

Proyecto Mecánica Analítica

Caos

MARÍA JOSÉ FONSECA VÁZQUEZ
Universidad de Guanajuato
11 de noviembre de 2018

Resumen

Se definió lo que es el caos, la forma en que fue identificado por primera vez, características de los sistemas caóticos, ejemplos de cada uno de ellos, atractores extraños, entre otros. También se complementó cada uno de los conceptos mencionados mediante el gráfico que lo ejemplifica, utilizando el lenguaje de programación Python.

Keywords: Caos, mapeos, atractor, exponentes de Lyapunov, Secciones de Poincaré, Fractales, Clima, Efecto mariposa

El caos consiste en un comportamiento aleatorio que ocurre en un sistema determinista (que depende de situaciones ocurridas en el pasado). Los sistemas caóticos comparten características comunes, siendo la más notoria que con condiciones iniciales que apenas difieren, se obtiene un resultado completamente distinto al anterior. El meteorólogo Lorentz se dio cuenta de ello mientras realizaba un análisis sobre las condiciones climáticas en un modelo a computadora. A partir de ahí, comenzó a haber un gran interés en la comunidad científica sobre otros sistemas con esas mismas características caóticas, y los resultados fueron sorprendentes: sistemas que en absoluto parecerían caóticos, presentan dichas propiedades. Por ejemplo, un péndulo puede llegar a presentar movimiento caótico dependiendo del sistema experimental en el que se encuentre; lo mismo pasa para los fluidos, que son tan complicados de modelar, incluso para describir el flujo de una llave que deja circular agua, pues no se puede expresar con exactitud el movimiento de cada una de sus partículas. Otro ejemplo, probablemente más conocido de sistema caótico, es el clima. Incluso sin tener conocimiento sobre el tema, es fácil imaginar que el clima se encuentra ampliamente ligado con la teoría del caos, pues a pesar del gran avance científico que se ha logrado

en los últimos años, las predicciones que se pueden encontrar en el día a día en los medios de comunicación para los cinco días próximos pueden ser totalmente distintas a lo que realmente ocurrirá; esto se debe a un fenómeno conocido como el *efecto mariposa*. Además de estos conceptos, se explicarán otros conceptos físicos que forman parte de la teoría del caos.

1. Mapeos

Un sistema dinámico consiste en una serie de posibles estados, junto con una regla que determina el estado presente en términos de estados pasados. Se requiere que la regla sea determinística, es decir, que podamos determinar el estado actual únicamente a partir de estados pasados.

Se tienen dos tipos de sistemas dinámicos: discretos y continuos en el tiempo. Un sistema discreto en el tiempo toma el estado actual como entrada y actualiza la situación produciendo un nuevo estado como salida. Un sistema continuo en el tiempo surge del límite de un sistema discreto, con tiempos cada vez más pequeños. La regla que gobierna al sistema en dicho caso es, por lo general, una ecuación diferencial, o una serie de ellas.

Una función cuyo espacio en dominio (entrada) y espacio en rango (salida) son los mismos, es llamado mapeo. Si x es un punto y f un mapeo, la órbita de x sobre f es la serie de puntos $\{x, f(x), f^2(x), \dots\}$, y un punto fijo (o estable) es aquel p , tal que $f(p) = p$.

En un mapeo, a la gráfica de la órbita se le llama *gráfica de Cobweb*. [1] Por ejemplo, en la Figura 1 se puede observar la órbita de la función $f(x) = 2.3x(1 - x)$.

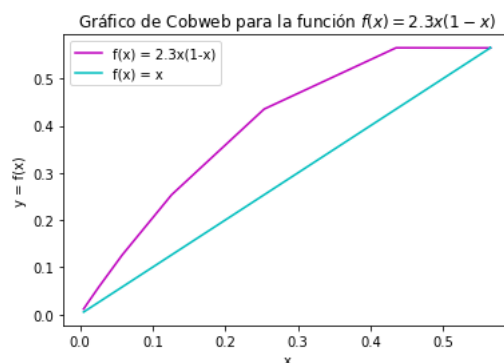


Figura 1: Ejemplificación de la gráfica de Cobweb.

Como puede observarse, se graficó también la función $y = x$. Los dos puntos en los que se cruzan ambas gráficas corresponden a los puntos estables de la órbita.

Un mapeo unidimensional iterado está basado en una función de variable real que toma la forma

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

Por ejemplo, la Figura 1, mostrada anteriormente, corresponde a un mapeo unidimensional, más específicamente, es un caso particular del llamado *mapeo logístico*.

Los mapeos logísticos tienen la forma $x_{n+1} = rx_n(1 - x_n)$. A continuación, se realizará un análisis detallado para este mapeo, tomando $x \in [0, 1]$ y $0 \leq r \leq 4$.

Ya que se observó en la primer imagen un ejemplo de la órbita de dicho mapeo, se procederá de una manera diferente. La Figura 2 muestra qué sucede con el valor de $f(x)$ conforme va ocurriendo un mayor número de iteraciones.

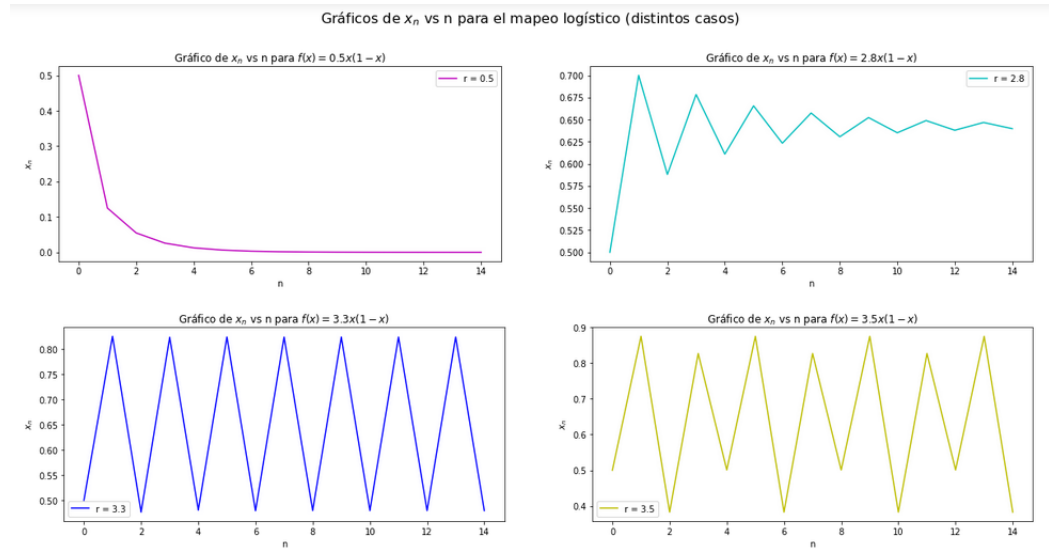


Figura 2: Gráficos de x_n vs n para distintos valores de r y la misma condición inicial

Como se puede observar, en el primer caso ($r = 0.5$), los valores son acotados

poco a poco. Para el siguiente ($r = 2.8$), nuevamente se acercan a un punto, sin embargo, parecen estar alternados, es decir, para un valor de n se obtiene un x_n mayor al que se acota, y en la siguiente iteración se obtiene uno menor, pero después de cierto número de iteraciones, se obtiene el mismo valor en todos los casos, como si se tuviera un *periodo uno* en las iteraciones. En el siguiente caso ($r = 3.3$), se obtiene un gráfico con *periodo dos* en las iteraciones, es decir, $x_n = x_{n+2}$. Finalmente, para el último caso ($r = 3.5$), se obtiene un *periodo cuatro* en las iteraciones, pues $x_n = x_{n+4}$.

¿Existe alguna relación entre todo esto? La respuesta es sí. La *periodicidad de iteraciones* de éstos y otros valores de r tienen la forma 2^n , y se ejemplificaron las primeras de ellas. Como los primeros dos valores de r anteriormente mostrados (0.5 y 2.8) tienen *periodicidad uno*, se toma mayor interés a partir del valor en que la periodicidad 2 aparece. De acuerdo a cálculos realizados hace algunos años, se han encontrado los valores en los que un nuevo periodo aparece, siendo los primeros de ellos los mostrados a continuación:

- $r_1 = 3$ (periodo 2)
- $r_2 = 3.449...$ (periodo 4)
- $r_3 = 3.54409...$ (periodo 8)
- $r_4 = 3.5644...$ (periodo 16)
- $r_5 = 3.568759...$ (periodo 32)

Al observar los valores anteriores, es fácil observar que existe una convergencia entre estos valores. Dicho valor (que corresponde a la potencia 2^∞), es $r_\infty = 3.569946....$ [2] Ahora, lo más natural es preguntarse qué sucede justo después de dicho valor. La respuesta suena simple, pero es mucho más complicada de lo que parece: **caos**. A continuación se muestra un diagrama de bifurcación para el mapeo logístico, que se puede observar en las Figuras 3, 4 y 5.

En la Figura 3, se muestra la vista completa de dicho diagrama. La interpretación de dicha gráfica tiene que ver con los valores r_i mencionados con anterioridad. Si se observa con detenimiento, para los primeros valores de r no hay bifurcación, sino que dicha *separación* comienza a partir del valor 3, que es lo que se esperaba, pues el periodo dos en las iteraciones del mapeo comienza en $r = 3$. La siguiente bifurcación, o periodo de iteración de 4, es en aproximadamente $r = 3.4$, el siguiente en aproximadamente $r = 3.54$, etcétera, que corresponden con los valores predichos. Además de ello, existen secciones *blancas* donde aparentemente no hay puntos.

