



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

# ANÁLISIS DE LOS MÉTODOS NUMÉRICOS EN ECUACIONES DIFERENCIALES ORDINARIAS UTILIZANDO MATHEMATICA

## *Analysis of the numerical methods in ordinary differential equations using mathematica*

JAIME SEGARRA-ESCANDÓN<sup>1</sup>

*Recibido: 22 de octubre de 2019. Aceptado: 22 de diciembre de 2019*

DOI: <http://dx.doi.org/10.21017/rimci.2020.v7.n13.a72>

### RESUMEN

En esta investigación, el objetivo principal es realizar el análisis comparativo de los métodos numéricos (Euler Explícito, Runge Kutta 4 y LocallyExact) para la resolución de ecuaciones diferenciales. Para cumplir con el objetivo de este estudio se utilizó el sistema de ecuaciones diferenciales del modelo Lotka-Volterra y se usó el software matemático Wolfram Mathematica. Para realizar la comparación de los métodos numéricos se resolvió el modelo Lotka-Volterra utilizando el comando NDSolve de Mathematica, este resultado se comparó con los métodos Euler Explícito, Runge Kutta 4 y LocallyExact. Los resultados obtenidos de los diagramas de fase y la tabla de puntos de las interacciones indican que el método Runge Kutta 4 tiene mayor precisión, seguido por el método LocallyExact. El método Euler Explícito se aleja de manera considerable del resultado de NDSolve.

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

### ABSTRACT

In this research, the main objective is to perform the comparative analysis of numerical methods (Explicit Euler, Runge Kutta 4 and LocallyExact) for the resolution of differential equations. To fulfill the purpose of this study, the system of differential equations of the Lotka-Volterra model was used and the mathematical software Wolfram Mathematica was used. To perform the comparison of the numerical methods the Lotka-Volterra model was solved using the NDSolve command of Mathematica, this result was compared with the Methods Explicit Euler, Runge Kutta 4 and LocallyExact. The results obtained from the phase diagrams and the point table of the interactions indicate that the Runge Kutta 4 method has greater precision, followed by the LocallyExact method. The explicit Euler method draws considerably away from the result of NDSolve.

**Key words.** 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]. Existen diversas investigaciones que estudian la modelización y las aplicaciones de las

ecuaciones diferenciales utilizando software informático[2-5]. También, existen varios estudios relacionados con el proceso de enseñanza y aprendizaje de las ecuaciones diferenciales ordinarias [1,6-8].

Además, es importante considerar que en la actualidad los investigadores y profesores han ma-

<sup>1</sup> Departamento de Matemáticas e Informática de la Seguridad, Universidad Rovira i Virgili, Tarragona, España. ORCID: <http://orcid.org/0000-0003-4304-2385>. Correo electrónico: [jaimerodrigo.segarr@urv.cat](mailto:jaimerodrigo.segarr@urv.cat)

nifestado el interés de incorporar herramientas informáticas en el proceso de enseñanza y aprendizaje de las matemáticas. El uso de la tecnología en la modelación matemática es importante y muy relevante, porque brindan ayuda a los estudiantes en la visualización de relaciones de la matemática, permitiendo establecer representaciones exactas de configuraciones geométricas, porque en este contexto, los jóvenes tienen la oportunidad de mover partes de estas configuraciones y observar las invariantes que se producen en una construcción matemática [9].

El uso de las herramientas tecnológicas en clase de matemáticas permite a los estudiantes mejorar la comprensión del concepto, además se sugiere cambiar la instrucción estándar con la metodología magistral por la instrucción experimental basada en la utilización de las TIC (tecnología de la información y comunicación). 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 [10-11].

En los últimos años la matemática ha recibido un impacto positivo en la introducción de la tecnología. Las calculadoras gráficas, ordenadores, internet, están cambiando la enseñanza de los contenidos de la matemática, dado que su gran capacidad y rapidez en el cálculo, y la facilidad que brindan para lograr representaciones gráficas, permiten que los estudiantes reflexionen sobre los procesos matemáticos y de esta manera mejoren sus habilidades y capacidades. La tecnología no debería utilizarse como sustituto de los conocimientos e intuiciones básicos, sino que puede y debería usarse para potenciarlos. 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 [12].

