UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

Facultad de Ingeniería de Sistemas e Informática

Proyecto Final de Análisis Numérico

Simulación Numérica y Métodos Iterativos

Curso:	Análisis Numérico
Docente:	Dr. Richard Cubas Becerra
Integrantes:	Diego Sotelo
	Alexis Gonzales
	Paolo Villavicencio
	Álvaro Salazar
Fecha:	18 de noviembre de 2025

Índice

Resumen

Palabras clave: Lotka-Volterra, Runge–Kutta, Python, Manim, Simulación.

1. Introducción

1.1. Descripción del Problema:

El modelado matemático de sistemas ecológicos constituye un pilar esencial dentro de la biología teórica. Entre los fenómenos más estudiados destaca la dinámica depredador–presa, caracterizada por oscilaciones cíclicas en las poblaciones de ambas especies. En este tipo de interacción, un incremento en la población de presas (por ejemplo, liebres) genera una mayor disponibilidad de alimento, permitiendo que los depredadores (como los linces) aumenten su población. Con el tiempo, la mayor presión de depredación reduce la abundancia de presas, lo que posteriormente provoca una disminución en la población de depredadores por escasez de alimento, permitiendo que el ciclo se reinicie. Este comportamiento cíclico, ampliamente documentado en la ecología matemática, ha sido señalado como uno de los fundamentos históricos del desarrollo de la ecología teórica moderna (Kingsland, 2015).

1.2. Planteamiento del Proyecto:

Para traducir esta dinámica biológica en un modelo cuantitativo, este proyecto se basa en el sistema formulado independientemente por Alfred J. Lotka (1925) y Vito Volterra (1926). Ambos propusieron un sistema de ecuaciones diferenciales ordinarias (EDOs) acopladas que hoy se conoce como el modelo de Lotka-Volterra, que aunque relativamente simple en su planteamiento, capta la esencia de la dependencia mutua y oscilatoria entre las especies. Dado que este sistema general no ofrece una solución analítica simple para tiempos arbitrarios, el proyecto busca resolverlo numéricamente y desarrollar un simulador computacional como herramienta de exploración.

1.3. Modelo y Método:

Para resolver el sistema de EDOs del modelo de Lotka-Volterra, se requiere un método numérico robusto. Métodos simples como el de Euler tienen ventajas de sencillez, pero tienden a acumular errores de truncamiento en simulaciones prolongadas. Por ello, este trabajo opta por emplear el método de Runge-Kutta de cuarto orden (RK4), reconocido por su buen equilibrio entre precisión y coste computacional: al evaluar la pendiente en varios puntos intermedios, mejora significativamente la aproximación respecto a los métodos de orden inferior. El objetivo es implementar este algoritmo desde cero en Python, generar los datos de simulación y visualizar la evolución de las poblaciones en el tiempo.

2. Objetivos

2.1. Objetivo general

Crear un programa que simule la evolución de las poblaciones (depredador-presa) a lo largo del tiempo, aplicando los métodos numéricos estudiados en el curso para representar los resultados mediante gráficos y una animación.

2.2. Objetivos específicos

- Analizar el modelo matemático de Lotka-Volterra que describe la interacción entre depredadores y presas.
- Desarrollar una implementación propia del método Runge-Kutta de 4º orden (RK4) para la solución numérica del sistema de EDOs, aplicando la teoría de la Unidad IV del curso.
- Codificar un programa en Python que emplee el método RK4 para calcular la evolución de las poblaciones a través del tiempo.
- Diseñar visualizaciones de datos para ilustrar los resultados, incluyendo un gráfico de población contra tiempo y un diagrama de fase que muestre la relación entre especies.
- Evaluar el comportamiento del ecosistema mediante la experimentación con distintos parámetros iniciales (ej. tasas de natalidad o eficiencia de caza) y analizar su impacto en la estabilidad del sistema.

