Curvas de Lissajous

Trabajo Final - Informática

- **Facultad de Ciencias Exactas y Naturales - UNLPam**

- **Profesor:** Willging Pedro

- **Autor:** Delorenzi Tomás

- **Fecha:** 23/08/2023

Resúmen

Las curvas de Lissajous, resultado de la superposición de dos movimientos armónicos perpendiculares, han trascendido los confines de la física para influir de manera significativa en el arte. En el ámbito científico, estas curvas desempeñan un papel fundamental en la comprensión de fenómenos ondulatorios y resonancia, siendo utilizadas para analizar frecuencias y fases. No obstante, su atractiva y elegante simetría también ha cautivado a artistas visuales, inspirando creaciones que exploran la intersección entre las matemáticas y la estética, y demostrando cómo la belleza matemática puede trascender la disciplina y enriquecer nuestra apreciación del mundo que nos rodea. Con el objetivo de mostrar un codigo de Python sencillo y simple el usuario podrá interactuar para aprender como se forman estas distintas figuras.

Palabras Clave:

- Movimiento armónico simple
- Física
- Oscilaciones
- Python

Introducción

Las curvas de Lissajous, nombradas en honor al matemático francés Jules Antoine Lissajous (1822-1880), son patrones visuales que resultan de la interacción de dos oscilaciones armónicas perpendiculares entre sí. Estas intrigantes figuras geométricas, que datan del siglo XIX, han sido utilizadas en diversas disciplinas como la física, la ingeniería y la música para ilustrar la relación entre frecuencias y fases en sistemas vibracionales. Las curvas de Lissajous han perdurado como un ejemplo clásico de la belleza matemática y la conexión entre la ciencia y el arte.

Desarrollo

Veamos el movimiento de una particula que posee dos grados de libertad. Supongamos que esta partícula esta sujeta a una fuerza restauradora que es proporcional a la distancia de la particula al origen, donde ubicamos el centro de atracción, y que está dirigida hacia el mismo:

$$\overrightarrow{F} = -k \cdot \overrightarrow{r}$$

expresión que, en coordenadas polares, se descompone en

$$F_x = -krcos(\theta) = -kx$$

$$F_y = -krsen(\theta) = -ky$$

dando como resultado las siguientes ecuaciones de movimiento

$$\ddot{x} + \omega_0^2 x = 0$$

$$\ddot{y} + \omega_0^2 y = 0$$

donde definimos $\omega_0^2 = \frac{k}{m}$. Resolviendo estas ecuaciones diferenciales encontramos que las soluciones son

$$x(t) = A\cos(\omega_0 t + \alpha)$$

$$y(t) = B\,cos(\omega_0 t + \beta)$$

Se encuentra que el movimiento se compone de oscilaciones armonicas en cada una de las dos direcciones, de la misma frecuencia, pero probablemente de amplitues y fases iniciales diferentes (dependera de las condiciones iniciales en las que parta nuestro sistema).

En *general*, las oscilaciones bidimensionales no tienen por qué poseer la misma frecuencia en los movimientos según las direcciones **x** e **y**, de forma que nuestras soluciones encontradas se escriben ahora como

$$x(t) = A\cos(\omega_x t + \alpha)$$

$$y(t) = B\, cos(\omega_y t + \beta)$$

y la trayectoria deja de ser una elipse, sino una de las llamadas **curvas de Lissajous** (fueron demostradas por primera vez en 1857 por Jules Lissajous)

Estas curvas serán cuando el movimiento se repita sobre sí mismo a intervalos regulares de tiempo, lo cual sólo será posible cuando las frecuencias ω_x y ω_y sean $conmensurables$, o sea cuando $\frac{\omega_x}{\omega_y}$ sea una fraccion racional. En caso de que este cociente no sea una fracción racional, la curva será $abierta$; es decir, la partícula no pasara dos veces por el mismo punto a la misma velocidad.