En este trabajo investigativo se plantea el siguiente objetivo: realizar la comparación de los métodos numéricos (Euler explícito, Runge Kutta 4 y LocallyExact) para la resolución de ecuaciones diferenciales utilizando el software de Mathematica.

## II. REVISIÓN DE LA LITERATURA

Esta sección presenta una breve revisión de la literatura de la solución numérica y la enseñanza de las ecuaciones diferenciales ordinarias mediante el uso de paquetes informáticos.

En [13] se plantea que el tratamiento que actualmente se da a la enseñanza de la modelización y de las aplicaciones a situaciones de la vida real se debe a dos motivos: la dificultad conceptual de la modelización y necesidad de conocimientos matemáticos de los que los estudiantes no disponen, y concepción personal de cada profesor respecto a la matemática aplicada y su posición en el ámbito de las matemáticas. Otro resultado importante que presentan los autores fue que la mayoría de los profesores optan por la enseñanza instrumental de las ecuaciones diferenciales apoyándose en tres argumentos: pobre nivel de competencia de los estudiantes; escasa capacidad de razonamiento matemático y pobre pensamiento relacional; sencillez de la enseñanza de técnicas frente a la dificultad de enseñar a resolver problemas; y poco tiempo dedicado a la planificación de la materia.

En su investigación [14] ha mostrado 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 de [8] explora 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.

En [15] en su investigación, aprendizaje y modelización de ecuaciones diferenciales. Presenta la descripción del proceso de modelación, el análisis de los manuales comúnmente usados en clases, y la implementación de una situación experimental con tareas no habituales, lo cual permitió la identificación y la influencia de las praxeologías en el proceso de aprendizaje de los estudiantes. Un resultado relevante de los autores es el enfatizar la importancia de la construcción del modelo pseudo-concreto adecuado para establecer posteriormente los modelos físico y/o matemático que sean pertinentes la problemática en cuestión.

La investigación de [16] estudia la aproximación a la solución de ecuaciones diferenciales por métodos no convencionales y el software Mathematica. El método analizado es el de Euler, los autores indican que su investigación reviste una utilidad más didáctica que práctica, ya que no se persigue mejorar los métodos numéricos existentes para aproximar las soluciones de las ecuaciones diferenciales, sino mejorar los niveles de comprensión sobre la naturaleza de las ecuaciones, sus soluciones, los límites y la convergencia.

Por otro lado, [17] presenta una experiencia de la integración del aprendizaje basado en proyectos cortos de investigación, el modelado matemático y las TICs en un curso de Ecuaciones Diferenciales Aplicadas, logrando la interpretación de cada uno de los casos de acuerdo con la cinética del modelo y las diversas perturbaciones según el contexto planteado. Los autores comprobaron el poder de algunas herramientas computacionales para plantear y resolver sistemas de ecuaciones diferenciales mediante algoritmos de aproximación, así como la importancia de resaltar las posibles aplicaciones de las ecuaciones diferenciales en el área académica de los estudiantes.

En otra investigación [18] presentan una investigación de nuevas tecnologías en la enseñanza de las ecuaciones diferenciales de primer orden. En este estudio incorporaron el uso del software Mathematica en la enseñanza de las ecuaciones diferenciales a los estudiantes de ingeniería. Se evidencia que, en el tema de ecuaciones diferenciales, el rendimiento académico de los estudiantes que utilizaron material didáctico mediante el uso del software Mathematica es mayor, con relación al es-

tudiante que no lo utilizaron; esto fue posible mostrar mediante el promedio de notas obtenidos en cada corte. Los autores indican que, es importante motivar a los docentes de matemáticas a que se incorpore nuevas tecnologías como Mathematica en el proceso de enseñanza y aprendizaje de los currículos, esto con el fin de impulsar cambios en los procesos de enseñanza.

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 muestran deficiencias en la evolución de la interfaz de las aplicaciones utilizadas, cuyo uso provoca fenómenos didácticos importantes.