3. Marco teórico y Metodología

3.1. Modelos Matemáticos

Los métodos numéricos permiten aproximar soluciones cuando las formas analíticas son inviables. En este trabajo se consideran tanto sistemas lineales como ecuaciones diferenciales ordinarias.

$$A\mathbf{x} = \mathbf{b}, \quad \frac{dy}{dt} = f(t, y)$$

3.2. Ecuación Diferencial Lotka-Volterra

El modelo depredador-presa de Lotka-Volterra se define mediante un sistema que incluye las siguientes dos ecuaciones diferenciales ordinarias:

$$\begin{cases} \dfrac{dx}{dt} = \alpha x - \beta xy & \text{(Ecuación de la presa)} \\ \dfrac{dy}{dt} = -\gamma y + \delta xy & \text{(Ecuación del depredador)} \end{cases} \tag{1}$$

donde las constantes positivas representan:

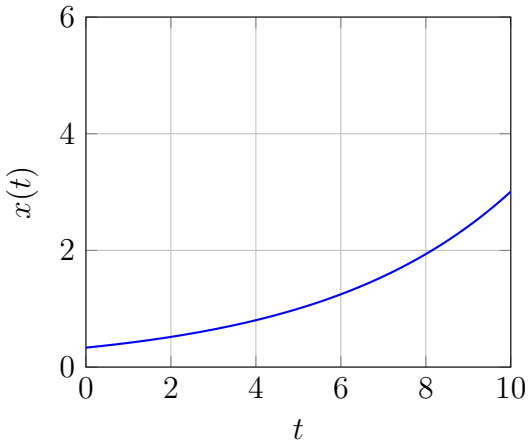
- α : tasa de crecimiento de las presas.
- β : tasa de encuentros presa–depredador que reduce la población de presas.
- γ : tasa de mortalidad de los depredadores en ausencia de presas.
- δ : incremento en la población de depredadores debido al consumo de presas.

Los puntos de equilibrio del sistema se obtienen igualando las derivadas a cero:

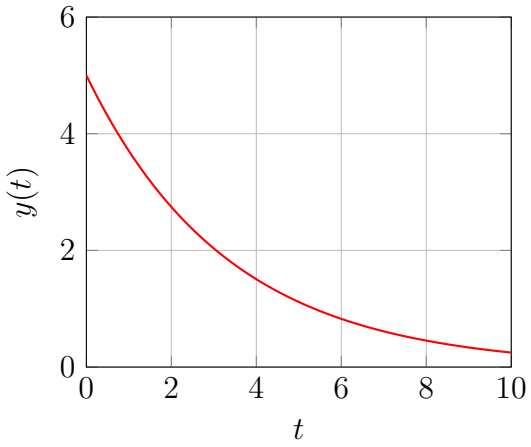
$$(x^*, y^*) = (0, 0), \qquad (\bar{x}, \bar{y}) = \left(\frac{\gamma}{\delta}, \frac{\alpha}{\beta}\right).$$

El equilibrio trivial $(0, 0)$ corresponde a la extinción de ambas poblaciones, mientras que $(\bar{x}, \bar{y})$ describe un equilibrio no trivial donde coexisten presas y depredadores, alrededor del cual surgen órbitas cerradas (curvas ovoidales) en el plano de fases.

Visualización cualitativa del modelo Lotka–Volterra



(a) Evolución de la población de presas cuando no hay depredadores.



(b) Evolución de la población de depredadores cuando no hay presas.

Figura 1: Escenarios extremos del modelo Lotka–Volterra: solo presas y solo depredadores.