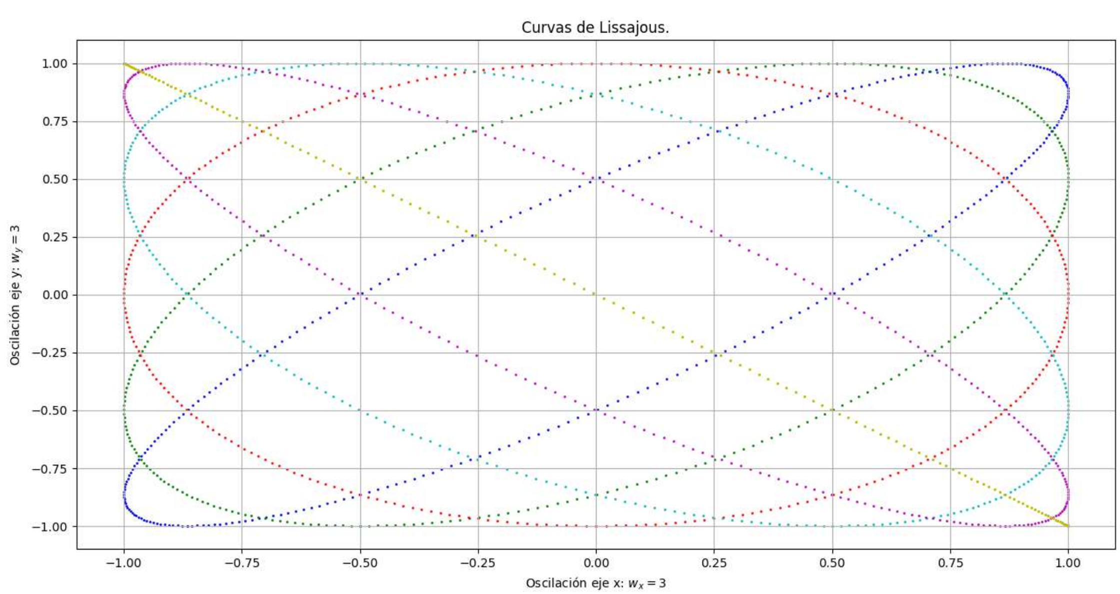
Es interesante señalar que el oscilador bidimensional constituye un modelo de sistema en el cual un cambio infinitesimal puede dar como resultado un tipo de movimiento cualitativamente distinto.

Cuando las pulsaciones de los movimientos en las direcciones e sean diferentes, la forma de la curva de Lissajous resultante depende mucho de la diferencia de fase $\delta \equiv \alpha - \beta$. A continuación se muestra el codigo.

In [ ]:
```
import matplotlib.pyplot as plt
from numpy import cos, pi, linspace

def plot_lissajous(ax, a, b, delta, valor_delta=None, color=None):
    t = linspace(-pi, pi, 300)
    x = cos(a * t + delta)
    y = cos(b * t)
    ax.scatter(x, y, c=color, s=1)

    if valor_delta is not None:
        ax.set_xlabel(f'Oscilación eje x: $w_x = {a}$')
        ax.set_ylabel(f'Oscilación eje y: $w_y = {b}$')
    else:
        ax.set_xlabel(f'Oscilación eje x: $w_x = {a}$, $\delta = {delta:.2f}$')
        ax.set_ylabel(f'Oscilación eje y: $w_y = {b}$')

# Opciones del menú
print("1. Ver figuras con deltas predefinidos")
print("2. Ingresar un delta específico")
print("3. Ingresar relación de frecuencias (a:b)")
print("4. Salir")
```

```python
while True:
    opcion = input("Ingrese el número de la opción deseada: ")

    if opcion == "1":
        # Deltas predefinidos y colores para cada curva
        deltas = [pi/6, pi/3, pi/2, pi*2/3, pi*5/6, pi]
        colors = ['b', 'g', 'r', 'c', 'm', 'y']
        fig, ax = plt.subplots()
        for i in range(len(deltas)):
            delta = deltas[i]
            color = colors[i]
            plot_lissajous(ax, 3, 3, delta, valor_delta=delta, color=color)
        plt.grid(True)
        plt.title(f'Curvas de Lissajous.')
        plt.show()
    elif opcion == "2":
        delta_usuario = float(input("Ingrese el valor del delta: "))
        fig, ax = plt.subplots()
        plot_lissajous(ax, 3, 3, delta_usuario, valor_delta=delta_usuario)
        plt.grid(True)
        plt.title(f'Curvas de Lissajous. $\delta = {delta_usuario:.4f}$')
        plt.show()
    elif opcion == "3":
        a = float(input("Ingrese el valor de 'a': "))
        b = float(input("Ingrese el valor de 'b': "))
        delta = float(input("Ingrese el valor del delta: "))
        fig, ax = plt.subplots()
        plot_lissajous(ax, a, b, delta)
        plt.grid(True)
        plt.title(f'Curvas de Lissajous.')
        plt.show()
    elif opcion == "4":
        print("Saliendo...")
        break
    else:
        print("Opción inválida. Por favor, seleccione una opción válida.")
```