El resto del artículo se estructura de la siguiente manera. A continuación, se presenta una breve introducción de ecuaciones diferenciales y Mathematica, luego se presenta la metodología utilizada en esta investigación. Seguidamente, se exponen los resultados obtenidos a partir del análisis de cada método propuesto. El artículo finaliza con la discusión y las conclusiones.

### III. ECUACIONES DIFERENCIALES Y MATHEMATICA

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 lo tanto, es un campo muy importante para los estudiantes. 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 encuentran 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 métodos sistemáticos y eficaces que nos lleven a una aproximación numérica de las soluciones [16]. 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.

### A. Mathematica

Es un programa desarrollado por Wolfram Research ([www.wolfram.com](http://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 [20].

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.

### B. 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 [20].

NDSolve tiene una serie de formas de expresión:

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

La línea anterior permite calcular una solución numérica para ecuaciones diferenciales ordinarias, para una función  $\mu$  con la variable independiente  $x$  en el rango  $x_{\min}$  a  $x_{\max}$

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

Resuelve ecuaciones diferenciales parciales eqns sobre una región rectangular.

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

Resuelve las ecuaciones diferenciales parciales eqns sobre la región  $\Omega$ .

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

Resuelve las ecuaciones diferenciales parciales dependientes del tiempo eqns sobre la región  $\Omega$ .

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

Resuelve las funciones  $\mu_i$ .

## IV. METODOLOGÍA

Para cumplir con el objetivo de esta investigación, se utilizan el modelo de Lotka-Volterra. Las ecuaciones de Lotka-Volterra, también conocidas como ecuaciones depredador-presa, son un sistema de dos ecuaciones diferenciales de primer orden, acopladas, autónomas y no lineales, que se usan para describir dinámicas de sistemas biológi-

cos en el que dos especies interactúan, una como presa y otra como depredador [21].

$$\begin{aligned}\frac{dx}{dt} &= x(a - by) \\ \frac{dy}{dt} &= y(-c + dx)\end{aligned}\quad (1)$$

Concretamente, para realizar este estudio se selecciona el modelo presa-depredador, en la que se representa a la población de linces y conejos de un bosque al norte de Canadá llevada a cabo por la empresa Hudson Bay. La Tabla 1 muestra los resultados de los índices de linces y conejos elaborada por la empresa Hudson Bay entre los años 1900 y 1921.

Tabla I. Conejos y linces en miles

Año	Conejos	Linces	Año	Conejos	Linces
1900	30	4	1911	40.3	8
1901	47.2	6.1	1912	57	12.3
1902	70.2	9.8	1913	76.6	19.5
1903	77.4	35.2	1914	52.3	45.7
1904	36.3	59.4	1915	19.5	51.1
1905	20.6	41.7	1916	11.2	29.7
1906	18.1	19	1917	7.6	15.8
1907	21.4	13	1918	14.6	9.7
1908	22	8.3	1920	16.2	10.1
1909	25.4	9.1	1921	24.7	8.6
1910	27.1	7.4			

Con los datos de la Tabla 1 y con los valores iniciales  $x(0) = 40$ ;  $y(0) = 5$ , el modelo presa-depredador queda definido como:

$$\begin{aligned}\dot{x} &= x(0.4 - 0.018y) \\ \dot{y} &= y(0.023x - 0.8)\end{aligned}\quad (2)$$

Para calcular las  $I$  (invariantes) del sistemas (2), se realiza el proceso de separación de variables, se obtiene la ecuación (3).

$$I(x, y) = 0.8 \log x - 0.023x + 0.4 \log y - 0.018y \quad (3)$$

Considerando las ecuaciones (2) y siguiendo las referencias de [20,22-23], se analiza los métodos numéricos de resolución de ecuaciones diferencia-

les utilizando funciones de Mathematica. Para los cálculos respectivos se utiliza el valor de  $h = 1/10$ .