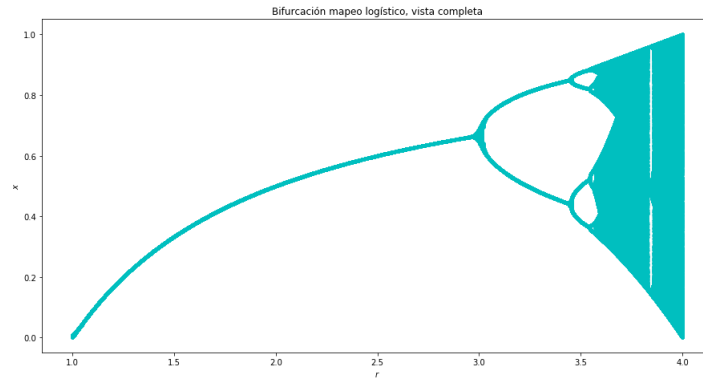


Figura 3: Diagrama de bifurcación mapeo logístico.

Ahora, como se observa en la Figura 4, si se hace un zoom en la primera parte en que la bifurcación ocurre, entonces es posible observar que en las secciones *blancas* sí existen puntos. Este comportamiento no era de esperar, pues al inicio todo parecía estable.

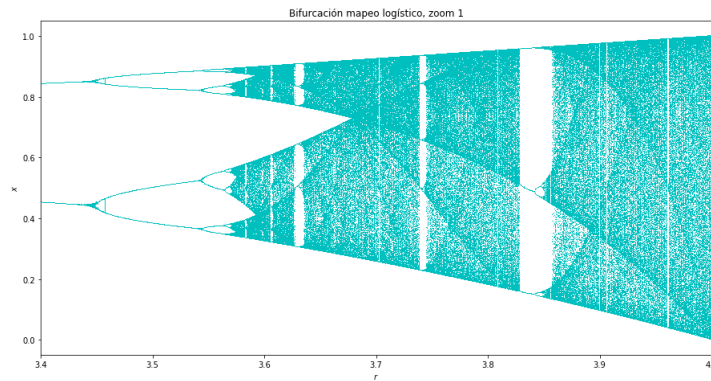


Figura 4: Diagrama de bifurcación mapeo logístico, secciones 'blancas'

Por último, la Figura 5 permite observar lo que convierte a este mapeo en *caótico*. Conforme se hace más y más zoom, es posible observar que dentro de esas secciones blancas lo que se encuentra es nuevamente el gráfico del mapeo logístico mostrado en la Figura 3, y si se sigue haciendo zoom y se aumentaran las iteraciones graficadas, así como los intervalos para la variable r , entonces se observaría infinitamente dicho patrón. Todo esto surge después del considerado r_{∞} , por lo que a dicho r se le puede denotar como *límite de estabilidad*.

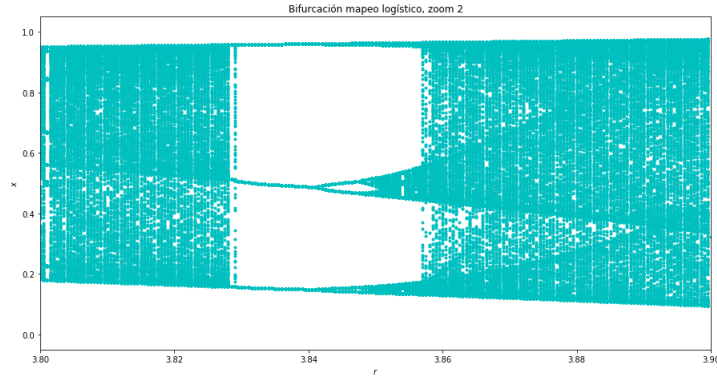


Figura 5: Zoom de las secciones 'blancas' del diagrama de bifurcación.

2. Exponentes de Lyapunov

Una órbita caótica es aquella que todo el tiempo continúa teniendo un comportamiento caótico, sin importar la cantidad de iteraciones que pasen, sin periodicidad ni puntos fijos. El sistema caótico debe, además, mostrar una dependencia muy sensible ante condiciones iniciales, en el sentido de que órbitas vecinas se separan rápidamente a pesar de tener condiciones iniciales muy similares. [2]

En cualquier lugar de la órbita, existen puntos arbitrariamente cercanos que se alejarán durante las siguientes iteraciones. Esta irregularidad está cuantificada por los *números de Lyapunov* y por los *exponentes de Lyapunov*.

Definimos un número de Lyapunov como la tasa promedio de divergencia o convergencia por iteración de los puntos cercanos a lo largo de la órbita, y a un exponente de Lyapunov como el logaritmo natural del número de Lyapunov.

El caos está definido como un exponente de Lyapunov mayor a cero. Por ello, *cualquier sistema que contenga al menos un exponente de Lyapunov negativo es definido como un sistema caótico*[3].

Sea f un mapeo suave perteneciente al campo de los reales. El número de Lyapunov está definido como

$$L(x_1) = \lim_{n \rightarrow \infty} (|f'(x_1)| |f'(x_2)| \dots |f'(x_n)|)^{1/n} \quad (2)$$

Y como el exponente de Lyapunov (λ) corresponde a su logaritmo, se tiene

$$\lambda(x_1) = \lim_{n \rightarrow \infty} \frac{[|f'(x_1)| |f'(x_2)| \dots |f'(x_n)|]}{n} \quad (3)$$

donde x_1 corresponde a la condición inicial.

Para ejemplificar dicho concepto, se realizó el cálculo y la gráfica correspondiente a los exponentes de Lyapunov para el mapeo logístico, pues es una función cuya naturaleza inicial estable y caótica después de cierto valor de r se conocen. El resultado se puede apreciar en la Figura 6.

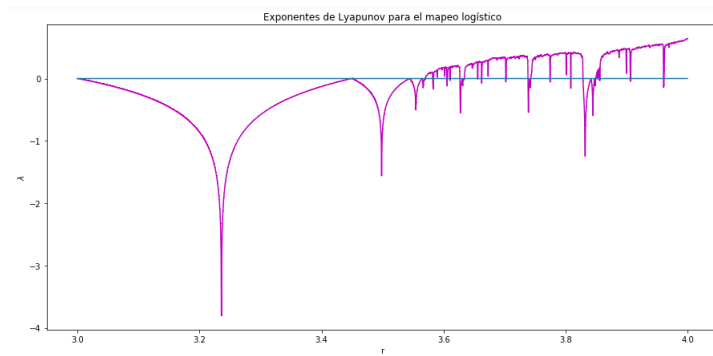


Figura 6: Exponentes de Lyapunov para el mapeo logístico.

Como se puede observar, y comparando con las Figuras correspondientes al diagrama de bifurcación del mapeo logístico, dicho mapeo se mantiene estable al inicio, pues los exponentes de Lyapunov se observan por debajo del eje x . Hay momentos en los que se toca al eje x , esto debido a que son los puntos que en la sección de mapeos se identificaron como r_n y permitían obtener un periodo de 2^n en el mapeo logístico. Por ejemplo, en $r = 3$ y $r = 3.45$, $\lambda = 0$, ya que son los puntos para los cuales se obtiene periodicidad 2 y 4.

Algo que también debe observarse es que efectivamente, conforme r va en aumento, se interseca al eje x más rápidamente, pues los valores de r para la periodicidad convergen al valor $3.568759\dots$. Después de ese valor, se observa que $\lambda \geq 0$, y era de esperar, pues según el diagrama de bifurcación permite observar que es ahí donde se comienza a tener una órbita caótica. Existen puntos para $r \geq 3.568759\dots$ en que $\lambda \leq 0$, y esto corresponde a las secciones blancas del diagrama de bifurcación, pues si se recuerda lo mencionado con anterioridad, en esos lugares se obtienen órbitas aparentemente estables, pues se obtiene el diagrama de bifurcación una y otra vez, aunque justo al tenerse el valor menor a cero, éste comienza a variar rápidamente, pues es realmente otra manifestación del caos para este mapeo.

3. Atractores Extraños

Un atractor se refiere a un subconjunto μ_0 de un espacio donde el *flujo* es globalmente contraído, mapeando sobre el mismo μ_0 después de diversas iteraciones. Puede ser un punto fijo, una órbita periódica, aperiódica o una combinación de ellas. El caso más interesante es el atractor recurrente aperiódico, al que se le conoce como *atractor extraño*.

El concepto de atractor extraño fue introducido por Ruelle y Takens en su escrito *On the Nature of Turbulence*. La palabra extraño hace referencia a que dicho tractor no es un punto fijo, un ciclo límite o uno de los conceptos obtenidos hasta esa fecha.

3.1. Atractor de Hénon

El mapa de Hénon es una manera sencilla para modelar flujos que se encuentran cerca de una bifurcación o puntos fijos, y es un prototipo de la dinámica que conduce al caos determinista. Debido a su simplicidad, el mapa de Hénon proporciona una herramienta muy útil para llevar a cabo una exploración detallada de la dinámica caótica.

Para observar el atractor correspondiente a este mapeo, se tienen las siguientes expresiones:

$$x_{n+1} = 1 - ax_n^2 + by_n \quad (4)$$

$$y_{n+1} = x_n \quad (5)$$

Este es un mapeo bidimensional, que puede ser escrito también con la siguiente relación:

$$x_{n+1} = 1 - ax_n^2 + bx_n - 1 \quad (6)$$

La Figura 7 permite observar dicho atractor.

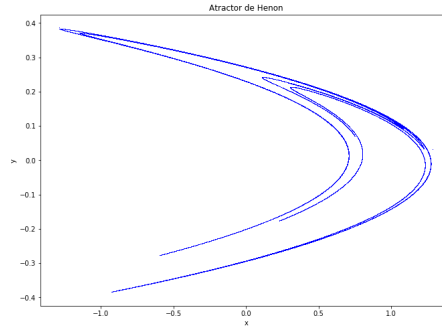


Figura 7: Ejemplificación del atractor de Hénon.

