

UNIVERSIDAD DE SONORA

LICENCIATURA EN FÍSICA

FÍSICA COMPUTACIONAL I

---

## Evaluación 2

---

*Alumno:*

José Gabriel Navarro I.

*Profesor:*

Carlos Lizarraga Celaya

28 de Abril de 2018



# 1 Introducción

En el presente reporte se habla acerca de la segunda evaluación de la materia Física Computacional I, que abarca el uso de Jupyter Lab, utilizando distintas librerías para la solución de un sistema de ecuaciones y sus posteriores graficas. En la presente evaluación, se resuelve el sistema atractor de Lorenz.

El atractor de Lorenz es un concepto que fue introducido por Edward Lorenz en 1963. Lorenz descubrió por primera vez el caos por accidente, cuando estaba realizando un modelo matemático para la convección atmosférica. Es por eso que este es un sistema dinámico determinista tridimensional no lineal derivado de las ecuaciones simplificadas de convección que se producen en las ecuaciones dinámicas de la atmósfera terrestre. Las ecuaciones son:

$$\begin{aligned}\dot{x} &= a(y - x) \\ \dot{y} &= x(b - z) - y \\ \dot{z} &= xy - cz\end{aligned}$$

Utilizando el código proporcionado por el blog de Geoff Boeing, se pudo hacer un análisis de este sistema para distintos parámetros de  $a$ ,  $b$  y  $c$ , en nuestro caso: sigma, beta y rho. En donde se podrá observar las distintas graficas producidas y su interpretación correspondiente.

# 2 Resultados

Se utilizó el código proporcionado por Geoff Boeing para analizar el atractor de Lorenz utilizando Python, en donde se vio como hacer un gif y otros tipos de graficas. Se realizaron tres ejemplos distintos con distintos valores para sigma, beta y rho. A continuación se presentan los resultados obtenidos para distintos valores.

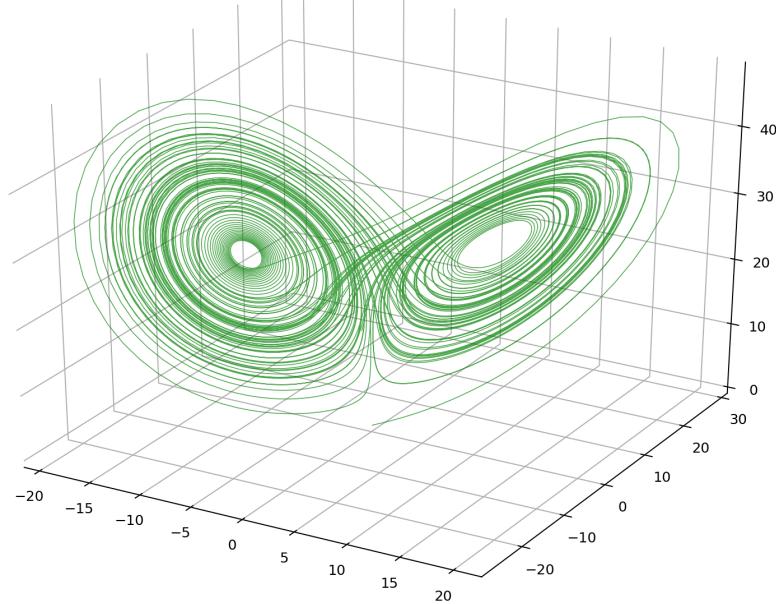
## 2.1 Ejemplo 1

Los valores para el primer ejemplo fueron los siguientes:

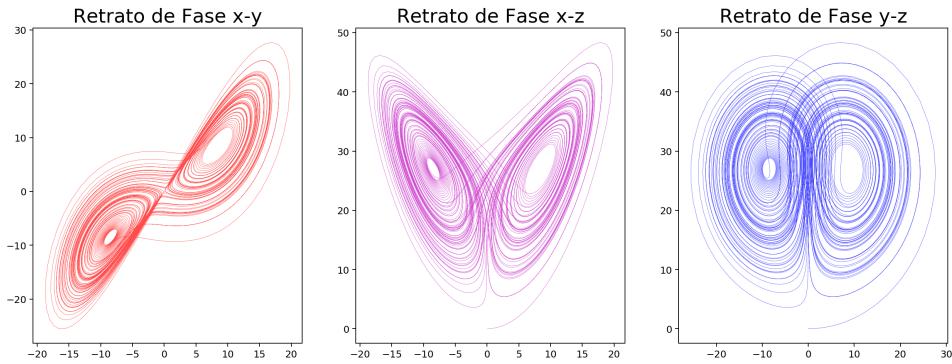
$$\begin{aligned}sigma &= 10 \\ beta &= 8/3 \\ rho &= 28\end{aligned}$$