## V. RESULTADOS

En este apartado, primero se instala todos los paquetes necesarios para realizar los cálculos de los métodos plateados. Seguidamente, se desarrolla el modelo (2) utilizando el modelo de Lotka-Volterra. A continuación, se desarrolla cada uno de los métodos: Euler explícito, Runge Kutta 4 y LocallyExact. Finalmente, se realiza la comparación de los métodos. Concretamente, para la comparación de los métodos se utiliza el comando NDSolve de Mathematica. En esta investigación se utiliza la versión de Mathematica 10.4.0.

### A. 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 las ecuaciones del modelo (2) y la ecuación (3) de las invariantes en las variables respectivas en Mathematica, por último, se procede a crear una función que dibuja las orbitas del modelo (2) con el método seleccionado.

Paquetes necesarios para utilizar los componentes de NDSolve:

```
Needs["DifferentialEquations`NDSolveProblems`"];
Needs["DifferentialEquations`NDSolveUtilities`"];
Needs["DifferentialEquations`InterpolatingFunctionAnatomy`"];
```

Introducción del modelo (2) y las invariantes:

```
solucion
= NDSolveProblem[{{Subscript[Y, 1]'[T] =
= Y1[T](0.4 - 0.018Y2[T]),
Subscript[Y, 2]'[T] =
= (-0.8
+ 0.023Y1[T]) Y2[T]}, {Y1[0] =
= 30, Y2[0] == 5,
```

```
{Y1[T], Y2[T]}, {T, 0, 20}, {}, {0.4Log[Y2[T]]
- 0.018Y2[T] + 0.8Log[Y1[T]]
- 0.023Y1[T]}, {}];
```

Función Lotka-Volterra:

```
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];
```

## B. Métodos numéricos para la resolución de ecuaciones diferenciales

En este apartado se detalla la solución en Mathematica de cada uno de los métodos numéricos para la resolución de ecuaciones diferenciales ordinarias.

### Método 1: Euler explícito

Euler propuso uno de los primeros métodos para resolver ecuaciones diferenciales con problemas de valores iniciales, este método es muy sencillo en sus cálculos. Este método permite aproximar soluciones del problema de valor inicial.

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

$$y(x_0) = y_0$$

Para resolver el sistema (2) se utiliza la función `ExplicitEuler` de Mathematica, el método Euler Explícito que se encuentra en el comando `NDSolve`.

```
Meteuler = NDSolve[ecuacion, Method
-> "ExplicitEuler", StartingStepSize -> step]
grafico = LotkaVolterraPlot[MetEuler, vars, time]
```

La Fig. 1 muestra la solución del modelo (2), utilizando el método de Euler Explícito.

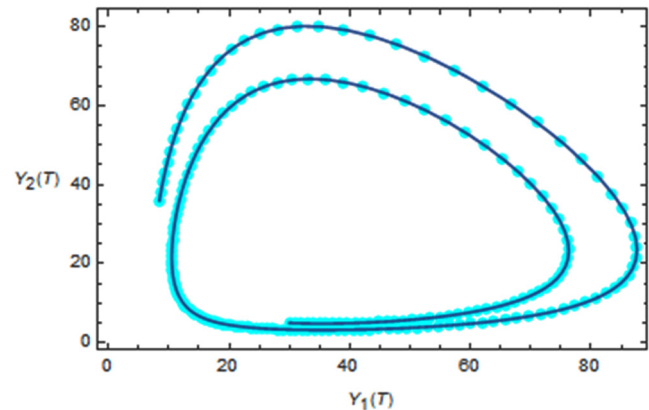


Fig. 1. Modelo Lotka-Volterra, método Euler Explícito.  
Nota: Fuente propia del estudio.

Se aplica la función de Lotka-Volterra creada en el apartado anterior para calcular las ecuaciones diferenciales, en este caso se ingresa la ecuación que se desea resolver y se realiza la respectiva ejecución de la función. Se utiliza el método de Euler explícito para desarrollar el modelo (2), la solución del método de Euler Explícito para las ecuación (2) con un tamaño de paso de  $h= 1/10$  con valores iniciales (40,5) se muestra en la Figura 1. Se traza las aproximaciones numéricas de los primeros 100 pasos con la aplicación del método numérico de Euler, todos con tamaños de pasos constantes. En la Figura 1 se observa que el método de Euler explícito muestra un comportamiento cualitativo incorrecto ya que las soluciones numéricas son en espiral hacia afuera o hacia adentro.

