se presenten geometrías complicadas en las soluciones de las ecuaciones diferenciales que lo describen, Ruelle y Takens denominaron al espacio fase de estas geometrías como un espacio fase caótico. Sin embargo, si estos sistemas dinámicos, a parte de ser caóticos, presentaban movimientos en ciertas regiones del espacio fase, se les definió como atractores extraños.

Existen múltiples relaciones, funciones, mapeos, modelos en donde se concluya que un sistema es caótico. Muchas técnicas permiten estudiar el caos utilizando primeros principios y a partir de bases científicas. Lo importante es que, a pesar de que se trate de caos, se pueda llegar a un orden.

I. MAPEOS

Un **sistema dinámico** es un sistema en el que interviene la variación en el tiempo, ya que su estado depende de éste. El estado final de un sistema dinámico es determinado por su estado anterior.

Aquellos sistemas dinámicos en los que el tiempo es discreto y no continuo, son conocidos como **mapeos**. Básicamente, a una función $y(x)$ se le asigna un valor para la variable x , lo cual genera una solución para y . Lo interesante es que este valor para y será el siguiente para x y así repetidamente.

Los mapeos surgen de varias maneras: como herramientas para analizar ecuaciones diferenciales, como modelos de un fenómeno natural -en algunos fenómenos puede considerarse el tiempo discreto- y como simples ejemplos de **caos** [1].

En esta sección nos limitaremos a analizar mapeos de una dimensión, los cuales se definen como una función de una sola variable real y tienen la siguiente forma:

$$x_{n+1} = f(x_n) \quad (1)$$

Dada la ecuación 1 y un valor inicial x_0 , se pueden encontrar los valores x_1, x_2, \dots , los cuales se conocen como la **órbita** del mapeo que empieza en x_0 .

Existen funciones en las que por más valores de x_n que se introduzcan, la función se satura y siempre se obtiene un mismo valor. A este valor se le conoce como **punto fijo**. Formalmente, si un valor x_p satisface $x_{n+1} = f(x_p) = x_p$, entonces x_p es un punto fijo.

Un ejemplo de mapeo es el de la función $x_{n+1} = 2x_n(1 - x_n)$, donde los puntos fijos son los que satisfacen la ecuación $x_p = 2x_p(1 - x_p)$. En este caso, los puntos fijos son $x_p = 1/2$ y $x_p = 0$.

Los puntos fijos de un mapeo pueden ser **estables** o **inestables**. Para determinar la **estabilidad** de un punto fijo, se tiene que saber si la órbita del mapeo es atractiva o repulsiva a este punto. Si es atractiva, la órbita se

estanca en el punto fijo, es decir, los valores x_0, x_1, x_2, \dots tienden al punto x_p , por lo tanto el punto fijo es estable; sin embargo, si es repulsiva, los valores x_0, x_1, x_2 se alejan de x_p , lo que significa que es inestable.

Existen diagramas conocidos como **telarañas** o **cobwebs** en inglés, en los cuales se representa la órbita de un mapeo. En este tipo de diagramas se puede saber si un punto fijo es estable o inestable. En la figura 1 se puede observar la órbita del mapeo $x_{n+1} = 2x_n(1 - x_n)$, el cual tiene dos puntos fijos: 0 y 0.5. Como puede verse en la figura, el punto fijo $x_p = 0$ es un punto inestable, ya que la órbita se aleja; sin embargo, $x_p = 0.5$ es un punto estable, debido a que la órbita se estanca en este valor.

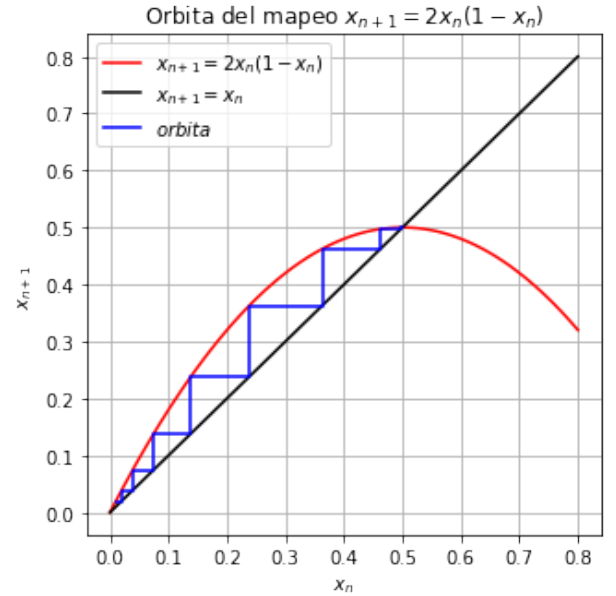


Figura 1. Gráfica de la órbita del mapeo $x_{n+1} = 2x_n(1 - x_n)$. El código hecho en Python para graficar la órbita de este mapeo se muestra en el apéndice A.1.

Una forma más general de saber si un punto fijo es estable o inestable es considerar una órbita que se aproxime al punto fijo x_p , es decir, $x_n = x_p + \delta_n$. Para determinar si δ_n crece o decrece cuando n incrementa podemos hacer una expansión en serie de Taylor del mapeo $x_{n+1} = f(x_n)$ alrededor de x_p :

$$x_{n+1} = f(x_n) = f(x_p + \delta_n) = f(x_p) + f'(x_p)\delta_n + O(\delta_n^2) \quad (2)$$

pero $x_{n+1} = x_p + \delta_{n+1}$ y $f(x_p) = x_p$, por lo tanto, la ecuación 2 se reduce a

$$\delta_{n+1} = f'(x_p)\delta_n + O(\delta_n^2) \quad (3)$$

Si el término $O(\delta_n^2)$ es muy pequeño, obtenemos un nuevo mapeo: $\delta_{n+1} = f'(x_p)\delta_n$. De esta forma, podemos

encontrar las soluciones del mapeo escribiendo algunos términos: $\delta_1 = f'(x_p)\delta_0$, $\delta_2 = f'(x_p)\delta_1 = (f'(x_p))^2\delta_0$, $\delta_3 = f'(x_p)\delta_2 = (f'(x_p))^3\delta_0$ y así, por inducción, la expresión final es $\delta_n = (f'(x_p))^n\delta_0$. Si $|f'(x_p)| < 1$, entonces δ_n tiende a 0 cuando n tiende a infinito, por lo tanto x_p es un punto fijo estable; de lo contrario, si $|f'(x_p)| > 1$, entonces δ_n tiende a infinito cuando n tiende a infinito, por lo tanto, x_p es un punto fijo inestable.

Otro ejemplo de mapeos son los **mapeos logísticos**, los cuales se definen con la siguiente función:

$$x_{n+1} = rx_n(1 - x_n) \quad (4)$$

La ecuación 4 tiene la forma de una parábola que abre hacia abajo, similar a la función del ejemplo anterior con $r = 2$.

Si $0 \leq r < 1$, entonces x_n tiende a cero cuando n tiende a infinito, debido a que x_n obtiene valores más pequeños en cada iteración. Además, el mapeo en la ecuación 4 tiene el punto fijo $x_p = 0$, lo que implica que $f'(x_p) = r(1 - 2x_p) = r > 0$, por lo tanto $x_p = 0$ es estable. Si $1 < r < 3$ se observa que x_n crece y oscila hasta que alcanza un valor constante; sin embargo, si $r > 3$, x_n también crece pero luego oscila alrededor de un punto (véase la figura 2).

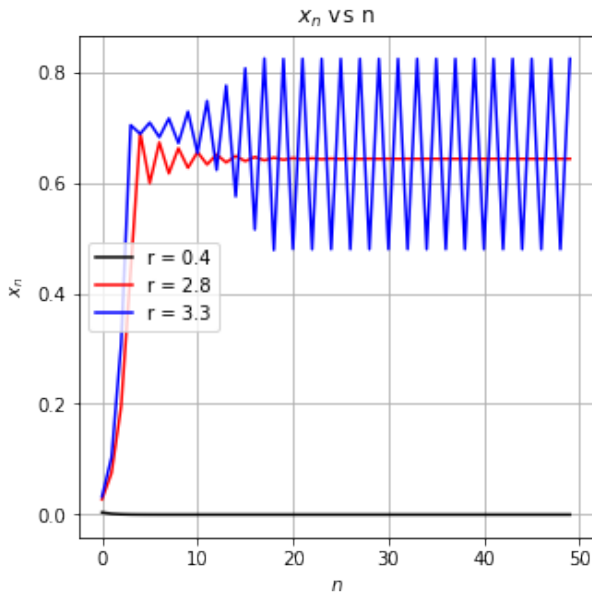


Figura 2. Gráfica de x_n contra el número de iteraciones n . El código hecho en Python para este grafico se muestra en el apéndice A.2.

Cuando $r = 3.3$, los puntos fijos son $x_{p1} = 0$ y $x_{p2} = 0.6969...$ [1], los cuales son inestables, por lo tanto, no atraen órbitas. En realidad, se observa que la

órbita del mapeo logístico es atraída hacia un punto p_1 , pero en la siguiente iteración es atraída por p_2 . Debido a lo anterior, se observa que la órbita se alterna entre estos dos puntos. Matemáticamente, esto luce como $p_2 = f(p_1)$ y $p_1 = f(p_2)$. Concluimos que se trata de una periódica oscilación entre x_{p1} y x_{p2} que atrae órbitas y, por lo tanto, generan una **órbita periódica**. A esto se le conoce como la **segunda iteración de f** y es definida como

$$f^{(2)} = f(f(x)) \quad (5)$$

Lo que implica que

$$p_1 = f^{(2)}(p_1) \quad (6)$$

$$p_2 = f^{(2)}(p_2) \quad (7)$$

Así, si p satisface $f^{(k)}(p) = p$, entonces p es un punto periódico de periodo k y la órbita que inicia en este punto es conocida como órbita de periodo k [3].