Las graficas obtenidas fueron la de fase en 3D, y para cada plano, además de la posición contra el tiempo. La fase en 3D es:

### Atractor de Lorentz diagrama de fase

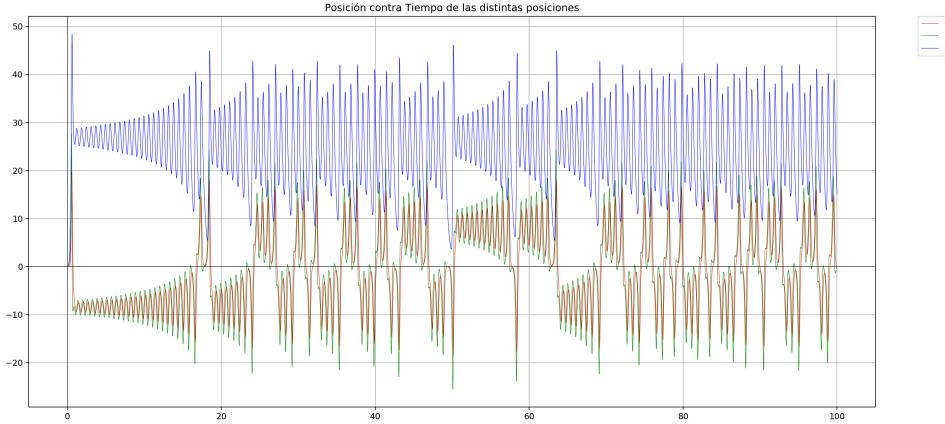


En donde podemos observar como tiene una forma del número ocho pero doblado. Este modelo demuestra el caos atado a un ciertos valores para los parámetros y su atractor es fractal. Si analizamos el retrato de fase para cada plano obtenemos:



Como podemos ver, para cada plano xy, xz, yz se tiene una distinta forma, donde en xy, se observa la figura de un ocho aplastado. Mientras que en x-z y y-z aparecen unas "mariposas".

La posición contra tiempo de cada una de las variables x, y, z resultó:



Podemos observar como la posición en  $z$  oscila sobre cero, es decir, solo toma valores positivos, mientras que  $x$  y  $y$  siguen mas o menos el mismo patrón y si pasan por la posición 0 en algún tiempo. Cabe mencionar que como el sistema es caótico, el sistema nunca pasará por una posición que ya haya atravesado.

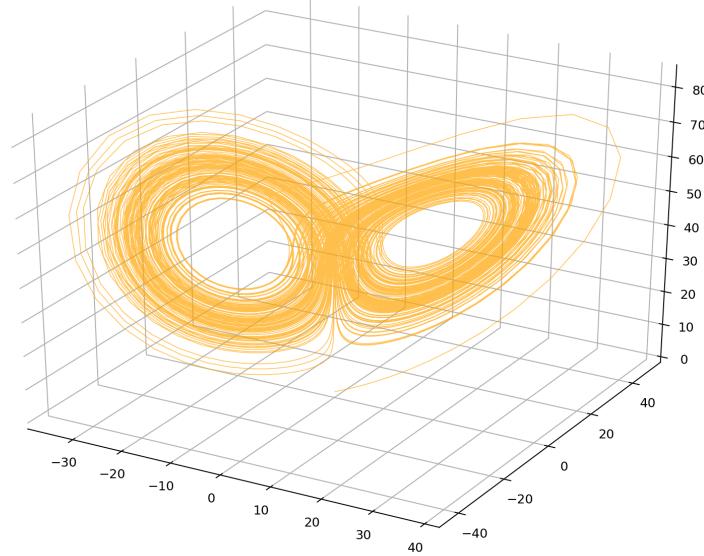
## 2.2 Ejemplo 2

Los valores para el segundo ejemplo fueron los siguientes:

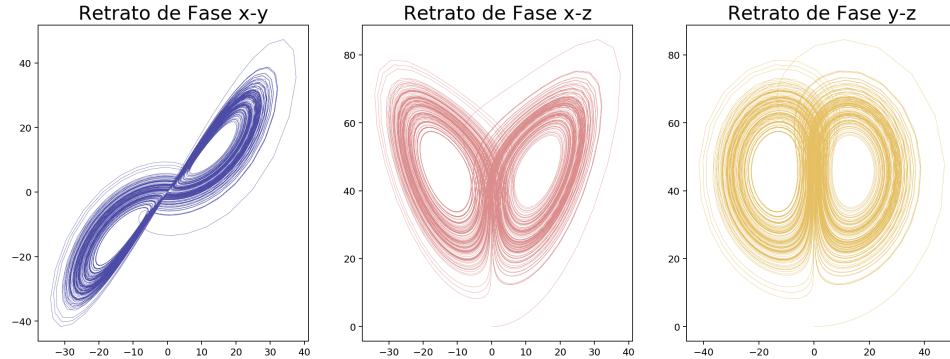
$$\begin{aligned} \sigma &= 28 \\ \beta &= 4 \\ \rho &= 46.92 \end{aligned}$$

