



<https://creativecommons.org/licenses/by/4.0/>

MÉTODOS NUMÉRICOS RUNGE-KUTTA Y ADAMS BASHFORTH-MOULTON EN MATHEMATICA

Runge-Kutta and Adams Bashforth-Moulton numerical methods with mathematica

JAIME SEGARRA-ESCANDÓN¹

Recibido:09 de junio de 2020. Aceptado:30 de junio de 2020

DOI: <http://dx.doi.org/10.21017/rimci.2020.v7.n14.a81>

RESUMEN

En este estudio, el objetivo principal es realizar el análisis de los métodos numéricos Runge-Kutta y Adams Bashforth-Moulton. Para cumplir con el objetivo se utilizó el sistema de ecuaciones diferenciales del modelo Lotka-Volterra y se usó el software matemático Wolfram Mathematica. En los resultados se realiza la comparación de los métodos RK4, AB4 y AM4 con el comando NDSolve utilizando el modelo Lotka-Volterra. Los resultados obtenidos en los diagramas de fase y la tabla de puntos de la iteración indicaron que el método RK4 tiene mayor precisión que los métodos AB4 y AM4.

Palabras clave. Ecuaciones diferenciales; Enseñanza de las matemáticas; Tecnología; Mathematica.

ABSTRACT

In this research, the main objective is to study the Runge -Kutta and Adams Bashforth- Moulton numerical methods. To fulfill the purpose of this study, the Systems of differential equations of the lotka-Volterra model was used and the mathematical software Wolfram Mathematica was used. In the results, the RK4, AB4 and AM4 methods are compared with the NDSolve command using the Lotka-Volterra model. The results obtained in the phase diagrams and the table of points of the iteration indicated that the RK4 method has higher precision than the AB4 and AM4 methods.

Keywords . Differential equations; Mathematics training; Technology; Mathematica.

I. INTRODUCCIÓN

LAS ECUACIONES diferenciales ordinarias (EDO) posibilitan describir fenómenos basados en la variación y, por tanto, permiten modelizar y resolver problemas procedentes de contextos muy diversos [1]. Debido a la importancia de la aplicación de las ecuaciones diferenciales, diversos investigadores estudiaron la modelización y las aplicaciones de las ecuaciones diferenciales utilizando software informático (e.g., [2], [3], [4], [5]). Por otro lado, algunos investigadores estudiaron el proceso de enseñanza aprendizaje de las ecuaciones diferenciales ordinarias (e.g., [1], [6], [7], [8]).

Adicionalmente, se tiene que el uso de la tecnología en la modelación matemática es importante y muy relevante, porque brinda ayuda a los usuarios a la visualización de relaciones de la matemática, permitiendo establecer representaciones exactas de congruaciones geométricas, porque en este contexto, los usuarios tienen la oportunidad de mover partes de estas congruaciones y observar las invariantes que se producen en una construcción matemática [9].

Sin duda, la tecnología, está transformando la educación notablemente, ha cambiado tanto la forma de enseñar como la forma de aprender y por

¹ Departamento de Matemáticas e Informática de la Seguridad, Universidad Rovira i Virgili, Tarragona, España. ORCID: 0000-0003-4304-2385. Correo electrónico: jaimerodrigo.segarra@urv.cat

supuesto el rol del maestro y el estudiante, al mismo tiempo que cambian los objetivos formativos para los estudiantes dado que estos tendrán que formarse para utilizar, usar y producir con los nuevos medios, además, el docente tendrá que cambiar sus estrategias de comunicación y asumir su función de facilitador del aprendizaje de los alumnos en entornos cooperativos y la utilización de software para ayudarlos a planificar y alcanzar los objetivos [10]. El implementar en la clase de matemáticas el uso de la tecnología puede permitir que los estudiantes mejoren la oportunidad de reflexionar sobre sus conjeturas en relación con los valores y el movimiento generado en las construcciones[11], [12].

El uso de herramientas computacionales para la enseñanza matemática trata de crear un conjunto de algoritmos que posibiliten la explotación de los medios técnicos disponibles, poniéndolos en función de informar, controlar, dirigir y evaluar la actividad del estudiante, de modo que éste pueda alcanzar los objetivos previstos [13]. Uno de los principios que plantea NCTM (National Council of Teachers of Mathematics) es que la tecnología es esencial en la enseñanza y el aprendizaje de las matemáticas; influye en las matemáticas que se enseñan y estimula el aprendizaje de los estudiantes [14].

Los investigadores mencionados anteriormente, resaltaron la importancia de estudiar las ecuaciones diferenciales ordinarias y la aplicación de la tecnología en las EDO. Por tanto, el objetivo de esta investigación es estudiar la precisión de los métodos numéricos Runge-Kutta y Adams Bashforth-Moulton. Para cumplir con el objetivo se plantea dos preguntas de investigación:

1. ¿Cuál es la evolución de precisión de los métodos de un paso (Runge-Kutta) y multipaso (Adams Bashforth-Moulton)?
2. ¿Cuál es el método (Runge-Kutta 4, Adams Bashforth-Moulton 4) que más se acerca a la solución del método NDSolve de Mathematica utilizando un modelo de Lotka-Volterra?

II. REVISIÓN DE LA LITERATURA

Esta sección presenta una breve revisión de la literatura de la enseñanza y solución numérica de

las ecuaciones diferenciales ordinarias mediante el uso de paquetes informáticos.

En [15], se realizó una investigación con un grupo de estudiantes que han recibido una formación tradicional del concepto, utilizan sus conocimientos matemáticos para resolver problemas y responder a cuestiones relacionadas con las EDO, para continuar con el diseño e implementación de un Módulo de Enseñanza para la introducción de las EDO, en un ambiente de resolución de problemas, haciendo uso de TIC (calculadora VoyageTM200). El uso de la herramienta tecnológica, la calculadora VoyageTM200, y el modelo de trabajo en el aula dieron un mayor protagonismo al estudiante en su propio proceso de aprendizaje.

Por otro lado, [16] estudió las relaciones entre los elementos que intervienen en el proceso de aprendizaje de ecuaciones diferenciales de primer orden, antes y después de la incorporación de herramientas tecnológicas de apoyo. Los resultados mostraron que el uso de las TIC como instrumento mediador produce un cambio de actitudes positivas y favorables hacia las aplicaciones de las ecuaciones diferenciales de primer orden, apoyando la adquisición de conocimientos, logrando estimular la creatividad de los estudiantes y contribuyendo en el diseño de nuevas estrategias metodológicas en el ámbito de la Educación Superior.

En su investigación, [17] mostraron algunas de las ventajas del uso de los paquetes Dsolve y NDSolve del software Mathematica, y en particular sus opciones gráficas para fortalecer la enseñanza de conceptos importantes. El autor recomienda que el alumno utilice algún software matemático como herramienta para realizar cálculos o ejemplificar conceptos gráficos toda vez que ya ha aprendido las técnicas de solución de ecuaciones y es capaz de realizar los cálculos a mano.

En otra investigación, [8] exploró las fortalezas y debilidades del paquete libre Maxima como herramienta para la enseñanza de ecuaciones diferenciales ordinarias. Se realiza una comparación entre los softwares Maxima, Maple y Mathematica. Entre las deficiencias de Maxima comparado con Maple y Mathematica, se identifica que existieron varios problemas en: tratar de resolver ecuaciones de orden alto; en resolver ecuaciones reducibles a

la ecuación de Bessel; no cuenta con la opción para resolver EDOs; utilizando soluciones en series de potencias; y también presentó algunos problemas para calcular transformadas de Laplace inversa.

Por su parte, [18] utilizó las Tecnologías de la Información y Comunicación como estrategia didáctica, particularmente en la matemática universitaria. Se presentó la experiencia de la introducción de las TICs en un curso de Cálculo II y con una implementación de ocho años de evolución. Las diferentes actividades fueron clasificadas como laboratorios, actividades complementarias y proyecto corto, las cuales están relacionadas y permiten tener diferentes escenarios para introducir, comprender e integrar los diferentes temas del curso. En conjunto, las diferentes actividades con la introducción de las TICs han funcionado como un elemento clave para promover la motivación de los estudiantes y otorgarles un papel dinámico con la guía del docente.

En una investigación más reciente, [19] pretenden que los estudiantes adquirieran capacidades en el empleo de tecnologías para que resolvieran diferentes ecuaciones diferenciales de las que se estudian, así como obtener la gráfica de su solución. Analizaron y manejaron aplicaciones como Differential Equations, Wolfram, Desmos, Photomath. Con ello, elaboraron una situación didáctica mediante la cual interactuaron estudiantes con las aplicaciones en la resolución de ecuaciones diferenciales y en la graficación. Los resultados mostraron deficiencias en la evolución de la interfaz de las aplicaciones utilizadas, cuyo uso provoca fenómenos didácticos importantes.

III. MARCO TEÓRICO

En este apartado se van a tratar conceptos de las EDO y los métodos numéricos que se utilizan en esta investigación. Posteriormente, se abordarán conceptos del software Mathematica.

