Escuela Politécnica Nacional

Métodos Numéricos

Paralelo: GR1CC

Proyecto

Segundo Bimestre

Tema: Nanodron

David Arciniegas
Ángel Falcón
Anthony Goyes
Ozzy Loachamín

2024 – A

Índice

1. Objetivos

- Desarrollar una simulación visual de la trayectoria de un nanodron mediante la solución de un sistema de ecuaciones diferenciales no lineales.
- Implementar un modelo matemático utilizando el método de Euler para resolver las ecuaciones que describen el movimiento del nanodron.
- Crear una interfaz gráfica interactiva que permita al usuario ajustar parámetros clave del sistema (α, β, γ) y observar los efectos en la trayectoria del nanodron.

2. Introducción

En este proyecto, se presenta el desarrollo de una simulación para estudiar el comportamiento dinámico de un nanodron utilizando un sistema de ecuaciones diferenciales no lineales, presentes en las ecuaciones (1), (2) y (3). Estas ecuaciones capturan la relación entre diferentes variables que afectan el movimiento del nanodron en un espacio tridimensional. El objetivo principal es proporcionar una herramienta visual interactiva que permita analizar cómo diferentes parámetros influyen en la trayectoria del nanodron. Para lograr esto, se ha implementado un método numérico (Euler) que resuelve las ecuaciones diferenciales, y se ha desarrollado una interfaz gráfica para facilitar la interacción del usuario con el sistema [1].

$$\frac{dx}{dt} = \alpha(y - x) \tag{1}$$

$$\frac{dy}{dt} = x(\beta - z) - y \tag{2}$$

$$\frac{dz}{dt} = xy - \gamma z \tag{3}$$

Donde x, y y z representan las coordenadas espaciales del nanodron, y α, β y γ son parámetros ajustables que afectan la dinámica del sistema. El sistema que describe el movimiento del nanodron es el *Sistema de Lorenz* (también conocido como Atractor de Lorenz) [2], el mismo que está compuesto por ecuaciones diferenciales ordinarias (EDOs) no lineales que tiene aplicación en el modelamiento de sistemas caóticos. Se trata de un sistema dinámico determinista tridimensional no lineal.

Este conjunto de ecuaciones diferenciales describen el movimiento de un fluido en convección, donde α, β y γ son parámetros que afectan el comportamiento del sistema. Lorenz desarrolló este modelo matemático para describir el movimiento de las masas de aire en la atmósfera, partiendo de una versión simplificada de las ecuaciones de convección térmica derivadas de las ecuaciones atmosféricas. Al estudiar los patrones climáticos, Lorenz comenzó a observar que estos no siempre cambiaban como se había previsto [3].

2.1. Sistema Dinámico

Un sistema dinámico es una forma de modelar cómo cambian las variables a lo largo del tiempo. En lugar de simplemente mirar un valor en un instante dado, se estudia cómo evolucionan los valores de las variables **a lo largo del tiempo**. En otras palabras, se trata de un conjunto de reglas que describen cómo cambian los estados del sistema con el tiempo.

2.2. Sistema Determinista

Un sistema determinista es uno en el que, dado un estado inicial, **el futuro del sistema está completamente determinado** y **no hay azar** involucrado. En otras palabras, si conoces exactamente el estado inicial del sistema, puedes predecir su comportamiento futuro con precisión. No hay incertidumbre o aleatoriedad en la evolución del sistema.

2.3. Sistema No Lineal

El sistema de Lorenz es no lineal porque las ecuaciones que describen cómo cambian las variables incluyen términos no lineales. En otras palabras, las ecuaciones contienen productos de las variables y no solo sumas o restas simples. Esto introduce complejidades adicionales en el comportamiento del sistema, como la posibilidad de caos.

Dado el sistema presentado anteriormente, ecuaciones (1), (2) y (3); el sistema es no lineal debido a las ecuaciones (2) y (3). Esto se debe a que se encuentra el producto de las variables xz y xy, respectivamente. Mientras que en un sistema lineal cada término debería involucrar solo a una de las variables o una derivada de ellas. Al tener estos productos se introduce una complejidad adicional que hace que el sistema no sea lineal.

2.4. Sistema Caótico

El sistema de Lorenz es un excelente ejemplo de un sistema dinámico que tiene un comportamiento caótico. Edward Norton Lorenz fue un matemático y meteorólogo estadounidense que tras realizar simulaciones numéricas en su computador en 1961, utilizó como entrada el dato 0,506 como una aproximación del dato 0,506127 y obtuvo dos escenarios climáticos completamente diferentes.