El retrato de fase en 3D es el siguiente:

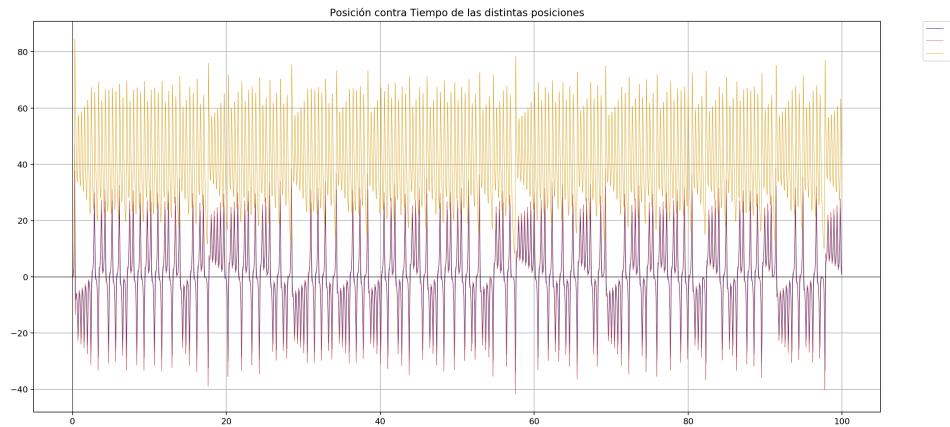
Atractor de Lorentz diagrama de fase



Como se puede observar, la gráfica es muy parecida a la del primer ejemplo, pero esta vez la figura no se "rellena", ahora si tiene la forma de un ocho. Analizando el retrato de fase para cada plano:



De igual manera, son muy parecidas a las del primer ejemplo, pero tienen el "agujero" en el centro (al menos para el caso del plano x-y), mientras que x-z y y-z aparecen las mariposas. Por último, las posiciones contra el tiempo:



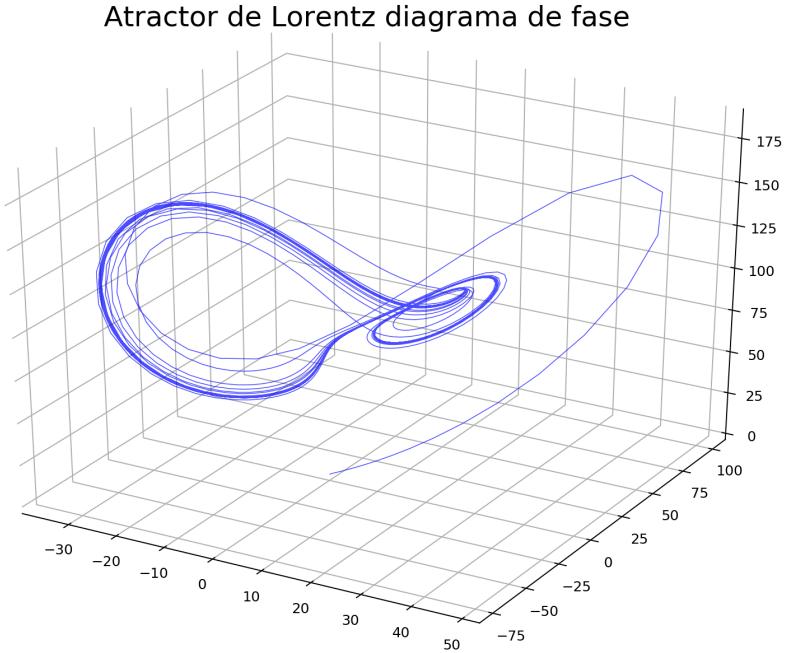
Como se puede observar, existe una gran relación entre el ejemplo 1 y el ejemplo 2, ya que la posición en  $z$  es mayor a 0, y para  $x$  y  $y$ , pasa por 0, pero ahora presencia mas caos.

