Modelamiento Fisico Taller 1

Kevin Cruz

12 de febrero de 2026

1. Introducción y Modelo Matemático

Se analiza el modelo de Rashevsky para la producción de inconformistas en una sociedad. Sea $x(t)$ la población total y $x_i(t)$ la población de inconformistas. El sistema se describe por:

$$\frac{dx}{dt} = (b-d)x \tag{1}$$
$$\frac{dx_i}{dt} = (b-d)x_i + rb(x - x_i) \tag{2}$$

donde b y d son tasas de nacimiento y muerte, y r es la fracción de descendencia inconformista proveniente de padres conformistas.

1.1. Deducción de la Ecuación para la Proporción $p(t)$

Definimos la proporción de inconformistas como $p(t) = \frac{x_i(t)}{x(t)}$. Para encontrar su dinámica, derivamos respecto al tiempo utilizando la regla del cociente:

$$\frac{dp}{dt} = \frac{d}{dt}\left(\frac{x_i}{x}\right) = \frac{x \cdot \frac{dx_i}{dt} - x_i \cdot \frac{dx}{dt}}{x^2} \tag{3}$$

Sustituyendo las ecuaciones (1) y (2) en la expresión anterior:

$$\frac{dp}{dt} = \frac{x\left[(b-d)x_i + rb(x-x_i)\right] - x_i\left[(b-d)x\right]}{x^2} \tag{4}$$

Expandiendo el numerador para simplificar:

$$\text{Num} = x(b-d)x_i + xrb(x-x_i) - x_i(b-d)x \tag{5}$$

Observamos que el término $x(b-d)x_i$ se cancela con $-x_i(b-d)x$. Esto es crucial, pues indica que el crecimiento natural proporcional (nacimientos y muertes "estándar") afecta a ambos grupos por igual y no altera la proporción. Nos queda:

$$\frac{dp}{dt} = \frac{rbx(x-x_i)}{x^2} = rb\left(\frac{x}{x} - \frac{x_i}{x}\right) \tag{6}$$

Finalmente, sustituyendo $p = x_i/x$, obtenemos la ecuación diferencial ordinaria (EDO) desacoplada:

$$\frac{dp}{dt} = rb(1-p) \tag{7}$$

Interpretación:

- **Independencia de d:** La tasa de mortalidad d desaparece porque reduce el numerador (x_i) y el denominador (x) en la misma proporción relativa. El cambio en la composición social $p(t)$ depende exclusivamente de la tasa de çonversión"de nuevos nacimientos (rb).
- **Equilibrio y Estabilidad:** El equilibrio ocurre cuando $\frac{dp}{dt} = 0$, es decir, $rb(1-p) = 0 \implies p^* = 1$. Dado que si $p < 1$, $\frac{dp}{dt} > 0$, la proporción tiende a crecer. El estado $p = 1$ es un **equilibrio estable**. Socialmente, esto implica que, bajo estas reglas de apareamiento aleatorio, la sociedad tiende inevitablemente a volverse completamente inconformista a largo plazo.

2. Solución Analítica

Resolvemos la ecuación (7) mediante separación de variables. Sea $k = rb$:

$$\int \frac{dp}{1-p} = \int k\,dt \implies -\ln(1-p) = kt + C \tag{8}$$

Despejando $p(t)$:

$$1 - p = e^{-kt-C} = Ae^{-kt} \implies p(t) = 1 - Ae^{-kt} \tag{9}$$

Usando la condición inicial $p(0) = p_0$, tenemos $p_0 = 1 - A$, por lo que $A = 1 - p_0$. La solución exacta es:

$$p(t) = 1 - (1 - p_0)e^{-rbt} \tag{10}$$

2.1. Evaluación Numérica

Dados los parámetros $p(0) = 0{,}01$, $b = 0{,}02$, $d = 0{,}015$, $r = 0{,}1$. Calculamos la constante combinada $k = rb = (0{,}1)(0{,}02) = 0{,}002$. La solución específica es:

$$p(t) = 1 - 0{,}99e^{-0{,}002t} \tag{11}$$

Evaluamos en $t = 50$:

$$p(50) = 1 - 0{,}99e^{-0{,}002(50)} = 1 - 0{,}99e^{-0{,}1} \approx 1 - 0{,}99(0{,}9048) \approx \mathbf{0{,}10421} \tag{12}$$