Entonces, se dice que un sistema es caótico si muestra algunas propiedades clave, entre ellas:

- **Sensibilidad a las condiciones iniciales:** Esto significa que pequeñas diferencias en el estado inicial del sistema pueden llevar a diferencias muy grandes en el comportamiento a largo plazo. En otras palabras, el sistema es extremadamente sensible a sus condiciones iniciales. En el sistema de Lorenz, esto se manifiesta en las trayectorias que divergen rápidamente.

- **Comportamiento no periódico:** Aunque el sistema de Lorenz es determinista, su comportamiento no se repite periódicamente, no parece seguir un patrón repetitivo regular.
- **Estructura fractal:** El atractor de Lorenz tiene una estructura fractal, lo que significa que su geometría es auto-similar a diferentes escalas. Esto es característico de sistemas caóticos y resalta la complejidad del comportamiento a medida que se examina en escalas cada vez más pequeñas.

Las constantes α, β y γ juegan un papel importante en el comportamiento del sistema y en la aparición del caos.

- **Efecto de α:** Controla cómo rápido el nanodron ajusta su posición entre dos ejes x y y.
  - Si α es alto, el nanodron ajusta rápidamente su posición. Esto puede hacer que su vuelo sea más estable y menos caótico.
  - Si α es bajo, el ajuste es más lento. Esto puede hacer que el vuelo sea más impredecible y caótico.
- **Efecto de β:** Afecta cómo el nanodron reacciona a una tercera dirección z, en combinación con la posición x.
  - Un valor alto de β puede hacer que el vuelo del nanodron sea más inestable y complicado.
  - Un valor bajo de β puede hacer que el vuelo sea más predecible y ordenado.
- **Efecto de γ:** Controla cuánto “pierde” el nanodron en la dirección z, es como si el nanodron estuviera perdiendo energía en esa dirección.
  - Si γ es alto, el nanodron pierde rápidamente energía en la dirección z. Esto puede ayudar a estabilizar el vuelo.
  - Si γ es bajo, el nanodron retiene más energía, lo que puede hacer que el vuelo sea más caótico y complicado.

2.5. Condiciones Iniciales

los valores específicos de los parámetros α, β y γ en el sistema de Lorenz pueden provocar cambios significativos en el comportamiento del sistema, y hay valores que son particularmente conocidos por su influencia en la dinámica del sistema.

- **Caso 01:** $\alpha = 0{,}1$, $\beta = 0{,}1$, $\gamma = 0{,}1$

  Cuando los parámetros α, β y γ se fijan en valores bajos como 0,1, el sistema tiende a comportarse de manera diferente en comparación con los valores clásicos.

  Comportamiento Esperado:

Estabilidad: Con estos valores bajos, el sistema puede no ser lo suficientemente inestable como para exhibir caos. En lugar de eso, podría mostrar un comportamiento más estable o incluso un comportamiento periódico, dependiendo de las interacciones entre los parámetros.

Dinámica: La dinámica del sistema puede ser más simple y menos caótica porque los valores bajos de α y β no inducen suficiente inestabilidad, y el bajo valor de γ reduce la disipación.

- **Caso 02:** $\alpha = 10$, $\beta = 28$, $\gamma = 8/3$

  Estos valores son los clásicos utilizados en el sistema de Lorenz, conocidos por generar el famoso atractor de Lorenz y mostrar comportamiento caótico.

  Comportamiento Esperado:

  Caos: Estos valores están en el rango que típicamente induce caos en el sistema. Con estos parámetros, el sistema exhibe trayectorias complejas y erráticas que llenan un conjunto fractal en el espacio de fases.

  Atractor de Lorenz: Bajo estos parámetros, el sistema produce el atractor de Lorenz, que es conocido por su estructura fractal y la forma característica de "mariposa", como se observa en la (fig. 1). Este atractor muestra cómo pequeñas variaciones en las condiciones iniciales pueden llevar a trayectorias completamente diferentes a lo largo del tiempo.