### 2.3 Ejemplo 3

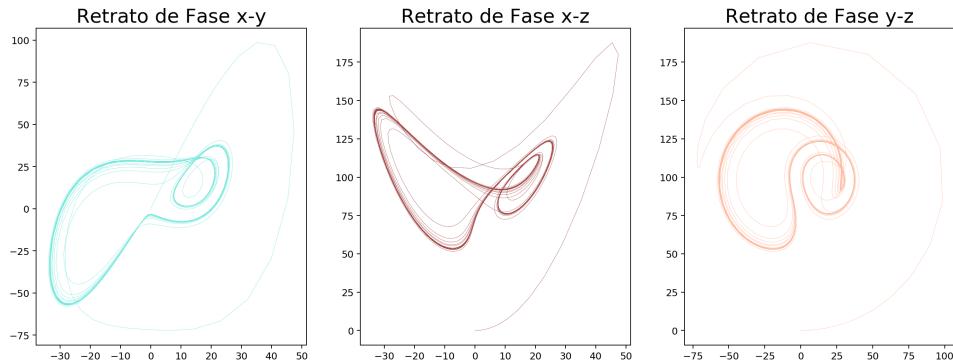
Los valores para el tercer ejemplo fueron los siguientes:

$$\begin{aligned} \sigma &= 10 \\ \beta &= 8/3 \\ \rho &= 99.96 \end{aligned}$$

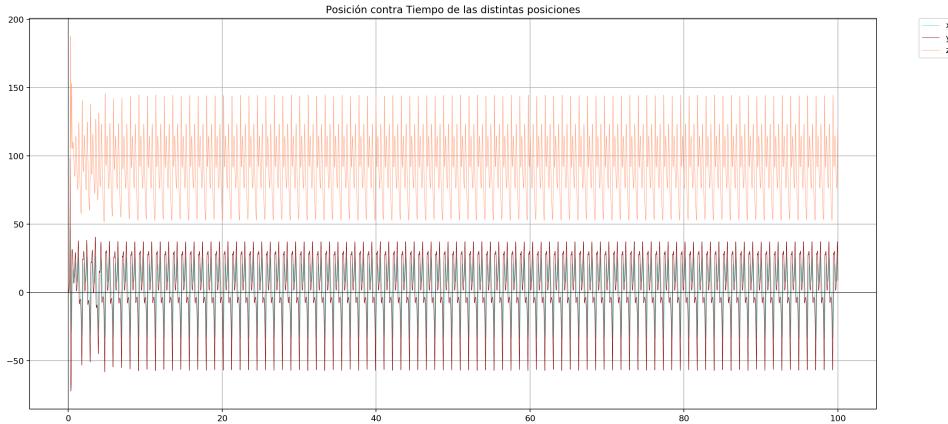
El retrato de fase en 3D es:



Para este ejemplo ya no se sigue un patrón como los anteriores, si no que se mueve por distintos octantes con una cierta inclinación hacia el eje de las x negativo y valores mas altos para z. Los valores son iguales que en el ejemplo 1, excepto rho, así que de esto se puede concluir que hasta con un pequeño cambio en los parámetros, el sistema puede entrar a mas caos. Si observamos los retratos de fase para cada plano tenemos:



A diferencia de los primeros dos ejemplos, se puede observar que este sistema es mas caótico, ya que no se puede observar ni la figura de un ocho, ni las mariposas, si no mas bien un patrón con forma circular u ovalada. Por último, analizando la posición contra tiempo:



El sistema se mueve mucho mas rápido y de una forma mas caótica que en casos anteriores, reforzando que si se mueve un solo parámetro, el sistema cambia totalmente. Algo que si tiene de parecido con los ejemplos anteriores, es que  $z$  siempre toma valores positivos, mientras que  $x$  y  $y$  toman tanto positivos como negativos.

### 3 Conclusiones

Al terminar esta evaluación me di cuenta que el utilizar herramientas como Python nos permite observar con mayor facilidad distintos sistemas y como es su funcionamiento. No solamente para su solución analítica, sino también para su interpretación mediante el uso de gráficas tanto en 3D como 2D.

A pesar de los distintos contratiempos dados en la evaluación, se pudo realizar todas las graficas, además de la animación o "gif" del retrato de fase. Esta evaluación se me hizo muy interesante, tanto la forma que se usa este lenguaje de programación, como el tema del que trato. Ver la animación del retrato de fase en 3D permite ver con mayor facilidad como evoluciona la posición con la velocidad y las planos coordenados complementan muy bien la información.

### 4 Bibliografía empleada

- Lorenz System. (26 de Abril de 2018). Recuperado de: [en.wikipedia.org/wiki/Lorenz\\_system](https://en.wikipedia.org/wiki/Lorenz_system)
- Animating the Lorenz Attractor with Python. (30 de Diciembre de 2016). Consultado el 26 de Abril de 2018. Recuperado de: [geoffboeing.com/2016/12/animating-lorenz-attractor-python/](http://geoffboeing.com/2016/12/animating-lorenz-attractor-python/)
- Visual Analysis of Nonlinear Dynamical Systems. (2016). Consultado el 26 de Abril de 2018. Recuperado de: [www.mdpi.com/2079-8954/4/4/37/htm#sec4-systems-04-00037](http://www.mdpi.com/2079-8954/4/4/37/htm#sec4-systems-04-00037)