Para este mapeo, se tienen dos puntos fijos: [6]

$$x_0 = \frac{-(1-b) \pm \sqrt{(1-b)^2 + 4a}}{2a} \quad (7)$$



Figura 8: Gráficos del eje x y y respecto al tiempo para el atractor de Hénon.

3.2. Atractor de Lorentz

Lorentz trabajó en modelos diversos, llegando a reducir las ecuaciones de Navier-Stokes con un par de 3 ecuaciones diferenciales ordinarias y no lineales que representan el cambio temporal en tres modos esenciales, uno de velocidad y dos de distribuciones de temperatura.

Las ecuaciones son:

$$\dot{x} = -\sigma x + \sigma y \quad (8)$$

$$\dot{y} = rx - y - xz \quad (9)$$

$$\dot{z} = -bz + xy \quad (10)$$

Se tienen tres parámetros: el *número de Prandtl* (σ), el *número de Rayleigh* r proporcional a la diferencia de temperatura entre el fondo del fluido y y la capa superior ΔT y la *medida geométrica* b , derivada de la geometría del sistema.

Los cálculos que Lorentz obtuvo los realizó con $r = 28$, $\sigma = 10$ y $b = 8/3$. En la Figura 9 se muestran los gráficos para las soluciones de $x(t)$, $y(t)$ y $z(t)$ respecto del tiempo para un intervalo de tiempo determinado. Al ser éste un atractor se observa que por ejemplo para el gráfico x vs t , el rango de la función toca dos puntos localizados en aproximadamente $\pm 8,48$ y son los valores hacia los cuales dicho eje, así como también el eje y son *atraídos* con mayor intensidad. En el eje z , el punto al que se siente atraído es el 27. Todo esto es más fácil de observar si se presta atención a las oscilaciones que presentan una amplitud menor, ya que dicha amplitud oscila alrededor de los puntos mencionados.

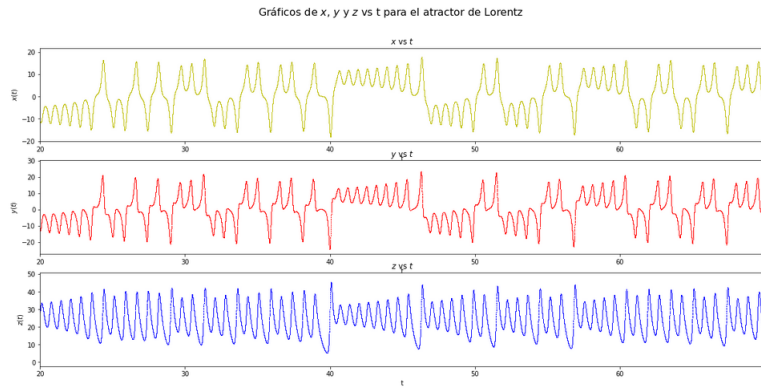


Figura 9: Ejes x , y y z en función del tiempo para el atractor de Lorentz.

En las Figuras 10, 11 y 12 se tienen proyecciones del atractor de Lorentz sobre dos ejes coordenados, y se observa que, para los tres casos, existen dos *órbitas* sobre las cuales se ve atraído al movimiento. Ese par de puntos C_1 y C_2 son puntos fijos inestables con coordenadas

$$(\pm \sqrt{b(r-1)}, \pm \sqrt{b(r-1)}, r-1)$$

y, para el caso de las condiciones iniciales utilizadas por Lorentz, corresponden a los valores mencionados anteriormete ($\pm 8,48$, $\pm 8,48$ y 27).

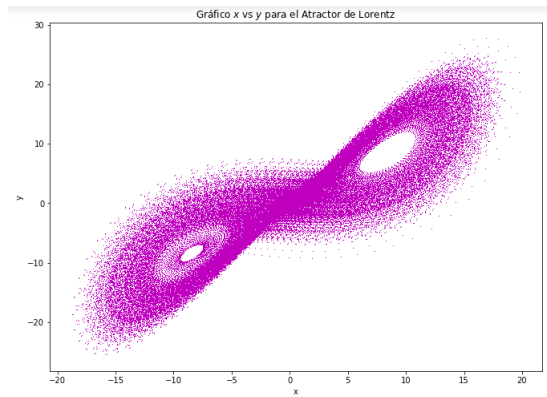


Figura 10: Eje x como función del eje y para el atractor de Lorentz.

C_1 y C_2 contienen un eigenvalor negativo y un par de eigenvalores complejos conjugados con una parte real positiva. Dichos eigenvalores determinan los cursos inestables de las trayectorias para los tres ejes coordenados. [4]

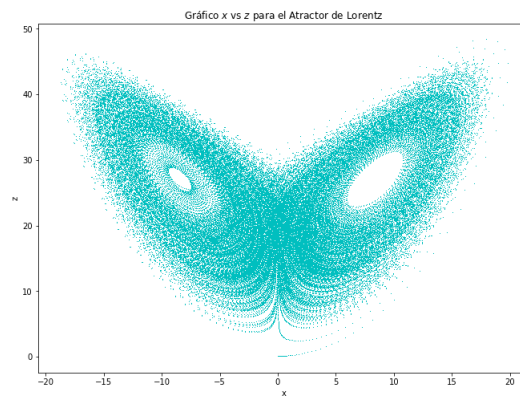


Figura 11: Eje x como función del eje Z para el atractor de Lorentz.

El eje x mide la velocidad de flujo, mientras que los ejes y y z denotan ciertas sobre la distribución de las temperaturas. [5]

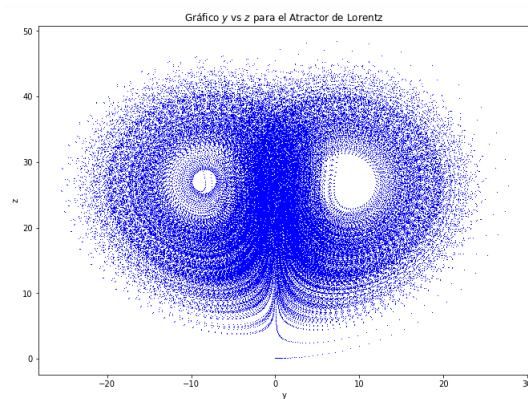


Figura 12: Eje y como función del eje z para el atractor de Lorentz.

3.3. Atractor de Peter de Jong

Se mostrará a continuación otro ejemplo de atractor extraño, el de Peter de Jong, en el que es muy visible el cambio del gráfico de acuerdo a las condiciones iniciales.

Se parte de las siguientes expresiones:

$$x_{n+1} = \sin(ay_n) - \cos(bx_n) \quad (11)$$

$$y_{n+1} = \sin(cx_n) - \cos(dy_n) [7] \quad (12)$$

El objetivo de esto es observar las diferencias de las Figuras 13 y 14, pues hubo una variación notable de condiciones iniciales y no se encuentra, a simple vista, una justificación para dicho cambio de valores.

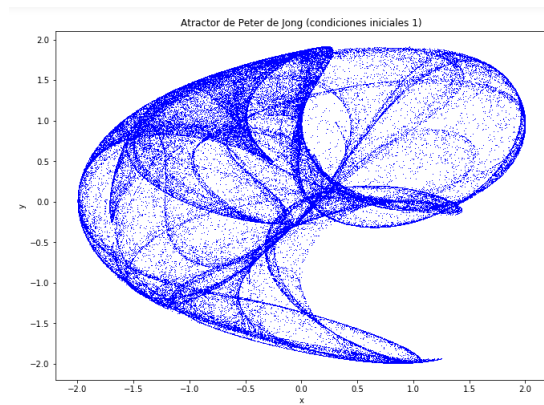


Figura 13: Atractor de Peter de Jong para primeras condiciones iniciales.

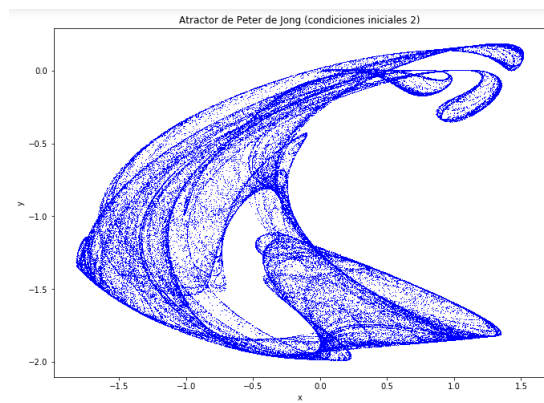


Figura 14: Atractor de Peter de Jong para segundas condiciones iniciales.

Ahora, para las variaciones de los ejes coordenados de acuerdo al número de iteraciones realizadas también se observa un cambio:

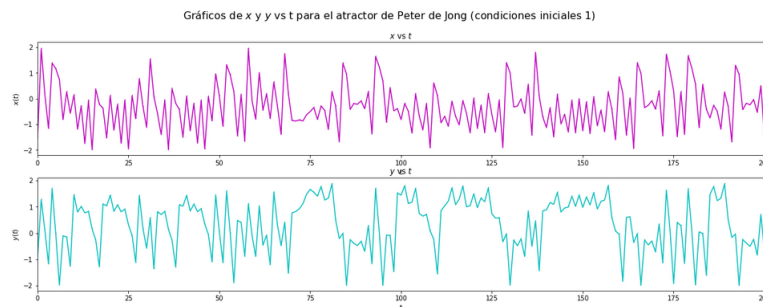


Figura 15: Atractor de Peter de Jong para primeras condiciones iniciales.