Comportamiento asintótico: Cuando $t \to \infty$, el término exponencial $e^{-0{,}002t} \to 0$, por lo tanto $\lim_{t\to\infty} p(t) = 1$. Esto confirma el análisis de estabilidad del equilibrio.

3. Métodos Numéricos

Para aproximar la solución numérica de la EDO $p'(t) = rb(1-p)$ en el intervalo $t \in [0, 50]$ con paso $h = 1$, definimos la función $f(t, p) = k(1-p)$, donde la constante combinada es $k = rb = 0{,}002$.

3.1. Método de Euler Explícito

La fórmula general es $p_{n+1} = p_n + hf(t_n, p_n)$. Sustituyendo nuestra función:

$$p_{n+1} = p_n + hk(1 - p_n) \tag{13}$$

Esta fórmula es explícita y directa, con un error de truncamiento local de orden $O(h^2)$.

3.2. Método de Taylor de Orden 2

La expansión de Taylor requiere la segunda derivada de $p(t)$. Derivamos la EDO original respecto a t:

$$p'(t) = k(1-p) \tag{14}$$

$$p''(t) = \frac{d}{dt}[k - kp] = -k\frac{dp}{dt} = -k[k(1-p)] = -k^2(1-p) \tag{15}$$

La fórmula iterativa de Taylor de orden 2 es:

$$p_{n+1} = p_n + hp'_n + \frac{h^2}{2}p''_n \tag{16}$$

Sustituyendo p' y p'':

$$p_{n+1} = p_n + hk(1-p_n) - \frac{h^2k^2}{2}(1-p_n) \tag{17}$$

Factorizando el término común $(1-p_n)$, obtenemos la fórmula computacional:

$$p_{n+1} = p_n + (1-p_n)\left(hk - \frac{h^2k^2}{2}\right) \tag{18}$$

3.3. Método del Trapecio Implícito

La fórmula del trapecio (Crank-Nicolson para EDOs) promedia las pendientes en t_n y t_{n+1}:

$$p_{n+1} = p_n + \frac{h}{2}[f(p_n) + f(p_{n+1})] \tag{19}$$

Sustituyendo $f(p) = k(1-p)$:

$$p_{n+1} = p_n + \frac{h}{2}[k(1-p_n) + k(1-p_{n+1})] \tag{20}$$

Para implementar este método, debemos despejar p_{n+1} algebraicamente. Expandimos los términos:

$$p_{n+1} = p_n + \frac{hk}{2}(1-p_n) + \frac{hk}{2} - \frac{hk}{2}p_{n+1} \tag{21}$$

Agrupamos los términos que contienen p_{n+1} a la izquierda:

$$p_{n+1} + \frac{hk}{2}p_{n+1} = p_n + \frac{hk}{2}(1-p_n) + \frac{hk}{2} \tag{22}$$

Factorizamos p_{n+1} y simplificamos el lado derecho:

$$p_{n+1}\left(1 + \frac{hk}{2}\right) = p_n + \frac{hk}{2}(2-p_n) \tag{23}$$

Finalmente, despejamos para obtener la fórmula iterativa explícita:

$$p_{n+1} = \frac{p_n + \frac{hk}{2}(2-p_n)}{1 + \frac{hk}{2}} \tag{24}$$

Esta formulación permite calcular p_{n+1} directamente sin necesidad de un método de búsqueda de raíces (como Newton-Raphson) en cada paso, gracias a la linealidad de la EDO original.

