

Simulación y visualización del péndulo no lineal amortiguado y forzado

Análisis: oscilaciones, transición a caos, mapas de Poincaré y
sensibilidad a condiciones iniciales

Resumen

Este documento resume las implementaciones y los resultados de simulaciones numéricas realizadas para el péndulo no lineal amortiguado y forzado. Se presentan los métodos numéricos usados (Euler–Cromer), las gráficas principales (ángulo y velocidad en función del tiempo para distintos regímenes, diferencias entre dos péndulos con condiciones iniciales casi idénticas, diagramas de fase y mapa de Poincaré) y la interpretación física de cada resultado.

Índice

1. Introducción	3
2. Método numérico	3
2.1. Euler–Cromer	3
2.2. Reinicio angular	4
3. Implementación (resumen)	4
3.1. Sistema físico (C++)	4
3.2. Bucle de integración (esquema)	4
4. Resultados y análisis	4
4.1. $\theta(t)$ para varios F_D	5
4.2. Comparación $\theta(t)$ con y sin reinicio ($F_D = 1,2$)	5
4.3. $\omega(t)$ para $F_D = 1,2$	6
4.4. Diferencia angular $\Delta\theta(t)$ entre dos péndulos (sensibilidad a condiciones iniciales)	6
4.5. Diagramas de fase (vs)	7
4.6. Mapa de Poincaré	7
5. Discusión	7
6. Archivos del proyecto	8

7. Conclusión

10

1. Introducción

Estudiamos el péndulo no lineal amortiguado y forzado, cuya ecuación de movimiento es

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l} \sin \theta - q \frac{d\theta}{dt} + F_D \sin(\Omega_D t). \quad (1)$$

Aquí θ es el ángulo, $\omega = d\theta/dt$ la velocidad angular; g la gravedad, l la longitud, q el coeficiente de amortiguamiento, F_D la amplitud de la fuerza de conducción y Ω_D su frecuencia.

Parámetros usados en las simulaciones:

Parámetro	Valor
g	9.8
l	9.8
q	0.5
Ω_D	2/3
Δt	0.04
Cond. iniciales	$\theta(0) = 0, 2, \omega(0) = 0$

El objetivo fue:

- Implementar integrador Euler–Cromer y usarlo para simular (1).
- Generar y guardar gráficas: $\theta(t)$, $\omega(t)$, comparación con/sin reinicio angular, evolución de $\Delta\theta$ entre dos péndulos cercanos, diagramas de fase y mapa de Poincaré.
- Interpretar los resultados: régimen amortiguado, respuesta forzada, y transición a caos para F_D grandes.

2. Método numérico

2.1. Euler–Cromer

El método Euler–Cromer (semi-implícito) para un sistema con estado (θ, ω) procede:

$$\omega_{n+1} = \omega_n + \dot{\omega}(\theta_n, \omega_n, t_n) \Delta t, \quad \theta_{n+1} = \theta_n + \omega_{n+1} \Delta t.$$

Ventaja: mejor estabilidad en osciladores que el Euler explícito. Se implementó de forma genérica en la librería estática (`libsimulacra.a`) y se usó la clase `EulerCromerIntegrator`.

2.2. Reinicio angular

Dado que θ es una variable angular, para propósitos de graficación se mantuvo θ en el intervalo $[-\pi, \pi]$ aplicando:

$$\text{si } \theta > \pi \Rightarrow \theta \leftarrow \theta - 2\pi; \quad \text{si } \theta < -\pi \Rightarrow \theta \leftarrow \theta + 2\pi.$$

Esto evita trazados con crecimiento continuo de θ cuando el péndulo completa giros enteros.

3. Implementación (resumen)

A continuación se muestran los fragmentos esenciales del código usados en las simulaciones. El proyecto contiene los ficheros de código que ya ejecutaste; aquí se resume la idea central.