### Método 2: Runge Kutta

Para la resolución numérica de ecuaciones diferenciales se tiene una variedad de métodos, entre ellos los métodos de Runge-Kutta que son un conjunto de métodos desarrollados por Carl David Tolmé Runge y Martin Wilhelm Kutta. Los métodos de Runge-Kutta forman una clase importante de métodos para la integración de ecuaciones diferenciales. En esta investigación se analiza el método Runge-Kutta de cuarto orden.

Para resolver estas ecuaciones se crea una función en Mathematica que resuelve un modelo de ecuaciones diferenciales ordinarias por el método Runge-Kutta orden 4.

```
RungeKutta4[___][Step[f_, t_, h_, y_, fxy_]]:
= Block[{yt, k1, k2, k3, k4},
k1 = fxy;
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];
yt = h (1/6 k1 + 1/3 k2 + 1/3 k3 + 1/6 k4);
{h, yt}];
```

La Fig. 2 muestra la órbita del modelo Lotka-Volterra utilizando el método Runge-Kutta de cuarto orden.

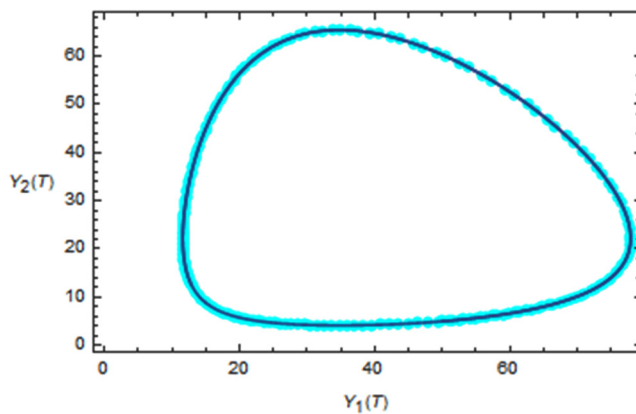


Fig. 2. Modelo Lotka-Volterra, método Runge-Kutta 4. Nota: Fuente propia del estudio.

La Fig. 2 muestra la solución de los métodos de Runge Kutta 4 para las ecuaciones (2) con un tamaño de paso de  $h=1/10$  con valores iniciales (40,5). Además, la Fig. 2 ilustra un diagrama de fase correcto para el modelo de Lotka-Volterra (2), estos métodos son bastante satisfactorios.

### Método 3: LocallyExact

Por último, se utiliza los métodos Splitting, estos integran ecuaciones diferenciales ordinarias, en este trabajo investigativo se utiliza el método Splitting LocallyExact, usa la evaluación numérica directa para resolver localmente la solución. En Mathematica se resuelve cada ecuación con el comando DSolve, seguidamente se utiliza una línea de comando en la que se utiliza el método LocallyExact.

```
DSolve[Y1, var, T]
DSolve[Y2, var, T]
LocallyExact = {"Splitting", "DifferenceOrder"
→ 1, "Equations" → {Y1, Y2}, "
Method" → {"LocallyExact", "LocallyExact"}];
```

La Fig. 3 muestra la órbita del método LocallyExact.

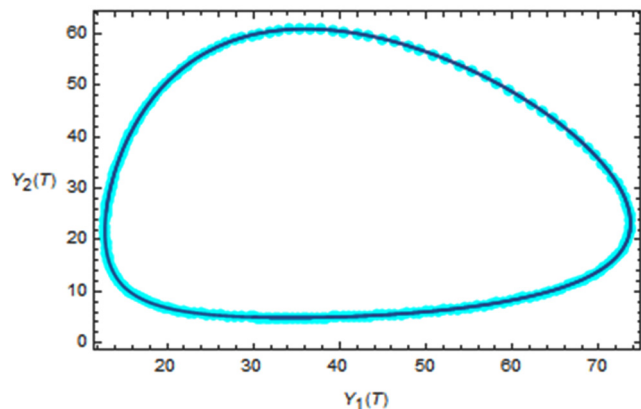


Fig. 3. Modelo Lotka-Volterra, método LocallyExact. Nota: Fuente propia del estudio.

