Simulación de Péndulos Acoplados con Interacción No Lineal, Amortiguamiento y Forzamiento

Análisis de Sincronización, Resonancia y Caos

[Romero Norman, Garzon Jorge]
ncromeror@correo.udistrital.edu.co]

Física Computacional 2
Profesor: John Hernán Díaz
Universidad Distrital Francisco José de Caldas

Octubre 2025

Resumen

En este trabajo se presenta una simulación computacional del movimiento de dos péndulos acoplados mediante una interacción no lineal cuadrática, sujetos a fuerzas de amortiguamiento y un forzamiento externo periódico. Se implementaron dos métodos de integración numérica (Runge-Kutta 4 y Euler-Cromer) para resolver el sistema de ecuaciones diferenciales. Se realizaron varios experimentos numéricos para analizar fenómenos complejos como la transferencia de energía, la sincronización de modos, la resonancia y la transición al caos. Los resultados obtenidos permiten caracterizar la rica dinámica del sistema y validar la robustez del software desarrollado.

Índice

1. Introducción

1.1. Motivación

La simulación de sistemas no lineales acoplados es fundamental para entender fenómenos físicos complejos en áreas como la mecánica, la óptica y la biología. El sistema de dos péndulos acoplados, amortiguados y forzados es un modelo paradigmático que, a pesar de su aparente simplicidad, exhibe una gran riqueza de comportamientos, incluyendo oscilaciones periódicas, transferencia de energía, sincronización y dinámica caótica. Este proyecto busca explorar dicha dinámica a través de herramientas computacionales.

1.2. Objetivos

- Implementar una simulación de dos péndulos acoplados, amortiguados y forzados en C++ utilizando Programación Orientada a Objetos (POO).
- Validar la precisión de los métodos de integración numérica (RK4 y Euler-Cromer) en el contexto de un sistema no conservativo.
- Analizar la transferencia de energía entre los péndulos bajo diferentes condiciones de acoplamiento.
- Estudiar el fenómeno de resonancia variando la frecuencia del forzamiento externo.
- Identificar y caracterizar regímenes de comportamiento periódico y caótico a través del análisis del espacio de fase.

2. Marco Teórico

El sistema está descrito por el siguiente par de ecuaciones diferenciales acopladas:

$$\ddot{\theta}_1 + \gamma\dot{\theta}_1 + \frac{g}{l}\sin(\theta_1) + \kappa(\theta_1 - \theta_2)^2 = F_0\cos(\omega t) \tag{1}$$

$$\ddot{\theta}_2 + \gamma\dot{\theta}_2 + \frac{g}{l}\sin(\theta_2) + \kappa(\theta_2 - \theta_1)^2 = F_0\cos(\omega t) \tag{2}$$

donde:

- θ_i: ángulo del péndulo i.
- g: aceleración gravitacional.
- l: longitud del péndulo.
- κ: constante de acoplamiento no lineal.
- γ: coeficiente de amortiguamiento.
- F_0: amplitud del forzamiento externo.
- ω: frecuencia angular del forzamiento.

2.1. Análisis Energético

Debido al amortiguamiento y forzamiento, el sistema no es conservativo. Sin embargo, el análisis de las energías individuales y de acoplamiento sigue siendo útil:

$$E_i = \frac{1}{2}ml^2\dot{\theta}_i^2 + mgl(1 - \cos\theta_i) \tag{3}$$

$$V_c = \frac{\kappa}{3}(\theta_1 - \theta_2)^3 \tag{4}$$

La energía total del sistema, $E_T(t) = E_1(t)+E_2(t)+V_c(t)$, no se conserva y su evolución temporal revela el balance entre la energía disipada y la energía inyectada por el forzamiento.

3. Métodos Numéricos

El sistema de dos ecuaciones de segundo orden se convierte en un sistema de cuatro ecuaciones de primer orden, con el vector de estado $\mathbf{y} = (\theta_1, \omega_1, \theta_2, \omega_2)$, donde $\omega_i = \dot{\theta}_i$.

3.1. Integración Runge-Kutta 4 (RK4)

Se implementó el método RK4 por su alta precisión y estabilidad, ideal para sistemas no lineales. El método avanza el estado del sistema en un paso de tiempo Δt mediante un promedio ponderado de cuatro evaluaciones de la derivada.

3.2. Integración Euler-Cromer

Para fines comparativos, se implementó el método de Euler-Cromer, un método simpléctico de primer orden que, aunque menos preciso que RK4, es más estable que el método de Euler estándar.