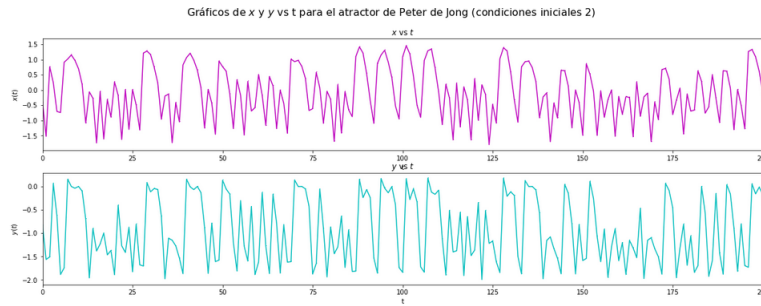


Figura 16: Atractor de Peter de Jong para segundas condiciones iniciales.

3.4. Atractor de Rossler

Otro ejemplo de atractor extraño es el llamado *atractor de Rossler*, cuyas proyecciones en los ejes coordenados se muestran en las Figuras 17, 18, 19 , y cuyas ecuaciones son:

$$\dot{x} = -(y + z) \quad (13)$$

$$\dot{y} = x + Ay \quad (14)$$

$$\dot{z} = B + xz - Cz \quad (15)$$

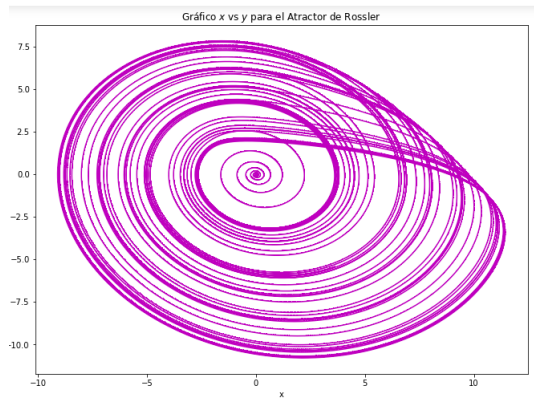


Figura 17: Eje x como función del eje y para el atractor de Rossler.

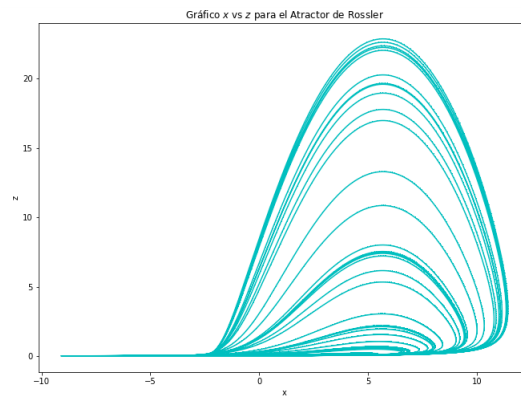


Figura 18: Eje x como función del eje z para el atractor de Rossler.

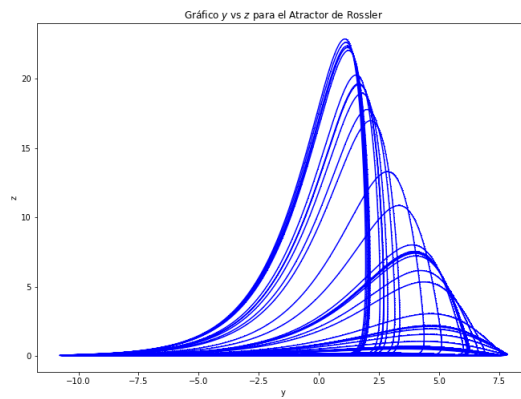


Figura 19: Eje y como función del eje z para el atractor de Rossler.

Por otro lado, la Figura 20 permite observar la variación de los tres ejes a lo largo del tiempo.

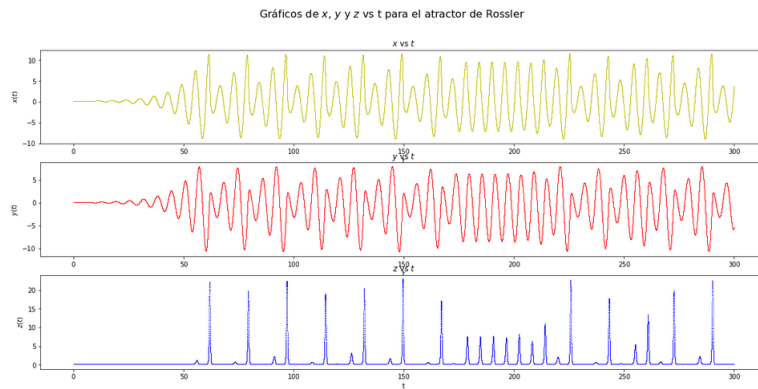


Figura 20: Variación de los tres ejes coordenados como función del tiempo para el atractor de Rossler.

4. Secciones de Poincaré

Las secciones de Poincaré simplifican la descripción geométrica de la dinámica de un sistema removiendo una de las dimensiones del mismo.

El punto clave es que esa simplicidad contiene la información esencial sobre periodicidad, bifurcaciones y caos.

Para un espacio tridimensional, la sección de Poincaré corresponde a un plano bidimensional elegido de manera que las trayectorias intersecten el plano transversalmente.

Si se tiene un espacio de N dimensiones, las secciones de Poincaré serán de $N - 1$ dimensiones. [8]

A continuación, se muestran las secciones de Poincaré para dos de los atractores ejemplificados en la sección anterior: el atractor de Lorentz y el atractor de Rossler.

La Figura 21 muestra la primer sección de Poincaré. El procedimiento seguido fue simple: se tenía el atractor de Lorentz con los valores para los ejes coordenados, y se tuvo que elegir un eje para igualar a cero, que en este caso fue el x . Entonces, del conjunto de datos de x que pertenecían al atractor de Lorentz, se verificó cuáles de ellos tenían el valor $x = 0$ y, cada vez que se encontraba uno, se agregaban valores a dos arreglos nuevos que corresponderían a los ejes y y z , graficando finalmente esos pares de datos.

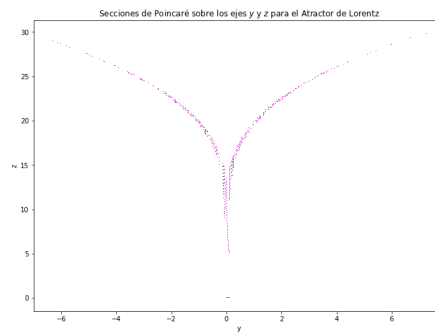


Figura 21: Sección de Poincaré para el atractor de Lorentz en $x = 0$.

Ahora, igualando el eje y a cero para observar los valores de los ejes x y z en dicho caso, se obtiene la Figura 22.

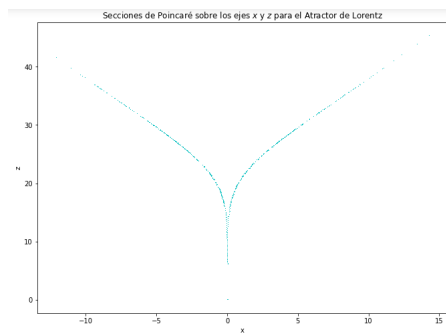


Figura 22: Sección de Poincaré para el atractor de Lorentz en $y = 0$.

Finalmente, si $z = 0$, se obtiene la Figura 23:

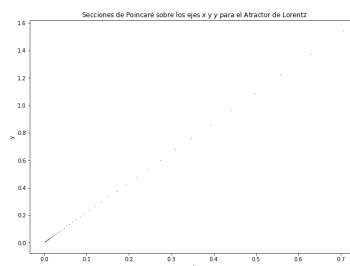


Figura 23: Sección de Poincaré para el atractor de Lorentz en $z = 0$.

Ahora, realizado los mismos tres cálculos para el atractor de Rossler, se obtienen los gráficos correspondientes a la Figuras 24, 25 y 26.

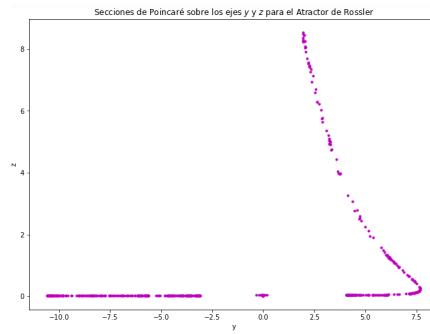


Figura 24: Sección de Poincaré para el atractor de Rossler en $x = 0$.

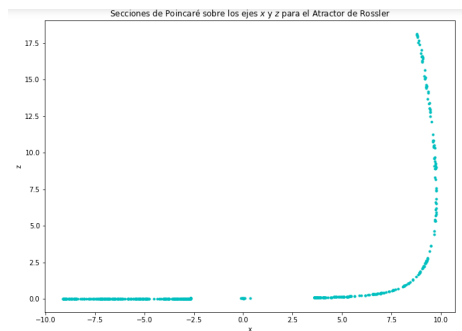


Figura 25: Sección de Poincaré para el atractor de Rossler en $y = 0$.

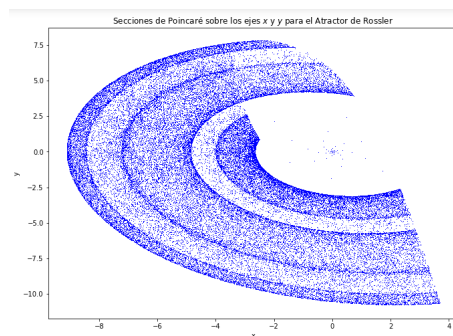


Figura 26: Sección de Poincaré para el atractor de Rossler en $z = 0$.