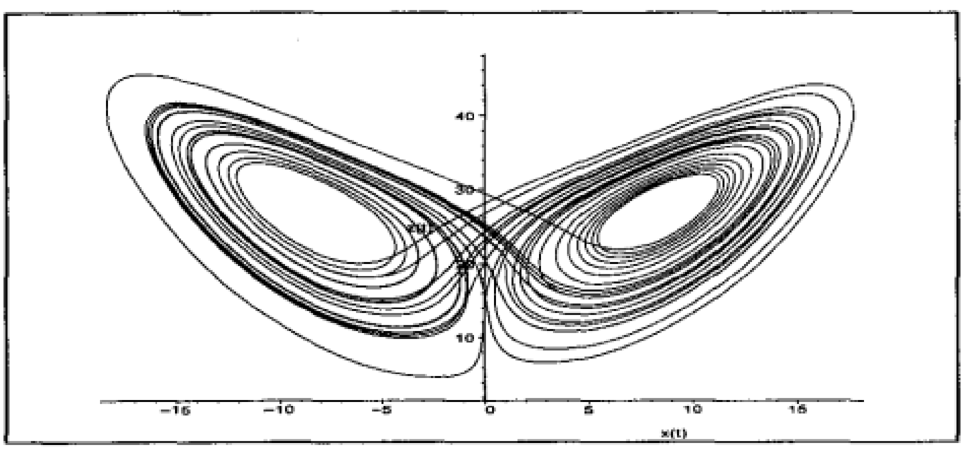

Figura 1: Mariposa de Lorenz [2].

2.6. Método de Euler

El método de Euler es uno de los métodos numéricos más sencillos para la resolución de EDOs. Se basa en la aproximación de la solución de la ecuación diferencial mediante una serie de pasos discretos. Dado un sistema de la forma:

$$\frac{dy}{dt} = f(\mathbf{y}, t) \tag{4}$$

Donde $\mathbf{y}(t)$ es un vector de funciones y $f(\mathbf{y}, y)$ define cómo cambia $\mathbf{y}$ con el tiempo, el método de Euler aproxima $\mathbf{y}(t)$ en intervalos discretos. El método de Euler se utiliza para aproximar la solución de estas ecuaciones de la siguiente manera:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \cdot f(y_n, t_n, \alpha, \beta, \gamma) \tag{5}$$

donde $\mathbf{y}_n = [x_n, y_n, z_n]$ es el vector de estado en el instante t_n, Δt tamaño del paso, y f es la función que describe la evolución del sistema.

3. Metodología

3.1. Descripción de la Solución

La solución propuesta consiste en modelar el movimiento del nanodron utilizando un sistema de ecuaciones diferenciales del tipo Lorenz, donde los parámetros (α, β, γ) representan constantes que afectan el comportamiento del sistema. El método de Euler fue seleccionado para resolver numéricamente estas ecuaciones debido a su simplicidad y eficiencia computacional en este contexto. Una vez calculadas las trayectorias, se utiliza la biblioteca Matplotlib para representar visualmente el movimiento en 3D.

3.2. Desarrollo Matemático

El modelo del nanodron está basado en un sistema de ecuaciones diferenciales del tipo "Sistema de Lorenz", ecuaciones (1), (2) y (3); por tal motivo, son difíciles de resolver y a menudo no tienen soluciones analíticas generales. Entonces, es necesario implementar métodos numéricos para estudiar el comportamiento del sistema. Estos métodos permiten aproximar las soluciones a partir de condiciones iniciales y parámetros específicos, pero no proporcionan una solución exacta. En este caso, se implementó el **método de Euler** (ecu. 5).

Por tal motivo, hay que seguir los siguientes pasos, se definió el estado inicial $\mathbf{y}_0 = [x_0, y_0, z_0]$, donde $\mathbf{y}_0$ representa las posiciones iniciales del nanodron en los tres ejes. Luego, se calculó los siguientes valores empleando las ecuaciones diferenciales para cada intercalo t_n, se calculó su $\mathbf{y}_{n+1}$ de la siguiente manera.

$$x_{n+1} = x_n + \Delta t \cdot \alpha(y_n - x_n) \tag{6}$$

$$y_{n+1} = y_n + \Delta t \cdot (x_n(\beta - z_n) - y_n) \tag{7}$$

$$z_{n+1} = z_n + \Delta t \cdot (x_n y_n - \gamma z_n) \tag{8}$$

Aquí, cada ecuación de la derecha representa cómo evoluciona cada coordenada del sistema en función de las anteriores.

Ahora, el siguiente paso es iterar sobre todos los pasos del tiempo. Por lo que, se repite para todos los valores de t_n, generando así una trayectoria aproximada para el nanodron en el espacio 3D.

3.3. Diagrama de Flujo/Pseudocódigo

La lógica del pseudocódigo sería la siguiente:

- Inicializar condiciones del sistema $(\alpha, \beta, \gamma, y_0, t)$
- Definir la función del modelo del nanodron.
- Implementar el método de Euler para resolver el sistema de ecuaciones diferenciales.
- Configurar la visualización en 3D de las trayectorias calculadas.
- Crear la interfaz gráfica con campos para la entrada de parámetros y un botón para ejecutar la simulación.
- Ejecutar la simulación y mostrar los resultados en una gráfica animada.