Ecuaciones diferenciales

Las ecuaciones diferenciales se utilizan para modelar y comprender problemas de la vida real. De hecho, es importante para las matemáticas ya que tiene fuerte relaciones con las funciones, derivadas e integrales. Por tanto, es un campo muy

importante para los estudiantes de ingeniería. Las ecuaciones diferenciales ordinarias (EDO) son herramientas básicas para los profesionales de especialidades relacionadas con la ciencia y la tecnología, como es el caso de los ingenieros informáticos, eléctricos, mecánicos entre otros. Esto es debido a que las EDO posibilitan describir fenómenos basados en la variación y, por tanto, permiten modelizar y resolver problemas procedentes de contextos muy diversos [1].

No existe un método general que nos dé una forma explícita para encontrar la solución de una ecuación diferencial. En la práctica, se encuentra ecuaciones específicas para las que no se conoce un método de resolución o para las cuales las formas explícitas de solución no son las adecuadas para los cálculos. Por estas razones, son tan importantes los métodos sistemáticos y eficaces que nos lleven a una aproximación numérica de las soluciones [20]. Para aproximar soluciones de ecuaciones diferenciales de primer orden existen métodos de: un paso (Taylor, Euler, Runge-Kutta), multipasos (Adams-Bashforth y Adams-Moulton) y los métodos de splitting y de composición. Estos problemas admiten una única solución bajo condiciones adecuadas de la función $f(x, y)$.

$$y' = f(x, y)$$

$$y(x_0) = y_0$$

En este estudio, de forma particular, se analiza los métodos de Runge-Kutta y los métodos Adams Bashforth-Moulton.

Métodos de Runge-Kutta (RK)

Los métodos Runge-Kutta son considerados métodos de un solo paso, probablemente uno de los procedimientos numéricos más utilizados, así como uno de los más precisos, usados para obtener soluciones aproximadas para un problema con valor inicial. Es importante indicar que los métodos de Runge-Kutta son generalizaciones de la fórmula básica de Euler. Por esta razón, se dice que el método de Euler es un método de Runge-Kutta de primer orden [21]. Se puede presentar la fórmula de Runge-Kutta como:

$$y_{n+1} = y_n + hF(x_n, y_n; h)$$

F depende del orden k elegido:

$$k = 1 \rightarrow F = T_1(x, y; h) = f(x, y)$$

$$k = 2 \rightarrow F = T_2(x, y; h) = f(x, y) + \frac{h}{2} \left[\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f(x, y) \right]$$

Los métodos de Runge-Kutta utilizan las siguientes ecuaciones:

Runge-Kutta de tercer orden (RK3):

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 4k_2 + k_3)$$

donde:

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf\left(x_n + \frac{h}{2}, y_n + \frac{1}{2}k_1\right)$$

$$k_3 = hf(x_n + h, y_n + k_1 + 2k_2)$$

Por otro lado, el método Runge-Kutta de cuarto orden (RK4) utiliza la siguiente ecuación:

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

donde:

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf\left(x_n + \frac{h}{2}, y_n + \frac{1}{2}k_1\right)$$

$$k_3 = hf\left(x_n + \frac{h}{2}, y_n + \frac{1}{2}k_2\right)$$

$$k_4 = hf(x_n + h, y_n + k_3)$$

Obsérvese, que los métodos RK4 tienen una k más que los métodos RK3.

Finalmente, se tiene el método de Runge-Kutta de quinto orden (RK5) de Butcher (1964) o de orden superior.

$$y_{n+1} = y_n + \frac{1}{90}h(7k_1 + 32k_2 + 12k_3 + 32k_4 + 7k_5)$$

donde:

$$k_1 = hf(x_n, y_n)$$

$$k_2 = f\left(x_n + \frac{h}{4}, y_n + \frac{1}{4}k_1h\right)$$

$$k_3 = hf\left(x_n + \frac{h}{4}, y_n + \frac{1}{8}k_1h + \frac{1}{8}k_2h\right)$$

$$k_4 = f\left(x_n + \frac{h}{2}, y_n - \frac{1}{2}k_1h + k_3h\right)$$

$$k_5 = f\left(x_n + \frac{3}{4}h, y_n + \frac{3}{16}k_1h + \frac{9}{16}k_4h\right)$$

$$k_6 = f\left(x_n + h, y_n - \frac{3}{7}k_1h + \frac{2}{7}k_2h + \frac{12}{7}k_3h - \frac{12}{7}k_4h + \frac{8}{7}k_5h\right)$$

Los métodos RK5 tienen dos k más que los métodos RK4, por lo que el coste computacional es mayor.

Métodos Adams Bashforth-Moulton

Los métodos Adams Bashforth-Moulton son los métodos lineales multipaso o continuos más utilizados. En este artículo se estudian los métodos Adams-Bashforth (AB2, AB3 y AB4) y Adams-Moulton (AM2, AM3 y AM4). Estos métodos tienen la forma:

$$z_{n+k} - z_{n+k-1} = h[\beta_0 f_n + \dots + \beta_k f_{n+k}]$$

La mayoría de los coeficientes α_{n+j} de los métodos Adams son cero, a excepción de α_{n+k} y α_{n+k-1} , estos valores se encuentran en un intervalo $[1, 1]$. Por otro lado, a los coeficientes β_{n+j} se deben proporcionar valores para poder lograr buenas aproximaciones.

Cuando el coeficiente $\beta_k = 0$ se obtienen los métodos Adams-Bashforth (métodos explícitos).

Por otro lado, cuando $\beta_k \neq 0$ se obtienen los métodos de Adams-Moulton (método implícito).

Métodos Adams-Bashforth

Los métodos de Adams-Bashforth son métodos explícitos, estos son métodos multipaso que permiten la resolución de ecuaciones diferenciales. Para calcular y_{i+1} hacen uso de $f(t_i, y_i)$. Existen varios métodos Adams-Bashforth, algunos se detallan a continuación:

El método de Adams-Bashforth de un paso es equivalente al método de Euler, tiene la siguiente expresión:

$$z_{n+1} = z_n + hf(t_n, z_n)$$

El método de Adams-Bashforth de orden dos se define con la siguiente expresión:

$$z_{n+2} = z_{n+1} + \frac{h}{2}(3f(t_{n+1}, z_{n+1}) - f(t_n, z_n))$$

Por otro lado, el método de Adams-Bashforth de orden tres se puede definir con la siguiente expresión:

$$z_{n+3} = z_{n+2} + \frac{h}{12}(23f(t_{n+2}, z_{n+2}) - 16f(t_{n+1}, z_{n+1}) + 5f(t_n, z_n))$$

Adams-Bashforth 4 (AB4). El método de Adams-Bashforth de cuarto orden tiene la siguiente expresión:

$$z_{n+4} = z_{n+3} + \frac{h}{24}(55f(t_{n+3}, z_{n+3}) - 59f(t_{n+2}, z_{n+2}) + 37f(t_{n+1}, z_{n+1}) - 9f(t_n, z_n))$$

Finalmente, se presenta la expresión del método Adams-Bashforth de orden cinco:

$$z_{n+5} = z_{n+4} + \frac{h}{720}(1901f(t_{n+4}, z_{n+4}) - 2774f(t_{n+3}, z_{n+3}) + 2616f(t_{n+2}, z_{n+2}) - 1274f(t_{n+1}, z_{n+1}) + 251f(t_n, z_n))$$

donde, f_n representa el valor $f(t_n, z_n)$, además, se puede ver que en cada uno de los métodos se debe conocer los valores de z_{n+1} y otros más anteriores dependiendo del orden.

Métodos Adams-Moulton

Los métodos Adams-Moulton son métodos implícitos. Estos son métodos similares a los métodos Adams-Bashforth. La principal diferencia entre los métodos Adams-Bashforth y Adams-Moulton, es que los métodos Adams-Bashforth son explícitos y los métodos Adams-Moulton son implícitos. En este método para calcular y_{n+1} hacen uso de $f(t_{y+1}, y_{i+1})$

A continuación, se muestran las ecuaciones de los métodos Adams-Moulton de 0, 1, 2, 3 y 4 orden: En este estudio para una mejor interpretación del método de Adams-Moulton de orden 3 le denominaremos AM4.

Adams-Moulton orden 0 (AM1)

$$z_n = z_{n-1} + hf(t_n, z_n)$$

Adams-Moulton orden 1 (AM2)

$$z_{n+1} = z_n + \frac{h}{2}(f(t_{n+1}, z_{n+1}) + f(t_n, z_n))$$

Adams-Moulton orden 2 (AM3)

$$z_{n+2} = z_{n+1} + \frac{h}{12}(5f(t_{n+2}, z_{n+2}) + 8f(t_n, z_n) - f(t_n, z_n))$$

Adams-Moulton orden 3 (AM4)

$$z_{n+3} = z_{n+2} + \frac{h}{24}(9f(t_{n+3}, z_{n+3}) + 19f(t_{n+2}, z_{n+2}) - 5f(t_{n+1}, z_{n+1}) + f(t_n, z_n))$$

Adams-Moulton orden 4 (AM5)

$$z_{n+4} = z_{n+3} + \frac{h}{720} (251f(t_{n+4}, z_{n+4}) + 646f(t_{n+3}, z_{n+3}) - 264f(t_{n+2}, z_{n+2}) + 106f(t_{n+1}, z_{n+1}) + f(t_n, z_n))$$

f_n representa el valor $f(t_n, z_n)$, además, se puede ver que en cada uno de los métodos, se debe conocer los valores de z .