En base a los resultados obtenidos puede obtenerse información al respecto, como por ejemplo, de acuerdo a la Figura 26, el eje que el atractor de Rossler cruza con mayor frecuencia es evidentemente el z , ya que se observa un trazo más completo.

5. Fractales

Los fractales son patrones complejos que muestran los mismos detalles o las mismas características a diferentes escalas; es decir, no se simplifican una vez magnificados. Esto significa que se puede alejar o hacer zoom una y otra vez y se encuentra el mismo patrón una y otra vez.

La geometría de fractales es usada para patrones producidos por caos. Como ejemplo para su diferenciación, las trayectorias para el problema de dos cuerpos, por ejemplo, contienen elipses, parábolas e hipérbolas, mientras que los atractores caóticos por lo general contienen atracciones repetidas después de cierta distancia o intervalo de tiempo, y estas atracciones son ejemplos de fractales.

Es posible encontrar fractales a lo largo de todo el espacio que nos rodea, como por ejemplo las espirales gigantes de las galaxias, árboles, ríos y canales, montañas, venas del cuerpo humano, flores, etcétera.

Existen diversos tipos de fractales, como lo son la espiral de Fibonacci, que consiste en sumar los dos números anteriores y el nuevo valor se suma con el de antes para seguir sumando siempre los dos anteriores. Después, de acuerdo a los valores obtenidos se va formando una figura geométrica que al unir sus vértices, crea una espiral.

A continuación, se mostrarán algunos ejemplos de fractales, tanto de ejemplos de la naturaleza como de modelos matemáticos que crean estas estructuras.

Nota: Los gráficos pertenecientes a esta sección (fractales) se basan en códigos existentes que sufrieron leves o nulas modificaciones, pues debido al nivel de complejidad para programarlos no fue posible realizarlos. Sin embargo, se cita la fuente original de cada uno de los códigos en el pie de imagen de cada caso.

5.1. Árboles

Como ya se mencionó, los árboles son de los ejemplos más sencillos que vienen a nuestra mente a la hora de pensar en fractales. La manera en que un árbol crece y se desarrolla es, a grosso modo, la siguiente: De una raíz sale el tallo, el cual, después de cierta distancia del mismo, se comienza a separar generando dos nuevas ramas. Conforme el tiempo pasa, de cada una de las ramas se genera una nueva, indefinidamente, como lo ilustra la Figura 27.

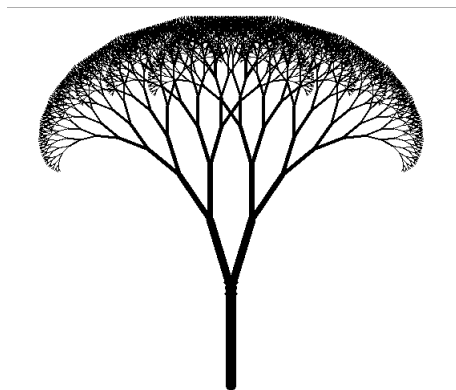


Figura 27: Gráfico de un árbol siguiendo la lógica de fractales. Código tomado de <https://stackoverflow.com/questions/39853005/drawing-a-fractal-tree-in-python-not-sure-how-to-proceed>

5.2. Triángulo de Sierpinski

Otro ejemplo de fractal es el conocido triángulo de Sierpinski. Consiste en partir de un triángulo equilátero y, a partir de él, trazar el triángulo interno (forado al unir los puntos centrales de cada arista) y después de ello, repetir dicho paso para cada uno de los siguientes triángulos formados. La Figura 28 muestra el resultado.

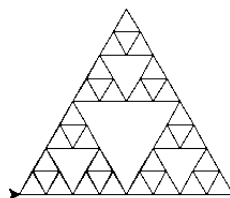


Figura 28: Gráfico del triángulo de Sierpinski. Código tomado de <https://stackoverflow.com/questions/25772750/sierpinski-triangle-recursion-using-turtle-graphics/35678092#35678092>

5.3. Curva de Dragón

Otra manera de crear fractales, es mediante la sustitución iterada, y dichos fractales se les conoce como *sistemas de funciones iteradas*. El ejemplo clásico es el conocido como curva de dragón. Se parte de una secuencia de cuatro

líneas, como un cuadrado, pero la arista de lado izquierdo o toca la parte superior, sino que del vértice inferior se dirige hacia abajo, geerando una especie de letra *P*, faltando cubrir la arista mencionada. Para la generación del fractal, conforme las iteraciones van pasando, la mitad de la figura (es decir, las primeras dos líneas) son sustituidas por la figura completa, y con la otra mitad pasa lo mismo, teniendo ahora una nueva figura con el doble de líneas. Se repite el proceso para todas las iteraciones siguientes.[9] La secuencia, después de cierto número de iteraciones, se muestra en la Figura 29.

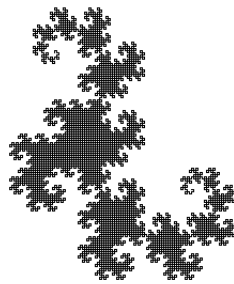


Figura 29: Fractal curva de dragón. Código tomado de <https://www.instructables.com/id/Dragon-Curve-Using-Python/>

6. Sistema Caótico Mecánico

Se explicará el experimento de Moon y Holmes. Se utilizó un aparato como el que se muestra en la Figura 30. Cuando el sistema se encontraba en reposo, entonces la viga de acero se podía encontrar en equilibrio en dos lugares distintos: pegado al imán izquierdo o derecho. Se hizo variar sinusoidalmente el voltaje aplicado, y el movimiento de la viga de acero presentó oscilaciones irregulares. Estas oscilaciones llegaron a ser caóticas.

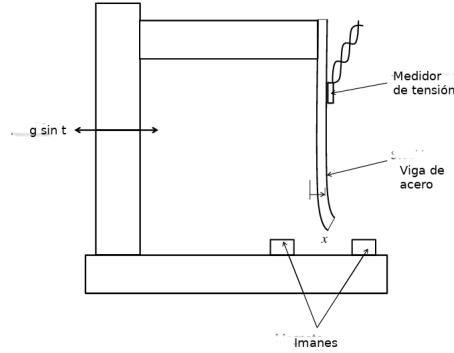


Figura 30: Sistema experimental experimento de Moon y Holmes.

A pesar de que el aparato luce bastante simple, se puede atribuir la irregularidad de las oscilaciones presentadas a situaciones como lo son excitación de modos de vibración de orden superior de la viga o ruido del dispositivo de agitación sinusoidal. Para evitar modelar ese sistema con ese tipo de consideraciones, partieron de un modelo establecido anteriormente, conocido como la *ecuación forzada de Duffing*, cuya expresión es[10]

$$\frac{d^2y}{dt^2} + \mu \frac{dy}{dt} + (y^3 - y) = g \sin t \quad (16)$$

Los primeros dos términos representan la inercia en la viga y las fuerzas disipativas, mientras que el tercero muestra los efectos de los imanes y a fuerza elástica. El término de lado derecho se debe a la variación sinusoidal.

Para realizar el análisis, se expresó la ecuación diferencial no lineal como dos ecuaciones diferenciales lineales, quedando de la siguiente manera:

$$\frac{dy}{dt} = s \quad (17)$$

$$\frac{ds}{dt} = g \sin(t) - (y^3 - y) - \mu s \quad (18)$$

La Figura 32 muestra el espacio fase del oscilador de Duffing que describe el movimiento de la vida del experimento de Duffing, mientras que la Figura 32 permite observar la variación de la variable y conforme avanza el tiempo.

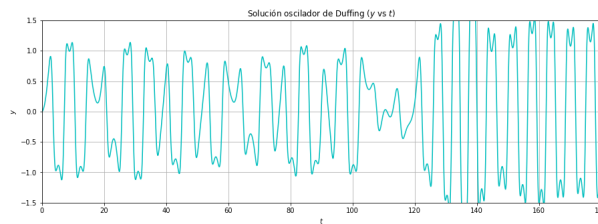


Figura 31: Sistema experimental experimento de Moon y Holmes.

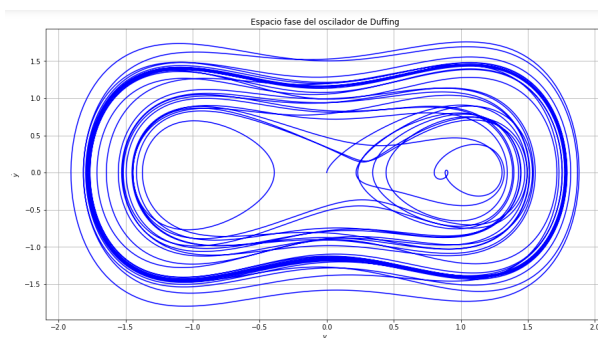


Figura 32: Espacio fase del oscilador de Duffing.

7. Tema Adicional: Predicción del Clima

A continuación, se hará mención a conceptos como la fuerza de coriolis, teoría del caos, efecto mariposa y la ecuación de advección de temperatura para describir, explicar la manera de modelar fenómenos climáticos.

Como se sabe desde hace varios años, el clima es un fenómeno muy difícil de modelar, es realmente caótico y una pequeñísima diferencia en las condiciones iniciales provocan un resultado increíblemente diferente. Tal es el caso, que Lorentz lo descubrió al realizar simulaciones sobre el clima y a un valor de las condiciones iniciales olvidó escribir los decimales últimos 3 decimales (de 6 en total).

