Asignatura: Ecuaciones Diferenciales Ordinarias.
Informe de la Tarea Investigativa II. Curso 2023, eds.

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

Daniel Machado

Grupo C211
Ciencia de la Computación
Facultad de Matemática y Computación
Universidad de La Habana. Cuba

Daniel Toledo

Grupo C211
Ciencia de la Computación
Facultad de Matemática y Computación
Universidad de La Habana. Cuba

Osvaldo Moreno

Grupo C211
Ciencia de la Computación
Facultad de Matemática y Computación
Universidad de La Habana. Cuba

José Antonio Concepción

Grupo C211
Ciencia de la Computación
Facultad de Matemática y Computación
Universidad de La Habana. Cuba

Adrián Navarro

Grupo C211
Ciencia de la Computación
Facultad de Matemática y Computación
Universidad de La Habana. Cuba

RESUMEN

El artículo Local Analysis of the Prey-Predator Model with Stage-Structure Prey and Holling Type Functional Responses realiza un análisis local exhaustivo del modelo depredador-presa con estructura de etapas en la población de presa y respuestas funcionales de Holling de tipo II y I. Los autores determinan los tres posibles equilibrios, analizan la estabilidad local a través de la matriz jacobiana, realizan simulaciones numéricas y obtienen varios resultados clave. Concluyen que el equilibrio trivial siempre es inestable, el equilibrio con predadores extintos es estable bajo ciertas condiciones y el equilibrio interior puede ser estable o inestable dependiendo de los parámetros. Las simulaciones numéricas corroboran estos resultados y muestran cómo las poblaciones evolucionan hacia uno de los tres puntos de equilibrio dependiendo de las condiciones iniciales y los valores de parámetros. En este trabajo se realiza un análisis crítico de los resultados obtenidos y se realizan sugerencias para mejorar artículo.

1. INTRODUCCIÓN

El artículo titulado "Local Analysis of the Prey-Predator Model with Stage-Structure Prey and Holling Type Functional Responses"por Dian Savitri fue publicado en 2019 en el Journal of Physics: Conference Series. Su factor de impacto tiene una puntuación de 0.227 en el año de publicación según el SCImago

Journal Rank. El artículo analiza el modelo depredador-presa con dos tipos de presa en estructura de etapas y un solo depredador. La población de presas se divide en adultos y juveniles. El depredador exhibe diferentes tasas de depredación para adultos e inmaduros. Se utilizan respuestas funcionales de Holling [1] tipo II para adultos y tipo I para juveniles. El estudio tuvo los siguientes objetivos: determinar los puntos de equilibrio del modelo, analizar la estabilidad local de los puntos de equilibrio utilizando la matriz jacobiana y autovalores y observar el comportamiento dinámico del modelo mediante simulaciones numéricas. Además utilizaron las siguientes técnicas para obtener resultados más rigurosos: análisis matemático para determinar los puntos de equilibrio y condiciones de estabilidad, cálculo de la matriz jacobiana y autovalores 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 estudio de la bifurcación queda propuesto para trabajos posteriores.

2. MODELO PROPUESTO

El artículo analiza previamente tres modelos matemáticos que abordan las interacciones presa-depredador con diferentes enfoques. El primer modelo propuesto por Huenchucona considera dos presas y un depredador, utilizando una función de respuesta de tipo Beddington-DeAngelis [2]. El modelo de Savitri y Abadi tiene en cuenta la estructura por etapas de la presa, utilizando funciones de respuesta diferentes para presas inmaduras y maduras [3]. Luego, el modelo de Castellanos y Chan-López estudia una cadena trófica de tres niveles, donde la presencia del depredador superior afecta la estabilidad del sistema [4]. Los sistemas de ecuaciones de dichos modelos (5, 6, 7 respectivamente) pueden ser consultadas en los Anexos A.

Estos modelos constituyen una base para el análisis del sistema propuesto que está dado por los siguientes sistemas de ecuaciones diferenciales donde se definen *x(t)*, *y(t)* y *z(t)* como la densidad de la población de presas jóvenes, presas adultas y depredadores respectivamente. La primera ecuación describe como las presas juveniles crecen con la tasa de crecimiento intrínseca r haciendo que aumente su población. La tasa de depredación para ambas poblaciones de presas es diferente. Se asume que el depredador puede asechar tanto a las presas juveniles como a las presas adultas en el modelo estructurado. La tasa de depredación del depredador utiliza el Holling tipo II para la especie de presa adulta y el Holling tipo I para la presa juvenil. La tasa de depredación se da en la forma original de la respuesta funcional de tipo II de Holling, donde η y m son el valor máximo de la tasa de reducción per cápita de y debido a z y el coeficiente de protección ambiental para la presa adulta, respectivamente. La tasa de cambio en las densidades de población de presas juveniles por las tasas de crecimiento intrínsecas se reducen por la tasa a la que las presas juveniles son consumidas por los depredadores. Las poblaciones de presas juveniles crecen con ecuaciones logísticas que se ven limitadas por la capacidad de carga *(k)* para mantener sus poblaciones.

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

El índice de depredación de presas juveniles es proporcional a la tasa de encuentro entre depredadores y presas juveniles, también conocido como la ecuación de Lotka-Volterra. Los parámetros $r,\beta,\alpha,\rho,m,\mu,\eta,\alpha_1,\eta_1$ son positivos. Los cambios en la densidad de población de las presas adultas representan el crecimiento de presas jóvenes a presas adultas. El parámetro η en el proceso de depredación provoca una disminución en la población de presas adultas, lo que se interpreta como la tasa de encuentros con presas adultas por unidad de densidad de depredadores. Sin depredación, las presas adultas experimentan un proceso natural de muerte. La tasa de depredación de los depredadores sigue la respuesta funcional del tipo II de Holling. La relación $\frac{m}{\eta}$ es el tiempo promedio de procesamiento de alimentos para el depredador.