Mathematica

Es un programa desarrollado por Wolfram Research (www.wolfram.com) y es utilizado en áreas científicas, de matemáticas, ingeniería y computacionales. Se trata de un sistema de álgebra computacional y de un lenguaje de programación de propósito general. Se divide en dos partes: el kernel o núcleo que desempeña los cálculos, y el front end o interfaz que despliega los resultados y permite al usuario interactuar con el núcleo como si fuera un documento. Mathematica ofrece una experiencia de clase interactiva que ayuda a los estudiantes a explorar y comprender conceptos, y les da a los docentes las herramientas que necesitan para crear fácilmente materiales de estudio, tareas y presentaciones [22].

Las características más importantes que se puede destacar de Mathematica son: la disponibilidad de bibliotecas de funciones elementales y especiales para matemáticas; soporte de matrices; herramientas numéricas y simbólicas para cálculo de variable continua o discreta; herramientas de visualización de datos en 2D y 3D; una colección de bases de datos; funcionalidad asintótica, incluye soluciones asintóticas a ecuaciones algebraicas y diferenciales, así como aproximaciones asintóticas a sumas; capacidades numéricas para la solución de ecuaciones diferenciales parciales de NDSolve; integrate, soporte grandes sistemas estructurados en Solve y soporte Reduce, DSolve

para ecuaciones diferenciales ordinarias no lineales de primer orden, etc.

NDSolve

La función NDSolve construye aproximaciones numéricas a la solución de ecuaciones diferenciales numéricas generales que se aproxima a la solución. Las ecuaciones diferenciales parciales implican dos o más variables independientes. La función NDSolve también puede resolver algunas ecuaciones diferenciales algebraicas (DAE), que suelen ser una combinación de ecuaciones diferenciales y algebraicas [23].

NDSolve tiene una serie de formas de expresión:

1. La siguiente expresión permite calcular una solución numérica para ecuaciones diferenciales ordinarias, para una función μ con la variable dependiente x en el rango x_{min} a x_{max} :

$$NDSolve[eqn, \mu, \{x, x_{min}, x_{max}\}]$$

2. Resuelve ecuaciones diferenciales parciales eqns sobre una región rectangular:
3. Resuelve las ecuaciones diferenciales parciales eqns sobre la región Ω :

$$NDSolve[eqn, \mu, \{x, y\} \in \Omega]$$

4. Resuelve las ecuaciones diferenciales parciales dependientes del tiempo eqns sobre la región Ω :

$$NDSolve[eqn, \mu, \{t, t_{min}, t_{max}\}, \{x, y\} \in \Omega]$$

5. Resuelve las funciones μ_i :

$$NDSolve[eqn, \{\mu_1, \mu_2, \dots\}, \dots]$$

IV. METODOLOGÍA

En este estudio se utiliza el modelo Lotka-Volterra. En este caso en concreto, para responder las preguntas, se utiliza un modelo de biología. El modelo Lotka-Volterra, por ejemplo, pueden analizar una ecuación de la biología que modela el crecimiento de las especies animales. Si una variable real $\mu(t)$ representa el número de individuos de una determinada especie en el tiempo t , la suposición más simple sobre su evolución es $\frac{\partial u}{\partial t} = u\alpha$ donde α es la tasa de reproducción. Una constante conduce a un crecimiento exponencial. En el caso de que más especies vivan juntas, las tasas de reproducción también dependerán del número de población de las otras especies [24]. En este estudio se utiliza la versión de Mathematica 11.3.

Modelo

Como se indicó anteriormente, en esta investigación se estudia los métodos Runge-Kutta y Adams Bashforth-Moulton, con un modelo en concreto planteado por [25]. Los investigadores plantearon, un modelo para dos especies con $x(t)$ siendo el número de depredadores, $y(t)$ el número de presas, que es denotado por el modelo de Lotka-Volterra:

$$\begin{aligned} \dot{x} &= x(y - 2) \\ \dot{y} &= y(1 - x) \end{aligned} \quad (1)$$

\dot{x} y \dot{y} representan diferenciación con respecto

al tiempo. Además, es importante indicar que la Ecuación 1 fue usado por Lotka en 1925 y por Volterra en 1927. Concretamente, este modelo se usó para estudiar la invasión parasitaria de especies de insectos. La Fig. 1 muestra el campo vectorial del modelo Lotka-Volterra para la Ecuación 1.

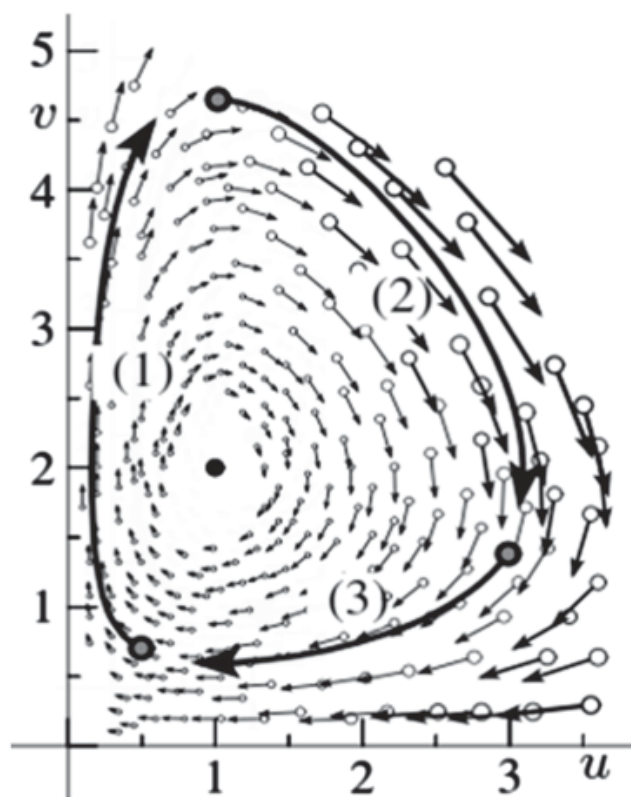


Fig. 1. Vector field, modelo Lotka-Volterra [25]

Para calcular las I (invariantes) del sistema de la Ecuación 1, se realiza el proceso de separación de variables, se obtiene la ecuación:

$$0 = \frac{1-x}{x} \dot{x} - \frac{y-2}{y} \dot{y} = \frac{d}{dt} I(x, y) \quad (2)$$

Considerando, la Ecuación 2, se obtiene la ecuación de las invariantes,

$$I(x, y) = \ln(x) - x + 2\ln(y) - y \quad (3)$$

Se debe considerar que, $I(x(t), y(t)) = \text{constante}$ para todo (t) .

Con la Ecuación 1 y siguiendo las referencias [23], [24], [25], [26] se analiza los métodos numéricos de resolución de ecuaciones diferenciales, utilizando funciones de Mathematica. Para los cálculos respectivos se utiliza el valor de $h = \frac{1}{10}$.

V. RESULTADOS

En este apartado, primero se verifica que estén todos los paquetes necesarios para realizar los cálculos de los métodos. Seguidamente, se resuelve la Ecuación 1 utilizando el modelo de Lotka-Volterra en Mathematica. A continuación, se resuelve la Ecuación 1 utilizando cada uno de los métodos: Runge-Kutta (RK2, RK3 y RK4) y Adams Bashforth-Moulton (AB2, AB3, AB4, AM2, AM3 y AM4). Finalmente, se realiza la comparación de los métodos. Concretamente, para la comparación de los métodos se utiliza el comando NDSolve de Mathematica, se comparan los métodos Runge-Kutta de cuarto orden (RK4), Adams-Bashforth de cuarto orden (AB4) y Adams-Moulton de tercer orden (AM4). Se utiliza el método de Adams-Moulton de tercer orden ya que este método tiene iteraciones similares al método AB4.