3.3. Parámetros de Simulación

Parámetro	**Símbolo**	**Valor Típico**
Paso de tiempo	Δt	0.01 s
Tiempo máximo	t_{max}	150 - 300 s
Constante de acoplamiento	κ	0.1 - 1.0
Coeficiente de amortiguamiento	γ	0.05 - 0.5
Amplitud de forzamiento	F_0	0.0 - 1.5
Frecuencia de forzamiento	ω	0.5 - 1.5

Cuadro 1: Parámetros típicos utilizados en las simulaciones.

4. Diseño del Software

4.1. Arquitectura POO

El software se desarrolló en C++17 siguiendo un paradigma de Programación Orientada a Objetos (POO). Se diseñó una clase `Pendulo` que encapsula el estado (θ, ω) y los parámetros de un péndulo individual. La lógica principal en `main.cpp` gestiona la interacción entre los dos objetos `Pendulo`, el bucle de simulación y la escritura de datos.

4.2. Diagrama de Flujo

El algoritmo principal sigue una secuencia lógica:

1. **Inicialización:** El usuario introduce los parámetros de simulación, físicos y condiciones iniciales de forma interactiva.
2. **Bucle Temporal:** Para cada paso de tiempo, se calcula la evolución del sistema utilizando el método numérico seleccionado (RK4 o Euler-Cromer).
3. **Almacenamiento:** El estado del sistema $(\theta_1, \omega_1, \theta_2, \omega_2)$ y las energías se guardan en un archivo `.csv`.
4. **Post-procesamiento:** Un script en Python se utiliza para leer los datos y generar los gráficos de análisis.

