Taller: Explorando Sistemas Dinámicos y Bifurcaciones

NICOLAS ANTONIO VARGAS ANGARITA.

BRYAN ALEXANDER RODRÍGUEZ HERNÁNDEZ.

1. ¿Qué es un sistema dinámico?

Un sistema dinámico es una regla que describe cómo cambia el estado de un sistema a lo largo del tiempo. El 'estado' puede ser un número (por ejemplo, población relativa), un vector (posición y velocidad), etc. La regla puede ser discreta (avanza por pasos) o continua (se describe por ecuaciones diferenciales).

2. Sistemas discretos vs continuos

- Sistemas discretos: el tiempo avanza en pasos enteros ($n = 0,1,2,...$). Ejemplo general: $x_{n+1} = f(x_n)$.

- Sistemas continuos: el estado cambia según ecuaciones diferenciales, por ejemplo: $\dot{x} = g(x,t)$.

3. Estabilidad y caos (resumen)

- Estabilidad: si se parte cerca de un punto de equilibrio, las soluciones se acercan a él con el tiempo.

- Caos: comportamiento no periódico, altamente sensible a condiciones iniciales; reglas deterministas pueden generar caos.

4. Ejemplos reales

- Crecimiento poblacional (modelo logístico).

- Epidemias (modelos SIR y variantes).

- Economía y mercados (modelos iterativos y retroalimentaciones).

Ecuación logística discreta

La ecuación logística discreta se escribe:

$x_{n+1} = r x_n (1 - x_n)$

Significado de los términos:

- x_n: población normalizada en el paso n (fracción entre 0 y 1).

- r: parámetro de crecimiento (controla la tasa).

- $x_n (1 - x_n)$: tasa de crecimiento con limitación por capacidad (evita crecimiento indefinido).

Si r aumenta mucho:

- Para $0 < r < 1$: la población tiende a 0.

- Para $1 < r < 3$: la población tiende a un punto fijo (equilibrio estable).

- Para $r > 3$: aparecen oscilaciones (duplicación de periodo) y, a partir de ciertos valores, caos.

Análisis matemático

1) Puntos fijos

Un punto fijo x^* satisface $x^* = r x^* (1 - x^*)$. Resolviendo se obtienen:

- $x^* = 0$

- $x^* = 1 - 1/r$ (para $x^* \neq 0$)

2) Derivada y estabilidad

La función $f(x) = r x (1 - x)$ tiene derivada $f'(x) = r (1 - 2x)$.

Un punto fijo x^* es estable si $|f'(x^*)| < 1$.

- En $x^* = 0$: $f'(0) = r \rightarrow$ estable si $0 < r < 1$.

- En $x^* = 1 - 1/r$: $f'(x^*) = -r + 2 \rightarrow$ estable si $1 < r < 3$.

Interpretación: cuando $1 < r < 3$ el sistema converge a un equilibrio estable. Al cruzar $r = 3$ ocurre una bifurcación y aparecen ciclos de periodo 2; aumentando r se sucede una cascada de duplicación de periodo y finalmente caos.

Programación y exploración (resumen de objetivos)

Objetivo: programar la ecuación logística y observar su evolución para distintos valores de r. En la práctica se simula iterando $x_{n+1} = r x_n (1 - x_n)$ a partir de una condición inicial x_0 y se grafica o analiza la serie.

Comportamiento esperado para valores de r concretos:

- r = 2.5: el sistema converge al punto fijo x ≈ 0.6 (estable).

- r = 3.2: el punto fijo deja de ser estable y aparece un ciclo de periodo 2 (oscila entre dos valores).

- r = 3.8: comportamiento típicamente caótico (sensibilidad a condiciones iniciales; trayectorias aparentemente aleatorias).

Diagrama de bifurcaciones (conceptos y respuestas)

1. ¿Dónde aparecen los puntos de bifurcación?

- La primera bifurcación (punto fijo → periodo 2) aparece en r = 3.

- A partir de r = 3 se produce una cascada de duplicación de periodo (2 → 4 → 8 → ...), acumulándose cerca de r ≈ 3.56995...

2. ¿En qué intervalos aparece el caos?

- Para r ≳ 3.56995 aparece caos persistente, aunque existen ventanas periódicas aisladas (por ejemplo, cerca de r ≈ 3.83 puede aparecer periodo-3).

3. ¿Qué significado real puede tener esto?

- Incluso reglas sencillas y deterministas pueden producir comportamientos impredecibles a largo plazo si el parámetro de crecimiento es alto. En biología, economía o ecología indica que altas tasas de crecimiento con limitaciones pueden generar oscilaciones y comportamientos complejos.