Se obtiene el siguiente pseudocódigo:

```
Definir función nanodrone_model(estado, tiempo, alfa, beta,
   gamma):
        Calcular dxdt, dydt, dzdt usando alfa, beta, gamma
        Retornar [dxdt, dydt, dzdt]

    Definir función euler_method(modelo, condiciones_iniciales
       , intervalo_tiempo, parametros):
        Inicializar matriz de soluciones
        Para cada paso en el intervalo de tiempo:
            Calcular el cambio en el estado usando la función
               nanodrone_model y el método de Euler
            Actualizar el estado
        Retornar matriz de soluciones

    Función run_simulation:
        Intentar:
            Leer y convertir alfa, beta, gamma,
               tiempo_simulación
            Definir condiciones iniciales
            Definir intervalo de tiempo
            Calcular soluciones usando el método de Euler
            Configurar la figura para la animación 3D
            Inicializar líneas y proyecciones para
               trayectorias
            Añadir puntos iniciales
            Configurar gráficos 3D y 2D
            Crear y mostrar la animación
        Capturar error:
            Mostrar mensaje de error

    Configurar interfaz gráfica:
```

```
        Crear ventana principal
        Añadir etiquetas y campos de entrada para parámetros
        Añadir botón para ejecutar simulación
        Configurar evento de clic para llamar a run_simulation
        Ejecutar bucle principal de la interfaz gráfica

Finalizar
```

Luego, se realizó la implementación en código. El método de euler se incorporó en la función `euler_method`, que toma como entrada la función que define el sistema (en este caso, `nanodrone_model`), las condiciones iniciales, el vector de tiempos, y los parámetros del sistema:

```
def euler_method(f, y0, t, args=()):
    n = len(t)
    y = np.zeros((n, len(y0)))
    y[0] = y0
    for i in range(1, n):
        dt = t[i] - t[i - 1]
        dy = np.array(f(y[i - 1], t[i - 1], *args))
        y[i] = y[i - 1] + dt * dy
    return y
```

3.4. Detalles importantes de la Implementación

La implementación se realiza en Python utilizando las bibliotecas numpy para el cálculo numérico, matplotlib para la visualización, y tkinter para la interfaz gráfica.

Se incluyen mecanismos de validación de entrada en la interfaz para asegurar que los parámetros (α, β, γ) ingresados sean numéricos.

La simulación permite comparar las trayectorias de dos posiciones iniciales distintas, A y B, para ilustrar la sensibilidad del sistema a las condiciones iniciales.

El sistema gráfico utiliza la animación para mostrar la evolución de las trayectorias en tiempo real, lo que permite una comprensión más intuitiva del comportamiento dinámico del nanodron.

4. Resultados

La simulación gráfica proporciona una representación visual clara del comportamiento del nanodron bajo diferentes configuraciones de (α, β, γ). Las trayectorias calculadas muestran cómo cambios en estos parámetros afectan la estabilidad y la forma de las órbitas. La herramienta desarrollada facilita la exploración interactiva de estos efectos, permitiendo al usuario experimentar con distintos valores y observar las respuestas del sistema en tiempo real.

Algunos de los resultados obtenidos se muestran a continuación. Se planteó el primer caso, en donde se asignan los valores $\alpha = 0{,}1$, $\beta = 0{,}1$, $\gamma = 0{,}1$. Se obtuvo como resultado la simulación presente en la (fig. 3).

