**1**

Como la parte real solo depende de los cosenos y la parte imaginaria solo depende de los senos Podemos escribir la siguiente expresión

**2**

A partir de la fórmula hallada previamente para describir la gráfica de cada epiciclo, podemos relacionar a cada Radio como la suma de componente real de Ck y otra parte imaginaria de Ck.

La fórmula que nos permitirá graficar la imagen deseada será:

Si tomamos como datos de entrada los Radios Rk provistos por el archivo txt, a N igual a la cantidad total de datos y ω0 como la frecuencia fundamental

* Rk = [598.165, 173.078, 60.656, 118.358, 33.712, 37.211, 1.909……..]
* ω*k* = ω0 k =
* N = 1000

El código en python que nos permite graficar la imagen es el siguiente:

import numpy as np

import matplotlib.pyplot as plt

def epiciclo\_sum(R, omega\_0, k):

suma = 0

for Ri, omegai in zip(R, omega\_0):

suma += np.sum([Ri \* (np.cos(omegai \* k) + 1j \* np.sin(omegai \* k))])

return suma

N = 1000

# Definir los valores para Ri y ωi

R = [598.165, 173.078, 60.656, 118.358, 33.712, 37.211,.......] # Radio de cada epiciclo

omega\_0 = [(2 \* np.pi)/N] # Frecuencia de cada epiciclo

# Definir el rango de tiempo

k = np.linspace(0, N, N)

# Calcular las coordenadas (x, y) para cada instante de tiempo k

coords = [epiciclo\_sum(R, omega\_0, ki) for ki in k]

# Obtener las coordenadas x e y por separado

x = np.real(coords)

y = np.imag(coords)

# Graficar

plt.figure(figsize=(6, 6))

plt.plot(x, y, label="Epiciclo")

plt.xlabel('x')

plt.ylabel('y')

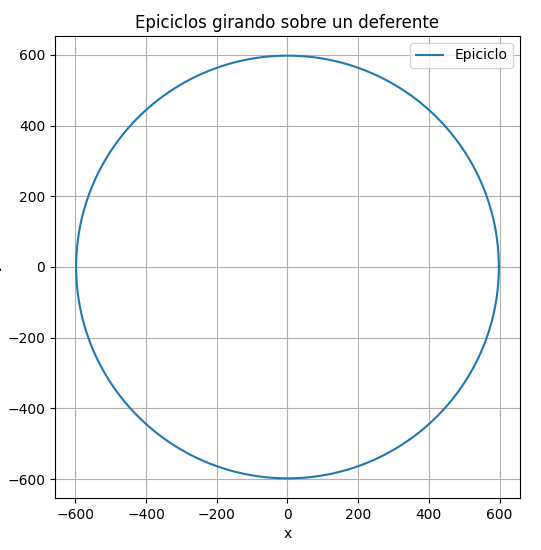
plt.title('Epiciclos girando sobre un deferente')

plt.axis('equal')

plt.grid()

plt.legend()

plt.show()

La imagen resultante es un círculo de radio pero sin los

giros que describen los epiciclos. Esto se debe a que el ω0 es

siempre el mismo en cada iteración k y generando esa única imagen.

Dicha figura está representada en una animación llamada **“Animación-epiciclo-01”** hecha en geogebra que se encuentra adjunta entre los archivos entregados.

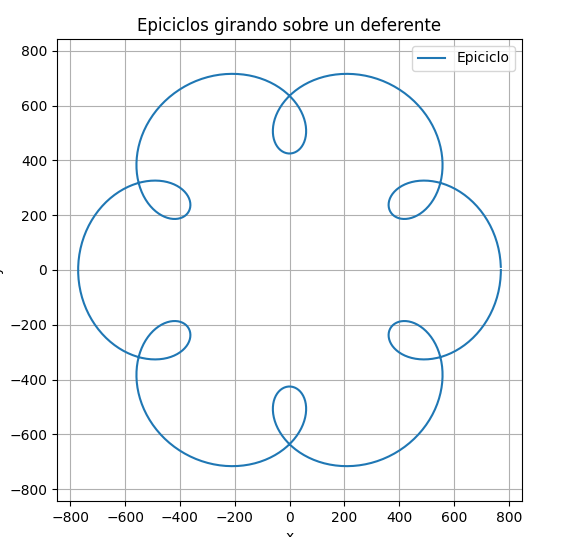
Ahora si quisiéramos que el ω0 escale entre diferentes valores, podemos escalar este valor por una constante que altere esta frecuencia en cada valor k.

Si por ejemplo ahora alteramos el código: omega\_0 = [(2 \* np.pi)/N]

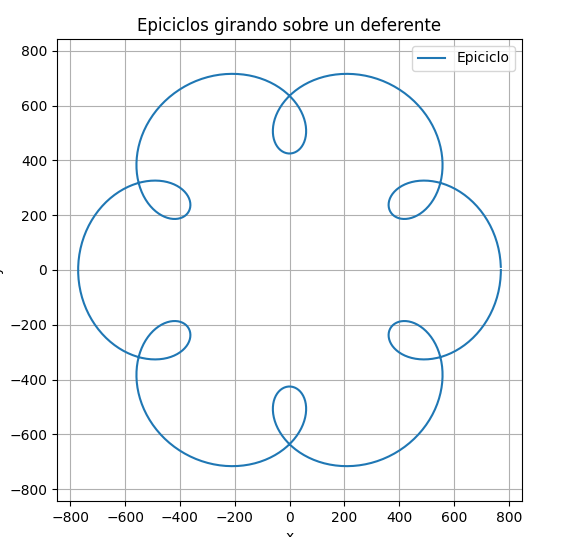
Por: omega\_0 = [(2 \* np.pi)/N,((2 \* np.pi)/N)\*7]

Ahora el ω0 irá alternando entre los valores y \*7

Ejecutando el archivo en python nuevamente con esta modificación obtenemos el siguiente resultado:



3)



En esta imagen resultante, podemos observar como existe una relación entre la constante por la cual se multiplicó el ω0 y los epiciclos generados.

El **7** que multiplica, provoca un aumento en la frecuencia por lo cual, un radio girara más rápido que el otro y por lo tanto generando la figura.

Además, el factor por el cual se multiplica, también guarda relación con el Nº de epiciclos que se originan (en este caso son 6) dado que en lo que el círculo grande realiza un periodo, el de menor radio realiza **7**.

Se puede llegar a la conclusión que con esta formula si necesitamos graficar N epiciclos, el factor por el cual se debe multiplicar el ω0 es N+1

Así si por ejemplo si queremos 20 epiciclos el factor a multiplicar es 21 obteniendo el resultado