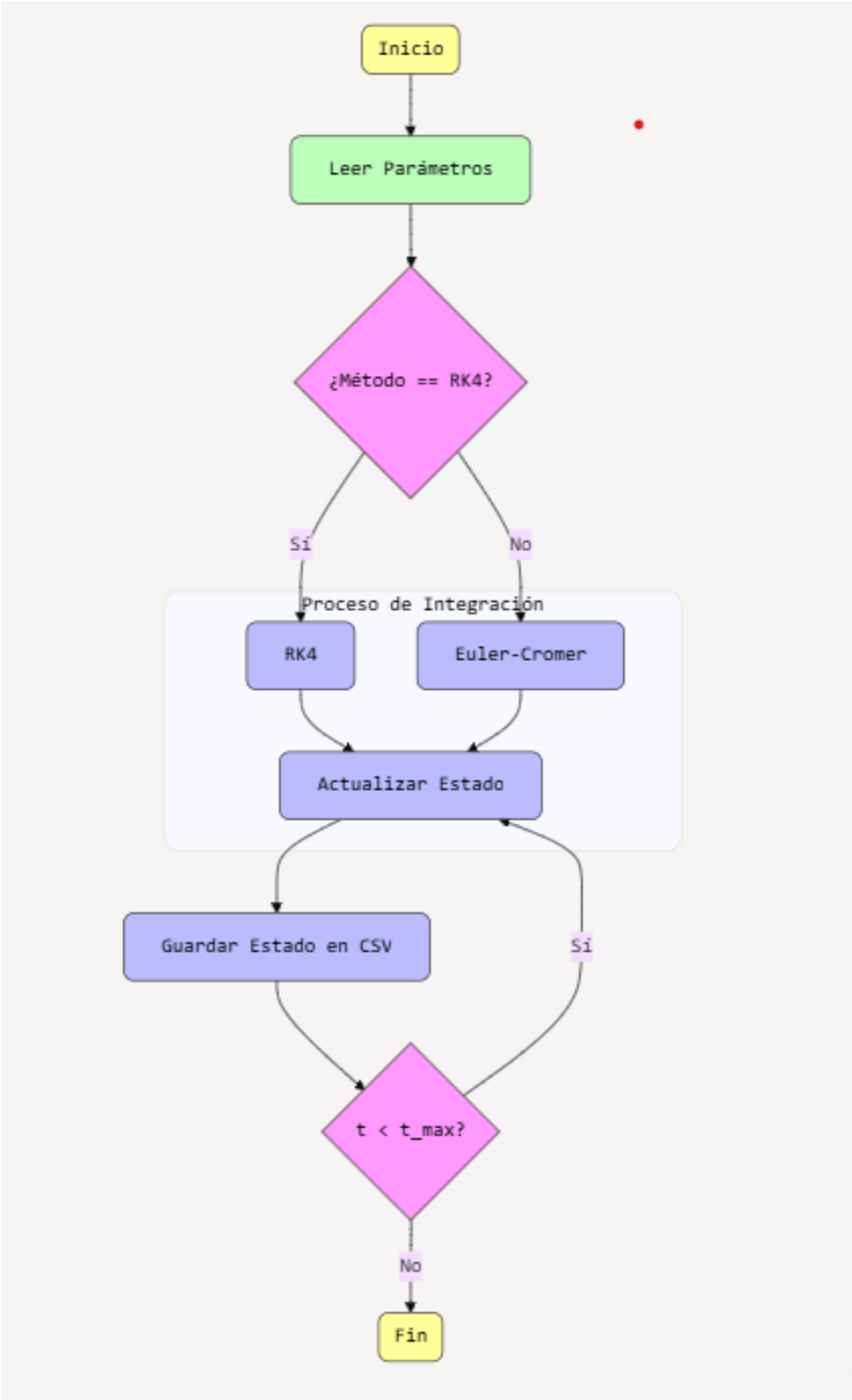

Figura 1: Diagrama de flujo del programa de simulación.

5. Resultados

5.1. Experimento 1: Transferencia de Energía

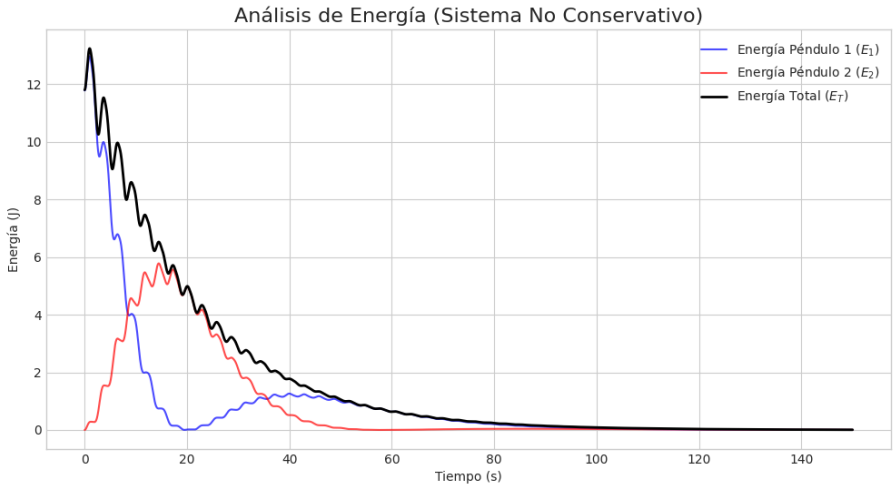

Figura 2: Evolución de las energías individuales y total para el caso no forzado. Se observa el fenómeno de batido (beats) donde la energía se transfiere periódicamente entre los péndulos.

5.2. Experimento 2: Resonancia

5.3. Experimento 2: Modos Sincronizados

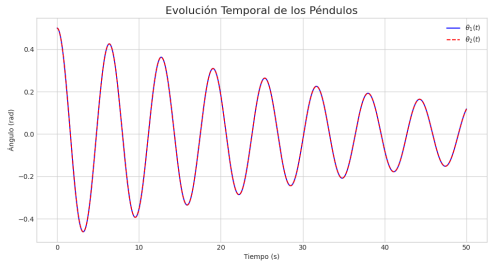

Figura 3: Modo simétrico ($\theta_1(0) = \theta_2(0)$). Los péndulos oscilan en fase.

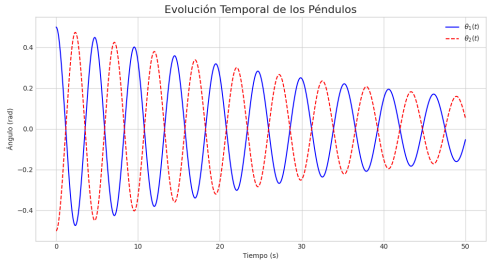

Figura 4: Modo antisimétrico ($\theta_1(0) = -\theta_2(0)$). Los péndulos oscilan en oposición.

(Asegúrate de que la numeración sea correcta)

-

6. Discusión

7. Conclusiones

1. Se implementó con éxito una simulación modular y reproducible en C++ para el sistema de péndulos acoplados, amortiguados y forzados.
2. El método RK4 demostró ser robusto para capturar la compleja dinámica del sistema, incluyendo regímenes caóticos.