Notamos lo mismo cuando $r = 5$, ya que la órbita se repite cada 4 iteraciones, por lo tanto la órbita es de periodo 4. Mientras más grande sea r , más se incrementará el periodo. Así, si denotamos a r_n como el valor donde x_n es periodo 2, entonces existe un valor de r_∞ en donde x_n es periodo ∞ . Este valor es aproximado a 3.569946... [2]. Sin embargo, si $r > r_\infty$, x_n no tiene un periodo definido y nunca es atraída por un punto en particular, lo cual puede observarse como algo caótico en la órbita del mapeo (véase la figura 3).

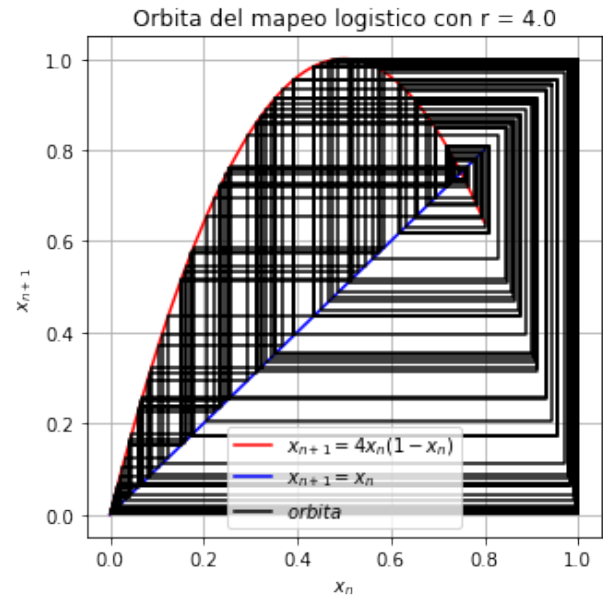


Figura 3. Gráfica de la órbita del mapeo logístico con $r = 4.0$. El código hecho en Python para este grafico se muestra en el apéndice A.3.

Es posible implementar un código de programación en Python para poder observar lo que sucede con x_n cuando n tiende a infinito para distintos valores de r . Para generarlo se necesitan dos ciclos, en uno de ellos debe variarse r y en el otro deben encontrarse los puntos x_n con ese valor de r dando un valor inicial x_0 . Lo que se debe de tomar en cuenta es que, al inicio de las órbitas, x_n no es atraído por un punto estable, es decir, x_n no llega a un valor constante. Por eso, al generar el código, lo que se debe graficar son los valores de x_n donde ya es constante", lo que podría ser a partir de $n = 300$, contra el valor de r correspondiente a esa órbita. En la figura 4 se muestran los valores obtenidos de x_n a partir de $n = 300$ cuando r tiene valores entre 3.4 y 4.

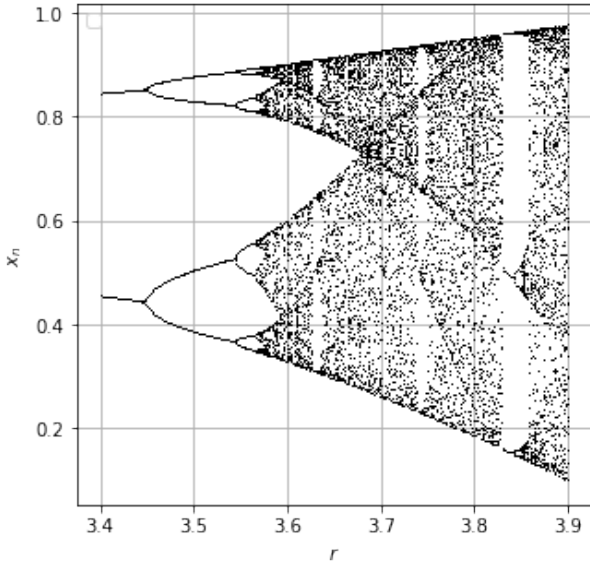


Figura 4. Gráfica de la órbita del mapeo logístico con diferentes r . El código hecho en Python para este grafico se muestra en el apéndice A.4.

Nótese, a partir de esta última gráfica, que x_n se encuentra entre dos valores para un r entre 3.4 y 3.45 aproximadamente, lo cual tiene sentido por lo analizado en la figura 2. No obstante, x_n está entre más valores si r incrementa hasta 3.569946. Luego, para r mayores a 3.569946 se observa una imagen caótica. A este tipo de diagramas, se les conoce como **diagramas de bifurcación**.

II. EXPONENTES DE LYAPUNOV

Para que un sistema sea realmente **caótico** debe ser sensible a condiciones iniciales [1].

Los exponentes de Lyapunov λ representan la divergencia entre las trayectorias de dos puntos cercanos. Para un sistema descrito por un mapeo general de una

dimensión $x_{n+1} = f(x_n)$, consideremos una condición inicial x_0 y una segunda condición muy cercana $x_0 + \delta_0$. Se espera que la diferencia entre los dos x_n crezca exponencialmente cuando n incrementa para predecir que el mapeo es caótico. Después de n iteraciones, la diferencia δ_n entre los dos valores x_n es aproximadamente

$$\delta_n = \delta_0 e^{n\lambda} \quad (8)$$

La diferencia inicial entre las órbitas es δ_0 , y después de una iteración, la diferencia δ_1 es

$$\delta_1 = f(x_0 + \delta_0) - f(x_0) \quad (9)$$

Después de n iteraciones la diferencia es

$$\delta_n = f^{(n)}(x_0 + \delta_0) - f^{(n)}(x_0) \quad (10)$$

Sustituyendo δ_n en la ecuación 10 obtenemos lo siguiente

$$e^{n\lambda} = \frac{f^{(n)}(x_0 + \delta_0) - f^{(n)}(x_0)}{\delta_0} \quad (11)$$

Tomando logaritmos por ambos lados de la ecuación y despejando λ , la ecuación anterior se reescribe como

$$\lambda = \frac{1}{n} \ln \left| \frac{f^{(n)}(x_0 + \delta_0) - f^{(n)}(x_0)}{\delta_0} \right| \quad (12)$$

Ahora, si aplicamos el límite cuando δ_0 tiende a cero en ambos lados de la ecuación, tomando en cuenta que λ es una constante (el límite de una constante es una constante), se obtiene la definición de derivada para $f^{(n)}$ evaluada en x_0 :

$$\lambda = \lim_{\delta_0 \rightarrow 0} \frac{1}{n} \ln \left| \frac{f^{(n)}(x_0 + \delta_0) - f^{(n)}(x_0)}{\delta_0} \right| = \frac{1}{n} \ln \left| \frac{df^{(n)}(x)}{dx} \right|_{x_0} \quad (13)$$

Recordemos que $f^{(n)}(x_0)$ es la iteración n de la función $f(x_0)$, es decir

$$f^{(n)}(x_0) = f(f(f(\dots f(x_0)))) \quad (14)$$

Luego, aplicando diferenciación por regla de la cadena, la derivada de $f^{(n)}(x)$ evaluada en x_0 es

$$\left. \frac{df^{(n)}(x)}{dx} \right|_{x_0} = \left. \frac{df}{dx} \right|_{x_{n-1}} \left. \frac{df}{dx} \right|_{x_{n-2}} \dots \left. \frac{df}{dx} \right|_{x_0} \quad (15)$$

Lo que implica que el logaritmo natural de la derivada de $f^{(n)}$ puede expresarse como una suma de logaritmos. Luego, si aplicamos el límite cuando $n \rightarrow \infty$, entonces el exponente de Lyapunov λ es

$$\lambda = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln \left| \frac{df(x_i)}{dx} \right| \quad (16)$$

De la ecuación 8 podemos notar que si λ es negativo, las dos órbitas eventualmente convergen, pero si λ es

positivo, las dos órbitas divergen y se presenta **caos**.

A partir de la ecuación obtenida para el exponente de Lyapunov, se puede calcular el valor de λ para un r particular del mapeo $x_{n+1} = rx_n(1 - x_n)$ y concluir si el sistema es caótico o no, dependiendo si λ es positivo o negativo. Así, podemos generar un código de programación en el cual se grafique el valor del exponente de Lyapunov contra el valor de r . Para lograrlo, inicialmente es necesario un valor fijo para r y tener una condición inicial para el mapeo x_0 . Después se hace el mapeo un número de 300 iteraciones, sólo para dejar que se estabilice (véase la sección de mapeos) y se obtenga una órbita periódica (si es que la tiene). Después, se programan otras 1000 iteraciones, se hace la operación $\ln(f'(x_n)) = \ln(r - 2rx_n)$ en cada iteración y se suma a los logaritmos de iteraciones anteriores. Conforme a la ecuación 16, el exponente de Lyapunov se obtiene dividiendo la suma total de logaritmos entre las 1000 iteraciones. Finalmente, este procedimiento se repite para el siguiente valor fijo de r y se grafica el valor de λ correspondiente a cada r . En la figura 5 se muestran los valores para λ cuando r tiene valores entre 3.0 y 4.0.

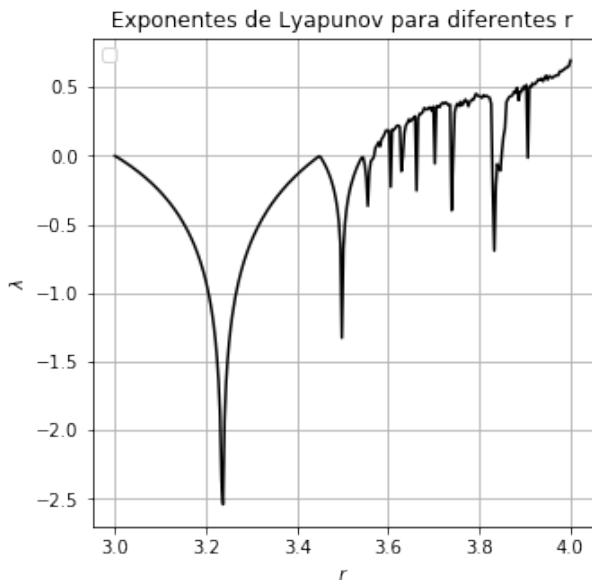


Figura 5. Valores del exponente de Lyapunov para distintos valores de r del mapeo logístico. El código hecho en Python para este gráfico se muestra en el apéndice A.5.