El parámetro ρ es la tasa de crecimiento de la población de depredadores. Se asume que la tasa de cambio en la población de depredadores disminuye siguiendo el modelo de Leslie-Gower [5]. El parámetro η_1 es la relación del crecimiento intrínseco dividido por el factor de conversión de depredación a presas adultas. Esto afecta la disponibilidad de alimentos favoritos por individuo, que está determinada por la capacidad de los recursos ambientales y es proporcional a la abundancia de sus alimentos favoritos. Se tiene además que $\rho - \frac{\eta_1}{m} < 0$ con $x = y = 0$, es decir, los depredadores disminuyen en el tiempo a falta de alimento.

2.1. Respuestas funcionales de Holling utilizadas

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.

3. PUNTOS DE EQUILIBRIO Y ESTABILIDAD LOCAL

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

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

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 $E_3 = (x^*, y^*, z^*)$.

Para realizar el análisis de los puntos de equilibrio hallamos la matriz Jacobina del modelo, la cual está dada por:

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

Dicha ecuación fue además verificada mediante la implementación de un script en Python (véase en los Anexos A). Para determinar los puntos de equilibrios y clasificarlos en estables o inestables, es necesario encontrar los valores propios de la matriz jacobiana. Estos valores propios son los que determinan la estabilidad del punto de equilibrio. Si todos los valores propios tienen parte real negativa, entonces el punto de equilibrio es estable. Si al menos uno de los valores propios tiene parte real positiva, entonces el punto de equilibrio es inestable.

Al realizar el cálculo de los auto valores en los distintos puntos de equilibrio obtenemos resultados discrepantes a los obtenidos por Savitri. Al evaluar la matriz jacobiana en E_1 obtuvimos:

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

Luego calculamos las raíces del polinomio característico de $J(E_1)$:

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

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.

Para el punto E_2 tenemos:

$$\det\left(J\left(\frac{k(r-\beta)}{r}, \frac{\beta k(r-\beta)}{\mu r}, 0\right) - \lambda I\right) = 0$$

$$J(E_2) = \begin{vmatrix} (\beta - r) - \lambda & 0 & \frac{\alpha k(\beta - r)}{r} \\ \beta & -\mu - \lambda & \frac{\eta\beta k(\beta - r)}{r\mu\left(\frac{\beta k(r-\beta)}{\mu r} + m\right)} \\ 0 & 0 & -\frac{\alpha_1 k(\beta - r)}{r} - \lambda \end{vmatrix} = 0 \tag{3}$$

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.

Para E_3 tenemos la siguiente matriz jacobiana:

$$J(x^*, y^*, z^*) = \begin{bmatrix} \mathrm{r} - \frac{2rx^*}{k} - \beta - \alpha z^* & 0 & -\alpha x^* \\ \beta & -\frac{\eta z^*}{y^*+m} + \frac{\eta y z^*}{(y^*+m)^2} - \mu & -\frac{\eta y^*}{y^*+m} \\ \alpha_1 z^* & \frac{\eta_1 z^{*2}}{(y^*+m)^2} & 2\rho z^* - \frac{2\eta_1 z^*}{y^*+m} + \alpha_1 x^* \end{bmatrix} \tag{4}$$

Hallando los coeficientes del polinomio característico aplicando el criterio de Routh-Horwitz (ver Apéndice D en [4]) se obtienen:

$$A_0 = 1$$
$$A_1 = -r + \tfrac{2rx^*}{k} + \beta + \alpha z^* + \tfrac{\eta z^*}{y^*+m} - \tfrac{\eta y^* z^*}{(y^*+m)^2} + \mu - 2\rho z^* + \tfrac{2\eta_1 z^*}{y^*+m} - \alpha_1 x^*$$
$$A_2 = \alpha\alpha_1 x^* z^* + \tfrac{\eta\eta_1 (z^*)^2 y}{(y^*+m)^3} + (-\tfrac{\eta z^*}{y^*+m} + \tfrac{\eta y z^*}{(y^*+m)^2} - \mu)(2\rho z^* - \tfrac{2\eta_1 z^*}{y^*+m} + \alpha_1 x^*) + (\mathrm{r} - \tfrac{2rx^*}{k} - \beta - \alpha z^*)(-\tfrac{\eta z^*}{y^*+m} + \tfrac{\eta y z^*}{(y^*+m)^2} - \mu + 2\rho z^* - \tfrac{2\eta_1 z^*}{y^*+m} + \alpha_1 x^*)$$
$$A_3 = -\alpha x^*((-\tfrac{\eta z^*}{y^*+m} + \tfrac{\eta y z^*}{(y^*+m)^2} - \mu)(\alpha_1 z^*) - (\beta)(\tfrac{\eta_1 z^{*2}}{(y^*+m)^2})) + (\mathrm{r} - \tfrac{2rx^*}{k} - \beta - \alpha z^*)((-\tfrac{\eta y^*}{y^*+m})(\tfrac{\eta_1 z^{*2}}{(y^*+m)^2}) - (-\tfrac{\eta z^*}{y^*+m} + \tfrac{\eta y z^*}{(y^*+m)^2} - \mu)(2\rho z^* - \tfrac{2\eta_1 z^*}{y^*+m} + \alpha_1 x^*))$$

donde A_0, A_1, A_2 y A_3 satisfacen:

$$P(\lambda) = A_0\lambda^3 + A_1\lambda^2 + A_2\lambda + A_3$$

y $A_0 > 0$, $A_1 > 0$, $A_2 > 0$, $A_3 > 0$ y $A_1A_2 - A_0A_3 > 0$ son las condiciones necesarias y suficientes para que las raíces de la ecuación $P(\lambda) = 0$ sean negativas o tengan partes reales negativas. Utilizando los valores de los parámetros brindados por el autor se obtiene que se cumplen las condiciones anteriores, por lo que concluyen que el punto E_3 es asintóticamente estable. Sin embargo, experimentando con una pequeña variación de los valores iniciales se obtienen resultados contradictorios pues la matriz de Hurwitz no es definida positiva. Para el parámetro $\eta = 0{,}6$ y los restantes valores de S2 (1) se llega a una aparente estabilidad $E_3 = (0{,}1194522936; 0{,}1667424272; 0{,}3393851516)$, pues se obtienen valores propios con parte real negativa $\lambda_1 = -0{,}0519514704367125 + 0{,}351175046207591i, \lambda_2 = -0{,}0519514704367125 - 0{,}351175046207591 i and \lambda_3 = -0{,}445962288036575$. Además, variando el valor de η ocurren cambios en las condiciones de estabilidad.