Para comenzar, se utilizará como ejemplo el fenómeno oceánico conocido como *El Niño*. Hace algunos años, fue descubierto en Perú que en algunas épocas del año el agua en el Pacífico Oriental se tornaba más caliente de lo habitual. No ocurría cada año, pero cuando sucedía, siempre pasaba en fechas cercanas a Navidad (de ahí el nombre de *El Niño*). Para entender cómo se produce, se puede pensar en el Océano Pacífico como una gran tina llena de agua. Gracias a la *fuerza de coriolis* y a la rotación de la Tierra,

los vientos cerca del Ecuador suelen soplar de Este a Oeste. Los vientos empujan al agua, de tal forma que el nivel del mar en Indonesia llega a ser usualmente medio metro más alto que en Perú. Como el agua cerca de la superficie, calentada gracias al Sol, es llevada hacia la dirección Oeste, se dirige una profunda y fría agua hacia el Este para reemplazarla. Si dichos vientos comienzan a soplar más y más fuerte cada vez, el agua tibia es empujada hacia el Oeste y la parte Este del Océano se torna más fría. A ese fenómeno es al que se le llama *La Niña*. Pero si dichos vientos cesan, esa enorme masa de agua tibia termina justo en Sudamérica, y ése es *EL Niño*. [11]

Sería fascinante poder conocer con exactitud cuándo va a llover, cuándo y dónde nevará, entre otros fenómenos meteorológicos. Sin embargo, no es tan sencillo. Como se mencionó para el ejemplo anterior, el fenómeno de *El Niño* no sucede cada año, ocurre entre cada 2 o 7 años, por lo que es algo menor a un ciclo, puede catalogar como un fenómeno periódico o un cuasi-ciclo.

A partir de la experiencia de Lorentz anteriormente mencionada, fue que dio origen la *teoría del caos*. Lorentz llegó a la conclusión de que no importa lo avanzadas que puedan llegar a ser las computadoras, jamás se va a poder predecir con exactitud el clima más allá de unos cuantos días después. En cualquiera de estos sistemas, el más mínimo cambio en las condiciones iniciales puede desencadenar resultados completamente inesperados. En base a esta conclusión, es que la famosa pregunta de Lorentz fue hecha: *¿El aleteo de las alas de una mariposa en Brasil pueden provocar un tornado en Texas?* A eso se le llama *Efecto Mariposa*, y básicamente hace mención (más que al aleteo como tal) a que variando mínimamente el estado climático actual se obtendrán cosas inimaginables. Dichos parámetros que pueden variar pueden ser el brillo del Sol, la temperatura, la presión, el viento, la humedad, entre otros.

Por otro lado, existen diversas ecuaciones diferenciales que se utilizan para realizar predicciones climáticas, una de ellas es la conocida *ecuación de advección de temperatura*. La expresión matemática que la comprende es

$$\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} = 0 \quad (19)$$

El primer término de lado izquierdo corresponde a la tasa de cambio de temperatura (T) como función de tiempo (t). Dicho término depende de la advección de temperatura en las direcciones Este-Oeste (x) y Norte-Sur (y) a causa del viento (que tienen componentes u y v en las direcciones x y

y, respectivamente). Los términos de advección son calculados mediante la multiplicación de los gradientes de temperatura por las componentes correspondientes del viento.

Dicha ecuación puede ser convertida a una forma más sencilla de resolver a computadora, quedando de la siguiente manera:

$$T^{n+1} = T^n - \frac{\Delta t}{\Delta x} (u^n (T_x^n - T_{x-1}^n) + v^n (T_{y+1}^n - T_y^n)) \quad (20)$$

En esta expresión, los superíndices indican el nivel del tiempo y los subíndices la posición horizontal.

Conclusión

A lo largo de la realización de este proyecto se obtuvo una gran cantidad de conocimientos nuevos sobre temas que parecieron realmente interesantes por varios motivos: permiten comprender la metodología seguida para analizar de una manera sencilla sistemas físicos que parecen complicados, ya que hasta el momento de los sistemas con los que se había trabajado ninguno había tenido la característica de ser caótico, sin embargo, ahora es posible comprender básicamente para cualquier sistema en qué momentos se comienza a presentar dicha característica. Tampoco se tenía clara la manera de graficar las proyecciones de un gráfico tridimensional en las intersecciones con uno de los ejes, y las secciones de Poincaré, a pesar de que fueron las más complicadas a mi parecer, me permitieron aprender nuevas funciones y métodos de Python, un lenguaje del que a pesar de haber tenido conocimientos previos, no eran tan buenos, y ahora realmente he mejorado en mis habilidades para programar en él.

Lo que más me agradó de haber realizado este trabajo es que normalmente en la materia de mecánica analítica no se hace tanto énfasis ni en la parte de programar (pues el mismo nombre cataloga a la materia como algo puramente teórico) ni en la teoría del caos, pues es necesario analizar varios temas que para su entendimiento requieren de varias sesiones dedicadas a ellas, y se descarta la parte de entender el caos; sin embargo, ahora siento tener más conocimientos que incluso pueden llegar a ser de un nivel de entendimiento más complejo.

Finalmente, debo mencionar que me pareció interesante el hecho de que sistemas que a simple vista parecen sencillos de modelar, sean realmente tan complicados de analizar, y la teoría del caos permite poder identificar las características de esos sistemas para, a pesar de no entenderlas a la perfección, por lo menos poderlas describir de una manera general y entendible.

8. Apéndice

```
Órbita de  $\frac{x^2 + 3}{9}$ .  
  
'''python  
def orbit(w):  
    #return (w * w + 3) / 9  
    return 2.3 * w * (1 - w)  
    #return (3 * w - math.pow(w,3) / 2  
  
f = open("OrbitEx.txt", "w+")  
i = 0  
w0 = 0.005  
#w0 = 4  
while i < 100 :  
    f.write(str(w0) + '\t' + str(orbit(w0)) + '\n')  
    w0 = orbit(w0)  
    i += 10  
f.close()  
  
i = np.linspace(0, 0.6, 5000)  
  
x, y = np.loadtxt('OrbitEx.txt', delimiter = '\t', unpack = True)  
plt.plot(x, y, 'm', label = 'f(x) = 2.3x(1-x)')  
plt.title('Gráfico de Cobweb para la función  $f(x) = 2.3x(1-x)$ ')  
plt.ylabel('y = f(x)')  
plt.xlabel('x')  
#plt.ylim(0, 0.6)  
#plt.xlim(0, 0.6)  
plt.plot(x, x, 'c', label = 'f(x) = x')  
plt.legend(loc='best')  
plt.show
```

```
Mapeo Logístico varios casos:  
  
'''python  
def orbit_0p5(w):  
    return 0.5 * w * (1 - w)  
def orbit_2p8(w):  
    return 2.8 * w * (1 - w)  
def orbit_3p3(w):  
    return 3.3 * w * (1 - w)  
def orbit_3p5(w):  
    return 3.5 * w * (1 - w)  
  
f = open("OrbitVarios.txt", "w+")  
i = 0  
w0p5 = 0.5  
w2p8 = 0.5  
w3p3 = 0.5  
w3p5 = 0.5  
  
while i < 15 :  
    f.write(str(i) + '\t' + str(w0p5) + '\t' + str(w2p8) + '\t' + str(w3p3) + '\t'  
    + str(w3p5) + '\n')  
    w0p5 = orbit_0p5(w0p5)  
    w2p8 = orbit_2p8(w2p8)  
    w3p3 = orbit_3p3(w3p3)  
    w3p5 = orbit_3p5(w3p5)  
    i += 1  
f.close()  
  
i = np.linspace(0, 0.6, 5000)  
n, o1, o2, o3, o4 = np.loadtxt('OrbitVarios.txt', delimiter = '\t', unpack = True)  
plt.plot(n, o1, 'm', label='r = 0.5')  
plt.plot(n, o2, 'c', label='r = 2.8')  
plt.plot(n, o3, 'b', label='r = 3.3')  
plt.plot(n, o4, 'y', label='r = 3.5')  
plt.rcParams['figure.figsize'] = (15, 7)  
plt.title('Gráfico de  $x_n$  vs n para  $f(x) = rx(1-x)$ ')  
plt.ylabel('xn')  
plt.xlabel('n')  
plt.legend(loc='best')  
plt.show
```

```

Diagrama de Bifurcación
```python
import random
def orbit(a, b):
 return a * b * (1 - b)
f = open("Bifurcation.txt", "w+")
r, x = 1, 0
while r <= 4:
 x = random.uniform(0, 1)
 iter = 0
 while iter <= 100:
 x = orbit(r, x)
 iter += 1
 while iter <= 500:
 f.write(str(r) + '\t' + str(x) + '\n')
 x = orbit(r, x)
 iter += 1
 r += 0.0005
f.close()
x, y = np.loadtxt('Bifurcation.txt', delimiter = '\t', unpack = True)
plt.figure(figsize=(15,26))
plt.suptitle('Diagrama de bifurcación para el mapeo logístico, diferentes límites del eje x', fontsize=16)
plt.subplot(3, 1, 1)
plt.plot(x, y, 'c', linestyle="", marker=",")
plt.title('Bifurcación mapeo logístico, vista completa')
plt.ylabel('x')
plt.xlabel('r')
plt.subplot(3, 1, 2)
plt.plot(x, y, 'c', linestyle="", marker=",")
plt.title('Bifurcación mapeo logístico, zoom 1')
plt.ylabel('x')
plt.xlabel('r')
plt.xlim(3.4, 4)
plt.subplot(3, 1, 3)
plt.plot(x, y, 'c')
plt.title('Bifurcación mapeo logístico, zoom 2')
plt.ylabel('x')
plt.xlabel('r')
plt.xlim(3.8, 3.9)
plt.show()

```

Gráfico de Cobweb para el mapeo logístico en un determinado  $r$ .