Análisis crítico

¿Por qué sistemas deterministas pueden generar caos?

- Porque las iteraciones no lineales amplifican pequeñas diferencias en las condiciones iniciales, provocando divergencia exponencial entre trayectorias cercanas (sensibilidad a condiciones iniciales).

Diferencias entre orden, caos y homeostasis:

- Orden: comportamiento predecible (por ejemplo, convergencia a punto fijo).

- Caos: comportamiento no predecible a largo plazo aun cuando la dinámica es determinista; alta sensibilidad a condiciones iniciales.

- Homeostasis: regulación que mantiene variables cerca de un punto de equilibrio (equivalente en muchos casos a estabilidad).

Relaciones con fenómenos reales:

- Clima: variables no lineales y acoplamientos pueden generar dinámica caótica (por eso la predicción a largo plazo es difícil).

- Mercado financiero: retroalimentaciones y alta ganancia pueden producir oscilaciones e irregularidades; la presencia de agentes humanos añade complejidad adicional.

Referencias

- May, R. M. (1976). Simple mathematical models with very complicated dynamics. Nature, 261, 459–467.

- Strogatz, S. H. (1994). Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering. Westview Press.

- Feigenbaum, M. J. (1978). Quantitative universality for a class of nonlinear transformations. Journal of Statistical Physics.

- Alligood, T. D., Sauer, T. D., & Yorke, J. A. (1996). Chaos: An Introduction to Dynamical Systems. Springer.

- G. Nicolis & I. Prigogine (1977). Self-Organization in Nonequilibrium Systems. Wiley.

- Artículo y recursos en línea: 'Logistic map' — se puede consultar para visualizaciones y detalles adicionales (por ejemplo en libros de texto y recursos educativos).

Instrucciones para ejecutar los códigos

Aquí se explica paso a paso cómo compilar y ejecutar los programas en C++ generados para este taller. Incluye instrucciones para Windows (con MinGW o WSL), Linux y macOS. Los archivos .cpp entregados son: logistic_sim.cpp y bifurcation.cpp.

Recomendaciones generales antes de empezar:

- Asegúrate de tener instalado un compilador de C++ (g++). En Windows puedes instalar MinGW o usar WSL; en macOS se puede usar Xcode Command Line Tools; en Linux normalmente está disponible en los repositorios (paquete build-essential).

- Guarda los archivos .cpp en la misma carpeta donde ejecutarás los comandos.

Pasos rápidos (Linux / macOS / WSL)

1. Abre una terminal (Terminal, Konsole, etc.).

2. Navega a la carpeta donde guardaste los archivos (.cpp). Por ejemplo:

   cd /ruta/a/la/carpeta

3. Compila el programa. Ejemplo para logistic_sim.cpp:

   g++ -std=c++11 logistic_sim.cpp -O2 -o logistic_sim

4. Ejecuta el programa:

   ./logistic_sim

5. El programa pedirá r, x0 y número de pasos; introduce valores (por ejemplo r=3.2, x0=0.2, pasos=500).

6. Se generará un archivo 'serie.csv' en la misma carpeta; puedes abrirlo con Excel o LibreOffice Calc para ver la serie, o usar Python/Matplotlib para graficar.

Pasos rápidos (Windows con MinGW)

1. Abre 'Command Prompt' o 'PowerShell' y navega a la carpeta con los archivos .cpp.

2. Compila:

```
g++ -std=c++11 logistic_sim.cpp -O2 -o logistic_sim.exe
```

3. Ejecuta:

```
.\logistic_sim.exe
```

4. Sigue las instrucciones en pantalla. Los archivos .csv se crearán en la carpeta actual.

Consejos para el diagrama de bifurcación

- El archivo 'bifurcation.csv' puede ser grande; si tu computadora es lenta, modifica 'r_step' a 0.005 o reduce 'samples' en el código para acelerar la generación.

- Para visualizar el diagrama: abre 'bifurcation.csv' y crea un diagrama de dispersión (scatter) con 'r' en el eje horizontal y 'x' en el eje vertical.

Solución de problemas comunes

- Si el compilador no se encuentra: revisa que g++ está instalado y en la variable PATH.

- Si el programa tarda mucho: reduce 'transient' y 'samples' en bifurcation.cpp, o aumenta 'r_step'.

- Si los archivos .csv no se abren en Excel: prueba abrirlos con LibreOffice Calc o importarlos desde Excel (datos → importar texto delimitado por comas).