4. SIMULACIONES NUMÉRICAS

Para las simulaciones numéricas se utilizaron los mismos parámetros propuestos por el artículo para comparar los resultados obtenidos. Dichos valores están dados por la Tabla 1.

Cuadro 1: Parámetros

	r	β	α	α_1	η	η_1	K	ρ	m	μ
S1	0.82	0.87	1.56	1.12	2.41	1.83	12	1.38	0.13	0.11
S2	1.32	0.87	1.16	0.72	1.6	0.41	2.8	0.78	0.23	0.11
S3	1.32	0.87	0.76	0.72	0.6	0.41	2.8	0.78	0.23	0.11
S3	1.32	0.87	1.16	0.72	0.12	0.41	2.8	0.78	0.23	0.11
S3	1.32	0.87	1.16	0.72	0.3095	0.41	2.8	0.78	0.23	0.11

4.1. Comparativas

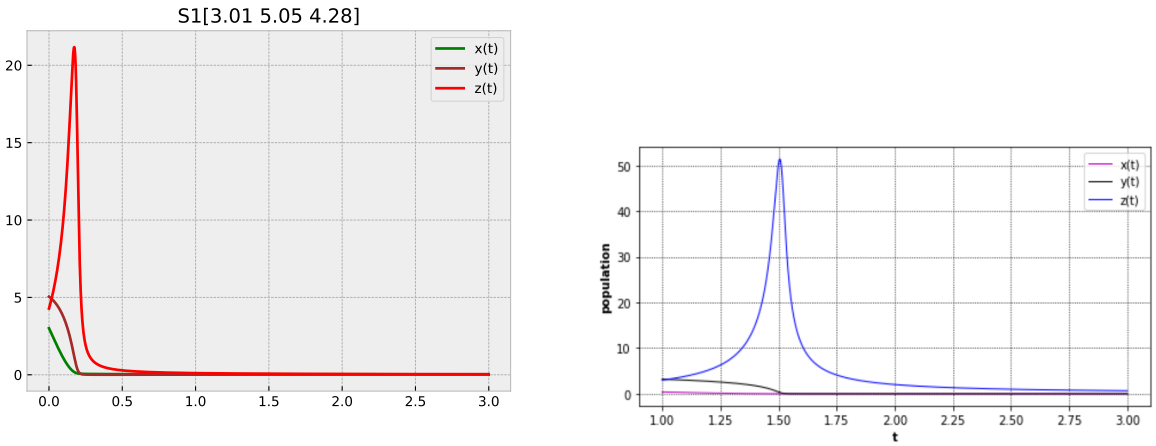

Figura 1: S1 con valores iniciales [3.01 5.05 4.28]

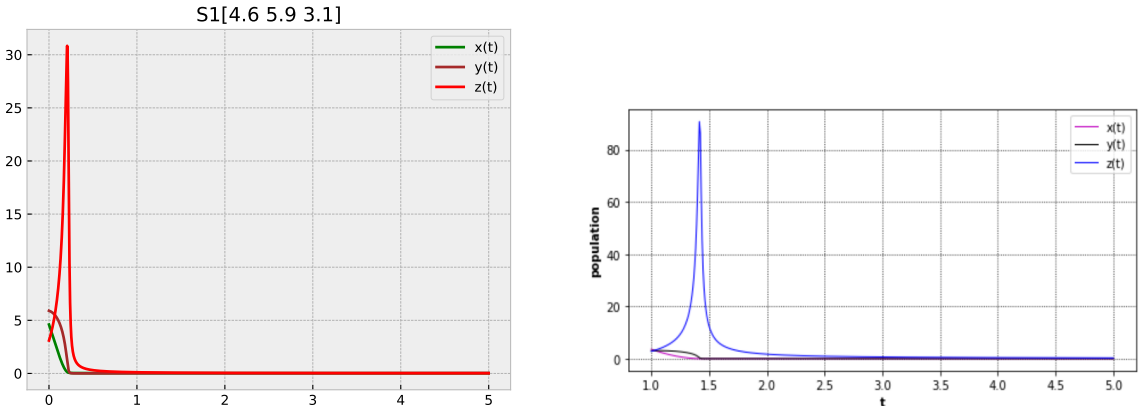

Figura 2: S1 con valores iniciales [4.6 5.9 3.1]

