

ANÁLISIS DE ESTABILIDAD Y SIMULACIÓN NUMÉRICA DEL SISTEMA  
PRESA-DEPREDADOR CON RESPUESTAS FUNCIONALES DE TIPO II DE HOLLING PARA  
PRESAS ADULTAS

Daniel Machado  
Daniel Toledo  
Osvaldo Moreno  
José A. Concepción  
Adrián Navarro  
C211

Mayo 2023

**Defensa del proyecto final.**

Facultad Matemática Computación

# TEMAS A TRATAR

- 1 INTRODUCCIÓN
- 2 MODELO PROPUESTO
- 3 SIMULACIONES NUMÉRICAS
- 4 CONCLUSIONES
- 5 BIBLIOGRAFÍA DEL TEMA

El artículo titulado "Local Analysis of the Prey-Predator Model with Stage-Structure Prey and Holling Type Functional Responses" por Dian Savitri analiza el modelo presa-depredador utilizando respuestas funcionales de Holling de tipo II para presas adultas y tipo I para presas jóvenes.

# OBJETIVOS

- Analizar los resultados del artículo
- Determinar los puntos de equilibrio del modelo.
- Analizar la estabilidad local de los puntos de equilibrio utilizando la matriz jacobiana y valores propios.
- Observar el comportamiento dinámico del modelo mediante simulaciones numéricas.

# TÉCNICAS UTILIZADAS

- Análisis matemático para determinar los puntos de equilibrio y condiciones de estabilidad.
- Cálculo de la matriz jacobiana y valores propios en cada punto de equilibrio.
- Aplicación del Criterio de Routh-Hurwitz para analizar la estabilidad del equilibrio interior.
- Simulaciones numéricas utilizando el método Runge-Kutta de cuarto orden y el método de Euler.

# MODELO PROPUESTO

El artículo analiza previamente tres modelos matemáticos que abordan las interacciones presa-depredador con diferentes enfoques. Se definen  $x(t)$ ,  $y(t)$  y  $z(t)$  como la densidad de la población de presas jóvenes, presas adultas y depredadores respectivamente.

# MODELO PROPUESTO

## MODELO MATEMÁTICO

$$\begin{aligned}\frac{dx}{dt} &= rx \left(1 - \frac{x}{k}\right) - \beta x - \alpha xz \\ \frac{dy}{dt} &= \beta x - \frac{\eta yz}{y + m} - \mu y \\ \frac{dz}{dt} &= \alpha_1 xz + \rho z^2 - \frac{\eta_1 z^2}{y + m}\end{aligned}\tag{1}$$

# RESPUESTAS FUNCIONALES DE HOLLING

## HOLLING TIPO I

Expresa la existencia de un aumento lineal en la tasa de captura del depredador con respecto a la densidad de presas, hasta alcanzar el valor en el que la tasa máxima de depredación permanece constante. Esta respuesta funcional es una función continua por partes, con una sección lineal y una sección constante.

## HOLLING TIPO II

Expresa un aumento en el consumo que se desacelera a medida que aumentan las presas consumidas, alcanzando la tasa máxima de consumo de depredadores de forma asintótica. La respuesta funcional de Holling tipo II es monotóna creciente, lo que significa que la tasa de consumo de los depredadores aumenta con la densidad de presas.



# PUNTOS DE EQUILIBRIO Y ESTABILIDAD LOCAL

Los puntos de equilibrio estan dados por aquellos que satisfagan la igualdad:

$$\frac{dx}{dt} = \frac{dy}{dt} = \frac{dz}{dt} = 0$$

# PUNTOS DE EQUILIBRIO Y ESTABILIDAD LOCAL

## POSTULADOS

Haciendo los cálculos pertinentes se obtienen como resultado los siguientes valores.  $E_1 = (0,0,0)$ ,  $E_2 = (\frac{k(r-\beta)}{r}, \frac{\beta k(r-\beta)}{\mu r}, 0)$  y  $E_3 = (x^*, y^*, z^*)$ .