Instalación de paquetes y modelo Lotka-Volterra

Primeramente, se realiza la instalación de algunos paquetes necesarios para la modelación del sistema Lotka-Volterra y para la solución de los métodos propuestos. A continuación, se introduce los datos de la Ecuación 1 y la Ecuación 3. Por último, se procede a crear una función que dibuja las orbitas del Ecuación 1, con el método seleccionado. Paquetes necesarios para utilizar los componentes de NDSolve:

```
***** script Mathematica *****
Needs["DifferentialEquations`NDSolveProblems`"];
Needs["DifferentialEquations`NDSolveUtilities`"];
Needs["DifferentialEquations`
`InterpolatingFunctionAnatomy`"];
*****
```

Se introducen el modelo Ecuación 1 y las invariantes Ecuación 3 en Mathematica, se almacena las ecuaciones en una variable llamada solución:

```
***** script Mathematica *****
```

```
solucion
= NDSolveProblem[{{Subscript[Y, 1]'[T] =
= Y1[T](-2 - Y2[T]),
Subscript[Y, 2]'[T] == (1 - Y1[T]) Y2[T]}, {Y1[0]
==
3, Y2[0] == 1,
{Y1[T], Y2[T]}, {T, 0, 10}, {}, {Log[Y1[T]] - Y1[T]
+ 2Log[Y2[T]]
-Y2[T]}, {}];
*****
```

Se procede a programar en Mathematica la función que resuelva el modelo Lotka-Volterra, el objetivo es presentar las orbitas de la Ecuación 1.

```
***** script Mathematica *****
```

```
LotkaVolterraOrbita[sol_, vars_, time_, opts_? OptionQ]:
= Module
[commonopts, data, data1, data2, ifuns, lplot, pplot],
ifuns = First[vars/.sol];
datos1 = Part[ifuns, 1, 0][["ValuesOnGrid"];
datos2 = Part[ifuns, 2, 0][["ValuesOnGrid"];
datos = Transpose[{data1, data2}];
componentes = Sequence[Axes-
> False, Frame-
> True, FrameLabel->
Join[Map[TraditionalForm, vars], {None, None}],
RotateLabel-> False];
plot
= ListPlot[data, Evaluate[FilterRules[{opts},
Options[ListPlot]]],
PlotStyle
-> {PointSize[0.02], RGBColor[0, 0, 4]}, Evaluate
[commonopts]];
plot2
= ParametricPlot[Evaluate[ifuns], time, Evaluate
[FilterRules[{opts},
Options[ParametricPlot]]], Evaluate[commonopts]];
Show[plot, plot2];
*****
```


Pregunta1. Método 1. Métodos de un paso: Métodos Runge-Kutta

Para resolver estas ecuaciones se crea funciones en Mathematica, que resuelven un modelo de ecuaciones diferenciales ordinarias por los métodos Runge-Kutta.

Runge-Kutta orden 2 (RK2)

El método de Runge-Kutta 2 evalúa dos veces a la función f , como se puede observar en el script desarrollado en Mathematica. Concretamente, este método realiza dos iteraciones, lo que requiere mayor coste que el método de Runge-Kutta de orden 1 o el método de Euler. Sin embargo, este método es más preciso que el método de Euler. En el algoritmo de matemática se puede ver una función con los parámetros necesarios para desarrollar el método RK2. Seguidamente, se utiliza el comando NDSolve utilizando el modelo de Lotka-Volterra y el método de RK2.

***** script Mathematica *****

```
RungeKutta2[___][\"Step\"[f_, t_, h_, y_, fxy_]]
:= Block[{yt, k1, k2},
```

```
k1 = f[t, y];
```

```
k2 = h * f[t + h, y + k1];
```

```
yt = 1/2 h(k1 + k2);
```

```
{h, yt}];
```

```
NDSolve[solucion, Method
→ RungeKutta2, StartingSte pSize
→ h]
```

```
LotkaVolterra[solucion2, var, tiempo]
```

La Fig. 2 muestra la solución del método RK2 para la Ecuación 1 con un tamaño de paso de $h = \frac{1}{2}$ con valores iniciales (3,1). Adicionalmente,

la Fig. 2 muestra que la representación gráfica (diagrama de fase) no se acerca a los valores esperados (Fig. 1), se puede observar que la gráfica no se cierra completamente. Además, no se unen los puntos, ya que es una aproximación.

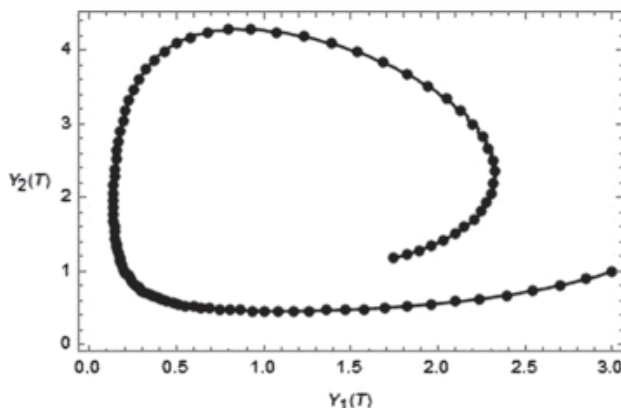


Fig. 2. Lotka-Volterra RK2.

Invariante Runge-Kutta 2

El método de Runge-Kutta 2 (RK2) tienen un error de truncamiento $O(h^3)$ y un error global $O(h^2)$. Para verificar el error producido por este método, en este estudio se analiza las invariantes. En este método a medida que aumenta t el error aumenta, el error aumenta considerando que h se mantiene igual, llegando a alcanzar valores de hasta 4 puntos cuando $t = 1$.

Runge-Kutta orden 3 (RK3)

El método de Runge-Kutta 3 (RK3) evalúa tres veces a la función f como se puede observar en el código desarrollado en Mathematica. Concretamente, el método de Runge-Kutta realiza tres pasos, lo que requiere mayor coste que el método de RK1 y RK2. Sin embargo, el método de RK3 son más precisos que los RK1 y RK2. En el código hay una función con los parámetros necesarios para desarrollar el método analizado en esta sección.

***** script Mathematica *****

```
RungeKutta3[___][["Step"[f_, t_, h_, y_, fxy_]]
:= Block[{yt, k1, k2, k3},
```

```
k1 = f[t, y];
k2 = f[t + 1/2 h, y + 1/2 hk1];
k3 = h * f[t + h, y - k1 + 2k2];
yt = 1/6 h(k1 + 4k2 + k3);
{h, yt}];
```

```
NDSolve[solucion, Method
→ RungeKutta3, StartingSte pSize
→ h]
```

```
LotkaVolterra[solucion, var, tiempo]
```

```
*****
```

La Fig. 3 muestra la solución del método RK3 para la Ecuación 1 con un tamaño de paso de $h = \frac{1}{10}$ con valores iniciales (3,1). Adicionalmente, la Fig. 3 muestra que la representación gráfica (diagrama de fase) no se cierra y no se unen los puntos finales. Sin embargo, se puede decir que se observa un mejor diagrama que el RK2, ya que se está ajustando a la gráfica esperada (Fig. 1).

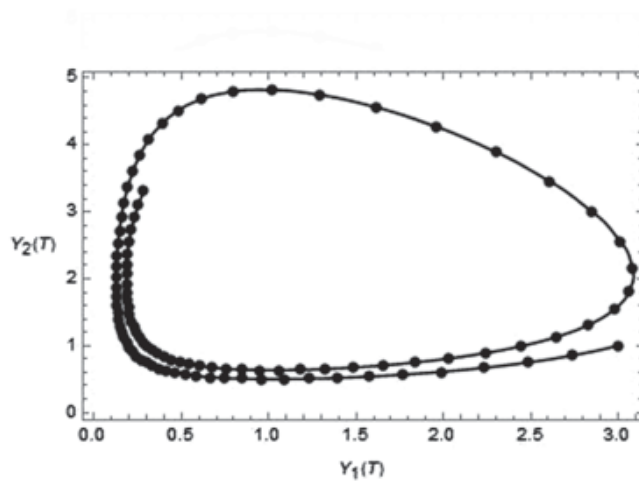


Fig. 3. Lotka-Volterra RK3.

Invariante Runge-Kutta 3

El método RK3 tienen un error de truncamiento $O(h^4)$ y un error global $O(h^3)$. Para verificar el error producido por el método RK3 se analiza la invariante del método. La

invariante del método RK3 indica que a medida que aumenta t el error aumenta, llegando a alcanzar valores de hasta 0.4 puntos cuando $t = 10$. Se puede concluir que el error de este método es considerablemente inferior que el método RK2.

Runge-Kutta orden 4

El método de Runge-Kutta 4 (RK4) evalúa cuatro veces a la función f , como se puede observar en el algoritmo desarrollado en Mathematica. Concretamente, el método de Runge-Kutta realiza cuatro pasos, lo que requiere mayor coste que el método de RK1, RK2 y RK3. Sin embargo, el método de RK4 son más precisos que los RK1, RK2 y RK3. A continuación, se presenta el algoritmo del método de RK4 en Mathematica.

```
***** script Mathematica *****
```

```
RungeKutta4[___][["Step"[f_, t_, h_, y_, fxy_]]
:= Block[{yt, k1, k2, k3, k4},
```

```
k1 = f[t, y];
k2 = f[t + 1/2 h, y + 1/2 hk1];
k3 = f[t + 1/2 h, y + 1/2 hk2];
k4 = f[t + h, y + hk3];
yt = h(1/6 k1 + 1/3 k2 + 1/3 k3 + 1/6 k4);
{h, yt}];
```

```
*****
```

La Fig. 4 muestra la solución del método RK4 para la Ecuación 1 con un tamaño de paso de $h = \frac{1}{10}$ con valores iniciales (3,1). Adicionalmente, la Fig. 4 muestra que la representación gráfica (diagrama de fase) cierra completamente y se unen los puntos finales. Además, se ajusta a la gráfica esperada (Fig. 1).