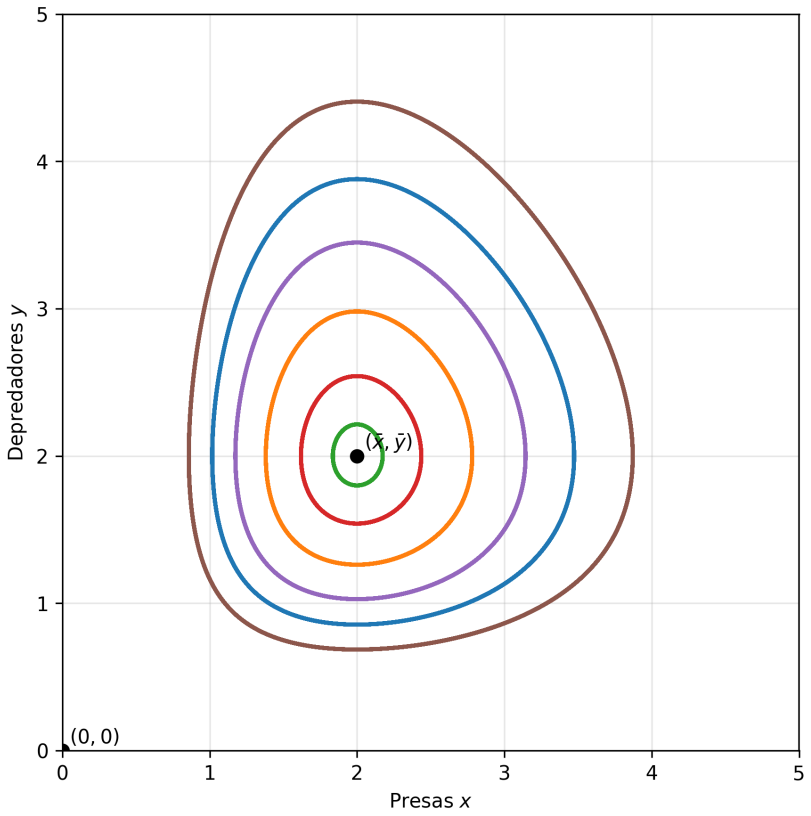


Figura 2: Plano de fases del modelo Lotka–Volterra con órbitas cerradas irregulares alrededor del punto de equilibrio no trivial.

3.3. Runge–Kutta de Cuarto Orden

El método de Runge–Kutta de cuarto orden (RK4) es una técnica numérica ampliamente utilizada para resolver ecuaciones diferenciales ordinarias (EDO) de manera precisa y estable, sin necesidad de calcular derivadas de orden superior.

Para un sistema de ecuaciones como este, el método RK4 calcula las poblaciones en el siguiente instante de tiempo $t+h$ (con paso h) mediante las siguientes fórmulas:

$$P_{t+h} = P_t + \frac{h}{6}(k_{1P} + 2k_{2P} + 2k_{3P} + k_{4P}),$$
$$D_{t+h} = D_t + \frac{h}{6}(k_{1D} + 2k_{2D} + 2k_{3D} + k_{4D}).$$

Al aplicar el método RK4 al modelo Depredador–Presa, se observa que las poblaciones de presas y depredadores oscilan periódicamente. Cuando la población de presas aumenta, los depredadores disponen de más alimento y también crecen; sin embargo, al incrementarse los depredadores, la población de presas disminuye, lo que causa una posterior reducción en los depredadores. Estas oscilaciones se repiten en el tiempo, representando un equilibrio dinámico entre ambas especies.