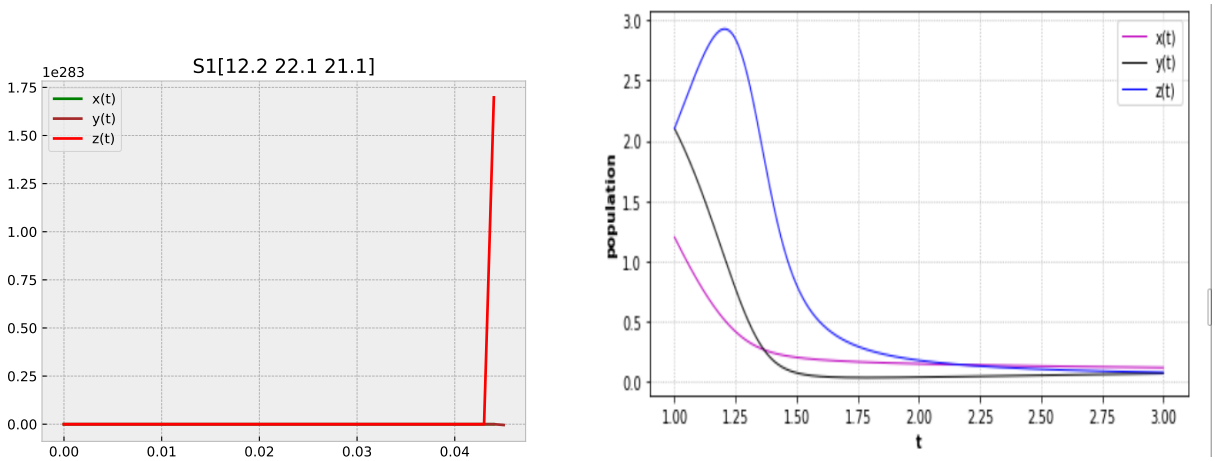

Figura 3: S1 con valores iniciales [12.2 22.1 21.1], se experimentaron problemas de overflow.

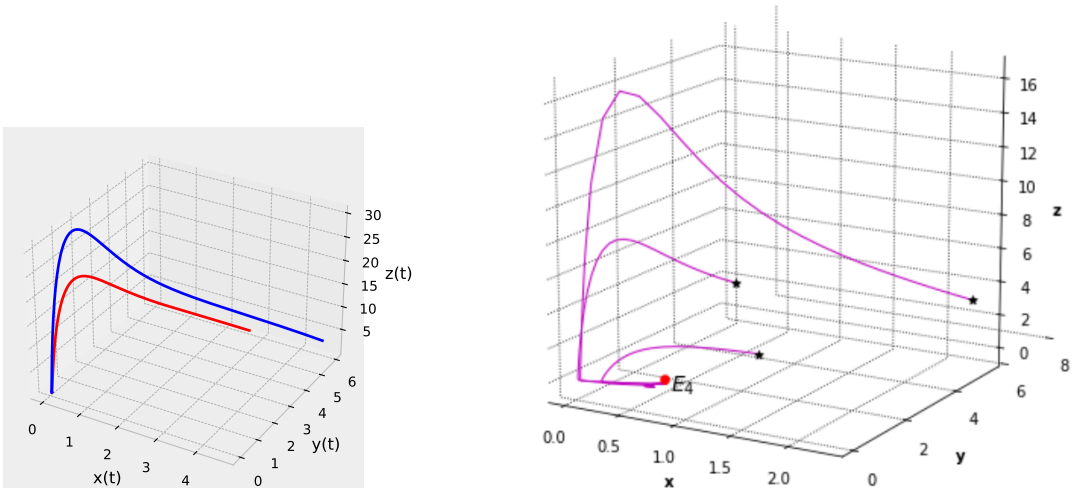

Figura 4: S1 3D, se experimentaron problemas de overflow.

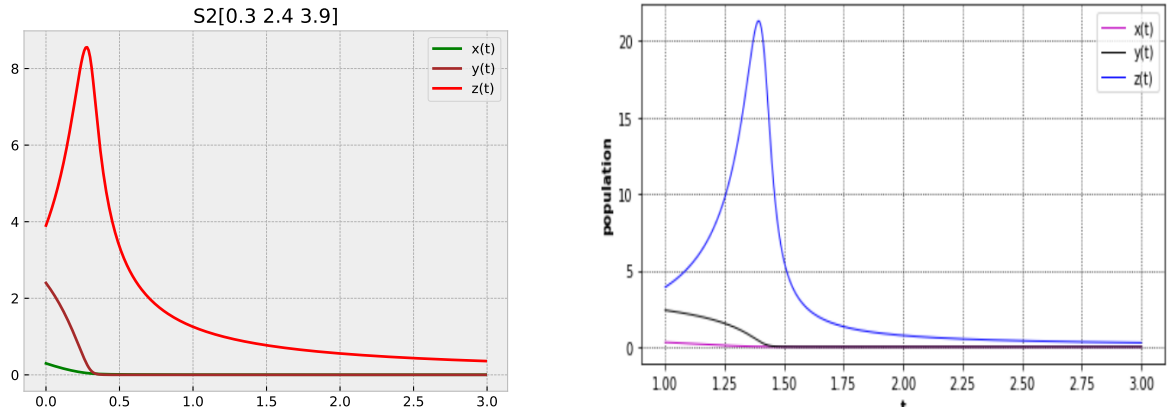

Figura 5: S2 con valores iniciales [0.3 2.4 3.9]

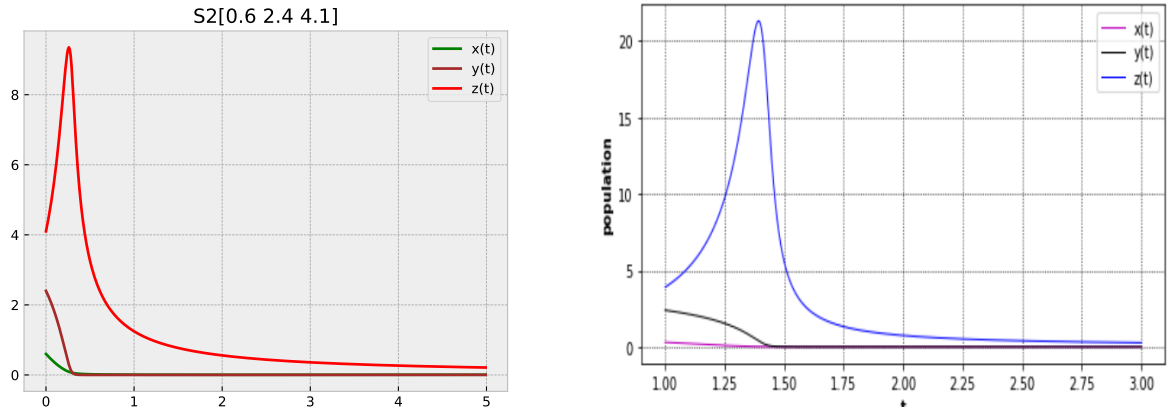

Figura 6: S2 con valores iniciales [0.6 2.4 4.1]