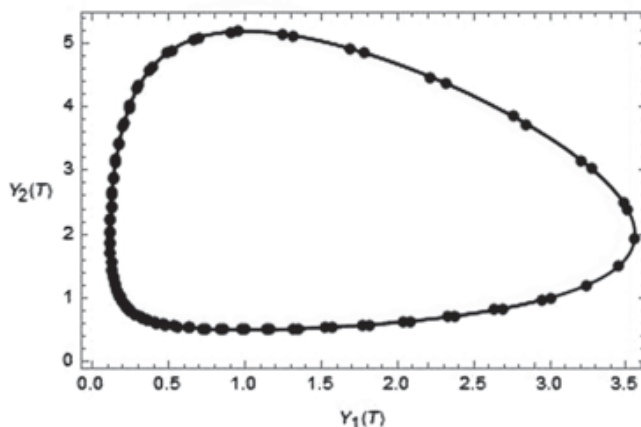


Fig. 4. Lotka-Volterra RK4.

***** script Mathematica *****

Invariante Runge Kutta 4

El método RK4 tienen un error de truncamiento $O(h^5)$ y un error global $O(h^4)$. Para verificar el error producido por el método RK4 se analiza la invariante del método. La invariante del método RK4 indica que a medida que aumenta t el error aumenta, llegando a alcanzar valores de hasta 0.00010 puntos cuando $t = 10$. Si se compara con los métodos RK1, RK2 y RK3, se puede decir que el método RK4 es más preciso. Sin embargo, el método RK4 tiene mayor coste computacional, que debería ser analizado.

Pregunta1. Método 2. Métodos multipasos: Métodos Adams-Bashforth

Los métodos de Adams-Bashforth son métodos multipaso explícitos. Los métodos Adams-Bashforth utilizan otros métodos para calcular y_n . Concretamente, en este estudio, para calcular y_n se utiliza el método RK4.

Método Adams-Bashforth 2 (AB2)

El método de AB2 evalúa 8 veces a la función f , esto debido a que se utiliza el método de RK4, este método evalúa f $4n$ (f es evaluado 4 veces), como se puede ver en el algoritmo desarrollado en Mathematica.

```
***** script Mathematica *****
AdamsBashforth2[___]["Step"[f_, t_, h_, y_, fxy_]]
:= Block[{yt, k1, k2, k3, k4},

vec[0] = y;
For[i = 0, i < 3, i ++,
k1 = f[t, vec[i]];
k2 = f[t + 1/2 h, y + 1/2 h k1];
k3 = f[t + 1/2 h, y + 1/2 h k2];
k4 = f[t + h, y + h k3];
vec[i + 1] = h(1/6 k1 + 1/3 k2 + 1/3 k3
+ 1/6 k4);];
yt = vec[2] + ((h/2)
* (3f[t + 1, vec[2]]
- f[t, vec[1]]));
{h, yt}];
```

En el algoritmo anterior presentado en Mathematica, se puede observar que para calcular y_n se utiliza el método RK4, en este caso cada valor de y_n se almacena en un vector, para no realizar varias veces el método de RK4, se agrega un bucle. Por otro lado, la Fig. 5 muestra la solución del método de AB2, para la Ecuación 1 con un tamaño de paso de $h = \frac{1}{10}$ con valores iniciales (3,1). Adicionalmente, la Fig. 5 muestra que la representación gráfica (diagrama de fase) una los puntos finales y es similar a la Fig. 1. Estos resultados eran los esperados ya que se utiliza el método RK4.

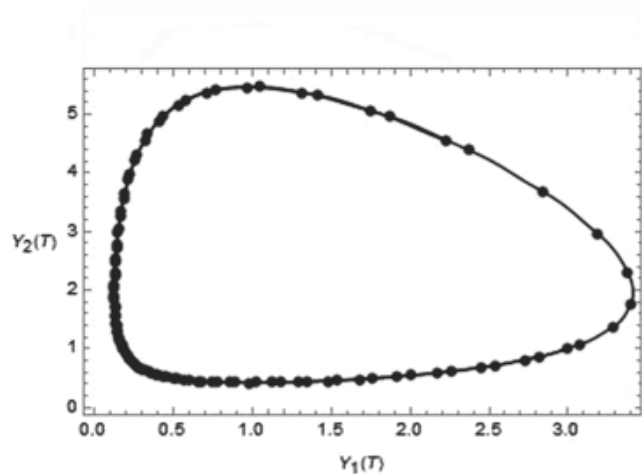


Fig. 5. Lotka-Volterra AB2.

Invariante AB2

El método de AB2 tienen un error local $O(h^3)$. Para verificar el error producido por el método AB2 se analiza la invariante del método. La invariante del método AB2 indica que aumenta el error en el intervalo $t[0,1]$, después el error disminuye en un intervalo aproximado de $t(1,3]$, convirtiéndose en ciclos de aumento y disminución. El error máximo alcanzado es un valor aproximado a 0.42 puntos cuando $t = 1$ y $t = 8$. Cuando $t = 10$ el valor de error aproximado es de 0.04 puntos.

Método Adams-Bashforth 3 (AB3)

El método de AB3 evalúa 12 veces a la función f , ya que utiliza el método de RK4 y este método evalúa $f 4n$, como se puede observar en el algoritmo de Mathematica.

***** script Mathematica *****

```
AdamsBashforth3[___][\"Step\"[f_, t_, h_, y_, fxy_]]
:= Block[{yt, k1, k2, k3, k4},
vec[0] = y;
For[i = 0, i < 4, i ++,
k1 = f[t, vec[i]];
k2 = f[t + 1/2 h, y + 1/2 hk1];
k3 = f[t + 1/2 h, y + 1/2 hk2];
k4 = f[t + h, y + hk3];
vec[i + 1] = h(1/6 k1 + 1/3 k2 + 1/3 k3
+ 1/6 k4);];
yt = vec[3] + ((h/12)
* (23f[t + 2, vec[3]]
- 16f[t + 1, vec[2]]
+ 5f[t, vec[1]]));
{h, yt};
```

De igual manera que el método AB2 y_n se calcula utilizando el método RK4, en este caso cada valor de y_n se almacena en un vector y el bucle aumenta en 1 con referencia al método AB2.

Por otro lado, la Fig. 6 muestra la solución del método de AB3 para la Ecuación 1 con un tamaño de paso de $h = \frac{1}{10}$ con valores iniciales (3,1). Adicionalmente, se puede decir que es similar a la Fig. 1.

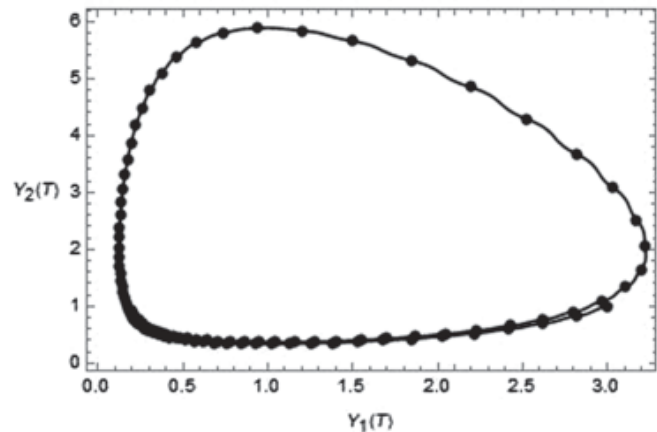


Fig. 6. Lotka-Volterra AB3. Nota: Fuente propia del estudio.

Invariante AB3

El método de AB3 tienen un error local $O(h^4)$. Para verificar el error producido por el método AB3 se analiza la invariante del método. La invariante del método AB3 indica que a medida que aumenta t en el intervalo $[0,3]$ el error aumenta, alcanzado un valor de 0.60 puntos, en los demás intervalos el error máximo alcanzado es aproximadamente de 0.40 puntos. Cuando $t = 10$ el valor del error es muy pequeño.

Método Adams-Bashforth 4 (AB4)

El método de AB4 evalúa 16 veces a la función f , ya que se utiliza el método de RK4, este método evalúa $f 4n$, como se puede observar en el algoritmo desarrollado en Mathematica.

***** script Mathematica *****

```
AdamsBashforth4[___][\"Step\"[f_, t_, h_, y_, fxy_]]
:= Block[{yt, k1, k2, k3, k4, y1, y2, y3, y4},
vec[0] = y;
For[i = 0, i < 5, i ++,
k1 = f[t, vec[i]];
k2 = f[t + 1/4 h, y1 + 1/4 hk1];
k3 = f[t + 3/8 h, y2 + 3/8 hk1 + 3/8 hk2];
k4 = f[t + h, y3 + 3/4 hk1 + 3/8 hk2 + 1/8 hk3];
vec[i + 1] = h(1/20 k1 + 9/40 k2 + 45/80 k3 + 16/40 k4);];
yt = vec[5];
{h, yt};
```

```

k2 = f[t + 1/2 h, y + 1/2 hk1];
k3 = f[t + 1/2 h, y + 1/2 hk2];
k4 = f[t + h, y + hk3];
vec[i + 1] = h(1/6 k1 + 1/3 k2 + 1/3 k3
+ 1/6 k4);];

```

```

yt = vec[4] + ((h/24)
* (55f[t + 3, vec[4]]
- 59f[t + 2, vec[3]]
+ 37f[t + 1, vec[2]]
- 9f[t, vec[1]]));

```

```
{h, yt};
```

```
*****
```

De igual manera que el método AB2 y AB3 se calcula y_n utilizando el método RK4, en este caso cada valor de y_n se almacena en un vector y el bucle aumenta en 1 en referencia al método AB3.