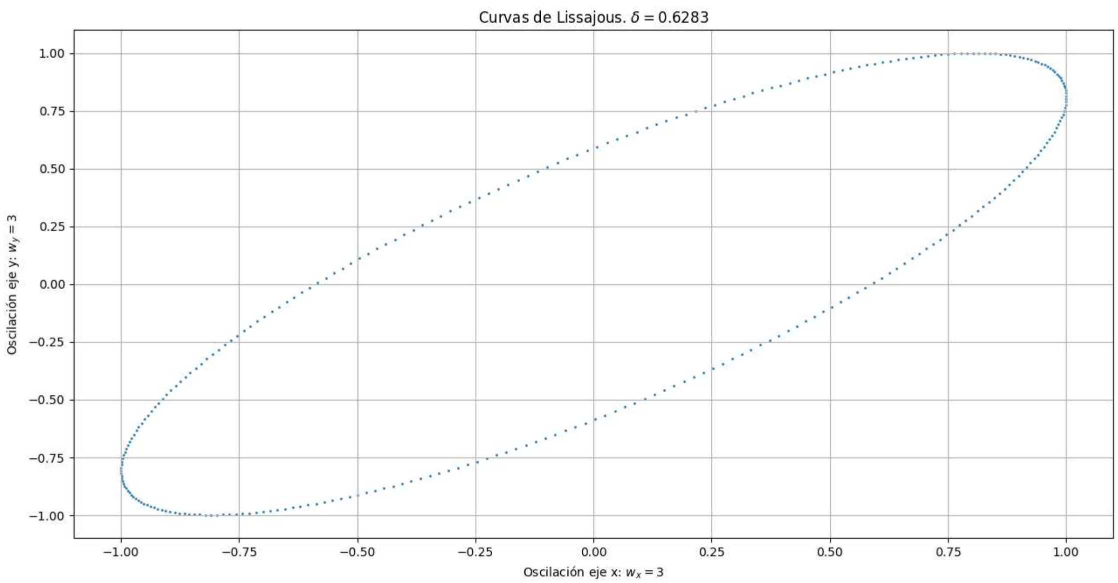
Opcion 1: Patrones predefinidos.