3.1. Sistema físico (C++)

```
1 State PendulumSystem::derivatives(const State& s, double
  t) const {
2     double theta = s[0], omega = s[1];
3     double dtheta = omega;
4     double domega = -(g/l) * sin(theta) - q * omega + F_D
      * sin(Omega_D * t);
5     return State(std::vector<double>{dtheta, domega});
6 }
```

Listing 1: Derivadas del péndulo (función used como IPhysicalSystem)

3.2. Bucle de integración (esquema)

```
1 for (t = 0; t <= tmax; t += dt) {
2     integrator.step(system, state, t, dt); // actualiza
      state y t
3     // opcional: ajustar theta a [-pi,pi]
4     // guardar datos para graficar
5 }
```

Listing 2: Paso dentro del bucle principal usando Euler-Cromer

4. Resultados y análisis

A continuación se muestran las gráficas generadas por los scripts y su interpretación.

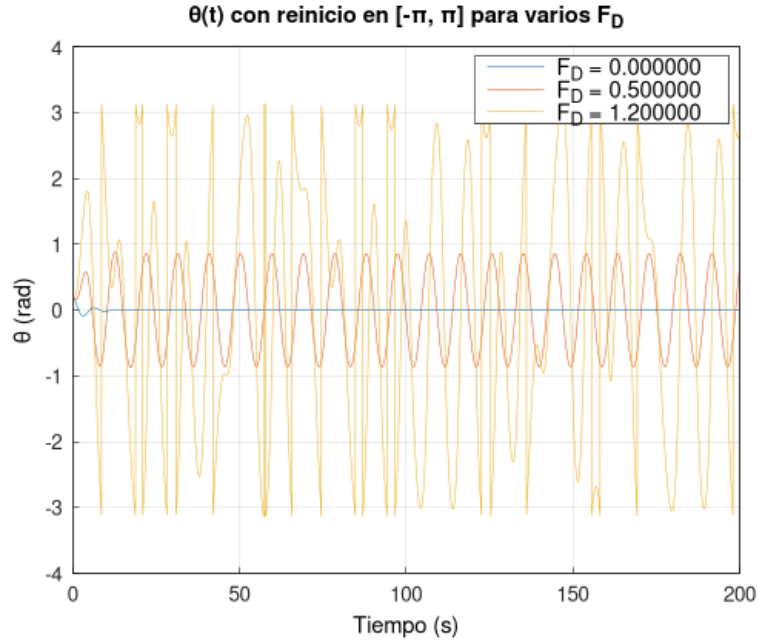


Figura 1: Comportamiento de $\theta(t)$ con reinicio en $[-\pi, \pi]$ para $F_D = 0, 0,5, 1,2$.

4.1. $\theta(t)$ para varios F_D

Interpretación:

- $F_D = 0$: las oscilaciones se amortiguan y el péndulo se detiene. La frecuencia de las oscilaciones transitorias se aproxima a la frecuencia natural $\Omega = \sqrt{g/l}$.
- $F_D = 0,5$: tras un transitorio el péndulo sincroniza con la fuerza externa, oscilando con Ω_D . Movimiento periódico estable.
- $F_D = 1,2$: aparecen saltos verticales en el gráfico por el reinicio del ángulo (cuando el péndulo da la vuelta completa). El comportamiento a largo plazo no es periódicamente repetitivo: indicio de caos.

4.2. Comparación $\theta(t)$ con y sin reinicio ($F_D = 1,2$)

Interpretación: los saltos verticales en la versión con reinicio son artefactos del mapeo angular (no discontinuidades físicas); la trayectoria sin reinicio muestra las vueltas completas. Ambos reflejan la misma dinámica subyacente.