A partir de la figura 5, puede observarse que para valores de r menores a 3.57, el exponente de Lyapunov es negativo, por lo tanto, el sistema es estable. Sin embargo, para $r > 3.57$, se observan valores para λ negativos, por lo tanto, el sistema es caótico. Esto se relaciona con lo concluido en la sección de mapeos, ya

que en la figura 4, la órbita del mapeo logístico es estable para $r < 3.57$, pero en r mayor a 3.57 se presenta caos.

III. FRACTALES

Un **fractal** es una figura geométrica complicada y no importa cuánto se reduzca el intervalo de observación, seguirá siendo complicada. El término fractal se le asocia a B. Mandelbrot, quien definió a los fractales como geometrías complicadas con repetición de estas geometrías en escalas más pequeñas. Los fractales son utilizados comúnmente para observar patrones en sistemas caóticos.

Uno de los fractales más conocidos, es el **conjunto de Cantor** o **polvo de Cantor**. Esta estructura inicia con un intervalo cerrado $S_0 = [0,1]$, donde la longitud del segmento inicial es 1. Luego, en la primera iteración, se remueve la tercera parte que está a la mitad, por lo tanto, $S_1 = [0,1/3] \cup [2/3,1]$. En la segunda iteración, se quita la tercera parte que está a la mitad de los dos segmentos de S_1 , por lo tanto $S_2 = [0,1/9] \cup [2/9,3/9] \cup [6/9,7/9] \cup [8/9,1]$; y así repetidamente para los siguientes S_n . En la figura 6 se muestra el conjunto de Cantor con 3 iteraciones. Sabemos que S_1 contiene dos segmentos con longitud de $(1/3)$, S_2 tiene 4 segmentos con longitud de $(1/9)$ y por inducción, S_n consiste de 2^n segmentos, cada uno de longitud $(1/3)^n$, por lo tanto, la longitud de S_n es $(2/3)^n$.



Figura 6. Conjunto de Cantor. Puede observarse que en cada iteración se obtienen más segmentos con menor longitud.

El conjunto de Cantor es una estructura que nunca termina, ya que aunque se haga un aumento en la figura, se seguirán observando segmentos a los que se les quita la tercera parte que está a la mitad para obtener más segmentos, aunque estos nuevos segmentos serán más difusos.

Consideremos un número N elementos con una cierta longitud característica l , necesarios para formar un volumen, una superficie o una recta. Se busca que exista un número D que nos proporcione la escala existente entre N y l , de forma que Nl^D esté acotado y sea 1 si es que los elementos están normalizados [5].

Recordemos que, para el caso del polvo de Cantor, el número de segmentos del conjunto S_n es 2^n y la longitud de cada segmento está dada por $(1/3)^n$ y el número de segmentos es 2^n . Se busca entonces la siguiente relación:

$$\lim_{n \rightarrow \infty} Nl^D = \lim_{n \rightarrow \infty} 2^n \left(\frac{1}{3}\right)^{nD} = 1 \quad (17)$$

Esta igualdad sólo se cumple si $2(1/3)^D = 1$, ya que el límite provocaría que el término $2(1/3)$ sea infinito o cero. Así, el valor de D donde Nl^D está acotado es

$$D = \frac{\ln(1/2)}{\ln(1/3)} = 0,631 \quad (18)$$

A este valor se le conoce como **dimensión de medida** D_m [5].

Otra definición útil es la **dimensión topológica** D_T , la cual es un valor entero que coincide con la dimensión del espacio de soporte [5]. Así, la dimensión topológica es igual a cero para un punto, igual a 1 para una recta, 2 para una superficie y 3 para un volumen. Esto es necesario conocer, ya que un objeto se considera fractal si su dimensión topológica es menor que su dimensión de media.

En esta sección, es importante mencionar la **dimensión de caja**, la cual se relaciona con un elemento llamado **box-counting**.

Básicamente, el box-counting se refiere a la cantidad de cajas necesarias de m dimensiones con longitud δ que pueden cubrir un objeto de un espacio de m dimensiones. Como ejemplos pueden ser segmentos con longitud δ necesarios para cubrir una recta L , cuadrados con área l^2 que cubren una superficie A o cubos en un volumen V .

Si N es la cantidad de elementos (cajas) con longitud δ , para una curva de longitud L , $N(\delta) \propto L/\delta$; para una superficie de área A , $N(\delta) \propto A/\delta^2$; entonces un conjunto K tiene dimensión D si la cantidad de elementos (cajas) que cubren a ese conjunto está dada por

$$N(\delta) = C(1/\delta)^D \quad (19)$$

Por lo tanto, la dimensión D del conjunto K es

$$D = \frac{\ln N(\delta) - \ln C}{\ln(1/\delta)} \quad (20)$$

Si consideramos que δ es muy pequeña, el segundo término del numerador de la ecuación anterior se vuelve

despreciable. Así, si tomamos el límite cuando $\delta \rightarrow 0$, definimos la dimensión de cada D_b del conjunto K como

$$D_b = \lim_{\delta \rightarrow 0} \frac{\ln N(\delta)}{\ln(1/\delta)} \quad (21)$$

si es que el límite existe. Lamentablemente, no siempre puede realizarse el límite, debido a que no siempre se establece una función $N(\delta)$, por lo tanto, la forma de estimar D'_b es

$$D'_b = \frac{\ln N(\delta)}{\ln(1/\delta)} \quad (22)$$

donde D'_b es un valor cercano a D_b .

Para el caso del conjunto de Cantor, la dimensión de caja es

$$D'_b = \frac{\ln 2^n}{\ln \frac{1}{1/(1/3^n)}} = \frac{n \ln 2}{n \ln 3} = \frac{\ln 2}{\ln 3} = 0,631 \quad (23)$$

Nótese que D'_b coincide con el resultado obtenido para la dimensión de medida D_m . Además, debido que el conjunto Cantor está conformado por puntos discretos, la dimensión topológica correspondiente a este conjunto es $D_T = 0$. Del resultado obtenido en la ecuación 23, concluimos que $D_T < D_m = D'_b$, por lo tanto, el conjunto cantor es considerado como un fractal.

Otro fractal muy conocido es el **conjunto de Julia**. Este conjunto es definido por el siguiente mapeo

$$z_{n+1} = z_n^2 + c \quad (24)$$

donde $z = x + iy$ y $c = a + ib$ están compuestos de dos variables y son números complejos.

Si el número complejo c es igual a cero, entonces el punto fijo x_p correspondiente al mapeo $z_{n+1} = z_n^2 + c$ es igual a cero y es un punto fijo estable, por lo tanto, todos los puntos z_n son atraídos a $x_p = 0$. Sin embargo, si c es diferente de cero, no siempre se obtendrán puntos fijos estables para el mapeo. Debido a esto, nace el **conjunto de Mandelbrot**, en el cual solo se grafican los distintos valores de c en los cuales el mapeo $z_{n+1} = z_n^2 + c$ con $z_0 = 0$ converge a un punto fijo dentro de un intervalo que depende de los valores de la parte real e imaginaria de c . El plano donde se grafican estos puntos es el plano complejo.

Puede realizarse un código de programación para observar el fractal generado por el conjunto de Mandelbrot. En este código, debe variarse el número complejo c , donde su parte real a e imaginaria b , por ejemplo, puede ser un conjunto de números en el rango de -2 a 2.

Así, para un valor de c , debe encontrarse si el mapeo $z_{n+1} = z_n^2 + c$ converge a un punto fijo, en este caso, dentro del intervalo de -2 a 2. Si el valor de c hace que los valores z_n converjan a un punto dentro del intervalo, se grafica este valor en el plano complejo, y si no, no se grafica. La forma de delimitar qué valores se grafican y cuales no, es introduciendo un condicional que verifique la magnitud de z_n . Si su magnitud es mayor que 2, el valor de c es ignorable, pero si es menor que dos, se grafica. En la figura 7 se muestra el fractal del conjunto Mandelbrot con un intervalo de -2 a 2 en el plano complejo.

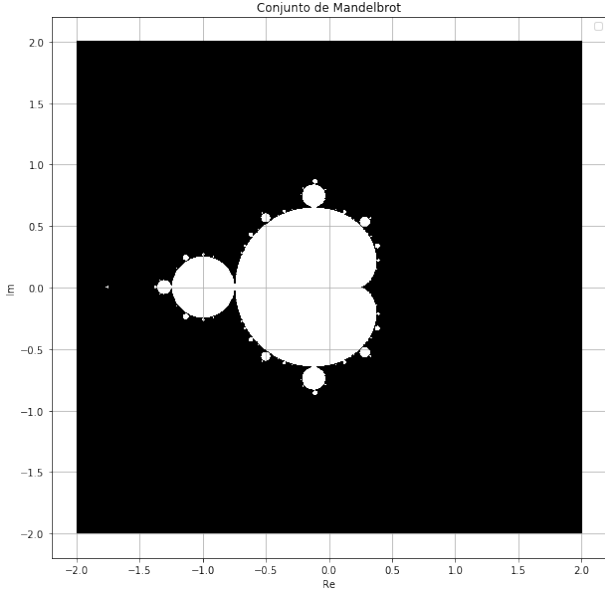


Figura 7. Conjunto de Mandelbrot. En este caso se graficaron todos los valores de c en los cuales el mapeo diverge, pero para graficar lo contrario, sólo se cambia la condición en el código. El código realizado en Jupyter se muestra en el apéndice A.6.