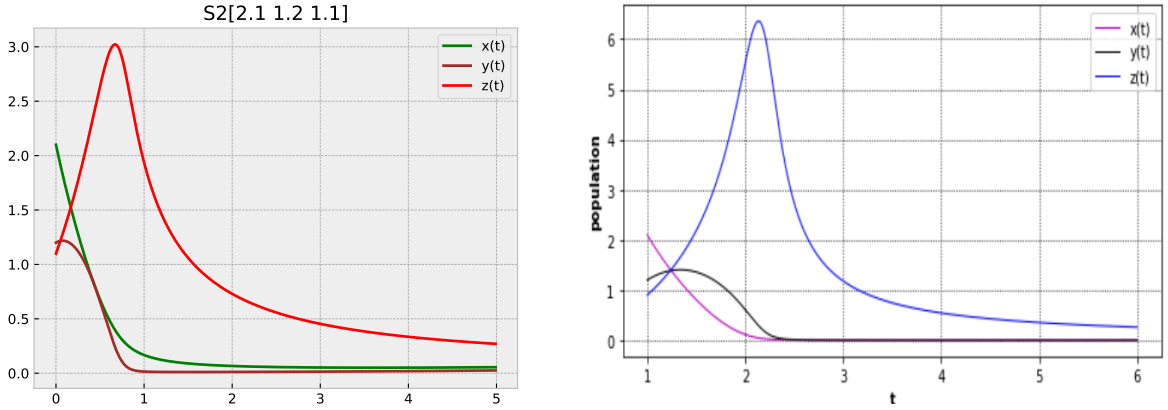

Figura 7: S2 con valores iniciales [2.1 1.2 1.1]

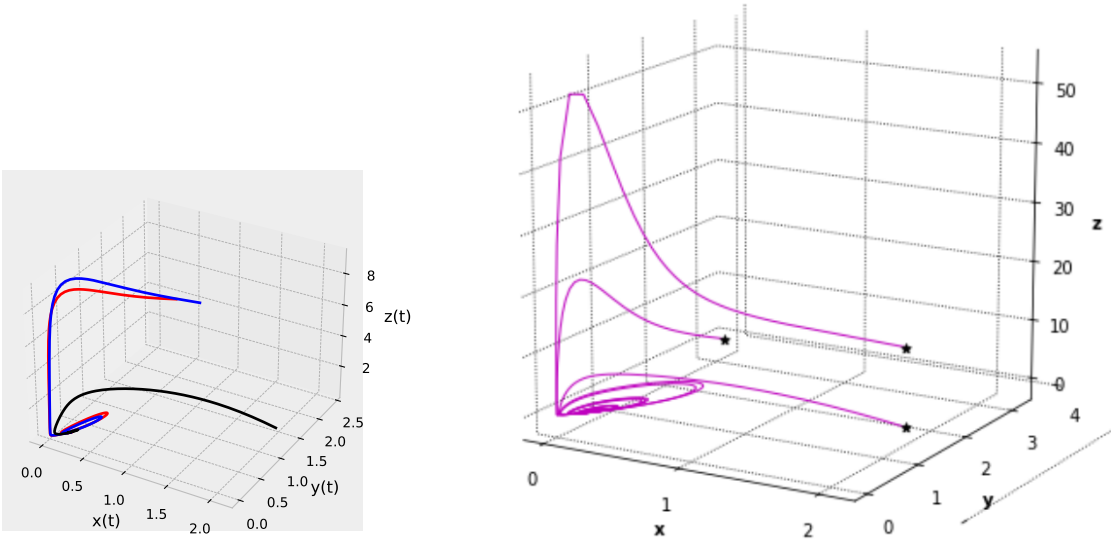

Figura 8: S2 con valores iniciales

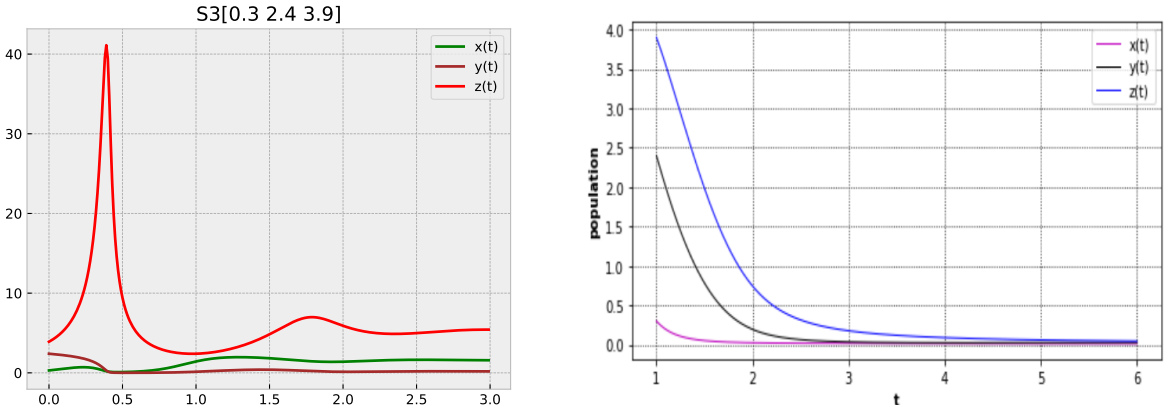

Figura 9: S3 con valores iniciales [0.3 2.4 3.9]