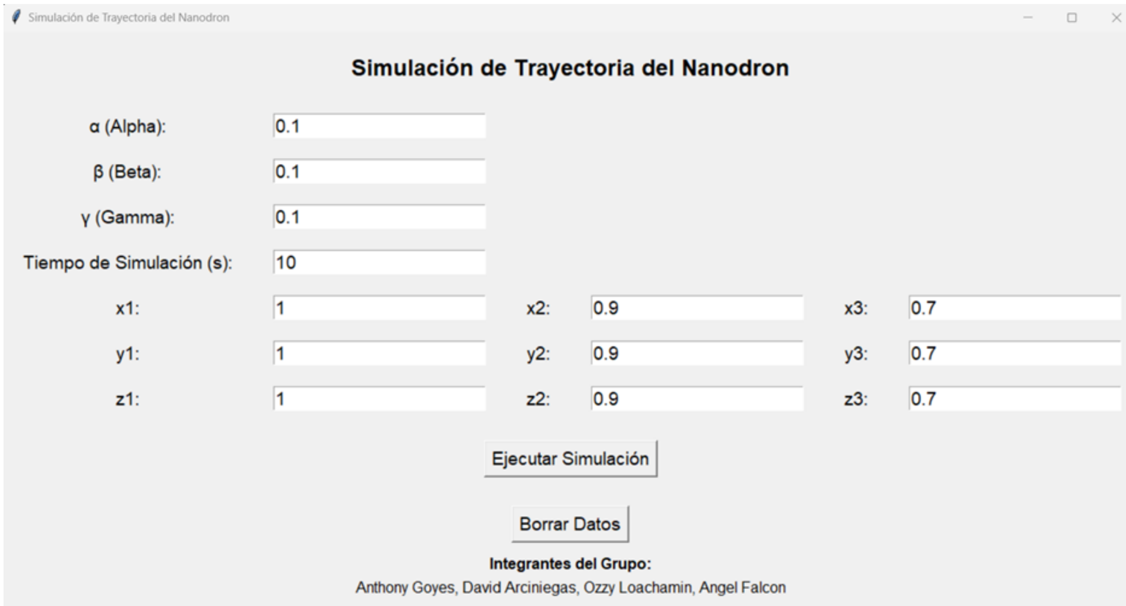

Figura 2: Interfaz del caso 01.

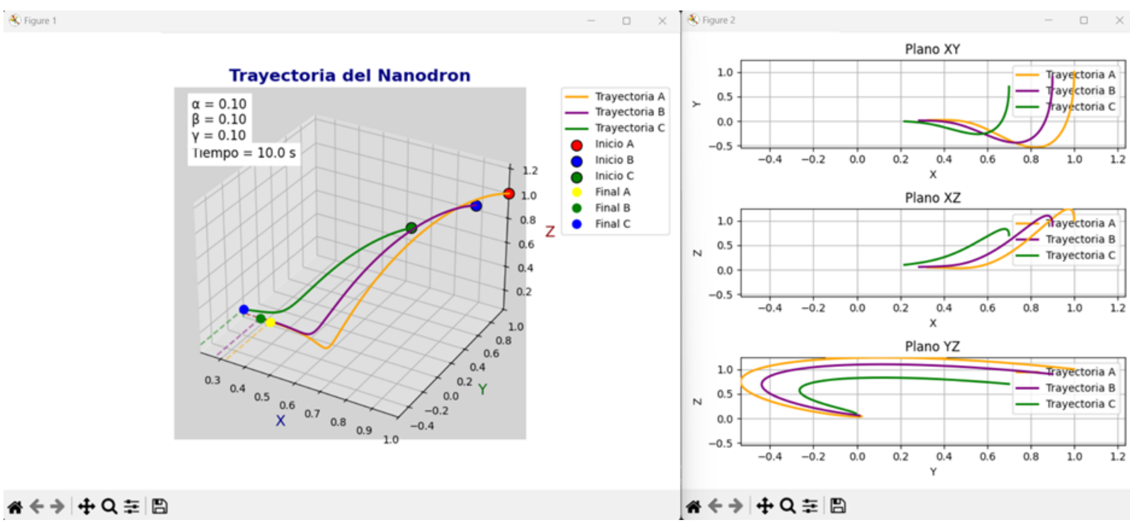

Figura 3: Simulación del caso 01.

Para el caso 02, se plantearon los valores que producen la *"Mariposa de Lorenz"*, $\alpha = 10$, $\beta = 28$, $\gamma = 8/3$. La gráfica de la simulación de estos parámetros se presentan en la (fig. 5).