Por otro lado, la Fig. 7 muestra la solución del método de AB4 para la Ecuación 1 con un tamaño de paso de $h = \frac{1}{10}$ con valores iniciales (3,1). Adicionalmente, la Fig. 7 muestra que la representación gráfica (diagrama de fase) una los puntos finales. Además, se puede decir que es similar a la Fig. 1, y a los métodos AB2 y AB3, esto debido a que utilizan el mismo método (RK4).

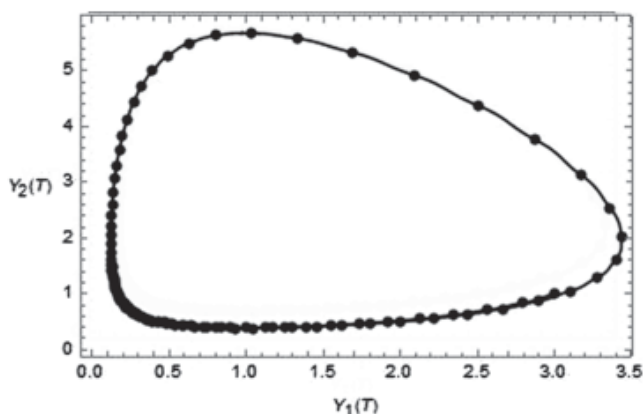


Fig. 7. Lotka-Volterra AB4.

Invariante AB4

El método de AB4 tienen un error local $O(h^5)$. Para verificar el error producido por el método AB4 se analiza la invariante del método. La invariante del método AB4 indica que el error máximo que se alcanza en los intervalos es aproximadamente de 0.42 puntos.

Pregunta1. Método 3. Métodos multipasos: Métodos Adams-Moulton

Los métodos de Adams-Moulton son métodos multipaso implícitos. Es decir, utilizan la fórmula para predecir un valor y_{n+1} , que a su vez utiliza la fórmula para obtener el valor corregido y_{n+1} . Es importante indicar que los métodos multipaso implícito no se usan en solitario, es decir se utilizan para mejorar las aproximaciones obtenidas por los métodos explícitos. La combinación de un método explícito con un método implícito se conoce como método predictor-corrector. Concretamente, en este estudio, para calcular los primeros pasos y_n se utilizan los métodos RK4. Por otro lado, como método predictor se utiliza los métodos de Adams-Bashforth y los métodos Adams-Moulton como corrector.

Métodos Adams-Moulton 2 (AM2)

El método de AM2 evalúa 8 veces a la función f , en este estudio se utiliza el método de RK4, este método evalúa $f 4n$, como se puede observar en el algoritmo de Mathematica. Adicionalmente, utiliza un método predictor, en este caso se utiliza el método de AB2. Finalmente, se utiliza el método Adams-Moulton 2 como corrector.

```

***** script Mathematica *****
AdamsBashMoulton2[___][\"Step\"[f_, t_, h_, y_, fxy_]]
:= Block[{yt, k1, k2, k3, k4, y1, y2},
vec[0] = y;
For[i = 0, i < 5, i ++,

```

```

k1 = f[t, vec[i]];
k2 = f[t + 1/2 h, y + 1/2 hk1];
k3 = f[t + 1/2 h, y + 1/2 hk2];
k4 = f[t + h, y + hk3];
vec[i + 1] = h(1/6 k1 + 1/3 k2 + 1/3 k3
+ 1/6 k4);

```

```
(*Befhorpredictor*)
```

```

y2 = vec[2] + h(3/2 f[t + 1, vec[2]]
- 1/2 f[t, vec[1]]);

```

```
(*AdmasMoultcorrector*)
```

```

yt = vec[1] + h/2 (f[t + 2, y2] + f[t + 1, vec[1]]);
(*Print[yt]; *)
{h, yt};

```

```
*****
```

La Fig. 8 muestra la solución del método de AM2 para la Ecuación 1, con un tamaño de paso de $h = \frac{1}{10}$ valores iniciales (3,1). La Fig. 8 muestra el diagrama de fase del método AM2.

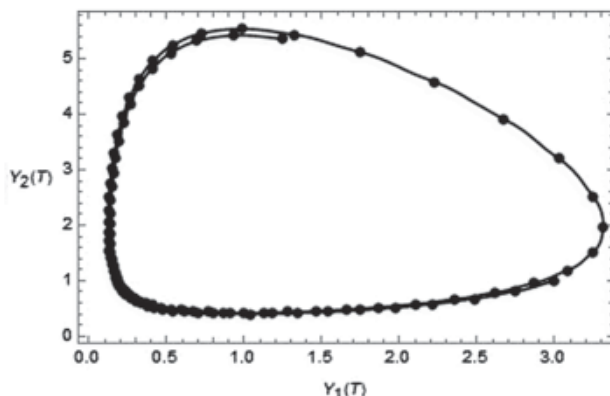


Fig. 8. Lotka-Volterra AM2

Invariante AM2

El método de AM2 tienen un error local $O(h^3)$. Para verificar el error producido por el método AM2 se analiza la invariante del método. La invariante indica que el error máximo se alcanza en el intervalo $[0, 2]$ y es de 0.36 puntos. En los siguientes intervalos los valores del error son menores.

Métodos Adams-Moulton AM3

El método de AM3 evalúa 12 veces a la función f , en este caso se utiliza el método de RK4, este método evalúa f $4n$, como se muestra en el algoritmo de Mathematica. Adicionalmente, se puede observar que se utiliza un método predictor (AB3). Además, se utiliza el método Adams-Moulton como corrector.

```
***** script Mathematica *****
```

```

AdamsBashMoulton3[___][["Step"[f_, t_, h_, y_, fxy_]]
:= Block[{yt, k1, k2, k3, k4, y1, y2, y3},
vec[0] = y;

```

```

For[ $i = 0, i < 5, i + +,$ 
k1 = f[t, vec[i]];
k2 = f[t + 1/2 h, y + 1/2 hk1];
k3 = f[t + 1/2 h, y + 1/2 hk2];
k4 = f[t + h, y + hk3];
vec[i + 1] = h(1/6 k1 + 1/3 k2 + 1/3 k3
+ 1/6 k4);]

```

```
(*ABpredictor*)
```

```

y3 = vec[3] + h(23/12 f[t + 2, vec[3]]
- 4/3 f[t + 1, vec[2]]
+ 5/12 f[t, vec[1]]);

```

```
(*AdmasMoultcorrector*)
```

```

yt = vec[2] + h(5/12 f[t + 3, y3]
+ 2/3 f[t + 2, vec[2]]
- 1/12 f[t + 1, vec[1]]);

```

```
(*Print[yt]; *)
{h, yt};
```

```
*****
```

La Fig. 9 muestra la solución del método de AM3 para la Ecuación 1 con un tamaño de paso de $h = \frac{1}{10}$, con valores iniciales (3,1). La Fig. 9 muestra el diagrama de fase del AM3. Se puede decir, que el método AM3 es similar a la Fig. 1.

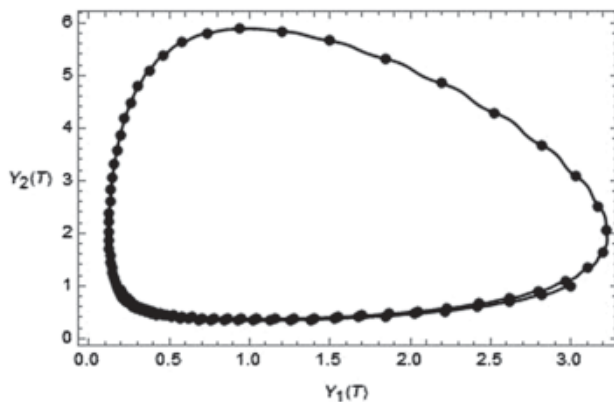


Fig. 9. Lotka-Volterra AM3

Invariante AM3

El método de AM3 tienen un error local $O(h^4)$. Para verificar el error producido por el método AM3 se analiza la invariante del método. La invariante del método indica que el error máximo alcanzado es en el intervalo $[0,3]$ con un valor aproximado de 0.60 puntos y en el intervalo $[7,10]$, el valor aproximado es de 0.45 puntos. Se puede ver que cuando $t = 10$, el valor del error baja considerablemente.

Métodos Adams-Moulton orden 4

El método de AM4 evalúa 16 veces a la función f , en este caso se utiliza el método de RK4, este método evalúa f $4n$, como se muestra en el algoritmo de Mathematica. Adicionalmente, se puede observar que se utiliza un método predictor (AB4) y el método AM4 como corrector.