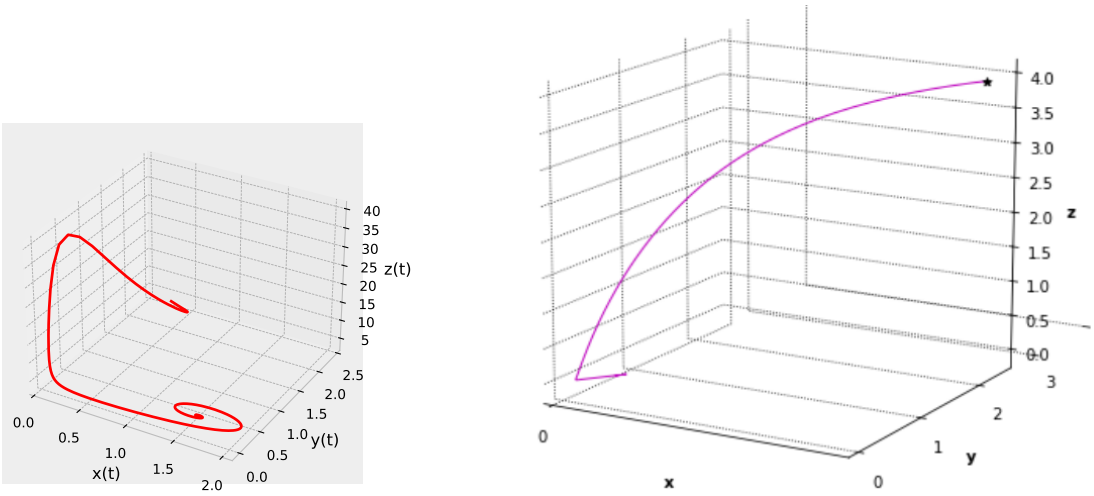

Figura 10: S3 3D

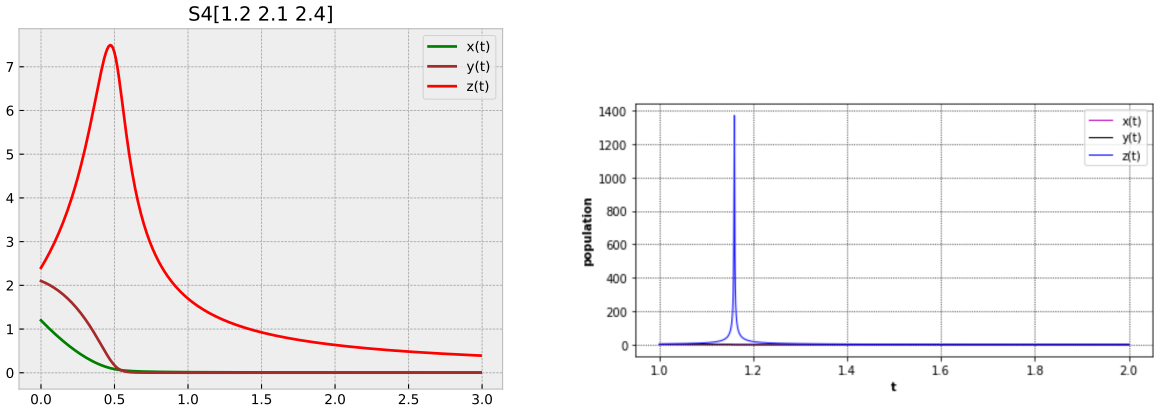

Figura 11: S4 con valores iniciales [1.2 2.1 2.4]