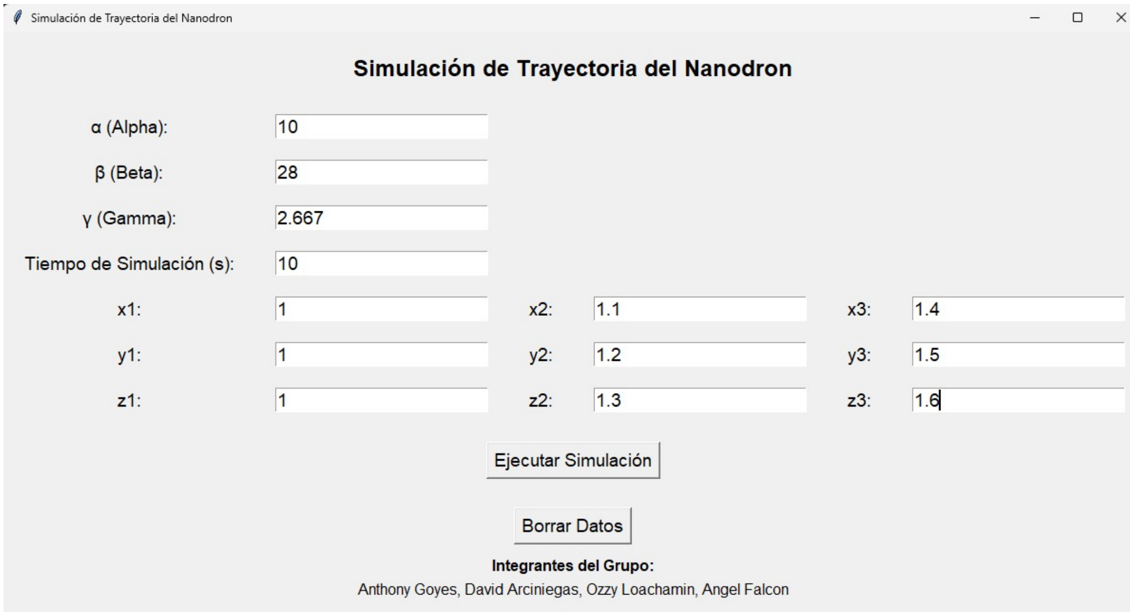

Figura 4: Interfaz del caso 02.

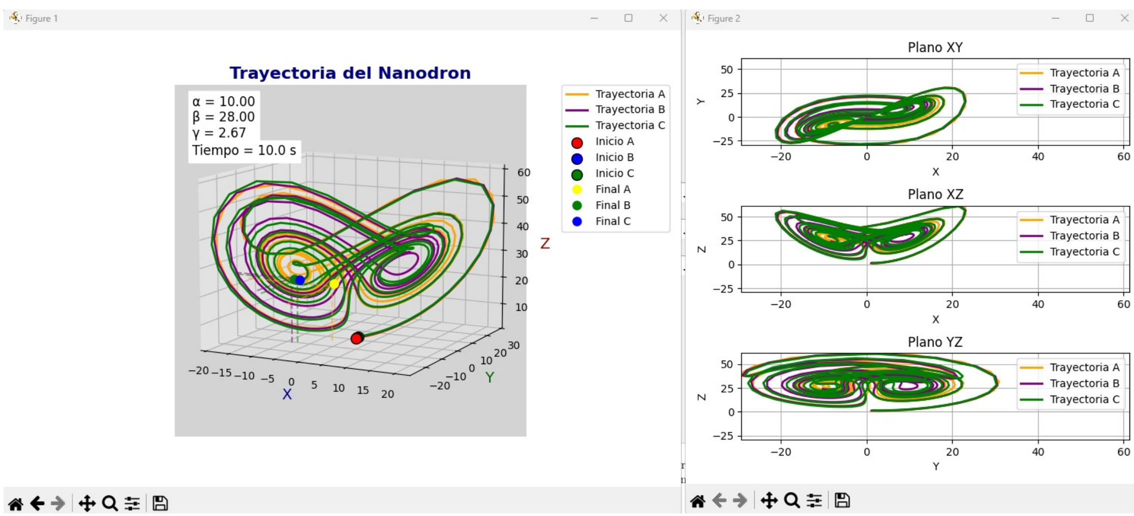

Figura 5: Simulación del caso 02.

Como se puede visualizar las condiciones ingresadas en α, β y γ de la interfaz de la (fig. 6), tienen casi los mismos valores para x, y y z lo que significa que cada conjunto de condiciones iniciales es idéntico. Como resultado, las tres trayectorias simuladas, aunque se calculan por separado para cada conjunto, seguirán la misma ruta y se verán como una sola trayectoria en la visualización, (fig. 7).