La Fig. 3 muestra la solución para los métodos de Splitting LocallyExact con valores iniciales (40,5), para desarrollar este método, primero se resuelve las ecuaciones del sistema (2) con el comando DSolve, resolviendo de esta manera cada ecuación diferencial independientemente. A partir de cada ecuación independiente se procede a aplicar el comando LocallyExact.

### Comparación de los métodos con NDSolve

Para realizar la comparación de los métodos se utiliza la solución del comando NDSolve de Mathematica. La Fig. 4 muestra el plano de fase del modelo Lotka-Volterra utilizando el comando NDSolve, de aquí en adelante se identifica como solución exacta.

La Fig. 5 muestra la comparación del método Euler Explícito y la solución de del comando NDSolve.

La Fig. 5 muestra las orbitas del método Euler explícito (turquesa) y NDSolve (rojo). En este, claramente se observa que el método de Euler se aleja demasiado a la solución exacta.

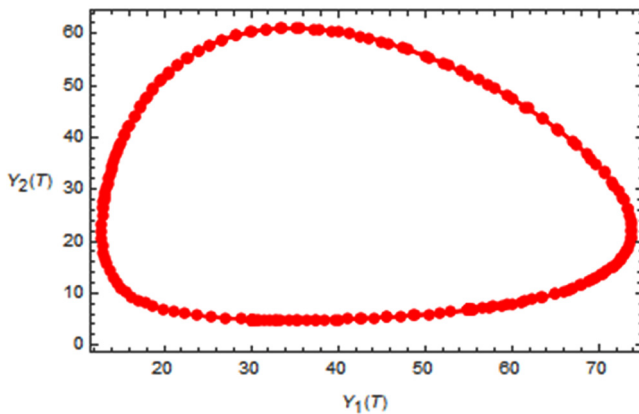


Fig. 4. Modelo Lotka-Volterra, método NDSolve.  
Nota: Fuente propia del estudio.

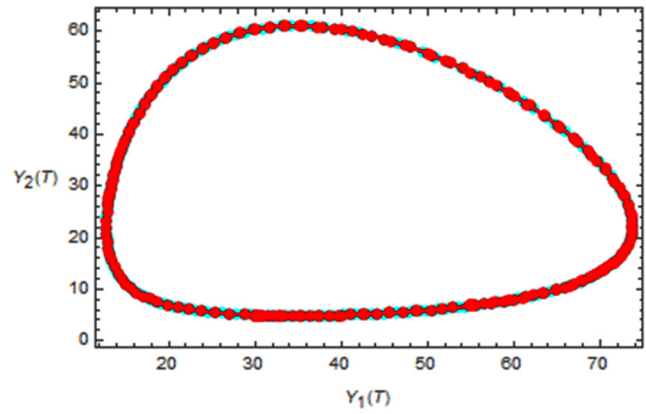


Fig. 6. Métodos de Runge Kutta y NDSolve.  
Nota: Fuente propia del estudio.

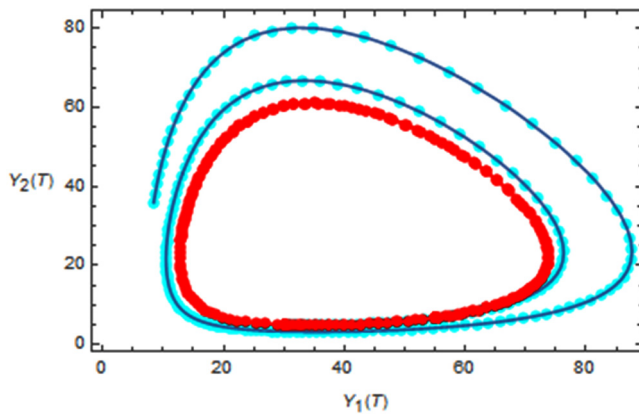


Fig. 5. Métodos de Euler Explícito y NDSolve.  
Nota: Fuente propia del estudio.