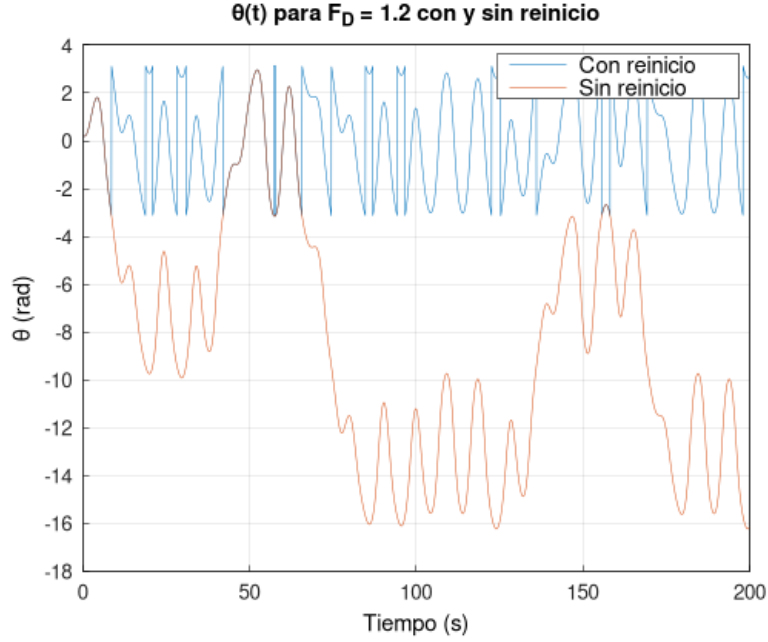


Figura 2: Comparación de $\theta(t)$ con reinicio en $[-\pi, \pi]$ y sin reinicio para $F_D = 1,2$.

4.3. $\omega(t)$ para $F_D = 1,2$

Interpretación: la velocidad muestra variaciones grandes y no periódicas en el régimen caótico, con amplitudes que cambian de forma irregular.

4.4. Diferencia angular $\Delta\theta(t)$ entre dos péndulos (sensibilidad a condiciones iniciales)

Interpretación física:

- En el caso no caótico ($F_D = 0,5$) la diferencia $\Delta\theta$ decae aproximadamente exponencialmente hacia cero: dos trayectorias cercanas convergen, indicando estabilidad y predictibilidad.
- En el caso caótico ($F_D = 1,2$) $\Delta\theta$ crece aproximadamente exponencialmente al comienzo: evidencia de *sensibilidad a condiciones iniciales* y un exponente de Lyapunov efectivo positivo. Tras suficiente tiempo $\Delta\theta$ puede saturar por límites angulares o por efectos de recorte/medición.

Nota sobre ajuste exponencial. En las gráficas se usó escala semilogarítmica (eje y log) para mostrar comportamiento lineal en esa escala — característico de crecimiento/decadencia exponencial $\Delta\theta \sim e^{\lambda t}$.

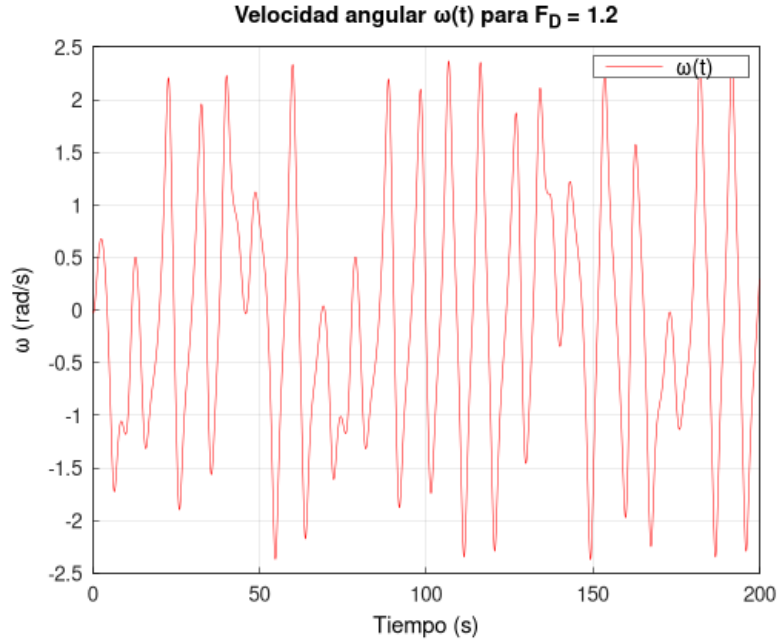


Figura 3: Velocidad angular $\omega(t)$ para $F_D = 1,2$.