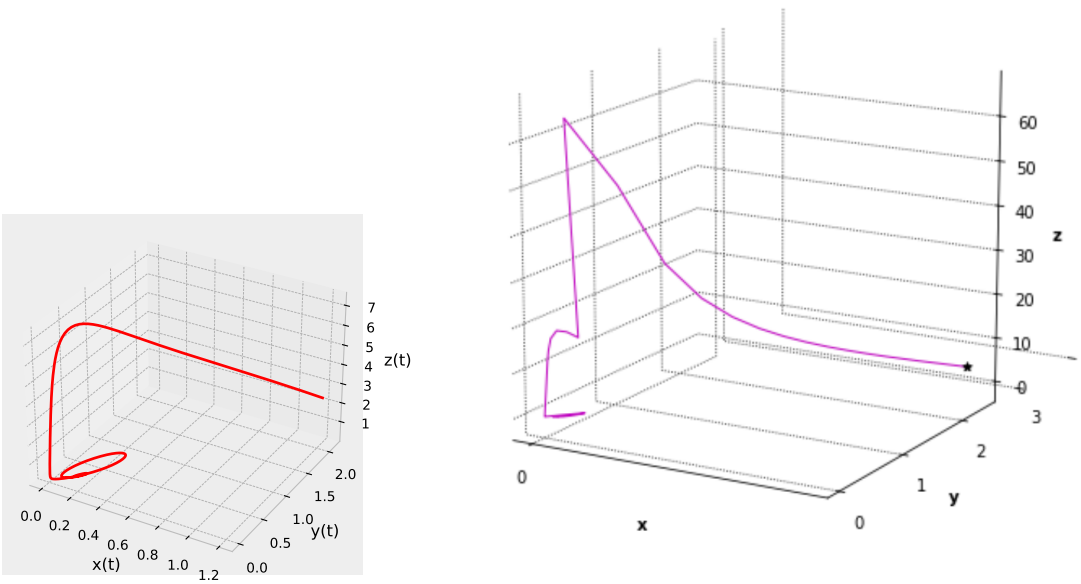

Figura 12: S4 3D

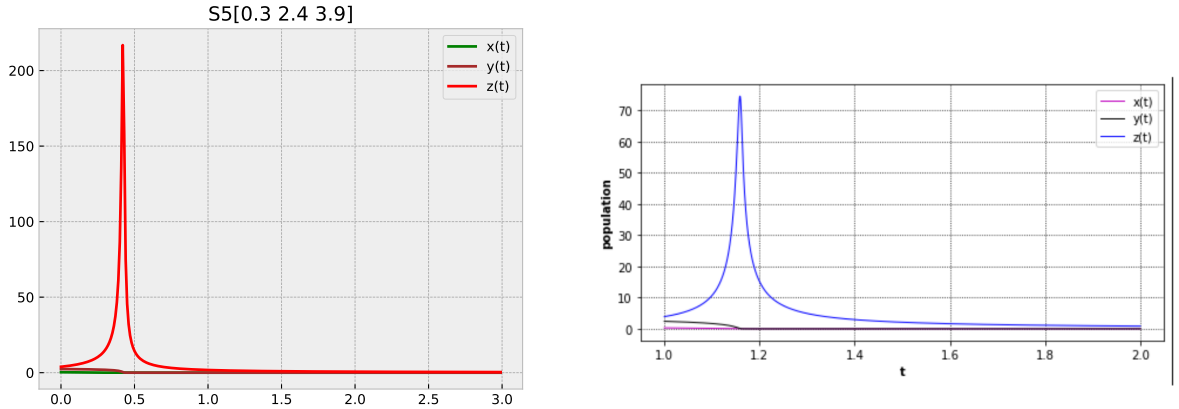

Figura 13: S5 con valores iniciales [0.3 2.4 3.9]

4.2. Métodos y algoritmos utilizados

Códigos del método de Euler y de Runge-Kutta utilizados para la solución del sistema:

```
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



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


# Parametros del sistema (positivos)
r = 0.82
alpha = 1.56
beta = 0.87
rho = 1.38
m = 0.13
mu = 0.11
eta = 2.41
alpha1 = 1.12
eta1 = 1.83
k = 12


def f1(x, y, z):
return x*(r*(k-x)/k-beta-alpha*z)
```

```
def f2(x, y, z):
return beta*x-y*((eta*z)/(y+m)-mu)


def f3(x, y, z):
return z*(alpha1*x+rho*z-(eta1*y)/(y+m))
```

CONCLUSIONES

Este estudio proporciona una comprensión más profunda de las complejas interacciones entre los depredadores y sus presas en los ecosistemas naturales. Los resultados obtenidos pueden ser útiles para predecir y manejar las poblaciones de presas y depredadores en diferentes entornos y para comprender mejor los efectos del cambio climático y otros factores ambientales en estas interacciones.

Durante este trabajo hemos aprendido sobre varios temas relacionados con las ecuaciones diferenciales y el modelado matemático. Hemos aprendido a resolver sistemas de ecuaciones diferenciales utilizando diferentes métodos numéricos, como el método de Euler o el método de Runge-Kutta. Además, hemos aprendido a modelar la interacción entre dos especies, analizando la dinámica de este modelo y estudiando su estabilidad. También hemos aprendido a utilizar las matemáticas para modelar problemas del mundo real y la importancia de los modelos matemáticos en varias disciplinas. Hemos utilizado diferentes programas de simulación numérica, como Python, para resolver ecuaciones diferenciales y visualizar los resultados, lo que nos ha ayudado a entender mejor los modelos matemáticos y hacer predicciones sobre el comportamiento de los sistemas. 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.

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. Este trabajo nos ha dado una buena base para continuar explorando estos temas en el futuro y desarrollar nuevas investigaciones en este campo.

REFERENCIAS

[1] C. S. Holling. «The Functional Response of Predators to Prey Density and its Role in Mimicry and Population Regulation». en. En: *Memoirs of the Entomological Society of Canada* 97.S45 (1965), págs. 5-60. ISSN: 0071-075X. DOI: 10.4039/entm9745fv. URL: https://www.cambridge.org/core/product/identifier/S0071075X00000862/type/journal_article.

[2] Manuel Falconi, Marcelo Huenchucona y Claudio Vidal. «Stability and global dynamic of a stage-structured predator–prey model with group defense mechanism of the prey». en. En: *Applied Mathematics and Computation* 270 (nov. de 2015), págs. 47-61. ISSN: 00963003. DOI: 10.1016/j.amc.2015.07.109. URL: https://linkinghub.elsevier.com/retrieve/pii/S0096300315010310.

[3] Dian Savitri y Abadi Abadi. «Dynamics and Numerical Simulation of Stage Structure Prey-Predator Models». en. En: (2018). DOI: 10.2991/icst-18.2018.168. URL: https://www.atlantis-press.com/article/55910953.

[4] Víctor Castellanos y Ramón E. Chan-López. «Existence of limit cycles in a three level trophic chain with Lotka–Volterra and Holling type II functional responses». en. En: *Chaos, Solitons & Fractals* 95 (feb. de 2017), págs. 157-167. ISSN: 09600779. DOI: 10.1016/j.chaos.2016.12.011. URL: https://linkinghub.elsevier.com/retrieve/pii/S0960077916303708.

[5] P. H. Leslie y J. C. Gower. «The properties of a stochastic model for the predator-prey type of interaction between two species». en. En: *Biometrika* 47.3-4 (1960). ISSN: 0006-3444, 1464-3510. DOI: 10.1093/biomet/47.3-4.219. URL: https://academic.oup.com/biometet/article-lookup/doi/10.1093/biomet/47.3-4.219.

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.

AGRADECIEMIENTOS

Agradecemos al claustro de profesores de la asignatura por su esfuerzo y tiempo empleado en nosotros.