Este conjunto presenta autosimilaridad, la cual es una de las propiedades más importantes para definir a un fractal. Si se hace un aumento en la imagen, podrá observarse la repetición de la figura 7 en una escala más pequeña.

IV. SECCIONES DE POINCARÉ

Existen sistemas físicos que son descritos con ecuaciones diferenciales difíciles de resolver analíticamente. Debido a esto, Poincaré desarrolló técnicas para conocer la naturaleza de un sistema y una de ellas fueron las llamadas **secciones de Poincaré**, ya que éstas proporcionan información útil para la dinámica de un sistema físico.

Un sistema dinámico descrito por una ecuación diferencial de la siguiente forma

$$\frac{dx(t)}{dt} = F(x, t) \quad (25)$$

describe trayectorias en el espacio fase y pueden ser fácilmente analizadas. El problema inicia cuando se tiene un espacio fase con dimensiones más grandes. En este caso, pueden utilizarse secciones de Poincaré para analizar el sistema.

Si al espacio fase de un sistema físico se colocan varios planos paralelos entre sí, los cuales se interponen en el camino de las trayectorias del sistema, los puntos que se proyectan en estos planos se les conoce como secciones de Poincaré. Para obtener estas secciones, se pueden generar mapeos que envíen los puntos proyectados del primer plano al segundo, del segundo plano al tercero y así repetidamente.

Básicamente, el recorrido de la trayectoria de un sistema físico en un espacio fase se intersecta con planos paralelos entre sí. Los puntos que se intersectan son conocidos como A_1, A_2, \dots, A_n [4], n dependiendo de cuántos cruzaron el plano. Las secciones de Poincaré se forman cuando este conjunto de puntos se proyectan en uno de los planos paralelos, y pueden formarse geometrías simples o complicadas. Cabe mencionar que las secciones que presenten geometrías simples indican que la ecuación diferencial del sistema físico puede resolverse analíticamente; sin embargo, una geometría complicada puede representar que el sistema es caótico.

En realidad, las secciones de Poincaré son útiles para reducir un sistema físico de N dimensiones a $N-1$ dimensiones con el objetivo de enfocarse sólo en dos soluciones del sistema.

Para un mejor entendimiento de esta sección, se analizarán las secciones de Poincaré para un péndulo amortiguado y forzado. La ecuación de movimiento de este sistema es

$$\frac{d^2\theta}{dt^2} = -\frac{b}{m} \frac{d\theta}{dt} - \frac{g}{l} \sin\theta + \frac{F_0}{m} \cos(\omega_d t) \quad (26)$$

donde l es la longitud inextensible del péndulo, g es la aceleración de la gravedad, m es la masa que cuelga en el péndulo, b es el coeficiente de la fuerza amortiguadora, F_0 es la amplitud y ω_d es la frecuencia angular de la fuerza externa aplicada.

Ahora, definimos $A = F_0/m$, $\omega_0^2 = g/l$ y $\mu = b/ml^2$, por lo tanto, la ecuación 26 se reescribe

$$\frac{d^2\theta}{dt^2} = -\mu\frac{d\theta}{dt} - \omega_0^2\sin\theta + A\cos(\omega_d t) \quad (27)$$

La ecuación diferencial de segundo orden anterior puede ser reducida a dos ecuaciones de primer orden haciendo la sustitución

$$\omega = \frac{d\theta}{dt} \quad (28)$$

Y así, la ecuación 26 cambia a

$$\frac{d\omega}{dt} = -\mu\omega - \omega_0^2\sin\theta + A\cos(\omega_d t) \quad (29)$$

Debido a que este sistema de ecuaciones diferenciales no es lineal, resolverlo analíticamente es muy complicado, pero numéricamente con un código en Jupyter es sencillo, simplemente utilizando la función *odeint* de Python.

El diagrama fase de tres dimensiones de un péndulo amortiguado y forzado se muestra en la figura 8. El eje x está representado por la variable θ , y por ω y el eje z contiene los valores de $\omega_d t$ con $\omega_d = 2\pi$. Nótese que los planos que intersectan el recorrido de la trayectoria son las secciones de Poincaré, las cuales proyectan los puntos en un plano x,y para un z fijo.

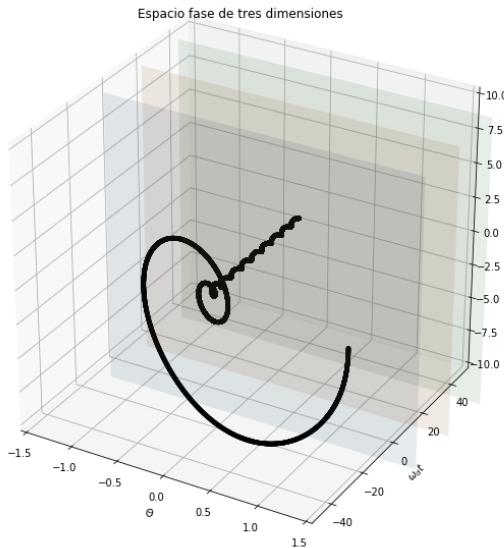


Figura 8. Diagrama fase en tres dimensiones de un péndulo amortiguado y forzado. En el eje x se muestran los valores obtenidos para $\theta(t)$, en z los valores de $\omega_d t$ y en y los valores para $\omega(t)$. El código realizado en Jupyter se muestra en el apéndice A.7, así como los valores utilizados para μ , ω_d y ω_0^2 .

En este caso, el diagrama fase de tres dimensiones resulta ser sencillo para analizar, y ofrece una gran

información sobre el movimiento del péndulo, pero esto no sucede en algunos sistemas físicos. Debido a eso, puede reducirse la dimensión del diagrama de fase, y observar el movimiento, en este caso, de dos variables utilizando las secciones de Poincaré.

Pueden dibujarse varios planos paralelos entre sí que intersecten el recorrido de la trayectoria para obtener las secciones de Poincaré. En este caso, puede realizarse un código que deje a la variable $z = \omega_d t$ como fija y obtener los puntos que se intersectan en este primer plano. Luego, generar otro valor para z y proyectar los puntos del anterior plano en este nuevo plano, y así sucesivamente. En la figura 9 se muestran los puntos proyectados en 50 planos paralelos separados por un tiempo de $3/50$ s.

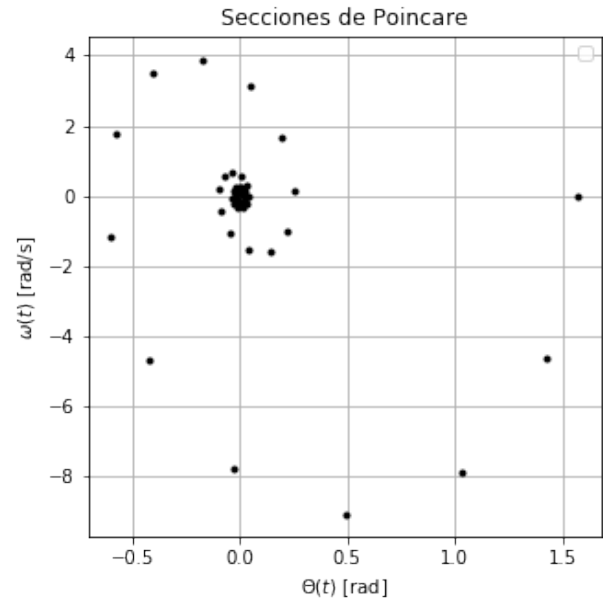


Figura 9. Secciones de Poincaré obtenidas al colocar 50 planos separados por $3/50$ s. En este caso, se proyectaron los puntos $x = \theta$ y $y = \omega$ en los planos. El código realizado en Jupyter se muestra en el apéndice A.8.

Nótese que la velocidad angular y el ángulo θ que describen el movimiento del péndulo amortiguado y forzado tienden a un valor muy aproximado a cero conforme el tiempo crece, aunque no se distingue si tienden a cero, para eso es necesario colocar más planos paralelos que originen más puntos. Esta sección de Poincaré en particular es útil para predecir si la velocidad y la posición del péndulo tienden a un valor definido o a tener valores caóticos.

Lo mismo puede realizarse asignando valores a cualquier otra variable para generar planos paralelos al plano

generado por las dos variables restantes. En la figura 10 se muestran los puntos proyectados en 70 planos dejando valores fijos para $x = \theta$, es decir, colocando planos paralelos al plano y,z .

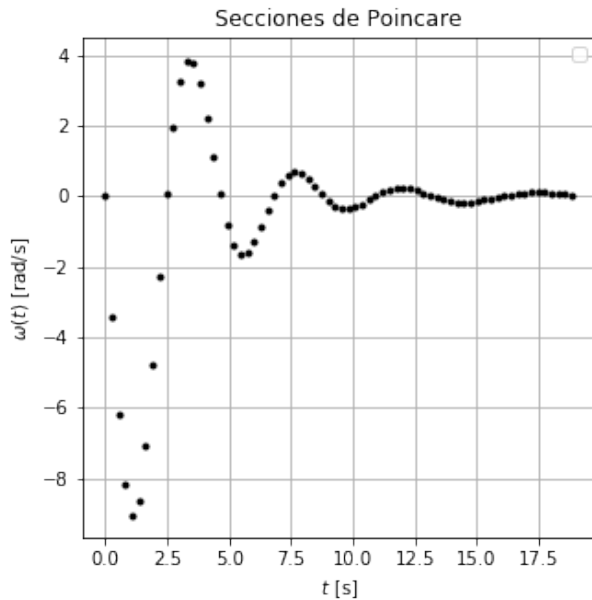


Figura 10. Secciones de Poincaré obtenidas al colocar 70 planos paralelos entre sí. En este caso, se proyectaron los puntos $y = y$ $z = \omega dt$ en los planos. El código realizado en Jupyter se muestra en el apéndice A.9.

De la figura 10 puede observarse que la velocidad angular tiende a un valor cercano a cero conforme el tiempo crece, es decir, la velocidad del péndulo es reducida conforme el tiempo avanza.