***** script Mathematica *****

```
AdamsBashMoulton4[___][\"Step\"[f_, t_, h_, y_, fxy_]]
:= Block[{yt, k1, k2, k3, k4, y1, y2, y3, y4},
vec[0] = y;
For[i = 0, i < 5, i ++,
k1 = f[t, vec[i]];
k2 = f[t + 1/2 h, y + 1/2 hk1];
k3 = f[t + 1/2 h, y + 1/2 hk2];
k4 = f[t + h, y + hk3]; vec[i + 1]
= h(1/6 k1 + 1/3 k2 + 1/3 k3
+ 1/6 k4);];
```

(*ABpredictor*)

```
y4 = vec[4] + ((h/24)
* (55f[t + 3, vec[4]]
- 59f[t + 2, vec[3]]
+ 37f[t + 1, vec[2]]
- 9f[t, vec[1]]));
```

(*Admascorrector*)

```
yt = vec[3] + ((h/24)(9f[t + 4, y4]
+ 19f[t + 3, vec[3]]
- 5f[t + 2, vec[2]]
+ f[t, vec[1]]));
```

(*Print[yt]; *){h, yt};

La Fig. 10 muestra la solución de método AM4 para la Ecuación 1 con un tamaño de paso de $h = \frac{1}{10}$ con valores iniciales (3,1). La Fig. 10 muestra el diagrama de fase del método AM4. Además, observando las gráficas se puede concluir que el método AM4 es similar a la Fig. 1 y a los métodos RK4 y AB4.

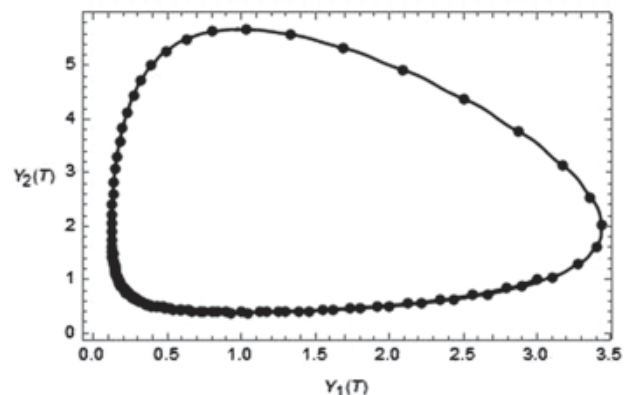


Fig. 10. Lotka-Volterra AM4

Invariante AM4

El método de AM4 tienen un error local $O(h^5)$. Para verificar el error producido por el método AM4 se analiza la invariante del método. La invariante indica que el error máximo se alcanza en el intervalo $[0,3]$, con un valor aproximado de

0.32 puntos y en el intervalo $[7,10]$ el valor aproximado es de 0.38 puntos. Cuando $t=10$, el valor del error baja considerablemente. Es importante indicar que los valores de las invariantes son similares al AM3 y AM2, esto debido a que ambos métodos utilizan el mismo método predictor y el método RK4.

Pregunta2. Comparación de los métodos

Para realizar la comparación de los métodos se utiliza la solución del comando NDSolve de Mathematica. La Fig. 11 muestra el plano de fase del modelo Lotka-Volterra de la Ecuación 1 utilizando el comando NDSolve, de aquí en adelante se identifica como solución NDSolve.

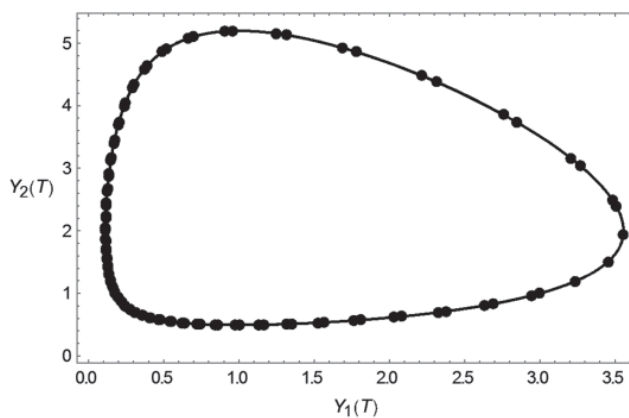


Fig. 11. Modelo Lotka-Volterra, método NDSolve

Por otro lado, la Fig. 12 presenta las órbitas de los métodos RK4 (azul), NDSolve (rojo), AB4 (verde) y AM4 (negro). En este se observa que el método RK4 se acerca considerablemente a la solución NDSolve. Adicionalmente, se puede observar que los métodos AB4 y AM4 se alejan en algunos puntos a la solución de NDSolve. Por otro lado, se puede observar que en algunos intervalos los puntos de los 4 métodos son similares, por ejemplo $[0.2, 0.4]$. Además, la Fig. 12 muestra que los métodos de Runge-Kutta tienen una exactitud mayor sobre los métodos de AB4 y AM4. Analizando los métodos AB4 y AM4 se puede decir que AM4 se acerca más a la solución de NDSolve.

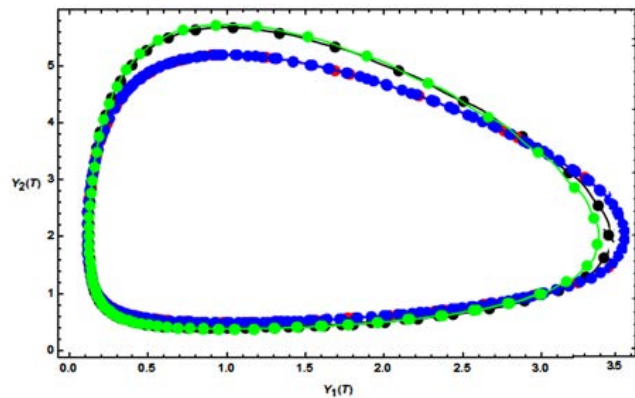


Fig. 12. Métodos RK4 (rojo), NDSolve (azul), AB4 (verde), AM4 Predictor-corrector (negro).

Invariantes

En este apartado, para una mejor comprensión de las diferencias de los valores obtenidos de la comparación de los métodos, se estudia la conservación de los diferentes métodos de la función $I(u, v)$. Concretamente, se analiza el error que proporciona cada uno de los métodos. Estos valores se presentan en un gráfico, lo que facilita la comparación visual de los resultados.

La Fig. 13 muestra la gráfica de las Invariantes de los métodos RK4 (rojo), AB4 (verde) y AM4 (negro). Específicamente, la Fig. 13 muestra que el método de RK4 tiene una excelente conservación, el error es mínimo. Se puede ver que en los métodos AB4 y AM4 en determinados rangos el error aumenta, aunque en algunos rangos el error disminuye considerablemente, se puede ver que el error que produce el método AM4 es inferior al método AB4, pero los valores no se diferencian de manera significativa.

VI. DISCUSIÓN Y CONCLUSIONES

El objetivo principal de este estudio fue analizar y comparar la solución numérica de los métodos numéricos Runge-Kutta 4 y Adams Bashforth-Moulton 4 con el modelo Lotka-Volterra. Además, se analizó la evolución del error producido por cada método cuando aumenta de orden. Para cumplir el objetivo se utiliza el software de Mathematica.

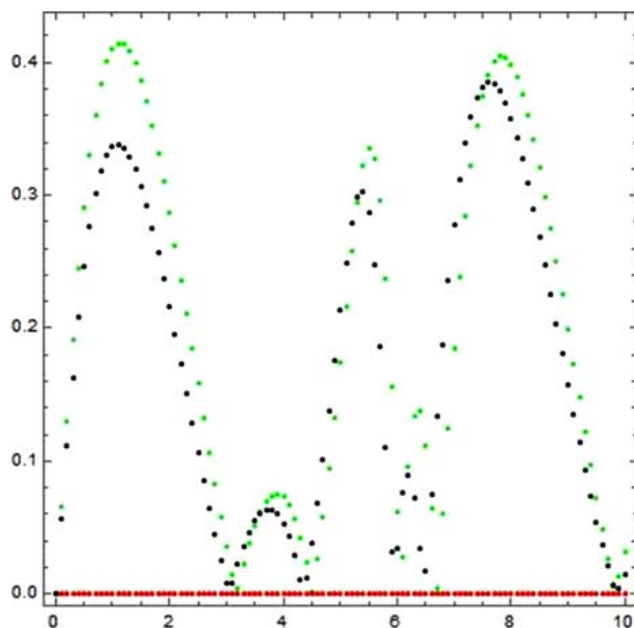


Fig. 13. Invariantes de los métodos RK4(rojo), AB4(verde) y AB (negro).

El primer método que se estudió fueron los métodos de un paso Runge-Kutta. Concretamente, se estudiaron los métodos de RK2, RK3 y RK4. En el caso de RK1 es equivalente al método de Euler por lo que no se le consideró en este estudio. El primer método que se estudió fue el método RK2, se puede observar claramente que no es similar a la solución de NDSolve, sin embargo, se puede decir que este método es más preciso que el método RK1 o Euler, aunque su coste computacional es mayor que el RK1. Adicionalmente, la invariante de este método indica un error muy elevado, hasta 4 puntos, se observa que a medida que aumenta t , el error aumenta considerablemente.