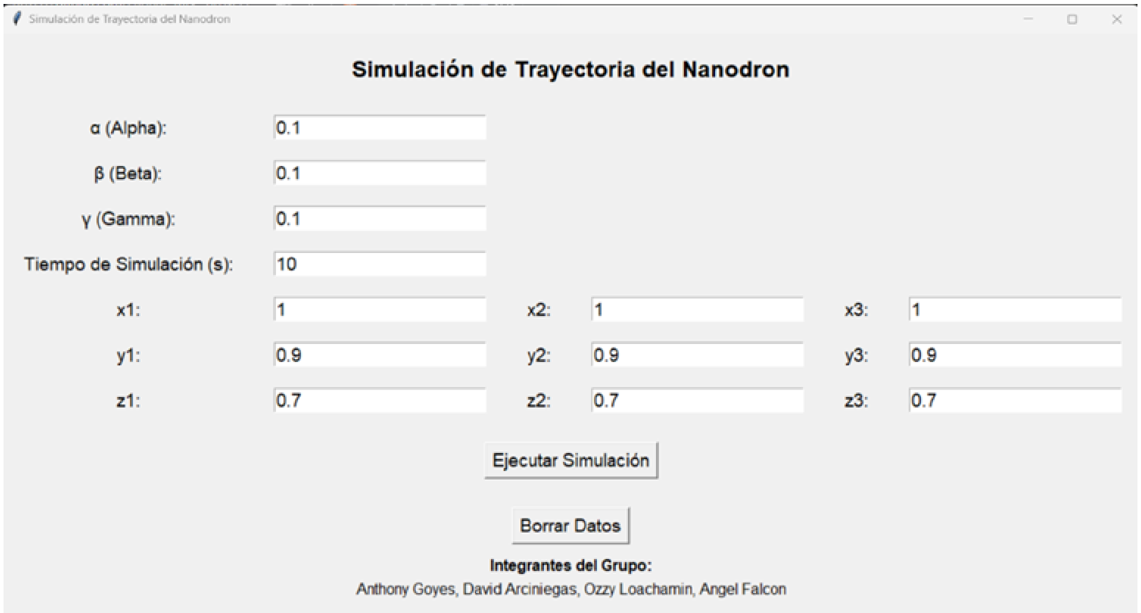

Figura 6: Interfaz del caso 03.

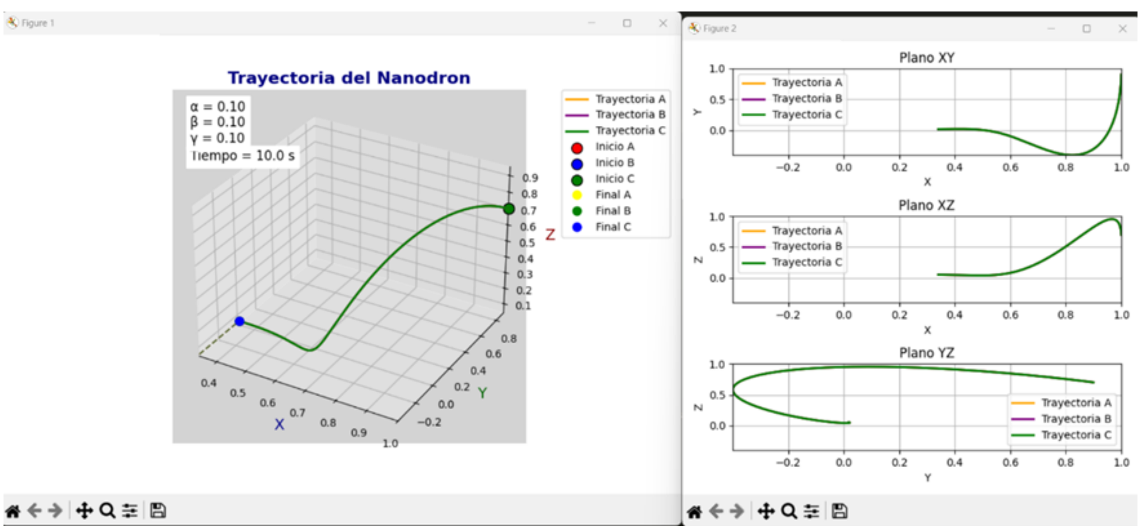

Figura 7: Simulación del caso 03.

Finalmente, se planteó el caso en el que α sea negativo (fig. 8), en este caso, se producen los resultados obtenidos en la (fig. 9).