A. ANEXOS

Modelo propuesto por Huenchucona, Falconi y Vidal

$$\begin{aligned}\frac{dx_1}{dt} &= rx_1\left(1-\frac{x_1}{K}\right)-\beta x_1-\frac{\beta_1 x_1 y}{1+m_1x_1^2+n_1x_2}\\ \frac{dx_2}{dt} &= \beta x_1-\frac{\beta_2 x_2 y}{1+m_2x_1+n_2x_2}-\mu_1 x_2\\ \frac{dy}{dt} &= \frac{\alpha_1\beta_1 x_1 y}{1+m_1x_1^2+n_1x_2}+\frac{\alpha_2\beta_2 x_2 y}{1+m_2x_1+n_2x_2}-\mu_2 y\end{aligned} \tag{5}$$

Modelo propuesto por Savitri y Abadi

$$\begin{aligned}\frac{dx_1}{dt} &= rx_1\left(1-\frac{x_1}{K}\right)-\beta x_1-\alpha x_1 y\\ \frac{dx_2}{dt} &= \beta x_1-\frac{\varepsilon x_2 y}{1+mx_2}-\mu_1 x_2\\ \frac{dy}{dt} &= \frac{\gamma\varepsilon x_2 y}{1+mx_2}-\mu_2 y\end{aligned} \tag{6}$$

Modelo propuesto por Castellanos y Chan-López

$$\begin{aligned}\frac{dx}{dt} &= \rho x\left(1-\frac{x}{k}\right)-a_1x\\ \frac{dy}{dt} &= ca_1xy-dy-\frac{a_2yz}{y+b_2}\\ \frac{dz}{dt} &= \alpha z^2-\frac{\beta z^2}{y+b_2}\end{aligned} \tag{7}$$

Script en Python para calcular la Jacobiana.

```
# Define the variables
x, y, z, r, k, b, a, n, m, u, a1, p, n1 = symbols('x y z r k b a n m u a1 p n1')

# Define the equations
```

```
eq1 = r*x*(1-x/k) - b*x - a*x*z
eq2 = b*x - (n*y*z)/(y+m) - u*y
eq3 = a1*x*z + p*z**2 - (n1*z**2)/(y+m)

# Verify if E2 is an equilibrium point
dx = sp.simplify(sp.Subs(eq1, (x,y,z), (((k*(r-b))/r),
 (b*k*(r-b))/(u*r), 0)).doit())
dy = sp.simplify(sp.Subs(eq2, (x,y,z), (((k*(r-b))/r),
 (b*k*(r-b))/(u*r), 0)).doit())
dz = sp.simplify(sp.Subs(eq3, (x,y,z), (((k*(r-b))/r),
 (b*k*(r-b))/(u*r), 0)).doit())
print("E2 is an equilibrium point:", {dx==0 and dy==0 and dz==0})

separator = '- - '*25
print(separator)

# Define the variables as a list
variables = [x, y, z]

# Define the equations as a list
equations = [eq1, eq2, eq3]

# Create the Jacobian matrix
Jacobian = Matrix(equations).jacobian(variables)
Jacobian = sp.simplify(Jacobian)
print("Jacobian matrix:")
print(Jacobian)

print(separator)

# Verify the local stability of the equilibrium point E1
E1=sp.Subs(Jacobian, (x,y,z), (0,0,0)).doit()
E1=sp.simplify(E1)
eigenvaluesE1 = list(sp.Matrix.eigenvals(E1, simplify=True, multiple=True))
print(f"Jacobian(E1): {E1}")
print(f"Eigenvalues of Jacobian(E1): {eigenvaluesE1}")

print(separator)

# Verify the local stability of the equilibrium point E2
E2=sp.Subs(Jacobian, (x,y,z), (((k*(r-b))/r), (b*k*(r-b))/(u*r), 0)).doit()
E2=sp.simplify(E2)
eigenvaluesE2 = list(sp.Matrix.eigenvals(E2).keys())
print(f"Jacobian(E2): {E2}")
print(f"Eigenvalues of Jacobian(E2): {eigenvaluesE2}")

print(separator)
```

```
# Checking the Routh-Hurwitz criterion for the E3 stability
A1=sp.simplify(-Jacobian[0,0]-Jacobian[1,1]-Jacobian[2,2])
print(f"A1: {A1}")

print(separator)

A2=sp.simplify(-Jacobian[0,1]*Jacobian[1,0]-Jacobian[0,2]*Jacobian[2,0]
-Jacobian[1,2]*Jacobian[2,1] +Jacobian[1,1]*Jacobian[2,2]
+Jacobian[0,0]*(Jacobian[1,1]+Jacobian[2,2]))
print(f"A2: {A2}")

print(separator)

A3=sp.simplify(Jacobian[0,2]*(Jacobian[1,1]*Jacobian[2,0]-Jacobian[1,0]
*Jacobian[2,1])+Jacobian[0,1]*(Jacobian[1,0]*Jacobian[2,2]
-Jacobian[1,2]*Jacobian[2,0])+Jacobian[0,0]*(Jacobian[1,2]*Jacobian[2,1]
-Jacobian[1,1]*Jacobian[2,2]))
print(f"A3: {A3}")

print(separator)

Hurwitz = Matrix([[0,0,0],[0,0,0],[0,0,0]])
Hurwitz[0,0]=A1
Hurwitz[0,1]=A3
Hurwitz[1,0]=1
Hurwitz[1,1]=A2
Hurwitz[2,1]=A1
Hurwitz[2,2]=A3

H1=A1
# H1Subs=sp.Subs(H1, (r, k, b, a, n, m ,u, a1, p, n1),
(0.82,12,0.87, 1.56, 2.41, 0.13, 0.11, 1.12, 1.38, 1.83)).doit()
print(f"H1: {H1}")

print(separator)

H2=A1*A2-A3
# H2Subs=sp.Subs(H2, (r, k, b, a, n, m ,u, a1, p, n1),
(0.82,12,0.87, 1.56, 2.41, 0.13, 0.11, 1.12, 1.38, 1.83)).doit()
print(f"H2: {H2}")

print(separator)

H3=A1*A2*A3-A3*A3
# H3Subs=sp.Subs(H3, (r, k, b, a, n, m ,u, a1, p, n1),
(0.82,12,0.87, 1.56, 2.41, 0.13, 0.11,
 1.12, 1.38, 1.83)).doit()
print(f"H3: {H3}")
```