El ejemplo anterior fue una forma de ejemplificar las secciones de Poincaré para un sistema dinámico en particular que depende de tres variables, pero en la naturaleza pueden encontrarse sistemas físicos que dependan de más de tres variables, en los cuales su diagrama fase sería muy complicado de graficar. Las secciones de Poincaré pueden mostrar el movimiento de una o dos variables del sistema, y así poder obtener información sobre éste reduciendo la complejidad de su diagrama fase.

V. ATRACTORES EXTRAÑOS

En secciones anteriores se han incluido términos en donde nos referimos a puntos que son atractores, ya sea de órbitas en un mapeo o de trayectorias en un sistema físico. En la sección de mapeos se analizaron los puntos fijos de un mapeo de una dimensión. Si un punto fijo era estable, entonces la órbita del mapeo era atraída hacia ese punto; y si era inestable, entonces la órbita se

alejaba de este punto.

Un atractor extraño presenta dos características importantes. Los recorridos de las trayectorias del atractor se encuentran dentro de un espacio acotado y aún así, se separan unas de otras de forma exponencial. Es como si la trayectoria del atractor extraño se estirara, se doblara debido a su sensibilidad a condiciones iniciales, nuevamente se estirara y así repetidamente, como si fuera una goma de mascar que se alarga y se dobla continuamente, donde no se unen las partes plegadas entre sí. En estos atractores se observan trayectorias que parecen ser atraídas hacia uno o más puntos y presentan geometrías complejas.

Un atractor extraño muy conocido es el **atractor de Lorenz**. Las ecuaciones diferenciales que conducen a este atractor son

$$\dot{x} = -\sigma x + \sigma y \quad (30)$$

$$\dot{y} = rx - y - xz \quad (31)$$

$$\dot{z} = -bz + xy \quad (32)$$

Estas ecuaciones describen a un fluido en donde líquido caliente fluye por encima de él y líquido frío fluye hacia abajo, lo que genera una corriente que puede ir en sentido horario o antihorario. Las letras σ , r y b son parámetros. Al parámetro σ se le conoce como el número de Prandtl y a r como el número de Reynolds [3]. En este sistema físico, x representa la velocidad de flujo de la corriente, el parámetro b representa el ancho de los rollos de flujo, y indica la diferencia de temperatura entre el fluido caliente y frío, y z es proporcional al cambio de temperatura desde su equilibrio.

Para valores de r entre $[-\infty, 1]$, el origen es un punto fijo estable; cuando r tiene valores entre $[1, 13.93]$ existen dos puntos fijos estables y el origen es un punto fijo inestable. Para $13.93 < r < 24.06$ se presentan trayectorias caóticas, pero no son atraídas a ningún punto. Luego, para $24.06 < r < 24.74$, se observa un atractor caótico con dos puntos fijos estables. Lorenz notó que para valores de $\sigma = 10$, $b = 8/3$, el sistema se vuelve caótico cuando el número de Reynolds $r > 24.74$, ya que se observa un atractor, pero los puntos fijos que eran estables ya no lo son. Los valores de r mencionados anteriormente, fueron obtenidos del capítulo 9 de la referencia bibliográfica 3.

Lorenz lo definió como caótico debido a que era muy sensible a las condiciones iniciales y porque las trayectorias que se generaban no convergían a una solución. En la figura 11 se muestra el diagrama fase del atractor de Lorenz, con condiciones iniciales de (1.4, 1.4, 28) para (x, y, z) en $t = 0$ y con $b = 8/3$, $\sigma = 10$ y $r = 28$.

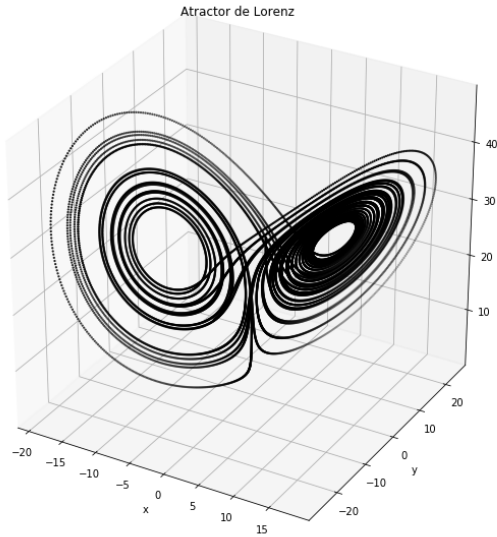


Figura 11. Atractor de Lorenz con $b = 8/3$, $\sigma = 10$ y $r = 28$. El código realizado en Jupyter se muestra en el apéndice A.10.

En la figura 11 puede observarse una geometría complicada para las trayectorias de las ecuaciones diferenciales antes mencionadas. Pareciese que las trayectorias son atraídas hacia un punto fijo, aunque en realidad no son atraídas a ninguno.

En esta sección, también es importante mencionar el **atractor de Rössler**. El científico alemán O. Rössler pudo encontrar un sistema de ecuaciones diferenciales no lineales que también generan un atractor caótico. Las ecuaciones que determinó son:

$$\dot{x} = -y - z \quad (33)$$

$$\dot{y} = x + ay \quad (34)$$

$$\dot{z} = b + z(x - c) \quad (35)$$

Cuando $a = b = 0,1$ y $c = 14$ puede observarse que se genera un atractor caótico, tal y como se muestra en la figura 12.

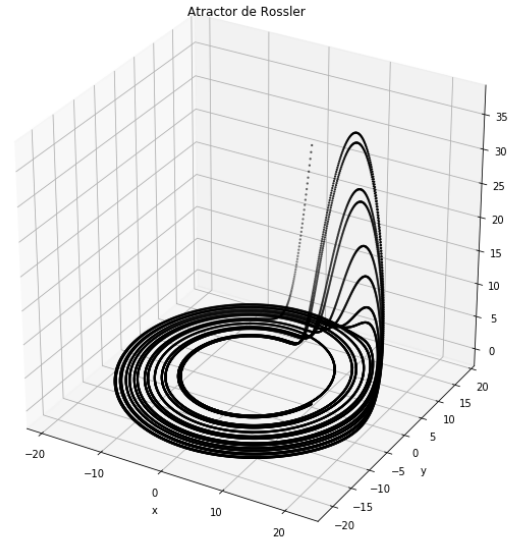


Figura 12. Atractor de Rössler con $a = b = 0,1$ y $c = 14$. El código realizado en Jupyter se muestra en el apéndice A.11.

Puede observarse cómo la trayectoria se estira en el eje z, luego se enrolla, vuelve al plano x,y y así sucesivamente.

VI. BIFURCACIONES

La trayectoria de un sistema dinámico puede cambiar al variar un parámetro. Los puntos fijos del sistema cambian su estabilidad o, en el caso de que ya no sean puntos fijos, se generan otros nuevos al variar este parámetro. La **bifurcación** es el cambio en la trayectoria de un sistema debido a la variación de un parámetro y los valores del parámetro donde ocurre el cambio se les conoce como **puntos de bifurcación**.

Existen dos tipos de bifurcaciones: el **nodo de silla de montar** o más bien **saddle-node** en inglés y la **bifurcación de doble periodo**. El saddle-node se genera debido que los puntos fijos de un sistema son creados o destruidos; mientras que la bifurcación de doble periodo se genera porque los puntos fijos pierden estabilidad. En la figura 13 se muestra el diagrama de bifurcación del mapeo $x_{n+1} = a - x_n^2$, con valores para a entre 0 y 1.3. Puede observarse a la izquierda un nodo de silla de montar y a la derecha dos bifurcaciones de doble periodo.

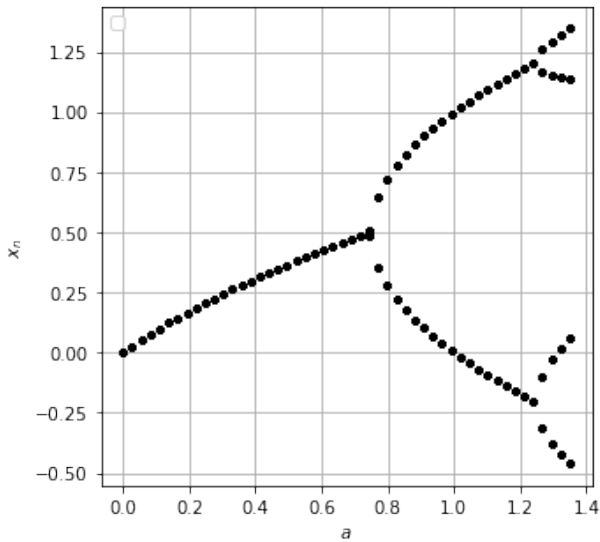


Figura 13. Diagrama de bifurcación para el mapeo $x_{n+1} = a - x_n^2$. El código realizado en Jupyter para este gráfico se muestra en el apéndice A.12.

El mapeo Hénon está definido por

$$h_a(x, y) = (a - x^2 + by, x) \quad (36)$$

donde a y b son dos parámetros. Si se elige a b como un parámetro fijo, ya sea con un valor de 0.4 y se grafica la bifurcación del mapeo de Hénon variando el parámetro a se obtiene la siguiente figura proyectada en el eje x :

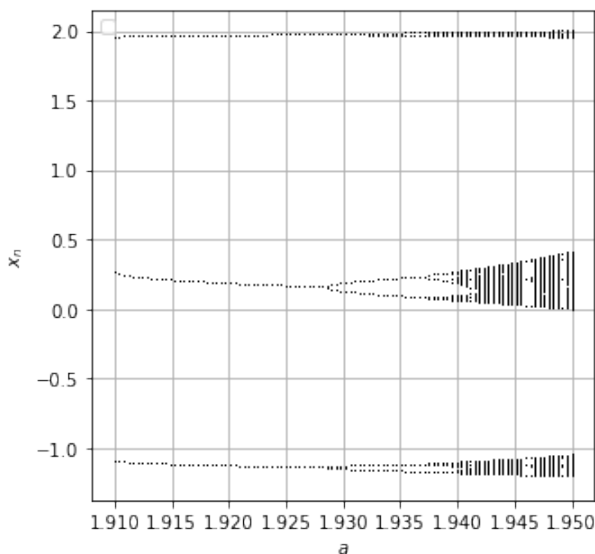


Figura 14. Diagrama de bifurcación para el mapeo de Hénon. El código realizado en Jupyter para este gráfico se muestra en el apéndice A.13.