Opcion 2: Permitir al usuario elegir el desfase entre ambas ondas

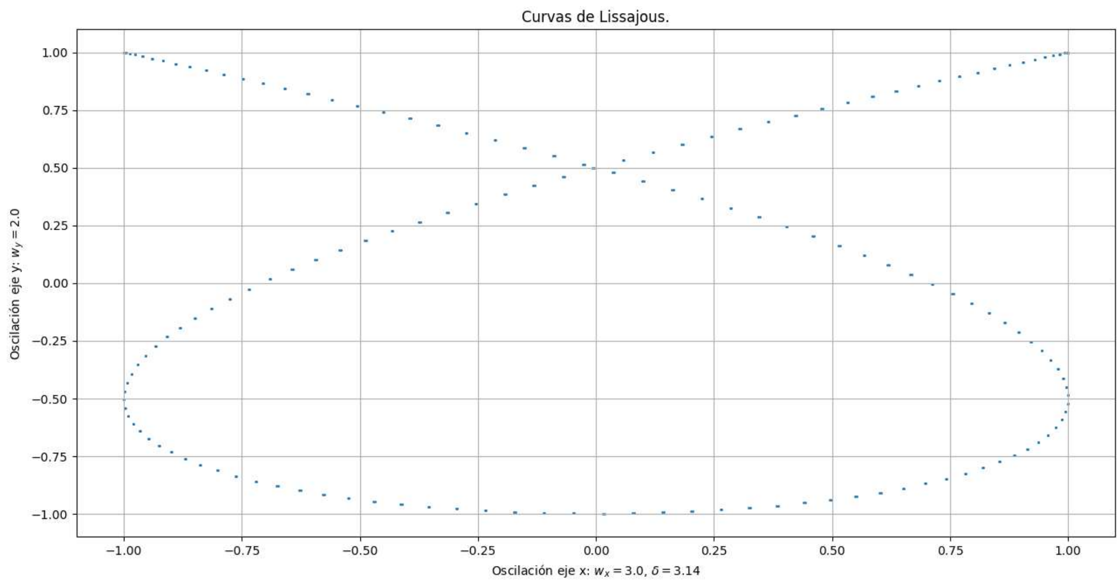
($\delta = \pi/5 \approx 0.6283$)

```
1. Ver figuras con deltas predefinidos
2. Ingresar un delta específico
3. Ingresar relación de frecuencias (a:b)
4. Salir
Ingrese el número de la opción deseada: 2
Ingrese el valor del delta: 0.6283
Ingrese el número de la opción deseada: 4
Saliendo...
```



Opcion 3: Permitir que el usuario escoja ω_x , ω_y y δ entre ambas ondas

```
1. Ver figuras con deltas predefinidos
2. Ingresar un delta específico
3. Ingresar relación de frecuencias (a:b)
4. Salir
Ingrese el número de la opción deseada: 3
Ingrese el valor de 'a': 3
Ingrese el valor de 'b': 2
Ingrese el valor del delta: 3.14
Ingrese el número de la opción deseada: 4
Saliendo...
```



Aplicaciones de las curvas de Lissajous

Física

En física, las curvas de Lissajous se utilizan para estudiar la relación entre dos ondas que se propagan en direcciones perpendiculares. Por ejemplo, en un osciloscopio, se pueden visualizar las curvas de Lissajous para estudiar la relación entre la amplitud y la frecuencia de una onda eléctrica y una onda magnética. Esta técnica es muy útil en el estudio de circuitos eléctricos y en la caracterización de señales eléctricas.

Música

Las curvas de Lissajous se utilizan en la creación de efectos visuales para conciertos y espectáculos de música electrónica. Al mostrar la relación entre las frecuencias de distintos sonidos, se pueden generar patrones complejos y sincronizados con la música. Estos patrones pueden ser utilizados para crear una experiencia visual única y enriquecedora para el público.

Diseño

En el diseño gráfico y de patrones, las curvas de Lissajous son muy útiles para crear diseños abstractos, simétricos y dinámicos. Las formas geométricas que se generan pueden ser utilizadas como base para la creación de patrones, diseños de logotipos y otros elementos visuales. Además, las curvas de Lissajous también se utilizan en la animación, donde pueden ser animadas para crear efectos visuales interesantes.

Matemáticas

En matemáticas, las curvas de Lissajous se utilizan en la geometría diferencial y en la teoría de las superficies mínimas. Por ejemplo, se pueden utilizar para construir superficies de revolución simétricas alrededor de un eje. Estas superficies son importantes en la física teórica y en la teoría de las cuerdas.

Ingeniería

En ingeniería, las curvas de Lissajous se utilizan en la ingeniería eléctrica para estudiar las relaciones entre la tensión y la corriente en los circuitos eléctricos. También se utilizan en la ingeniería mecánica para el diseño de máquinas que producen movimiento armónico. Además, las curvas de Lissajous también son utilizadas en la ingeniería acústica para estudiar la relación entre la fase y la amplitud de las ondas sonoras.

Arte

Las curvas de Lissajous se utilizan en la creación de arte generativo, donde las formas y patrones son generados por algoritmos y ecuaciones matemáticas. Estos algoritmos pueden ser utilizados para crear diseños complejos y dinámicos que cambian con el tiempo o en respuesta a las interacciones del usuario. El arte generativo es un campo en auge, y las curvas de Lissajous son una herramienta valiosa para los artistas que trabajan en este ámbito.