```

```python
def orbit(a, b):
    return a * b * (1 - b)

def dorbit(a, b):
    return math.log(abs(a - a * 2 * b))

f = open("LyapunovLogistic.txt", "w+")

r = 3.00
x = random.uniform(0, 1)

while r <= 4.00:
    itera = 0
    lnsun = 0.0

    while itera <= 300:
        x = orbit(r, x)
        itera += 1

    while itera <= 5000:
        x = orbit(r, x)
        lnsun += dorbit(r, x)
        #lnsun += 0.5
        itera += 1

    f.write(str(r) + '\t' + str(lnsun / 5000) + '\n')
    r += 0.0005

f.close()

x, y = np.loadtxt('LyapunovLogistic.txt', delimiter = '\t', unpack = True)
plt.rcParams['figure.figsize'] = (15, 7)
plt.plot(x, y, 'm', marker=",")
plt.plot(x, x-x)
plt.title('Exponentes de Lyapunov para el mapeo logístico')
plt.xlabel('r')
plt.ylabel('$\lambda$')
plt.show

```

```

Atractor de Lorentz

'''python
def model(s,t,u):
    x = s[0]
    y = s[1]
    z = s[2]
    sigma = 10
    r = 28
    b = 8/3

    dxdt = - sigma*x + sigma*y
    dydt = r*x - y - x*z
    dzdt = - b*z + x*y
    dsdt = [dxdt,dydt, dzdt]
    return dsdt

s0 = [0.001,0.001, 0.001]

n = 100000

t = np.linspace(0, 1000,n)

s = odeint(model,s0,t , args=(n,))

x = s[:, 0]
y = s[:, 1]
z = s[:, 2]

plt.figure(figsize=(11,8))
plt.plot(y,z,'b')
plt.title("Gráfico $y$ vs $z$ para el Atractor de Lorentz")
plt.ylabel('z')
plt.xlabel('y')
plt.show()

```

```

'''Atractor de Lorentz (Continuación)
'''python
plt.figure(figsize=(11,8))
plt.plot(x,z,'c', linestyle = "")
plt.title("Gráfico $x$ vs $z$ para el Atractor de Lorentz")
plt.ylabel('z')
plt.xlabel('x')
plt.show()

'''python
plt.figure(figsize=(11,8))
plt.plot(x,y,'m',linestyle="", marker = ',')
plt.title("Gráfico $x$ vs $y$ para el Atractor de Lorentz")
plt.ylabel('y')
plt.xlabel('x')
plt.show()

'''python
plt.figure(figsize=(20,9))
plt.suptitle('Gráficos de $x$, $y$ y $z$ vs $t$ para el atractor de Lorentz', fontsize=16)
plt.subplot(3, 1, 1)
plt.plot(t,x,'y',linestyle="", marker = ',')
plt.xlim(20, 70)
plt.title('$x$ vs $t$')
plt.ylabel('$x(t)$')
plt.xlabel('$t$')

plt.subplot(3, 1, 2)
plt.plot(t,y,'r',linestyle="", marker = ',')
plt.xlim(20, 70)
plt.title('$y$ vs $t$')
plt.ylabel('$y(t)$')
plt.xlabel('$t$')

plt.subplot(3, 1, 3)
plt.plot(t,z,'b',linestyle="", marker = ',')
plt.xlim(20, 70)
plt.title('$z$ vs $t$')
plt.ylabel('$z(t)$')
plt.xlabel('$t$')
plt.show()

```

```

Atractor de Rossler
'''python
def model(s,t,u):
    x = s[0]
    y = s[1]
    z = s[2]
    A = 0.2
    B = 0.2
    C = 5.7

    dxdt = - (y + z)
    dydt = x + A*y
    dzdt = B + x*z - C*z
    dsdt = [dxdt,dydt, dzdt]
    return dsdt

s0 = [0.001,0.001, 0.001]

n = 50000

t = np.linspace(0, 300,n)

s = odeint(model,s0,t , args=(n,))

x = s[:, 0]
y = s[:, 1]
z = s[:, 2]

plt.figure(figsize=(11,8))
plt.plot(x,y,'m', marker = ',')
plt.title("Gráfico $x$ vs $y$ para el Atractor de Rossler")
plt.ylabel('y')
plt.xlabel('x')
plt.show()
'''

```

```

Atractor de Rossler (continuación)
'''python
plt.figure(figsize=(11,8))
plt.plot(x,z,'c', marker = ',')
plt.title("Gráfico $x$ vs $z$ para el Atractor de Rossler")
plt.ylabel('z')
plt.xlabel('x')
plt.show()

'''python
plt.figure(figsize=(11,8))
plt.plot(y,z,'b', marker = ',')
plt.title("Gráfico $y$ vs $z$ para el Atractor de Rossler")
plt.ylabel('z')
plt.xlabel('y')
plt.show()

'''python
plt.figure(figsize=(20,9))
plt.suptitle('Gráficos de $x$, $y$ y $z$ vs $t$ para el atractor de Rossler', fontsize=16)
plt.subplot(3, 1, 1)
plt.plot(t,x,'y',linestyle="", marker = ',')
plt.title('$x$ vs $t$')
plt.ylabel('$x(t)$')
plt.xlabel('$t$')

plt.subplot(3, 1, 2)
plt.plot(t,y,'r',linestyle="", marker = ',')
plt.title('$y$ vs $t$')
plt.ylabel('$y(t)$')
plt.xlabel('$t$')

plt.subplot(3, 1, 3)
plt.plot(t,z,'b',linestyle="", marker = ',')
plt.title('$z$ vs $t$')
plt.ylabel('$z(t)$')
plt.xlabel('$t$')

plt.show()
'''

```

```

Secciones de Poincaré para los atractores de Lorentz y Rossler

'''python
def model(s,t,u):
    x = s[0]
    y = s[1]
    z = s[2]
    sigma = 10
    r = 28
    b = 8/3

    dxdt = -sigma*x + sigma*y
    dydt = r*x - y - x*z
    dzdt = -b*z + x*y
    dsdt = [dxdt,dydt, dzdt]
    return dsdt

s0 = [0.001,0.001, 0.001]
n = 100000
t = np.linspace(0, 1000,n)

s = odeint(model,s0,t , args=(n,))
eje_1 = []
eje_2 = []

for k in range(100000):
    if float(format(s[:, 0][k], '0.1f')) == 0:
        eje_1.append(s[:, 1][k])
        eje_2.append(s[:, 2][k])

# plot results
plt.figure(figsize=(11,8))
plt.plot(eje_1,eje_2,'m')
plt.title("Secciones de Poincaré sobre los ejes $y$ y $z$ para el Atractor de Lorentz")
plt.ylabel('z')
plt.xlabel('y')
#plt.legend(loc='best')
plt.show()
'''

```

```

'''python
eje_1 = []
eje_2 = []

for k in range(100000):
    if float(format(s[:, 1][k], '0.1f')) == 0:
        eje_1.append(s[:, 0][k])
        eje_2.append(s[:, 2][k])

# plot results
plt.figure(figsize=(11,8))
plt.plot(eje_1,eje_2,'c')
plt.title("Secciones de Poincaré sobre los ejes $x$ y $z$ para el Atractor de Lorentz")
plt.ylabel('z')
plt.xlabel('x')
#plt.legend(loc='best')
plt.show()
'''

'''python
eje_1 = []
eje_2 = []

for k in range(100000):
    if float(format(s[:, 2][k], '0.1f')) == 0:
        eje_1.append(s[:, 0][k])
        eje_2.append(s[:, 1][k])

# plot results
plt.figure(figsize=(11,8))
plt.plot(eje_1,eje_2,'k')
plt.title("Secciones de Poincaré sobre los ejes $x$ y $y$ para el Atractor de Lorentz")
plt.ylabel('y')
plt.xlabel('x')
#plt.legend(loc='best')
plt.show()
'''

```

```

'''python
def model(s,t,u):
    x = s[0]
    y = s[1]
    z = s[2]
    A = 0.2
    B = 0.2
    C = 5.7

    dxdt = - (y + z)
    dydt = x + A*y
    dzdt = B + x*z - C*z
    dsdt = [dxdt,dydt, dzdt]
    return dsdt
s0 = [0.001,0.001, 0.001]
n = 100000
t = np.linspace(0, 100000,n)
s = odeint(model,s0,t , args=(n,))

x = s[:, 0]
y = s[:, 1]
z = s[:, 2]

eje_1 = []
eje_2 = []

for k in range(100000):
    if float(format(s[:, 0][k], '0.1f')) == 0:
        eje_1.append(s[:, 1][k])
        eje_2.append(s[:, 2][k])

# plot results
plt.figure(figsize=(11,8))
plt.plot(eje_1,eje_2,'m')
plt.title("Secciones de Poincaré sobre los ejes $y$ y $z$ para el Atractor de Rossler")
plt.ylabel('z')
plt.xlabel('y')
plt.legend(loc='best')
plt.show()
'''

```

```

'''python
eje_1 = []
eje_2 = []

for k in range(100000):
    if float(format(s[:, 1][k], '0.1f')) == 0:
        eje_1.append(s[:, 0][k])
        eje_2.append(s[:, 2][k])

# plot results
plt.figure(figsize=(11,8))
plt.plot(eje_1,eje_2,'c.')
plt.title("Secciones de Poincaré sobre los ejes $x$ y $z$ para el Atractor de Rossler")
plt.ylabel('z')
plt.xlabel('x')
plt.legend(loc='best')
plt.show()
'''

'''python
eje_1 = []
eje_2 = []

for k in range(100000):
    if float(format(s[:, 2][k], '0.1f')) == 0:
        eje_1.append(s[:, 0][k])
        eje_2.append(s[:, 1][k])

# plot results
plt.figure(figsize=(11,8))
plt.plot(eje_1,eje_2,'b,')
plt.title("Secciones de Poincaré sobre los ejes $x$ y $y$ para el Atractor de Rossler")
plt.ylabel('y')
plt.xlabel('x')
plt.legend(loc='best')
plt.show()
'''