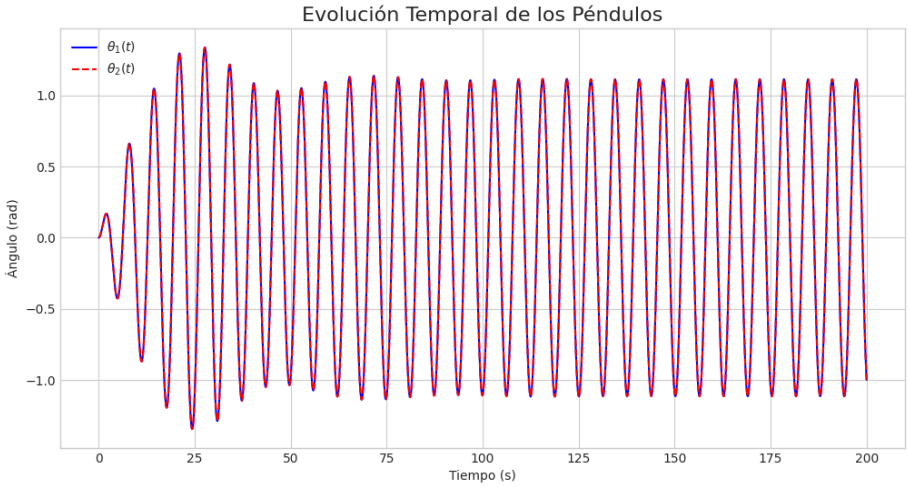

Figura 5: Amplitud de oscilación de θ_1 y θ_2 para una frecuencia de forzamiento en resonancia ($\omega = 1{,}0$ rad/s). La amplitud crece significativamente hasta alcanzar un estado estacionario donde la energía inyectada por el forzamiento es igual a la energía disipada por el amortiguamiento.

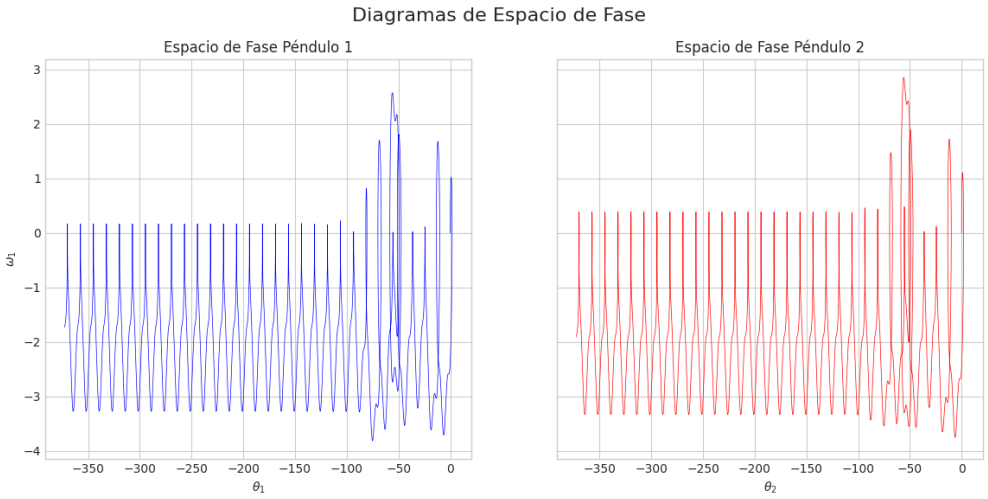

Figura 6: Diagrama de espacio de fase para un régimen de parámetros que induce caos ($F_0 = 1{,}5, \omega = 0{,}66$). Se observa un atractor extraño, caracterizado por trayectorias no periódicas que llenan una región del espacio de forma fractal. Esto contrasta con las elipses cerradas de un movimiento periódico.

3. Se validó numéricamente la transferencia de energía y la resonancia, fenómenos clave en sistemas oscilatorios acoplados.

4. Se identificó un atractor extraño en el espacio de fase para valores de forzamiento elevados, confirmando la capacidad del modelo para generar caos determinista.

5. El proyecto cumple con los estándares de un desarrollo de software científico, incluyendo documentación técnica (Doxygen) y análisis reproducible.

A. Código Fuente Principal

A.1. Clase Pendulo - Header

A.2. Bucle de Integración RK4

B. Instrucciones de Reproducción

Para compilar y ejecutar el proyecto, siga los siguientes pasos desde una terminal de Linux:

```
# 1. Compilar el proyecto
make

# 2. Ejecutar la simulación (modo interactivo)
make run

# 3. Generar los gráficos
make plot

# 4. Generar la documentación del código
make doc
```