# MATRIZ JACOBIANA

$$J(x, y, z) = \begin{bmatrix} r - \frac{2rx}{k} - \beta - \alpha z & 0 & -\alpha x \\ \beta & -\frac{\eta z}{y+m} + \frac{\eta y z}{(y+m)^2} - \mu & -\frac{\eta y}{y+m} \\ \alpha_1 z & \frac{\eta_1 z^2}{(y+m)^2} & 2\rho z - \frac{2\eta_1 z}{y+m} + \alpha_1 x \end{bmatrix}$$

# PUNTOS DE EQUILIBRIO Y ESTABILIDAD LOCAL E1

Al evaluar la matriz jacobiana en  $E_1$  obtuvimos:

Resultados obtenidos	Resultado de Savitri
$J(E_1) = \begin{bmatrix} r - \beta & 0 & 0 \\ \beta & -\mu & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$J(E_1) = \begin{bmatrix} (-1 + r) & 0 & 0 \\ \beta & -\mu & 0 \\ 0 & 0 & 0 \end{bmatrix}$

# PUNTOS DE EQUILIBRIO Y ESTABILIDAD LOCAL E1

$$\begin{vmatrix} r - \beta - \lambda & 0 & 0 \\ \beta & -\mu - \lambda & 0 \\ 0 & 0 & 0 - \lambda \end{vmatrix} = (r - \beta - \lambda)(\mu - \lambda)(\lambda) = 0$$

Los valores propios de la matriz  $J(E_1)$  son:  $\lambda_1 = r - \beta$  (el cual puede ser negativo si  $\beta > r$ ),  $\lambda_2 = -\mu$  y  $\lambda_3 = 0$ . Por lo tanto, el punto de equilibrio  $E_1$  es inestable.

# PUNTOS DE EQUILIBRIO Y ESTABILIDAD LOCAL E2

Para el punto  $E_2$  tenemos:

$$\begin{vmatrix} (\beta - r) - \lambda & 0 & \frac{\alpha k(\beta - r)}{r} \\ \beta & -\mu - \lambda & \frac{\eta \beta k(\beta - r)}{r\mu \left( \frac{\beta k(r - \beta)}{\mu r} + m \right)} \\ 0 & 0 & -\frac{\alpha_1 k(\beta - r)}{r} - \lambda \end{vmatrix} = 0$$

## PUNTOS DE EQUILIBRIO Y ESTABILIDAD LOCAL E2

Los valores propios de la matriz  $J(E_2)$  son  $\lambda_1 = \beta - r$ ,  $\lambda_2 = -\mu$  y  $\lambda_3 = -\frac{\alpha_1 k(\beta - r)}{r}$ . El autor plantea que  $E_2$  es estable bajo la condición  $r > \beta$ . Sin embargo, si  $r > \beta$  entonces  $-\frac{\alpha_1 k(\beta - r)}{r} > 0$  por lo que el punto de equilibrio  $E_2$  es inestable.

# PUNTOS DE EQUILIBRIO Y ESTABILIDAD LOCAL E3

Para  $E_3$  tenemos la siguiente matriz jacobiana:

$$J(x^*, y^*, z^*) = \begin{bmatrix} r - \frac{2rx^*}{k} - \beta - \alpha z^* & 0 & -\alpha x^* \\ \beta & -\frac{\eta z^*}{y^* + m} + \frac{\eta y z^*}{(y^* + m)^2} - \mu & -\frac{\eta y^*}{y^* + m} \\ \alpha_1 z^* & \frac{\eta_1 z^{*2}}{(y^* + m)^2} & 2\rho z^* - \frac{2\eta_1 z^*}{y^* + m} + \alpha_1 x^* \end{bmatrix}$$

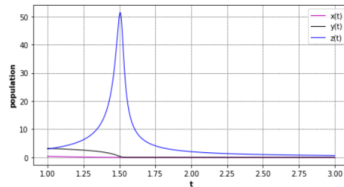
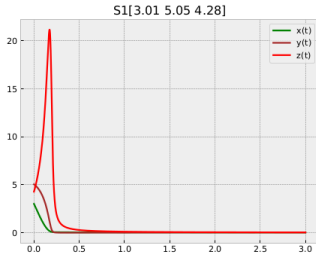


# PUNTOS DE EQUILIBRIO Y ESTABILIDAD LOCAL

Puntos de Equilibrio	Estabilidad Local
$E_1 = (0, 0, 0)$	Inestable
$E_2 = \left( \frac{k(r-\beta)}{r}, \frac{\beta k(r-\beta)}{\mu r}, 0 \right)$	Inestable
$E_3 = (x^*, y^*, z^*)$	Asintóticamente estable