4.5. Diagramas de fase (vs)

Interpretación:

- $F_D = 0,5$: órbita estable, cercana a un ciclo límite; la trayectoria es casi periódica.
- $F_D = 1,2$: estructuras complejas con múltiples curvas y envolventes; la trayectoria visita regiones del espacio de fase de forma no periódica.

4.6. Mapa de Poincaré

Interpretación: el mapa de Poincaré revela la estructura fractal/atrac-tor extraño en el régimen caótico: puntos dispersos que forman conjuntos con estructura, demostrando espacialmente la no periodicidad y sensibilidad.

5. Discusión

- Las simulaciones muestran claramente los tres comportamientos cuali-tativos: amortiguamiento sin conducción, respuesta periódica forzada, y caos para conducción fuerte.
- La presencia de crecimiento exponencial de $\Delta\theta$ en la fase caótica es indicativa de un exponente de Lyapunov positivo; en la fase no caótica el exponente es negativo (convergencia).

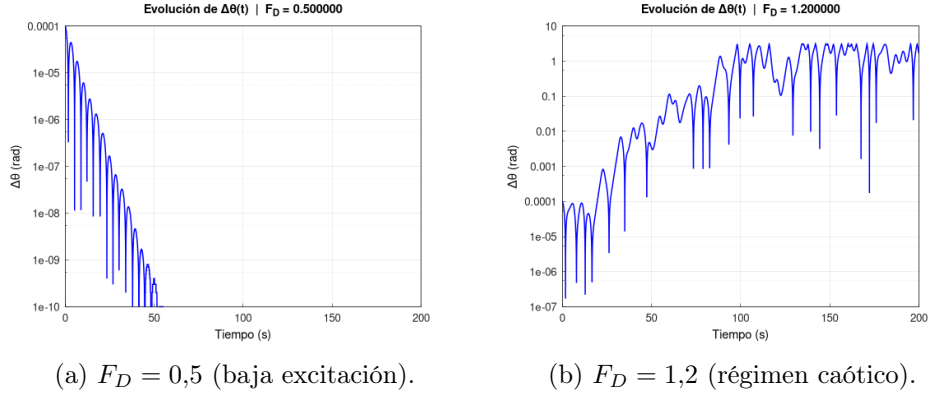


Figura 4: Evolución de $\Delta\theta(t) = |\theta_1 - \theta_2|$ donde $\theta_1(0) = 0,2$, $\theta_2(0) = 0,201$.

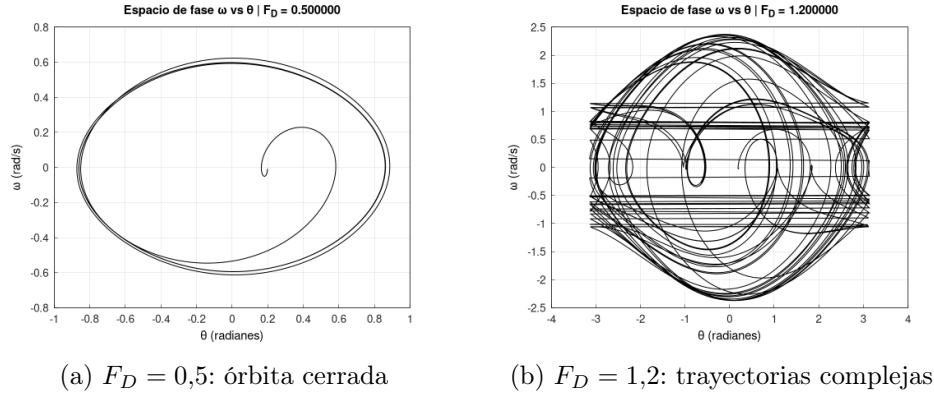


Figura 5: Diagramas de fase ω vs θ .

- El mapa de Poincaré es una herramienta poderosa para identificar estructuras en sistemas periódicamente forzados; simplifica la dinámica al muestrear en fase con la fuerza de conducción.
- Limitaciones numéricas: paso temporal finito, errores de truncamiento y del método Euler–Cromer; para estimaciones precisas de exponentes de Lyapunov se recomiendan métodos más cuidados (RK adaptativo, renormalización Gram–Schmidt, etc.).