Opcion 3: Caso especial

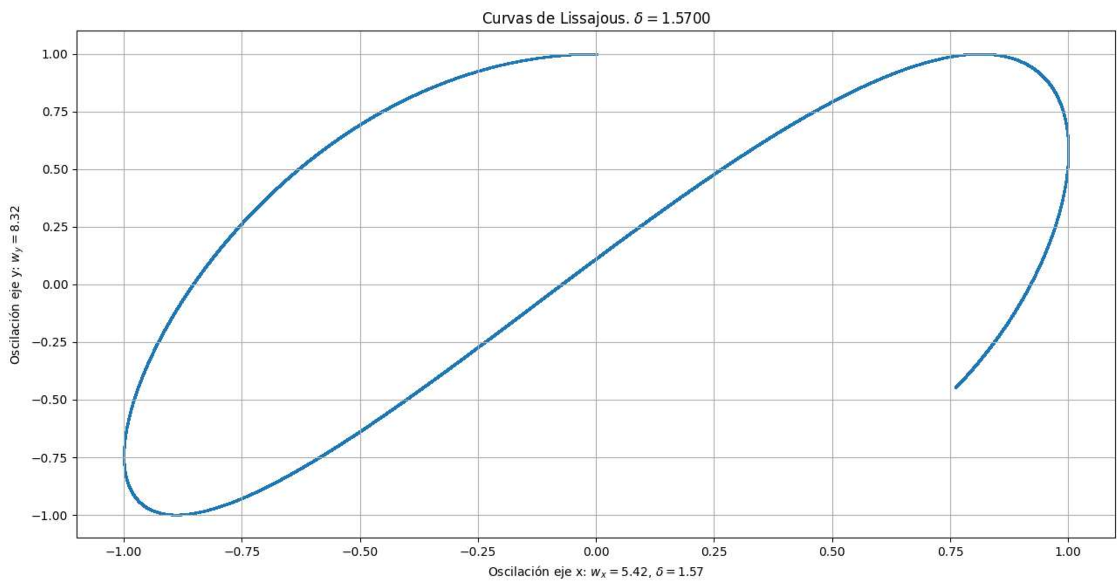
Vamos a tener una curva abierta cuando el cociente de ambas frecuencias no sea una fraccion racional. Asi vemos que si dejamos correr el tiempo (para esto he modificado el intervalo de tiempo en el que el programa toma valores para graficar los puntos de ambos cosenos) se ira pintando absolutamente todo el espacio. Lo que equivale a decir que nuestra particula pasara por cada punto del espacio si le damos el tiempo suficiente.