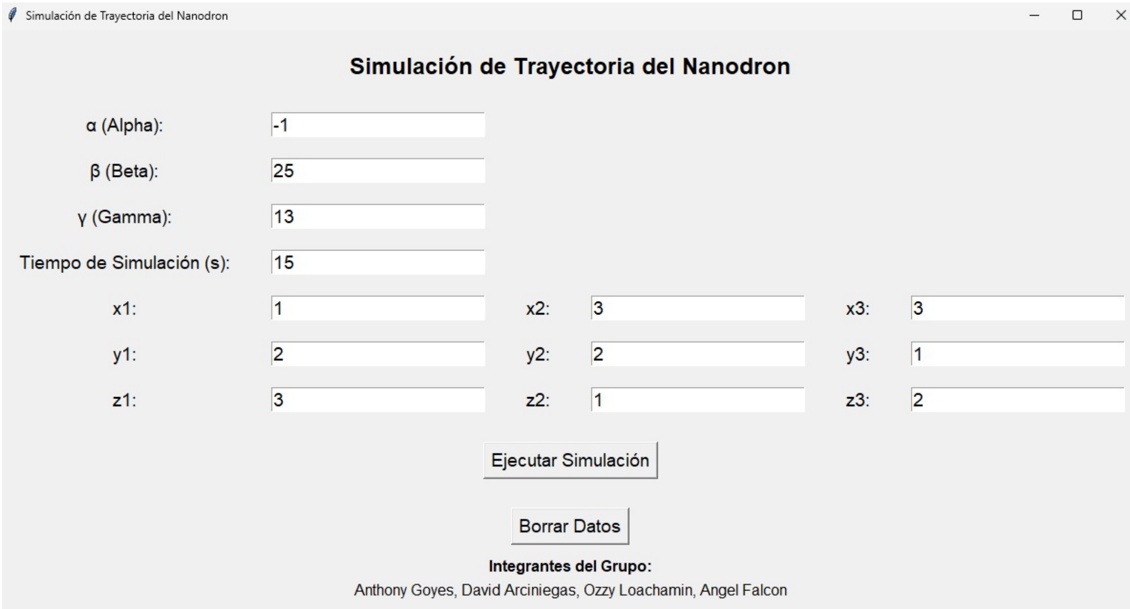

Figura 8: Interfaz para un ($\alpha = -1$).

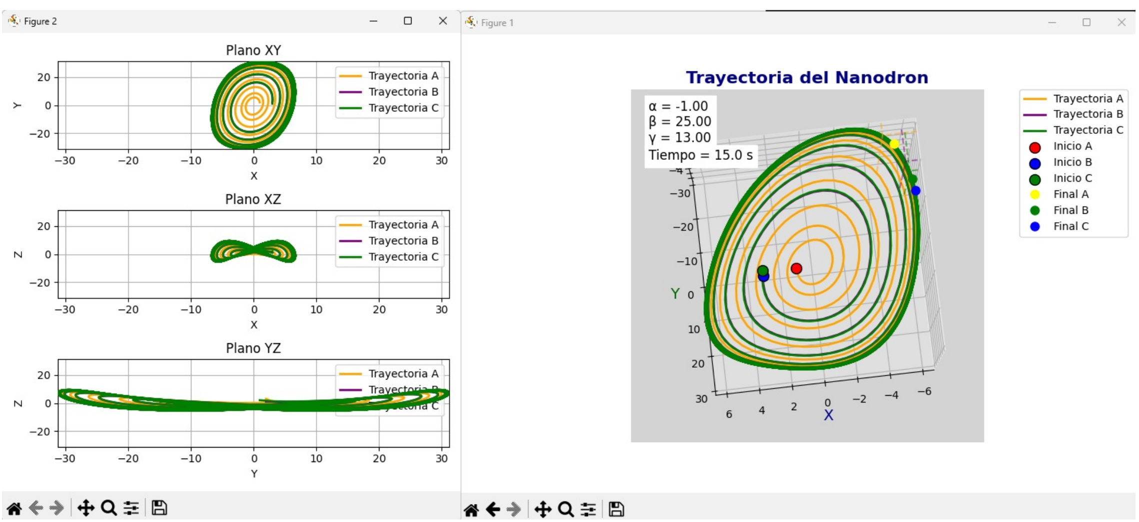

Figura 9: Simulación del caso negativo ($\alpha = -1$).

5. Conclusiones

Se logró desarrollar una simulación que muestra claramente la trayectoria del nanodron al resolver un sistema de ecuaciones diferenciales no lineales. La simulación permite ver cómo se mueve el nanodron en un espacio tridimensional.

Se creó una interfaz gráfica fácil de usar que permite a los usuarios cambiar los parámetros principales del sistema (α, β, γ), el tiempo de simulación y los puntos iniciales. Esto permite ver de forma interactiva cómo estos cambios afectan la trayectoria del nanodron, haciendo la simulación más interesante y educativa.

Este proyecto proporciona una buena base para futuras mejoras, como la implementación de métodos numéricos más avanzados o la inclusión de otros factores que puedan influir en el movimiento del nanodron. Además, la interfaz gráfica puede expandirse para incluir más opciones o controles.

El modelo de Lorenz utilizado para simular el vuelo del nanodron demostró un comportamiento caótico con alta sensibilidad a las condiciones iniciales, confirmando las predicciones teóricas del sistema dinámico.

6. Recomendaciones

Sería recomendable usar métodos numéricos más precisos, como Runge-Kutta, para mejorar la precisión de la simulación, especialmente en situaciones donde el método de Euler puede no ser suficiente por su simplicidad.

Realizar estudios adicionales para incorporar variables adicionales como la resistencia del aire y efectos de campo electromagnético, lo que podría ofrecer una simulación más precisa del vuelo del nanodron.

Referencias

[1] Fernández J. Métodos numéricos. Recuperado de: http://materias fi uba ar/6209/download/7-Numerico1 pdf. 2004.

[2] Gamez GM. Sistemas tipo lorenz. Master's thesis, Universidad de Sonora. Division de Ciencias Exactas y ...; 2006.

[3] Buitrago Puentes RH. El sistema y el atractor geométrico de Lorenz; 2010.