6. Archivos del proyecto

Los scripts y archivos relevantes (tal como se ejecutaron) incluyen:

- `src/pendulum.cpp` (simulación básica y gráficas)
- `src/pendulum_delta.cpp` (*evolucinde* $\Delta\theta$)

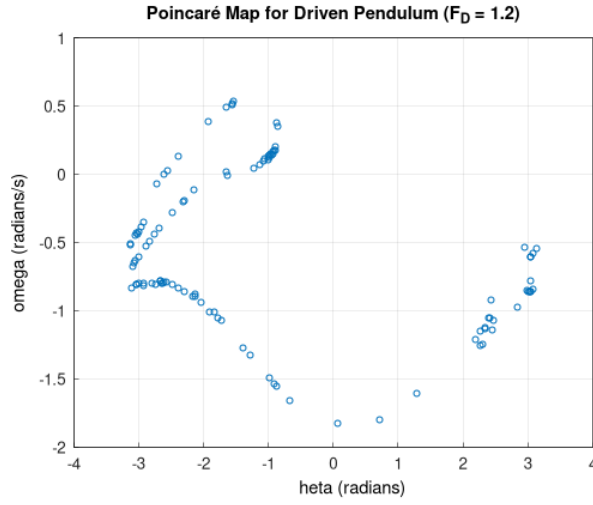


Figura 6: Mapa de Poincaré: puntos (θ, ω) muestreados en tiempos t_n tales que $\Omega_D t_n = 2\pi n$. $F_D = 1,2$.

- `src/pendulum_phasespace.cpp(diagramasde fase)`
- `src/pendulum_poincare.cpp(mapadePoincar)`
- `include/physim/core/*` (headers: `state.h`, `euler_cromer.h`, `system.h`, ...)
- Biblioteca estática: `lib/libsimulacra.a`
- Imágenes generadas (guardadas por los scripts):
 - `pendulum_thetafd.png`
 - `pendulum_thetacomparison.png`
 - `pendulum_omega.png`
 - `delta_thetaFD05.png`
 - `delta_thetaFD12.png`
 - `phase_spaceFD05.png`
 - `phase_spaceFD12.png`
 - `poincare_map.png`

7. Conclusión

Las simulaciones reproducen los resultados clásicos del péndulo forzado: para baja fuerza se obtiene comportamiento regular (convergencia y sincronía con la fuerza), mientras que para fuerzas grandes emerge comportamiento caótico con sensibilidad a condiciones iniciales. Las herramientas numéricas y visuales aplicadas (Euler–Cromer, diagramas de fase, mapas de Poincaré y curvas $\Delta\theta(t)$) permiten analizar y caracterizar adecuadamente estas transiciones.

Apéndice A: comandos de compilación

Ejemplo de línea de compilación (desde la raíz del proyecto, donde `lib/` contiene `libsimulacra.a`):

```
g++ -std=c++17 -Iinclude -Llib src/pendulum.cpp -lsimulacra -lmatplot -o pendulum
g++ -std=c++17 -Iinclude -Llib src/pendulum_delta.cpp -lsimulacra -lmatplot -o pendul
g++ -std=c++17 -Iinclude -Llib src/pendulum_phase_space.cpp -lsimulacra -lmatplot -o
g++ -std=c++17 -Iinclude -Llib src/pendulum_poincare.cpp -lsimulacra -lmatplot -o pen
```

Apéndice B: comentario final

Si quieres, puedo:

- Incluir en el documento los fragmentos de código completos (archivos `.cpp`) como anexos.
- Añadir los ajustes exponenciales y estimaciones del exponente de Lyapunov a partir de los datos numéricos.
- Generar una versión PDF ya compilada si me indicas dónde están las imágenes (o subir los archivos).

Documento generado automáticamente a partir de las simulaciones realizadas con la librería `simulacra` y `Matplotlib++`.