Código levemente cambiado

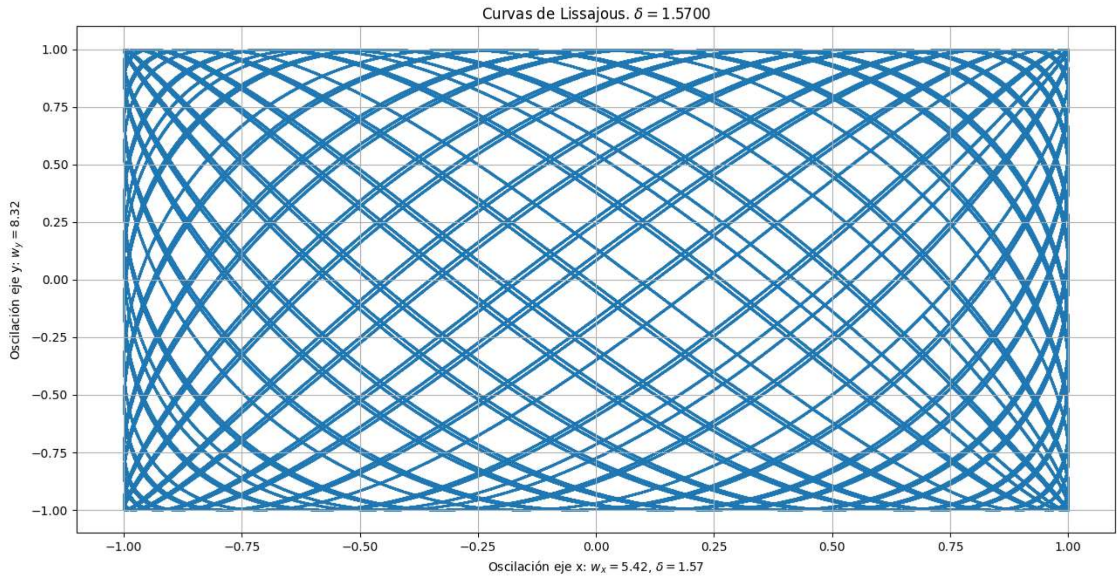
In [ ]:
```
import matplotlib.pyplot as plt
from numpy import cos, pi, linspace

def plot_lissajous(ax, a, b, delta, valor_delta=None, color=None):
    t = linspace(0, 1, 30000) # en este caso pongo que el intervalo de tiempo sera de 1 segundo.
    x = cos(a * t + delta)
    y = cos(b * t)
    ax.scatter(x, y, c=color, s=1)

    if valor_delta is not None:
        ax.set_xlabel(f'Oscilación eje x: $w_x = {a}$')
        ax.set_ylabel(f'Oscilación eje y: $w_y = {b}$')
    else:
        ax.set_xlabel(f'Oscilación eje x: $w_x = {a}$, $\delta = {delta:.2f}$')
        ax.set_ylabel(f'Oscilación eje y: $w_y = {b}$')

# Opciones del menú
print("1. Ver figuras con deltas predefinidos")
print("2. Ingresar un delta específico")
print("3. Ingresar relación de frecuencias (a:b)")
print("4. Salir")

while True:
    opcion = input("Ingrese el número de la opción deseada: ")

    if opcion == "1":
        # Deltas predefinidos y colores para cada curva
        deltas = [pi/6, pi/3, pi/2, pi*2/3, pi*5/6, pi]
        colors = ['b', 'g', 'r', 'c', 'm', 'y']
        fig, ax = plt.subplots()
        for i in range(len(deltas)):
            delta = deltas[i]
            color = colors[i]
            plot_lissajous(ax, 3, 3, delta, valor_delta=delta, color=color)
        plt.grid(True)
        plt.title(f'Curvas de Lissajous.')
        plt.show()
    elif opcion == "2":
        delta_usuario = float(input("Ingrese el valor del delta: "))
        fig, ax = plt.subplots()
        plot_lissajous(ax, 3, 3, delta_usuario, valor_delta=delta_usuario)
        plt.grid(True)
        plt.title(f'Curvas de Lissajous. $\delta = {delta_usuario:.4f}$')
```

```
        plt.show()
    elif opcion == "3":
        a = float(input("Ingrese el valor de 'a': "))
        b = float(input("Ingrese el valor de 'b': "))
        delta = float(input("Ingrese el valor del delta: "))
        fig, ax = plt.subplots()
        plot_lissajous(ax, a, b, delta)
        plt.grid(True)
        plt.title(f'Curvas de Lissajous.')
        plt.show()
    elif opcion == "4":
        print("Saliendo...")
        break
    else:
        print("Opción inválida. Por favor, seleccione una opción válida.")
```