```



```

Atractor de Hénon
'''
python
def nuevo(x_0, y_0):
    return 1 - a*math.pow(x_0, 2) + y_0, b*x_0
i = 0
x_0 = .1
y_0 = .3
V_X = []
V_Y = []
t = []
while i < 100000 :
    x_0, y_0 = nuevo(x_0, y_0)
    V_X.append(x_0)
    V_Y.append(y_0)
    t.append(i)
    i += 1
plt.figure(figsize=(11,8))
plt.plot(V_X, V_Y, 'b,')
plt.title('Atractor de Henon')
plt.ylabel('y')
plt.xlabel('x')
plt.show
'''
'''
python
plt.figure(figsize=(20,7))
plt.suptitle('Gráficos de $x$ y $y$ vs $t$ para el atractor de Henon', fontsize=16)
plt.subplot(2, 1, 1)
plt.plot(t,V_X,'m',linestyle="--", marker = ',')
plt.title('$x$ vs $t$')
plt.ylabel('$x(t)$')
plt.xlabel('$t$')
plt.xlim(0, 70)

plt.subplot(2, 1, 2)
plt.plot(t, V_Y,'c',linestyle="--", marker = ',')
plt.title('$y$ vs $t$')
plt.ylabel('$y(t)$')
plt.xlabel('$t$')
plt.xlim(0, 70)

plt.show()
'''

```

```

Atractor de Peter de Jong
'''
python
def nuevo_2(x_0, y_0):
    return math.sin(a * y_0) - math.cos(b * x_0), math.sin(c * x_0) - math.cos(d * y_0)
i = 0
x_0 = .1
y_0 = .3
a, b, c, d = -2, -2, -1.2, 2
V_X = []
V_Y = []
t = []
while i < 100000 :
    x_0, y_0 = nuevo_2(x_0, y_0)
    V_X.append(x_0)
    V_Y.append(y_0)
    t.append(i)
    i += 1
plt.figure(figsize=(11,8))
plt.plot(V_X, V_Y, 'b,')
plt.title('Atractor de Peter de Jong (condiciones iniciales 1)')
plt.ylabel('y')
plt.xlabel('x')
plt.show
'''
'''
python
plt.figure(figsize=(20,7))
plt.suptitle('Gráficos de $x$ y $y$ vs $t$ para el atractor de Peter de Jong (condiciones iniciales 1)', fontsize=16)
plt.subplot(2, 1, 1)
plt.plot(t,V_X,'m',linestyle="--", marker = ',')
plt.title('$x$ vs $t$')
plt.ylabel('$x(t)$')
plt.xlabel('$t$')
plt.xlim(0, 200)

plt.subplot(2, 1, 2)
plt.plot(t, V_Y,'c',linestyle="--", marker = ',')
plt.title('$y$ vs $t$')
plt.ylabel('$y(t)$')
plt.xlabel('$t$')
plt.xlim(0, 200)

plt.show()
'''

```

```

'''python
def nuevo_3(x_0, y_0):
    return math.sin(a * y_0) - math.cos(b * x_0), math.sin(c * x_0) - math.cos(d * y_0)

i, x_0, y_0 = 0, .1, .3
a, b, c, d = 2.01, -2.53, 1.61, -.33
V_X = []
V_Y = []
t = []
while i < 100000 :
    x_0, y_0 = nuevo_2(x_0, y_0)
    V_X.append(x_0)
    V_Y.append(y_0)
    t.append(i)
    i += 1

plt.figure(figsize=(11,8))
plt.plot(V_X, V_Y, 'b,')
plt.title('Atractor de Peter de Jong (condiciones iniciales 2)')
plt.ylabel('y')
plt.xlabel('x')
plt.show

'''python
plt.figure(figsize=(20,7))
plt.suptitle('Gráficos de $x$ y $y$ vs $t$ para el atractor de Peter de Jong (condiciones iniciales 2)', fontsize=16)
plt.subplot(2, 1, 1)
plt.plot(t,V_X,'m',linestyle="--", marker = ',')
plt.title('$x$ vs $t$')
plt.ylabel('$x(t)$')
plt.xlabel('$t$')
plt.xlim(0, 200)

plt.subplot(2, 1, 2)
plt.plot(t, V_Y,'c',linestyle="--", marker = ',')
plt.title('$y$ vs $t$')
plt.ylabel('$y(t)$')
plt.xlabel('$t$')
plt.xlim(0, 200)
plt.show()
'''

```

```

Arbol

'''python
import turtle
t = turtle.Turtle(shape="turtle")
t.lt(90)
lv = 13
l = 120
s = 17
t.width(lv)
t.penup()
t.bk(l)
t.pendown()
t.fd(l)
def draw_tree(l, level):
    width = t.width() # save the current pen width
    t.width(width * 3.0 / 4.0) # narrow the pen width
    l = 3.0 / 4.0 * l
    t.lt(s)
    t.fd(l)

    if level < lv:
        draw_tree(l, level + 1)
    t.bk(l)
    t.rt(2 * s)
    t.fd(l)

    if level < lv:
        draw_tree(l, level + 1)
    t.bk(l)
    t.lt(s)

    t.width(width) # restore the previous pen width

t.speed("fastest")

draw_tree(l, 2)

turtle.done()
'''

```

Triángulo de Sierpinski

```

'''python
import turtle

def sier(side, level):
    if level == 1:
        for i in range(3):
            turtle.fd(side)
            turtle.left(120)
    else:
        sier(side/2, level-1)
        turtle.fd(side/2)
        sier(side/2, level-1)
        turtle.bk(side/2)
        turtle.left(60)
        turtle.fd(side/2)
        turtle.right(60)
        sier(side/2, level-1)
        turtle.left(60)
        turtle.bk(side/2)
        turtle.right(60)

sier(200, 4)

if __name__ == '__main__':
    main()
    turtle.mainloop()
'''

```

Curva de Dragón

```

'''python
r = 'r'
l = 'l'
old = r
new = old
iteration=int(input('Enter iteration:'))
length=int(input('Enter length of each segment:'))
pencolor=input('Enter pen color:')
bgcolor=input('Enter background color:')
cycle=1
while cycle<iteration:
    new = (old) + (r)
    old = old[::-1]
    for char in range(0,len(old)):
        if old[char] == r:
            old = (old[:char]) + (l) + (old[char+1:])
        elif old[char] == l:
            old = (old[:char]) + (r) + (old[char+1:])
    new = (new) + (old)
    old = new
    cycle=cycle+1
printans=input('Display r/l form?(y/n):')
if printans=='y':
    print(new)
turtle.ht()
turtle.speed(0)
turtle.color(pencolor)
turtle.bgcolor(bgcolor)
turtle.forward(length)
for char in range(0,len(new)):
    if new[char] == (r):
        turtle.right(90)
        turtle.forward(length)
    elif new[char] == (l):
        turtle.left(90)
        turtle.forward(length)
'''

```

Ejemplo Sistema Mecánico Caótico

```
python
def ecu(c, t, v, g):
    (y, s) = c
    dcdt = (s, g * math.sin(t) - (math.pow(y, 3) - y) - v*s)
    return dcdt

def sln(t):
    res = odeint(ecu, c_0, t, args = (v, g))
    return res

g = .3
v = .1
t = np.linspace(0, 250, 10000)
c_0 = (0, 0)
plt.figure(figsize=(15,8))
plt.plot(sln(t)[:, 0], sln(t)[:, 1], 'b')
plt.title('Espacio fase del oscilador de Duffing')
plt.xlabel("$y$")
plt.ylabel("$\dot{y}$")
plt.grid()
plt.show()

python
plt.figure(figsize=(15,5))
plt.plot(t, sln(t)[:, 1], 'c')
plt.title('Solución oscilador de Duffing')
plt.grid()
plt.xlim(0, 180)
plt.ylim(-1.5, 1.5)
plt.xlabel("$t$")
plt.ylabel("$y$")
plt.show()
```

Referencias

1. CHAOS: An Introduction to Dynamical Systems - KATHLEEN T. ALLIGOOD, TIM D. SAUER & JAMES A. YORKE. (2000, Springer)
2. Studies in Nonlinearity - Steven H. Strogatz - Nonlinear Dynamics and Chaos (2001, Westview Press)
3. A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining Lyapunov Exponents from a Time Series," *Physica D*, Vol. 16, pp. 285-317, 1985.
4. [Texts on computational mechanics 7] G. Faust, M. Haase, J.H. Argyris - A an introduction for natural scientists & engineers (1994, North-Holland)
5. David Acheson - From calculus to chaos an introduction to dynamics (1997, Oxford University Press)
6. Haoran Wen, .^A Review Of The Hénon Map And Its Physical Interpretations", School of Physics. Georgia Institute of Technology.
7. Kevin MacLeod. "Strange Attractors: The De Jong Attractors", sitio web: www.algosome.com
8. Robert Hilborn - Chaos & nonlinear dynamics an introduction for scientists & engineers (2000, Oxford University Press)
9. Fractal Foundation - "Geometric Fractals". Sitio web: <https://fractalfoundation.org/OFC/OFC-index.htm>
10. Edward Ott - Chaos in dynamical systems (1993, Cambridge University Press)
11. National Ocean Service - What are El Niño and La Niña? Sitio web: <https://oceanservice.noaa.gov/facts/ninonina.html>
12. Department of Meteorology and Walker Institute - How do we predict Weather and Climate? Sitio web: http://www.met.reading.ac.uk/outreach/richardson_fancy.pdf