4. Análisis de Resultados Numéricos

Se implementaron los métodos en Python para un intervalo de 50 años con un paso $h = 1$. Los resultados obtenidos al final del intervalo se resumen en la Tabla 1.

Tabla 1: Comparación de resultados y error absoluto en $t = 50$ $(h = 1)$

Método	**Valor** $p(50)$	**Error Absoluto**	**Orden Global**
Solución Exacta	0.10421284	-	-
Euler Explícito	0.10398242	$2{,}30 \times 10^{-4}$	$O(h)$
Taylor Orden 2	0.10421238	$4{,}64 \times 10^{-7}$	$O(h^2)$
Trapecio	0.10421283	$1{,}15 \times 10^{-8}$	$O(h^2)$

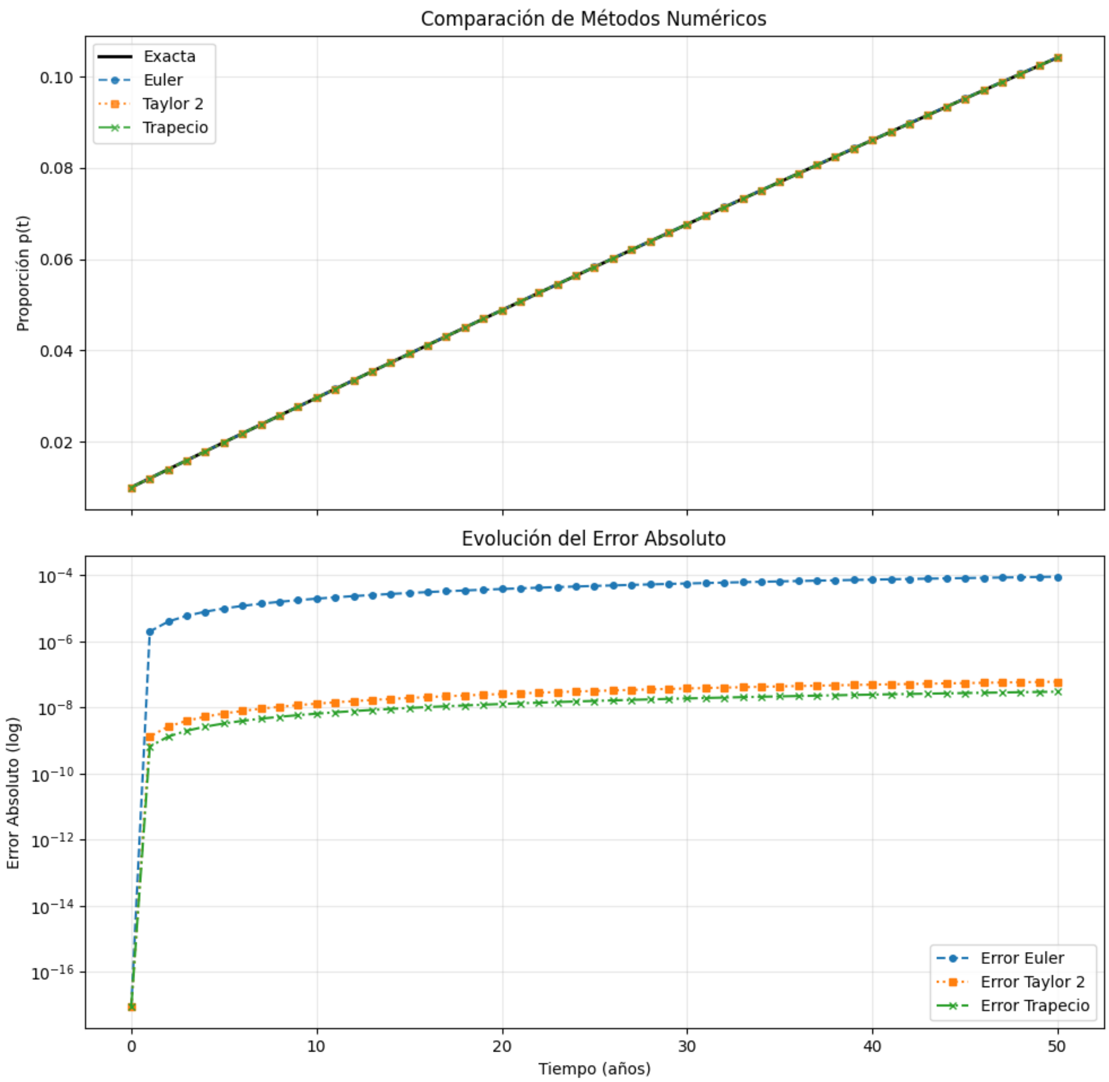

Figura 1: Comparación de las trayectorias de $p(t)$ y evolución del error absoluto en escala logarítmica.

4.1. Análisis de Precisión

Como se observa en la Figura 1, todos los métodos capturan la tendencia creciente de la población inconformista. Sin embargo, el gráfico de error absoluto revela diferencias críticas:

- **Euler:** Presenta el error más alto, estabilizándose cerca de 10^{-4}. Al ser un método de primer orden, su precisión es limitada para pasos de tiempo grandes ($h = 1$).
- **Taylor 2 y Trapecio:** Ambos métodos de segundo orden muestran una precisión superior (errores entre 10^{-7} y 10^{-8}). El método del **Trapecio** resulta ligeramente más preciso en este caso, lo cual es notable dado que es un método estable y robusto para diversos tipos de EDOs.

5. Comparativa de Eficiencia Computacional

Siguiendo el requerimiento del taller, se ejecutó un benchmark de 10^7 iteraciones en Python, C++ y Fortran[cite: 16]. Los tiempos de ejecución obtenidos se presentan en la Tabla **??** y la Figura 2.