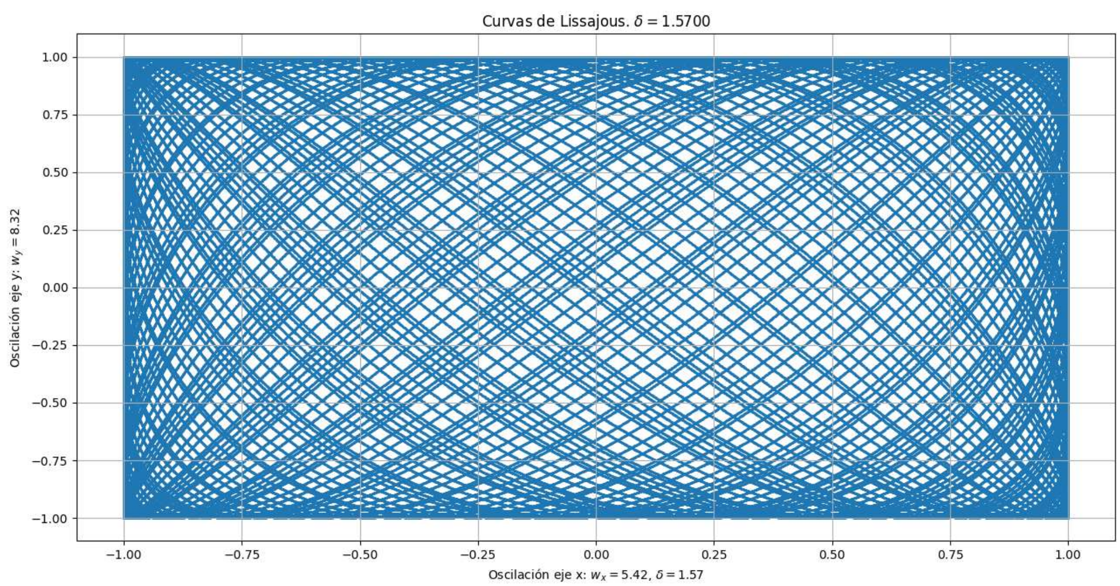
(0-1) seg



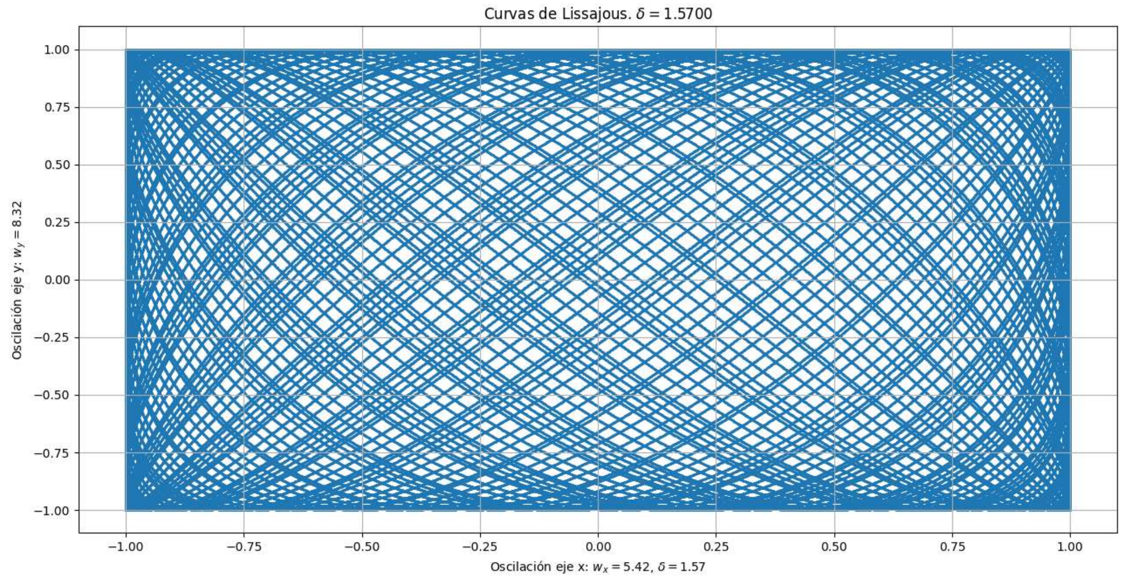
(0,30) seg



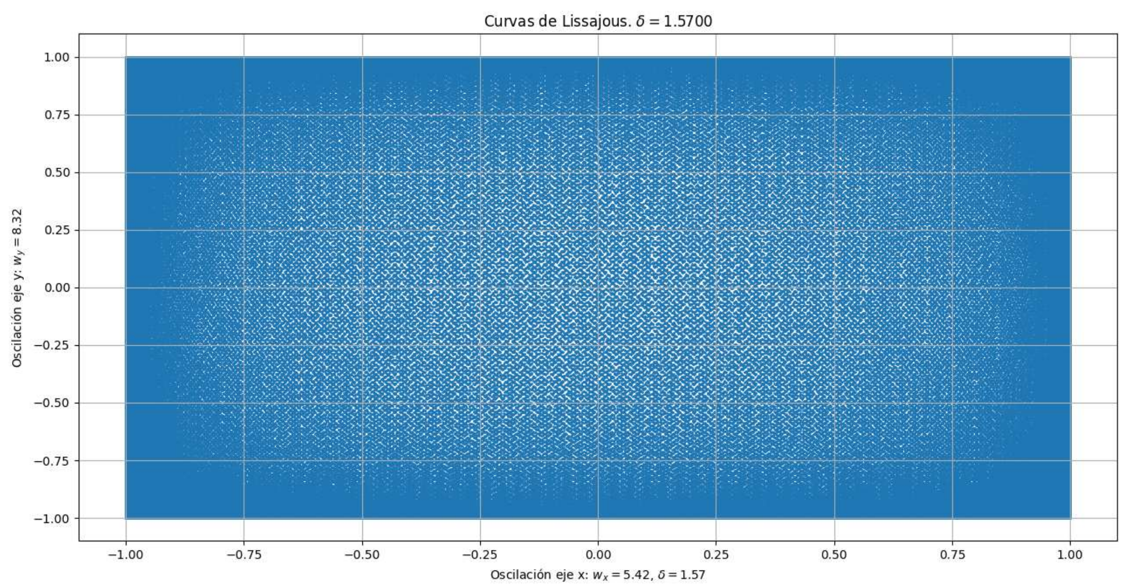
(0-60) seg



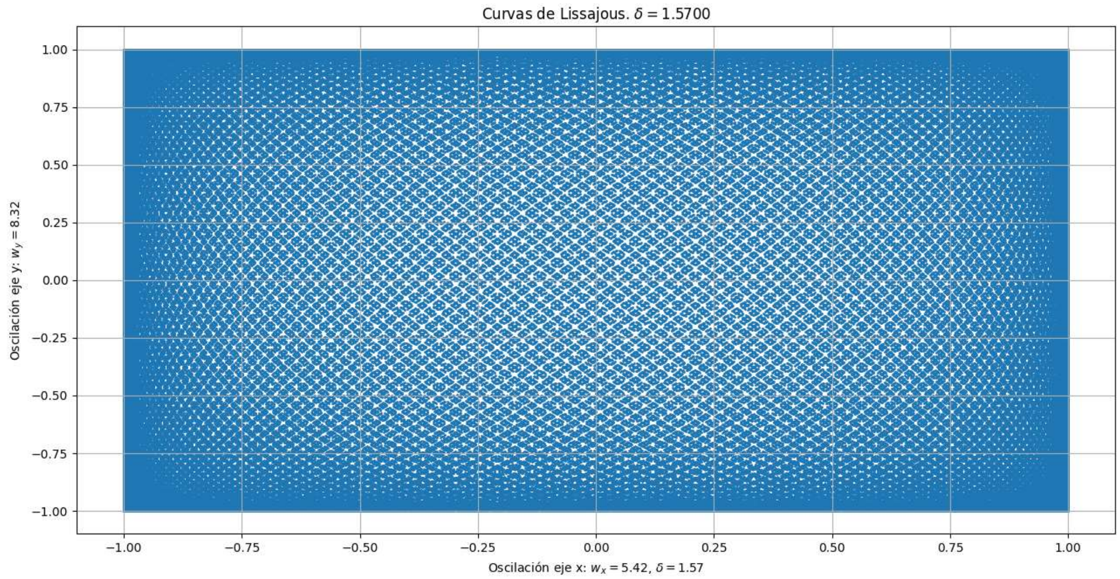
(0-1800) seg



(0-3600) seg



(0-86400) seg



Bibliografía

1. Marion J.B., Thornton S. (1991), Classical dynamics of particles and systems, Ed. Harcourt Brace Jocanovich, quinta edición.
2. https://matplotlib.org/devdocs/index.html