Puede observarse en la figura 14 que se obtienen tres bifurcaciones de tipo nodo de silla de montar en el

intervalo de a de 1.91 a 1.95.

Existen situaciones en las que un punto fijo no cambia al variar un parámetro de un sistema, aunque sí puede cambiar su estabilidad. A esto se le conoce como **bifurcación transcítica**. En este caso, no se observaría nodo de silla de montar o bifurcación de doble periodo en el diagrama, sino más bien se observa que la órbita tiende a un punto fijo y luego a otro, porque intercambian su estabilidad, y el gráfico está limitado a girar sobre los puntos fijos a medida que cambia el parámetro.

Existe otro tipo de bifurcación llamada **bifurcación de diapasón**, la cual se presenta en sistemas físicos en los cuales existe una simetría [1]. En este caso, los puntos fijos aparecen y desaparecen en pares simétricos. La bifurcación de diapasón se obtiene con el siguiente mapeo

$$x_{n+1} = rx_n - x_n^3 \quad (37)$$

Nótese que si se hace la sustitución de x_n por $-x_n$ y se multiplica por un signo menos los dos lados de la igualdad, se obtiene la misma ecuación. El diagrama resultante en este tipo de bifurcación es parecido a una horquilla, por eso el nombre de bifurcación de diapasón.

También existen **bifurcaciones imperfectas** y catástrofes. El siguiente mapeo genera una bifurcación imperfecta

$$x_{n+1} = h + rx_n - x_n^3 \quad (38)$$

Si $h = 0$, se obtiene una bifurcación de diapasón, pero cuando h es diferente de cero, se presentan imperfecciones. A este tipo de valores que generan imperfecciones en el diagrama de bifurcación se les conoce como **parámetros de imperfección**. En la figura 15 se muestra una bifurcación imperfecta tomando a h como fija y con un valor de 0.1.

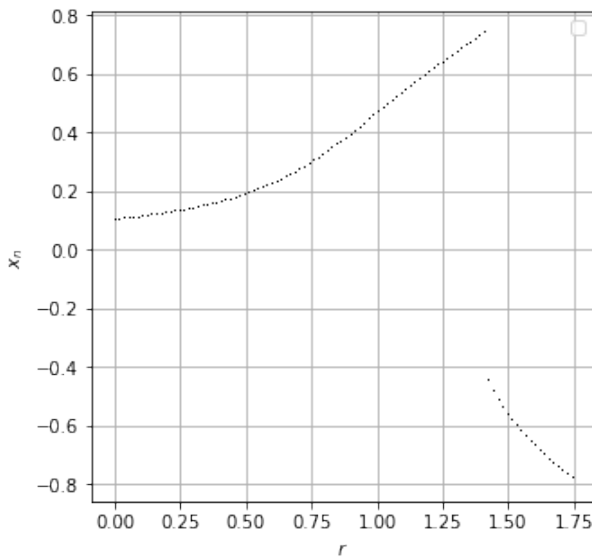


Figura 15. Diagrama de bifurcación imperfecta con $h = 0.01$. El código realizado en Jupyter para este gráfico se muestra en el apéndice A.14.

A partir de esta última figura, puede observarse cómo se separa el diagrama de bifurcación y se presentan dos partes desconectadas, generando una bifurcación imperfecta.

VII. PARTÍCULA AMORTIGUADA EN UN DOBLE POZO DE POTENCIAL

Un doble pozo de potencial es una región en donde se encuentran dos mínimos de la energía potencial, tal y como se muestra en la 16. En nuestro caso, consideremos una partícula de masa m en este tipo de regiones.

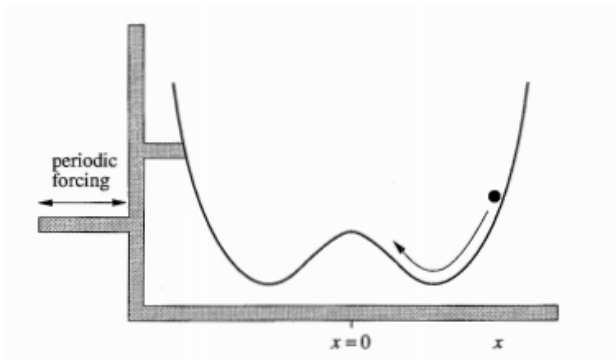


Figura 16. Partícula en un doble pozo de potencial [1].

Supongamos que existe una fuerza retardante proporcional a la velocidad de la partícula con coeficiente retardante k y una fuerza sinusoidal que sacude el sistema, es decir, una fuerza propulsora.

La región en forma de dos pozos tiene la misma forma de la energía potencial que depende de la posición, por

lo tanto la energía potencial de esta partícula está dada por

$$V(x) = \frac{1}{4}x^4 - \frac{1}{2}x^2 \quad (39)$$

Haciendo un análisis de fuerzas obtenemos la siguiente ecuación

$$m\ddot{x} = -km\dot{x} - m(x^3 - x) + A\cos\omega t \quad (40)$$

donde $km\dot{x}$ es la fuerza retardante, $m(x^3 - x)$ es la fuerza gravitacional que se obtiene al integrar el potencial y $A\cos\omega t$ es la fuerza sinusoidal propulsora. Si dividimos la ecuación entre m obtenemos

$$\ddot{x} + k\dot{x} - x + x^3 = F\cos\omega t \quad (41)$$

donde $F = A/m$.

Como puede observarse, la ecuación diferencial obtenida no es lineal y, por lo tanto, es complicado encontrar una solución analítica. No obstante, puede resolverse de forma numérica utilizando un código de programación. En la figura 17 se muestra la solución para la posición y la velocidad de la partícula que dependen del tiempo, tomando a $k = 0.25$, $\omega = 1$ y $F = 0.18$, con condiciones iniciales $x = v = 0$ cuando $t = 0$.

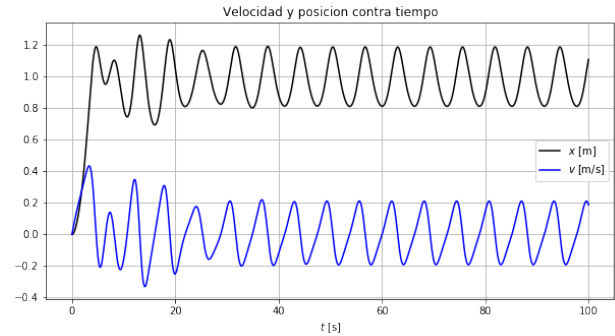


Figura 17. Posición y velocidad de una partícula en un doble pozo de potencial. El código realizado en Jupyter para este gráfico se muestra en el apéndice A.15.

A partir de esta figura, podemos concluir que para los valores dados de k , F y ω , la partícula presenta perturbaciones en su movimiento, pero después de un tiempo llega a presentar movimiento oscilatorio constante.

El movimiento caótico inicia cuando la amplitud de la fuerza propulsora es mayor. En las figuras 18 y 19 se muestra la posición y la velocidad de la partícula y su diagrama fase, respectivamente. En este caso, se tomó a $k = 0.25$, $\omega = 1$ y $F = 0.40$, con condiciones iniciales $x(0) = y(0) = 0$.

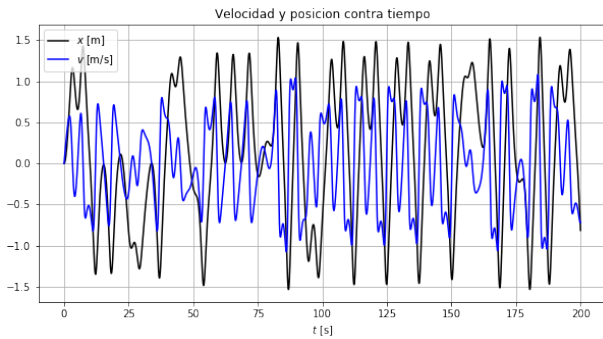


Figura 18. Posición y velocidad de una partícula en un doble pozo de potencial tomando a $F = 0,40$. El código realizado en Jupyter para este gráfico se muestra en el apéndice A.16.

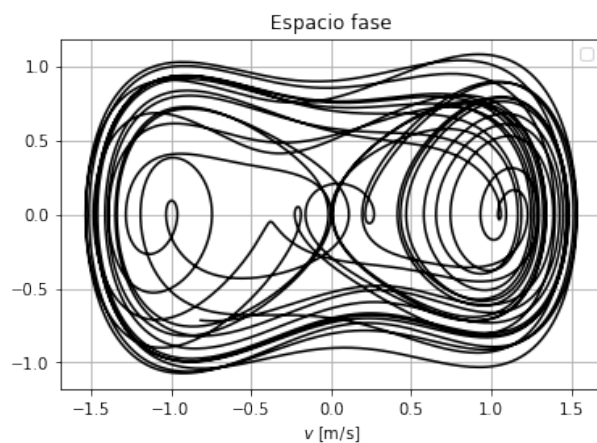


Figura 19. Espacio fase de una partícula en un doble pozo de potencial tomando a $F = 0,40$. El código realizado en Jupyter para este gráfico se muestra en el apéndice A.16.

Nótese la diferencia entre las soluciones con $F = 0,18$ y $F = 0,40$. En la 18 se muestra un movimiento caótico, la posición y la velocidad ya no son constantes ni siquiera en un periodo de tiempo. Luego, en la figura 19 puede observarse un espacio fase parecido a un atractor caótico, una figura distorsionada debido al movimiento del sistema.