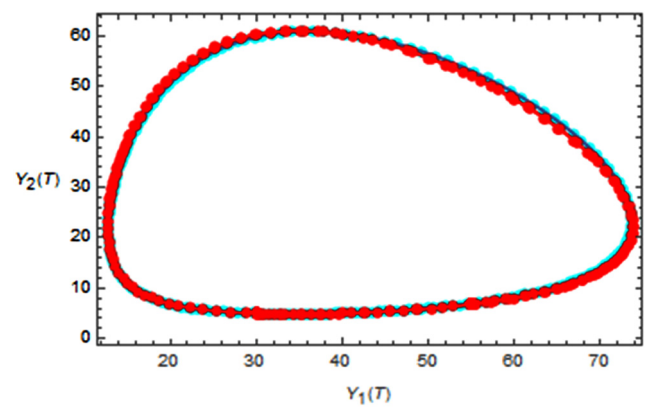


Fig. 7. Métodos de LocallyExact y NDSolve.  
Nota: Fuente propia del estudio.

Seguidamente, en la Fig. 6 se muestra la comparación del método Runge Kutta de cuarto orden y la solución exacta.

La Fig. 6 muestra que el método Runge Kutta (turquesa) y NDSolve (rojo), tienen idéntica solución. Se observa claramente que la mayoría de puntos de los dos métodos coinciden. Las variaciones entre estos pueden ser muy pequeños.

Por otro lado, la Fig. 7 muestra la comparación del método LocallyExact y la solución exacta.

La Fig. 7 presenta las orbitas de los métodos LocallyExact (turquesa) y NDSolve (rojo). En este se observa que los métodos de LocallyExact es acerca a la solución exacta, aunque existen ciertos puntos que se alejan levemente de la solución exacta.

La Fig. 8 muestra la comparación de los tres métodos Euler Explícito (turquesa), Runge Kutta 4 (negro) y LocallyExact (azul) con la solución exacta (rojo).

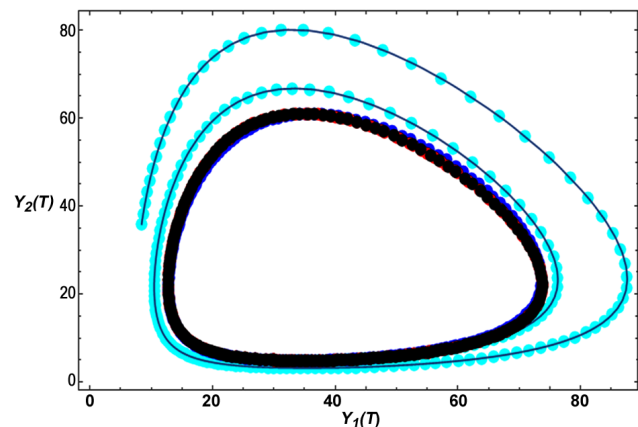


Fig. 8. Métodos de Euler Explícito, Runge Kutta, LocallyExact y NDSolve. Nota: Fuente propia del estudio.

La Fig. 8 muestra que los métodos de Runge-Kutta tienen una exactitud mayor sobre los métodos de Euler Explícito y LocallyExact. El método Euler Explícito es el que más se aleja de la solución exacta, este es un resultado esperado debido a la simplicidad de este método.



## 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 los valores de 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. 9 muestra la gráfica de las Invariantes de los métodos Euler Explícito (verde), Locally Exact (rojo) y Runge Kutta 4 (azul).

La Fig. 9 muestra que el método de Euler Explícito no tiene una buena conservación, el error invariante va aumentando. El método de Runge Kutta 4 y LocallyExact tienen una mejor conservación, por lo que, la Fig. 10 muestra los resultados de estos dos métodos en una escala menor.