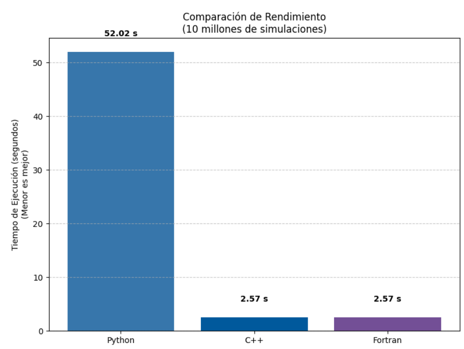

Figura 2: Comparativa de rendimiento entre lenguajes interpretados y compilados.

5.1. Análisis de Rendimiento

La diferencia de rendimiento es drástica: C++ y Fortran son aproximadamente 20 veces más rápidos que Python[cite: 17]. Esta disparidad se explica por:

- **Naturaleza del Lenguaje:** Python es un lenguaje interpretado; cada iteración del bucle implica una sobrecarga de inspección de tipos en tiempo de ejecución.
- **Compilación:** C++ y Fortran son lenguajes compilados que traducen el código directamente a instrucciones de máquina optimizadas, gestionando de forma eficiente el uso de registros del procesador durante los cálculos numéricos intensivos.

6. Conclusiones

El estudio del modelo de Rashevsky permite concluir que la proporción de inconformistas en esta sociedad es independiente de la tasa de mortalidad d y depende únicamente de la tasa de çonversión"por nacimientos rb. Matemáticamente, el sistema tiende a un equilibrio estable en $p^* = 1$, lo que sugiere una homogeneización total a largo plazo bajo las premisas dadas.

Desde la perspectiva computacional, mientras que Python ofrece una gran facilidad para la generación de gráficas y análisis rápido, el uso de lenguajes como Fortran o C++ sigue siendo indispensable en el cómputo científico cuando se requiere procesar volúmenes masivos de iteraciones o simulaciones complejas.