VIII. CONCLUSIONES

Los sistemas mecánicos de la naturaleza no son nada comparables a aquellos sistemas en donde se reducen variables y se desprecian términos. Está justificado pensar en ángulos pequeños para oscilaciones de un péndulo o despreciar fuerzas de fricción en una caída libre, con el objetivo de entender la dinámica de un sistema. Sin embargo, hay ocasiones en las que en un sistema físico deben tomarse en cuenta muchos factores para comprender la dinámica en su totalidad. He aquí donde suele presentarse el caos en los sistemas dinámicos. A lo largo de estas secciones, se ha podido concluir que es

factible analizar el movimiento de un sistema caótico, que puede categorizarse e incluso, medirse. No importa cómo sean las ecuaciones de movimiento de un sistema físico, hay muchas formas de saber si estamos ante la presencia de un sistema caótico y así, poder llegar a un **orden**.

IX. APÉNDICES

IX-A. A.1

Código para la órbita del mapeo $x_{n+1} = 2x_n(1 - x_n)$, se genera la figura 1.

```
from matplotlib import pyplot as plt
import numpy as np
import math
from scipy.integrate import odeint
```

```
x_n = []
x_nmas1 = []
x_0 = 0.01
x_x = []
x_y = []
j = 0
for i in range(300):
    x_nmas1.append(2*x_0*(1-x_0))
    x_n.append(x_0)
    x_0 = x_nmas1[i]
```

```
for i in range(300):
    if i % 2 == 0:
        if i==299:
            break
        x_x.append(x_n[j])
        x_y.append(x_n[j+1])
    else:
        j = j+1
        x_x.append(x_n[j])
        x_y.append(x_n[j])
```

```
plt.figure(figsize=(5,5))
x = np.linspace(0,0.8,num=100)
plt.plot(x, 2*x*(1-x), 'r')
plt.plot(x, x, 'k')
plt.plot(x_x, x_y, 'b-')
plt.legend()
plt.xlabel('$x_n$')
plt.ylabel('$x_{n+1}$')
plt.grid()
```

IX-B. A.2

Código para la gráfica de x_n contra el número de iteraciones n , genera la figura 2.

```
from matplotlib import pyplot as plt
import numpy as np
import math
from scipy.integrate import odeint

x_n = []
x_nmas1 = []
x_0 = 0.01
j = []

def fun(r):
    x_0 = 0.01
    for i in range(50):
        x_nmas1.append(r*x_0*(1-x_0))
        x_n.append(x_0)
        x_0 = x_nmas1[i]
        j.append(i)

plt.figure(figsize=(5,5))
fun(0.4)
plt.plot(j,x_nmas1,'k',label="r =0.4")
x_n = []
x_nmas1 = []
j = []
fun(2.8)
plt.plot(j,x_nmas1,'r',label="r=2.8")
x_n = []
x_nmas1 = []
j = []
fun(3.3)
plt.plot(j,x_nmas1,'b',label="r=3.3")
plt.legend()
plt.title('$x_n$ vs $n$')
plt.xlabel('$n$')
plt.ylabel('$x_n$')
plt.grid()
```

IX-C. A.3

Código para órbita del mapeo logístico con $r = 4$, genera la figura 3.

```
from matplotlib import pyplot as plt
import numpy as np
import math
from scipy.integrate import odeint

x_n = []
```

```
x_nmas1 = []
x_0 = 0.01
x_x = []
x_y = []
j = 0
for i in range(300):
    x_nmas1.append(4.0*x_0*(1-x_0))
    x_n.append(x_0)
    x_0 = x_nmas1[i]

for i in range(300):
    if i % 2 == 0:
        if i==299:
            break
        x_x.append(x_n[j])
        x_y.append(x_n[j+1])
    else:
        j = j+1
        x_x.append(x_n[j])
        x_y.append(x_n[j])
```

```
plt.figure(figsize=(5,5))
x = np.linspace(0,0.8,num=100)
plt.plot(x, 4.0*x*(1-x), 'r')
plt.plot(x,x,'b',label='$x_{n+1}=x_n$')
plt.plot(x_x,x_y,'k-',label='orbita')
plt.legend()
plt.title('Mapeo logístico r = 4.0')
plt.xlabel('$x_n$')
plt.ylabel('$x_{n+1}$')
plt.grid()
```

IX-D. A.4

Código de la gráfica de la órbita del mapeo logístico con diferentes r , genera la figura 4.

```
from matplotlib import pyplot as plt
import numpy as np
import math
from scipy.integrate import odeint

x_n = []
r_n = []
r = np.linspace(3.4,3.9,num = 300)
for k in range(300):
    x = 0.01
    for i in range(300):
        x = r[k]*x*(1-x)
```

```

for i in range(50):
    x = r[k]*x*(1-x)
    r_n.append(r[k])
    x_n.append(x)

plt.figure(figsize=(5,5))
plt.plot( r_n, x_n, 'k', label = '')
plt.legend()
plt.xlabel('$r$')
plt.ylabel('$x_n$')
plt.grid()

```

IX-E. A.5

Código de los exponentes de Lyapunov para distintos valores de r , genera la figura 5.

```

from matplotlib import pyplot as plt
import numpy as np
import math
from scipy.integrate import odeint

lndf = 0.0
r_n = []
lya_exp = []
r = np.linspace(3.0,4.0,num = 300)
for k in range(300):
    x = 0.001
    lndf = 0.0
    for i in range(300):
        x = r[k]*x*(1-x)
    for i in range(1000):
        x = r[k]*x*(1-x)
        lndf=math.log(np.abs(r[k]
        -2*r[k]*x))+lndf
    r_n.append(r[k])
    lya_exp.append(lndf/1000)

```

```

plt.figure(figsize=(5,5))
plt.plot( r_n, lya_exp, 'k-')
plt.title('Exponentes de Lyapunov
para diferentes r')
plt.legend()
plt.xlabel('$r$')
plt.ylabel('$\lambda$')
plt.grid()

```

IX-F. A.6

Código para el conjunto de Mandelbrot, genera la figura 7.

```

from matplotlib import pyplot as plt
import numpy as np
import math
from scipy.integrate import odeint

```

```

z_imag = []
z_real = []
z_i = []
z_r = []
z_re = np.linspace(-2,2,2000)
z_im = np.linspace(-2,2,2000)

```

```

for m in range(len(z_re)):
    for n in range(len(z_im)):
        c = complex(z_re[m],z_im[n])
        z = complex(0,0)
        for t in range(500):
            z = z*z + c
            if ((z.imag*z.imag)
            +(z.real*z.real)) > 4.0:
                z_real.append
                (z_re[m])
                z_imag.append
                (z_im[n])
                break

```

```

plt.figure(figsize=(10,10))
plt.plot(z_real, z_imag, 'k,')
plt.title('Conjunto de Mandelbrot')
plt.legend()
plt.xlabel('Re')
plt.ylabel('Im')
plt.grid()

```

IX-G. A.7

Código para el diagrama fase en tres dimensiones de un péndulo amortiguado y forzado, se genera la figura 8.

```

from mpl_toolkits import mplot3d
from matplotlib import pyplot as plt
import numpy as np
import math
from scipy.integrate import odeint

```

```

#Function
def fun_total(x, t, ome_0, miu, A,
omed):
    (theta, omega) = x
    dxdt = (omega, - (miu*omega)

```



```

- pow(ome_0,2)*np.sin(theta)
+ A*np.cos(omed*t))
return dxdt

#Function for numeric solution
def sol_total(t):
    sol = odeint(fun_total, x0, t,
        args = (ome_0,miu, A, omed))
    return sol

#Initial conditions
ome_0 = 3.0*np.pi
miu = 3.0*np.pi/2.0
omed = 2*np.pi
x0 = (np.pi/2.0,0.0)
A = 1.05
ordenada = [-80,-60,-40]
# Generates x, y and z
t = np.linspace(-5, 5, 10000)
xx, yy = np.meshgrid(range(4),
    range(20))

#Plotting in 3D
fig = plt.figure()
ax = plt.axes(projection='3d')

#Phase diagram
xline = np.array(sol_total(t)[: ,0])
yline = np.array(sol_total(t)[: ,1])
zline = np.array(omed*t)

ax.scatter3D(xline, zline, yline,
    color='black', linewidths=0.1);

for i in range(len(ordenada)):
    point = np.array([0, 0,
        ordenada[i]])
    normal = np.array([0, 0, 1])
# Plane equation a*x+b*y+c*z+d=0
# [a,b,c] is the normal, this case k
# d es la distancia donde corta a z
    d = -point.dot(normal)
# se calcula z
    z = (-normal[0] * xx - normal[1]
        * yy - d) * 1. /normal[2]
#para dibujar varios planos
    ax.scatter(xline, zline, yline,
        color='black', linewidths=0.1);
    ax.plot_surface(xx, z,
        yy, alpha=0.1, linewidth=0);

```

```

ax.set_title('Espacio fase de tres
dimensiones ');
plt.xlabel('$\Theta$ ');
plt.ylabel('$\omega_{dt}$ ');
axes.set_zlabel('$\omega_{dt}$ ');
fig.set_size_inches(10.5, 10.5)
axes = plt.gca()
axes.set_xlim([-1.5,1.5])
axes.set_zlim([-10,10])
axes.set_ylim([-50,50])