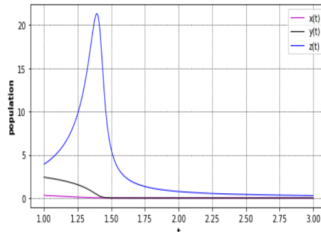
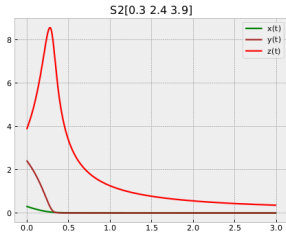
# SIMULACIONES NUMÉRICAS

S1 con valores iniciales [3.01 5.05 4.28]



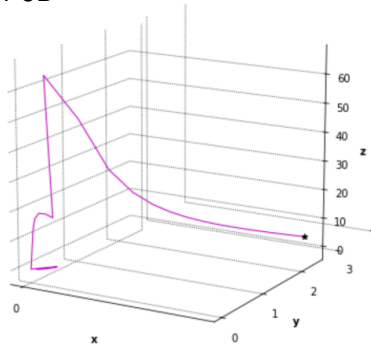
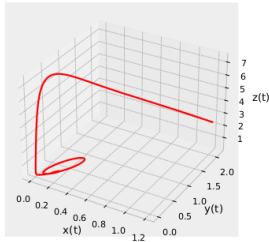
# SIMULACIONES NUMÉRICAS

S2 con valores iniciales [0.3 2.4 3.9]



# SIMULACIONES NUMÉRICAS

S4 3D



# SIMULACIONES NUMÉRICAS

Código del método de Euler en Python:

```
def euler(f, c, h, time):  
    x = [c]  
    t = [0]  
    n = int(time//h)  
  
    for i in range(n):  
        x.append(x[i] + h*f(t, x[i]))  
        t.append(t[i]+h)  
  
    x = np.array(x)  
    return t, x
```

# SIMULACIONES NUMÉRICAS

## Código del método de Runge-Kutta en Python:

```
def runge_kutta(f, c, h, time):  
    x = [c]  
    t = [0]  
    n = int(time//h)  
  
    for i in range(n):  
        k1 = f(t[i], x[i])  
        k2 = f(t[i] + h/2, x[i] + 0.5*h*k1)  
        k3 = f(t[i] + h/2, x[i] + 0.5*h*k2)  
        k4 = f(t[i] + h, x[i] + h*k3)  
        t.append(t[i] + h)  
        x.append(x[i] + h/6*(k1 + 2*k2 + 2*k3 + k4))  
  
    x = np.array(x)  
    return t, x
```

## CONCLUSIONES

- Este estudio proporciona una comprensión más profunda de las complejas interacciones entre los depredadores y sus presas en los ecosistemas naturales.
- Hemos aprendido a resolver sistemas de ecuaciones diferenciales utilizando diferentes métodos numéricos, como el método de Euler o el método de Runge-Kutta.
- Hemos estudiado la estabilidad de sistemas dinámicos, utilizando técnicas como el análisis de estabilidad lineal y el método de Routh-Hurwitz, y hemos desarrollado habilidades investigativas, como la capacidad de buscar y analizar información relevante, trabajar en equipo, elaborar y presentar informes técnicos, y utilizar herramientas matemáticas y de programación para resolver problemas.

## RECOMENDACIONES

Creemos que es importante seguir explorando los modelos matemáticos y las técnicas de simulación numérica para entender mejor los sistemas dinámicos y predecir su comportamiento en diferentes áreas, como la biología, la física, la ingeniería y la economía.



## BIBLIOGRAFÍA



C. Henry Edwards, David E. Penney y David Calvis.  
*Differential equations and boundary value problems:  
computing and modeling* . 4th ed. Upper Saddle River, NJ:  
Pearson Education, 2008. ISBN : 978-0-13-156107-6.



Ranjit Kumar Upadhyay y Satteluri R. K. Iyengar. *Introduction to mathematical modeling and chaotic dynamics*. eng. 2013. ISBN : 978-1-4398-9887-1 978-1-4398-9886-4.