El siguiente método estudiado fue el método RK3, en este caso el diagrama se va acercando a la solución del método de NDSolve. El método RK3 es más preciso que los métodos RK2 y RK1. Este resultado se puede evidenciar con el análisis de las invariantes del método RK3, el error alcanza los 0.4 puntos cuando $t=10$. Se puede decir que el método RK3 mejorará considerablemente en exactitud en relación con el método RK2.

Finalmente, se estudió el método RK4, se puede observar que la órbita cierra correctamente y

que la solución es similar a la del método NDSolve, de esta manera se puede decir que el método de RK4 tiene mayor exactitud que los métodos de Runge-Kutta de menor orden. Para fortalecer esta afirmación se estudió la invariante del método RK4, se puede observar que el error alcanza valores de 0.00010 puntos cuando $t=10$. Con este resultado se puede apreciar que el error se reduce considerablemente en comparación con los métodos de menor orden. Aunque, se debe considerar que el método RK4 tiene mayor coste computacional que los métodos RK1, RK2 y RK3. Coincidió con [25], los autores indicaron que el algoritmo para el método de Runge-Kutta de cuarto orden, es de uso extendido, y reconocido como una valiosa herramienta de cálculo, por la buena aproximación que produce.

Seguidamente, se estudia los métodos multi-paso. El primero que se estudió fueron los métodos Adams-Bashforth. En este caso este método necesita un método adicional. Concretamente, se utiliza el método RK4. En el método AB2 las órbitas se cierran correctamente y el diagrama es similar a la solución del método NDSolve. Adicionalmente, se puede observar que el error alcanza los 0.4 puntos en algunos intervalos.

Por otro lado, en el caso del método AB3, la solución es se asemeja al método de NDSolve. El error alcanza valores de hasta 0.60 puntos. Sin embargo, no se evidencia una evolución o una mejora notable entre los métodos AB2 y AB3.

Finalmente, se estudia el método AB4, al ver el diagrama de fase se evidencia que son similares a los métodos AB2 y AB3. El error producido por este método alcanza el valor de 0.40 puntos. Concretamente, no se evidencia una evolución notable de los métodos Adams-Bashforth cuando se aumenta el orden, estos resultados puede ser porque los tres métodos usan el método RK4.

Adicionalmente, se tienen los métodos corrector-predictor (métodos Adams-Moulton), en este estudio se utiliza el método de RK4, como método predictor los métodos Adams-Bashforth y como corrector los métodos Adams-Moulton. Las

órbitas del método AM4 cierran correctamente, a diferencia del método AM2 y AM3 que no se cierra correctamente. Las invariantes no muestran una evolución considerable de los métodos AM2, AM3 y AM4, ya que estos utilizan los mismos métodos (RK4, AB4).

Por último, para responder la segunda pregunta de investigación, se realiza una comparación de los métodos de RK4, AB4 y AM4 con la solución de NDSolve. Los resultados indican que el método de un paso RK4 es el que más se acerca a la solución del método de NDSolve de Mathematica. Por otro lado, los métodos AB4 y AM4 tienen una órbita similar, en determinados puntos se alejan de la solución NDSolve. Sin embargo, el método Adams-Moulton de orden (AM4) es más preciso que el método AB4. Coincidimos, con [25, 26], los autores en su investigación demostraron que el método RK4 es más preciso que otros métodos.

Como conclusión se puede decir que, los resultados obtenidos en este estudio indican que el método que mayor conservación y que más se acerca a la solución de NDSolve es el método de Runge-Kutta 4, este método tiene mayor precisión que el método de AB4 y AM4, así se puede observar en las orbitas comparativas y en las gráficas de las Invariantes.

Por otro lado, esta investigación quiere remarcar la importancia de la utilización de un software matemático en el desarrollo de los métodos numéricos para resolver ecuaciones diferencias, como se puede ver, estos métodos utilizan una serie de iteraciones que dificultan hacerlos manualmente.

En una futura línea de investigación se estudiará el coste computacional de estos métodos, con el propósito de realizar una comparación más equilibrada.

REFERENCIAS

- [1] J. Barbarán, J., J. Fernández, El análisis de errores en la resolución de ecuaciones diferenciales ordinarias. Una metodología para desarrollar la competencia matemática”, Enseñanza de las ciencias, Revista de investigación y experiencias didácticas, Vol. 32, Nº 3, pp. 173-186, 2014.
- [2] M. Artigue, Une recherche d'ingénierie didactique sur l'enseignement des équations différentielles du premier cycle universitaire, Cahier du séminaire de Didactique des Maths et de l'Informatique de Grenoble, pp. 183-206, 1989.
- [3] M. Kallaher, Revolutions in Differential Equations: Exploring ODEs with Modern Technology, Washington: The Mathematical Association of America, 1999.
- [4] C. Rasmuss, New directions in differential equations: a framework for interpreting students' understandings and difficulties, The Journal of Mathematical Behaviour, Vol. 20, pp. 55-87, 2011.
- [5] S. Arslan, Traditional instruction of differential equations and conceptual learning. Teaching Mathematics and its Applications, An International Journal of the IMA, Vol. 29, Nº 2, pp. 94-107, 2010.
- [6] M. Artigue, Functions from an Algebraic and Graphic Point of View: Cognitive Difficulties and Teaching Practices, An The concept of function. Aspects of epistemology and Pedagogy, Vol. 25, pp. 109-132, 1992.
- [7] S. Habre, Exploring students strategies to solve differential equations in a reformed setting, Journal of Mathematical Behavior, Vol. 18, pp. 445-472, 2000.
- [8] G. Ortigoza, Ecuaciones diferenciales ordinarias con Maxima, Educación matemática, vol. 21, Nº 21, pp. 134-677, 2009.
- [9] M. Santos, Potencial didáctico del software dinámico en el aprendizaje de las matemáticas, Avance y perspectiva, vol. 20, pp. 247-258, 2001.
- [10] O. Rincón, El diseño instruccional en la educación a distancia. Un acercamiento a los Modelos, Apertura, vol. 9, Nº 10, pp. 104-119, 2009.
- [11] N. Blackett, D. Tall, Gender and the versatile learning of trigonometry using computer software, The Proceedings of the International Group for the Psychology of Mathematics Education XV, vol. 1, pp. 144-151, 1991.
- [12] K. Moore, Trigonometry, technology, and didactic objects, Proceedings of the 31st annual meeting of the North American Chapter of the international Group for psychology of Mathematics Education, Atlanta, GA, 2009.
- [13] M. Fernández, Perfeccionamiento de la enseñanza-aprendizaje del tema límite de funciones con el

- uso de un asistente matemático., *Revista Latinoamericana de Investigación en Matemática Educativa*, vol. 3, N° 2, pp. 171-187, 2000.
- [14] NCTM, National Council of Teachers of Mathematics [NCTM]., *Principles and standards for school mathematics*. Reston, VA: NCTM, 2000.
- [15] J. Perdomo, Módulo de enseñanza para la introducción de las ecuaciones diferenciales ordinarias en un ambiente de resolución de problemas con tecnología, *Revista de Didáctica de las Matemáticas*, vol. 78, pp. 13-134, 2011.
- [16] O. Rincón, TIC en la enseñanza de las ecuaciones diferenciales de primer orden, *Revista Logos, Ciencia & Tecnología*, vol. 8, N° 1, pp. 89-100, 2016.
- [17] G. Ortigoza, Resolviendo ecuaciones diferenciales ordinarias con Maple y Mathematica, *Revista mexicana de física*, vol. 53, N° 2, pp. 155-167, 2007.
- [18] J. Molina-Mora, Experiencia basada en la triada TICs, enseñanza por proyectos y modelado para la enseñanza de sistemas de ecuaciones diferenciales, *Uniciencia*, vol. 29, N° 2, pp. 46-61, 2015.
- [19] C. Ríos, F. Cald Era, M. Ivette, V. Valenzuela, Fidelidad en el uso de app para la resolución de ecuaciones diferenciales, *Apertura*, vol. 11, N° 1, pp. 74-89, 2019.
- [20] M. Sandoval, C. Ruvalcaba, J. González, O. Chávez, F. García, Aproximación a la solución de ecuaciones diferenciales por métodos no convencionales y el software Mathematica, *Cultura Científica y Tecnológica*, vol. 47, 2015.
- [21] D. Zill, M. Cullen, *Ecuaciones Diferenciales*, México: Cengage Learning Editores, 2009.
- [22] Mathematica Wolfram, <https://www.wolfram.com/support>.
- [23] E. Hairer, C. Lubich, G. Wanner, *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, Springer Science and Business Media, 2006.
- [24] J. Segarra, Resolución numérica de ecuaciones diferenciales en Wolfram Mathematica, Universitat Jaume I, España, 2018.
- [25] D. Zill, *Differential equations with modeling applications*, México: International Thomson Editores. 2002.
- [26] J. Segarra, Análisis de los métodos numéricos en ecuaciones diferenciales ordinarias utilizando Mathematica. *Revista Ingeniería, Matemáticas y Ciencias de la Información*, vol. 7, no 13, p. 13-23. 2020.