```

IX-H. A.8

Código para las secciones de Poincaré obtenidas al colocar 50 palnos, genera la figura 9.

```

from mpl_toolkits import mplot3d
from matplotlib import pyplot as plt
import numpy as np
import math
from scipy.integrate import odeint

def fun_total(x, t, ome_0, miu, A,
    omed):
    (theta, omega) = x
    dxdt = (omega, - (miu*omega)
        - pow(ome_0,2)*np.sin(theta)
        + A*np.cos(omed*t))
    return dxdt

def sol_total(t):
    sol = odeint(fun_total, x0, t,
        args = (ome_0,miu, A, omed))
    return sol

ome_0 = 3.0*np.pi
miu = 3.0*np.pi/2.0
omed = 2*np.pi
x0 = (np.pi/2.0,0.0)
A = 1.05

plt.figure(figsize=(5,5))
t = np.linspace(0,3,50)
plt.plot(sol_total(t)[: ,0],
    sol_total(t)[: ,1], 'k.')
plt.legend()
plt.title('Secciones de Poincare ')
plt.ylabel('$\omega(t)$ [rad/s]')
plt.xlabel('$\Theta(t)$ [rad]')
plt.grid()

```

IX-I. A.9

Código para las secciones de Poincaré obtenidas al colocar 70 planos, genera la figura 10.

```
from mpl_toolkits import mplot3d
from matplotlib import pyplot as plt
import numpy as np
import math
from scipy.integrate import odeint
```

```
def fun_total(x, t, ome_0, miu, A, omed):
    (theta, omega) = x
    dxdt = (omega, - (miu*omega)
    - pow(ome_0,2)*np.sin(theta)
    + A*np.cos(omed*t))
    return dxdt
```

```
def sol_total(t):
    sol = odeint(fun_total, x0, t,
    args = (ome_0,miu, A, omed))
    return sol
```

```
ome_0 = 3.0*np.pi
miu = 3.0*np.pi/2.0
omed = 2*np.pi
x0 = (np.pi/2.0,0.0)
A = 1.05
```

```
plt.figure(figsize=(5,5))
t = np.linspace(0,3,70)
plt.plot(t, sol_total(t)[: ,1], 'k.')
plt.legend()
plt.title('Secciones de Poincare')
plt.ylabel('$\omega(t)$ [rad/s]')
plt.xlabel('$t$ [s]')
plt.grid()
```

IX-J. A.10

Código para el atractor de Lorenz, genera la figura 11.

```
from mpl_toolkits import mplot3d
from matplotlib import pyplot as plt
import numpy as np
import math
from scipy.integrate import odeint
```

```
def fun_total(f, t, sigma, b, r):
    (x, y, z) = f
    dfdt = (-sigma*x + sigma*y,
    r*x-y-x*z, -b*z + x*y)
```

```
return dfdt
```

```
def sol_total(t):
    sol = odeint(fun_total, f0, t,
    args = (sigma,b,r))
    return sol
```

```
f0 = (1.4,1.4,28)
r = 28
b = 8.0/3.0
sigma = 10.0
t = np.linspace(-20, 20, 30000)
fig = plt.figure()
ax = plt.axes(projection='3d')
xline = np.array(sol_total(t)[: ,0])
yline = np.array(sol_total(t)[: ,1])
zline = np.array(sol_total(t)[: ,2])
```

```
ax.scatter3D(xline, yline, zline,
color = 'black', linewidths=0.000,
marker='.');
ax.set_title('Atractor de Lorenz');
plt.xlabel('x');
plt.ylabel('y');
fig.set_size_inches(10.5, 10.5)
```

IX-K. A.11

Código para el atractor de Rössler, genera la figura 12.

```
from mpl_toolkits import mplot3d
from matplotlib import pyplot as plt
import numpy as np
import math
from scipy.integrate import odeint
```

```
def fun_total(f, t, a, b, c):
    (x, y, z) = f
    dfdt = (-y - z, x + a*y, b + z*(x-c))
    return dfdt
```

```
def sol_total(t):
    sol = odeint(fun_total, f0, t, args = (a,b,c))
    return sol
```

```
f0 = (1.4,14,28)
a = 0.1
b = 0.1
c = 14
t = np.linspace(-45, 45, 30000)
fig = plt.figure()
```

```

ax = plt.axes(projection='3d')
xline = np.array(sol_total(t)[: ,0])
yline = np.array(sol_total(t)[: ,1])
zline = np.array(sol_total(t)[: ,2])

ax.scatter3D(xline , yline , zline ,
color = 'black' , linewidths=0.000,
marker='.');
ax.set_title('Atractor de Rossler');
plt.xlabel('x');
plt.ylabel('y');
fig.set_size_inches(10.5, 10.5)

```

IX-L. A.12

Código que genera el diagrama de bifurcación para el mapeo $x_{n+1} = a - x_n^2$, genera la figura 13.

```

from matplotlib import pyplot as plt
import numpy as np
import math
from scipy.integrate import odeint

```

```

x_n = []
a_n = []
a = np.linspace(0,1.35,num = 50)
for k in range(len(a)):
    x = 0.00
    for i in range(300):
        x = a[k] - x**2
    for i in range(1000):
        x = a[k] - x**2
        a_n.append(a[k])
        x_n.append(x)

```

```

plt.figure(figsize=(5,5))
plt.plot( a_n, x_n, 'k.')
plt.legend()
plt.xlabel('$a$')
plt.ylabel('$x_n$')
plt.grid()

```

IX-M. A.13

Código para el diagrama de bifurcación del mapeo de Hénon, genera la figura 14.

```

from matplotlib import pyplot as plt
import numpy as np
import math
from scipy.integrate import odeint

```

```

x_n = []

```

```

y_n = []
a_n = []
b = 0.4
a = np.linspace(1.91,1.95,num = 100)
for k in range(len(a)):
    x = 0.01
    y = 0.01
    for i in range(2000):
        x = a[k] - x**2 + b*y
        y = x
    for i in range(500):
        x = a[k] - x**2 + b*y
        y = x
        a_n.append(a[k])
        x_n.append(x)

```

```

plt.figure(figsize=(5,5))
plt.plot( a_n, x_n, 'k,')
plt.legend()
plt.xlabel('$a$')
plt.ylabel('$x_n$')
plt.grid()

```

IX-N. A.14

Código para el diagrama de bifurcación imperfecta, genera la figura 15.

```

from matplotlib import pyplot as plt
import numpy as np
import math
from scipy.integrate import odeint

```

```

h = 0.1
r_n = []
x_n = []
r = np.linspace(0,3.5,num = 200)
for k in range(100):
    x = -0.9
    for i in range(500):
        x = h + r[k]*x - (x**3)
    for i in range(500):
        x = h + r[k]*x - (x**3)
        r_n.append(r[k])
        x_n.append(x)

```

```

plt.figure(figsize=(5,5))
plt.plot( r_n, x_n, 'k,')
plt.legend()
plt.xlabel('$r$')
plt.ylabel('$x_n$')
plt.grid()

```

IX-Ñ. A.15

Código para la posición y velocidad de una partícula en un doble pozo de potencial, genera la figura 17.

```
from matplotlib import pyplot as plt
import numpy as np
import math
from scipy.integrate import odeint

def fun_total(f, t, k, F, ome):
    (x, v) = f
    dfdt = (v, -k*v + x - x**3
    + F*np.cos(ome*t))
    return dfdt

def sol_total(t):
    sol = odeint(fun_total, f0, t,
    args = (k, F, ome))
    return sol

ome = 1.0
k = 0.25
F = 0.18
f0 = (0.0,0.0)

plt.figure(figsize=(10,5))
t = np.linspace(0,100,7000)
plt.plot(t, sol_total(t)[: ,0],
'k-',label='$x$ [m]')
plt.plot(t, sol_total(t)[: ,1],
'b-',label='$v$ [m/s]')
plt.legend()
plt.title('Velocidad y posicion
contra tiempo')
plt.xlabel('$t$ [s]')
plt.grid()
plt.show()
```

IX-O. A.16

Código para la posición, la velocidad y el espacio fase de una partícula en un doble pozo de potencial, genera las figuras 18 y 19.

```
from matplotlib import pyplot as plt
import numpy as np
import math
from scipy.integrate import odeint

def fun_total(f, t, k, F, ome):
    (x, v) = f
    dfdt = (v, -k*v + x - x**3
```

```
+ F*np.cos(ome*t))
return dfdt

def sol_total(t):
    sol = odeint(fun_total, f0, t,
    args = (k, F, ome))
    return sol

ome = 1.0
k = 0.25
F = 0.40
f0 = (0.0,0.0)

plt.figure(figsize=(10,5))
t = np.linspace(0,200,7000)
plt.plot(t, sol_total(t)[: ,0],
'k-',label='$x$ [m]')
plt.plot(t, sol_total(t)[: ,1],
'b-',label='$v$ [m/s]')
plt.legend()
plt.title('Velocidad y posicion
contra tiempo')
plt.xlabel('$t$ [s]')
plt.grid()
plt.show()

plt.plot(sol_total(t)[: ,0],
sol_total(t)[: ,1], 'k-')
plt.legend()
plt.title('Diagrama fase')
plt.xlabel('$x$ [m]')
plt.ylabel('$v$ [m/s]')
plt.grid()
plt.show()
```

REFERENCIAS

- [1] Strogatz, S. H. (1994). Nonlinear Dynamics and Chaos. United States of America: Perseus Books. Recuperado el 28 de Octubre de 2018
- [2] Hilborn, R. C. (2011). Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers (Segunda ed.). Oxford University Press, 2000.
- [3] Kathleen T. Alligood, T. D. (1996). CHAOS: An introduction to dynamical systems. New York: Springer.
- [4] Stephen T. Thornton, J. B. (s.f.). Classical Dynamics of Particles and Systems (5 ed.). (2. Brooks/Cole, Ed.)
- [5] Ricard V. Solé, Susanna C. Manrubia. Orden y caos en sistemas complejos. Edicions UPC
- [6] Robinson, R. C. (2004). An introduction to Dynamical systems: Continuous and Discrete. New Jersey: Pearson Prentice Hall.
- [7] P.H. Diamond, Notes by: Kurt Thompson. Phys 221A Lecture Notes - Lyapunov Exponents and their Relation to Entropy. April 28, 2017.