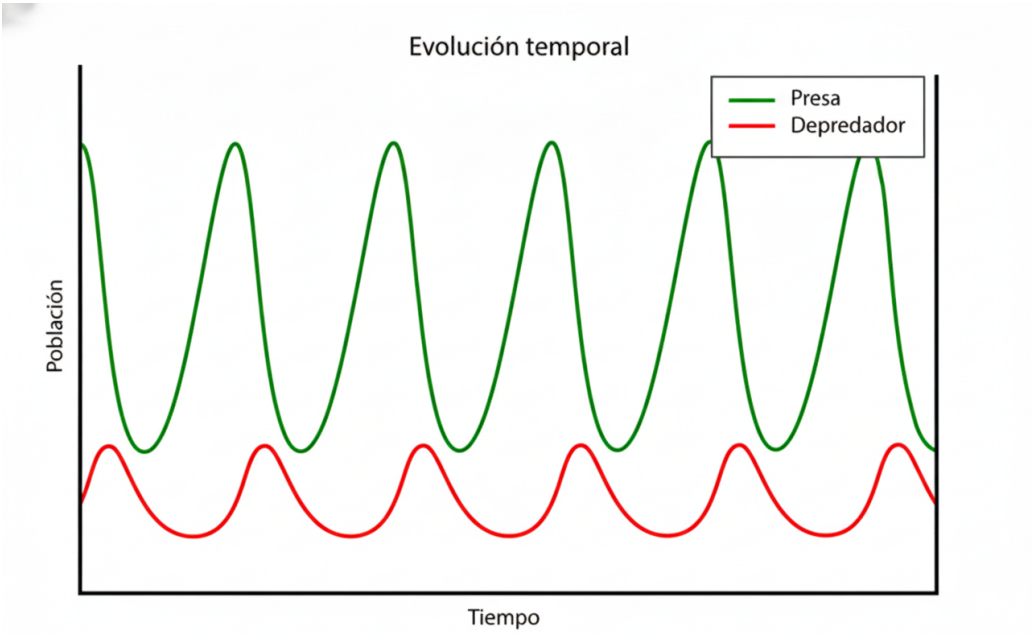


Figura 3: Evolución temporal de las poblaciones de presas (P) y depredadores (D), obtenida mediante el método de Runge–Kutta de 4° orden.

```
1
2  def runge_kutta4(f, y0, t0, tf, h):
3      t = np.arange(t0, tf+h, h)
4      y = np.zeros(len(t))
5      y[0] = y0
6      for i in range(0, len(t)-1):
7          k1 = f(t[i], y[i])
8          k2 = f(t[i] + h/2, y[i] + h*k1/2)
9          k3 = f(t[i] + h/2, y[i] + h*k2/2)
10         k4 = f(t[i] + h, y[i] + h*k3)
11         y[i+1] = y[i] + (h/6)*(k1 + 2*k2 + 2*k3 + k4)
12     return t, y
```

Listing 1: Método de Runge–Kutta 4to orden

3.4. Herramientas y Tecnologías

- **Python:** Lenguaje principal para la lógica del sistema.
- **Manim:** Utilizado para animaciones y visualización de procesos.

4. Desarrollo del simulador

Desarrollo y fases de prueba del proyecto seguidas

4.1. Investigación y Planificación:

Estudio del modelo y la teoría de RK4.

4.2. Implementación del Motor Numérico:

Creación de la función en Python que implementa el algoritmo RK4 y sus pruebas.

4.3. Desarrollo del Simulador:

Creación del bucle principal que simula el paso del tiempo y almacena los resultados en arreglos de NumPy.

4.4. Creación de las Visualizaciones:

Uso de Matplotlib para graficar los resultados almacenados

5. Resultados y análisis

5.1. Simulación base:

Presentación de los parámetros y condiciones iniciales usados.

- **Gráfico 1:** Población vs. Tiempo. (Insertar aquí tu gráfico). Discusión de los ciclos observados.
- **Gráfico 2:** Diagrama de Fase. (Insertar aquí tu gráfico). Discusión de la órbita estable.

5.2. Experimentación y Análisis del Sistema:

Ejecución de la simulación con diferentes parámetros

- **Experimento 1:** (Ej. ¿Qué pasa si α (natalidad de conejos) aumenta?). Mostrar gráficos y explicar el cambio)
- **Experimento 2:** (Ej. ¿Qué pasa si γ (mortalidad de zorros) disminuye?). Mostrar gráficos y explicar el cambio)

6. Conclusiones

7. Referencias bibliográficas

- Burden, R. L., & Faires, J. D. (2011). *Análisis Numérico*. Cengage Learning.
- Chapra, S. C., & Canale, R. P. (2015). *Métodos Numéricos para Ingenieros*. McGraw-Hill.
- Kingsland, S. E. (2015). *Alfred J. Lotka and the origins of theoretical population ecology*. Proceedings of the National Academy of Sciences, 112(30), 9493–9495.https://pmc.ncbi.nlm.nih.gov/articles/PMC4534218
- Kiusalaas, J. (2013). *Numerical Methods in Engineering with Python 3*. Cambridge University Press.
- Forrest, S. (s. f.).*Predator–Prey Models*. Dept. of Computer Science, University of New Mexico. Recuperado de https://www.cs.unm.edu/~forrest/classes/cs365/lectures/Lotka-Volterra.pdf
- Parker, A. E. (2021).*Runge–Kutta 4 (and other numerical methods for ODE's)* Ursinus College Digital Commons. Recuperado de https://digitalcommons.ursinus.edu/cgi/viewcontent.cgi?article=1007&context=triumphs

8. Anexos