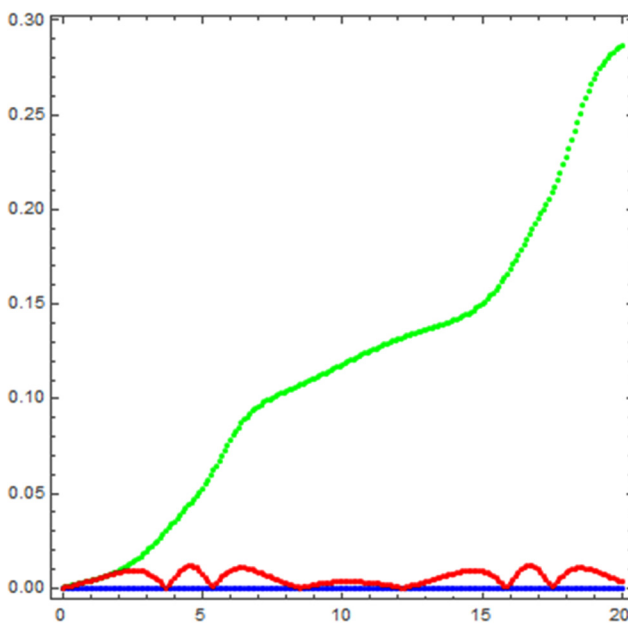


Fig. 9. Invariantes Métodos de Euler Explícito, Runge Kutta y LocallyExact. Nota: Fuente propia del estudio.

La Fig. 10 muestra claramente que el método de Runge Kutta 4 (azul) tiene mejor conservación que el método de LocallyExact (rojo).

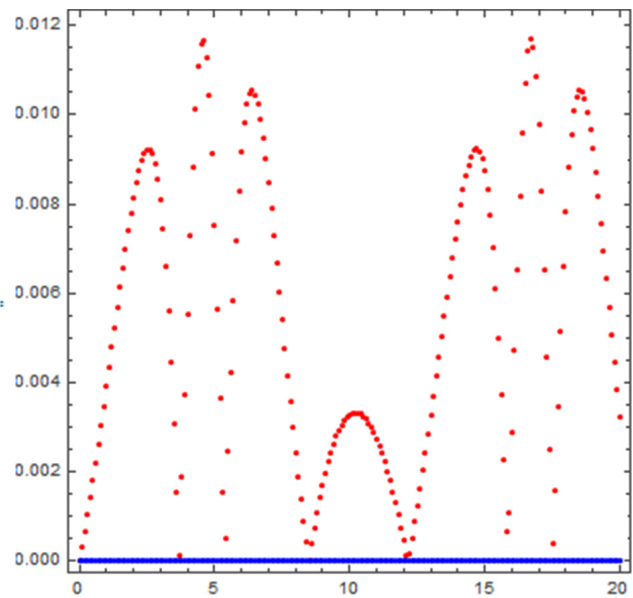


Fig. 10. Invariantes Métodos de Euler Explícito, Runge Kutta y LocallyExact. Nota: Fuente propia del estudio.

## VI. DISCUSIÓN Y CONCLUSIONES

El objetivo principal de este trabajo investigativo es analizar y comparar la solución numérica de ecuaciones diferenciales ordinarias utilizando los métodos Euler Explícito, Runge Kutta 4 y LocallyExact con el modelo Lotka-Volterra. Para cumplir el objetivo se utiliza el software de Mathematica. También, este estudio tiene el propósito que estudiantes universitarios utilicen estos resultados como complemento de formación.

El primer método que se analizó fue el método de Euler ya que es el más simple. Este método es el que más se aleja de la solución de NDSolve. Los valores de las Invariante se elevan considerablemente produciendo errores muy altos. Una ventaja de este método es que no consigue un esfuerzo computacional. Se coincide con la investigación de [24], estos autores reportan de manera similar que el valor calculado en el método de Euler se aleja apreciablemente del valor verdadero cuando el paso de integración es lo suficientemente grande, es por esto, que se deben adoptar pasos muy pequeños.

El segundo método utilizado en esta investigación es Runge Kutta 4, este método se acerca de una manera impresionante a los valores de la solución de NDSolve, en las orbitas se observa una

diferencia mínima con respecto a la solución exacta. También, se aprecia niveles muy bajos de error producido por la invariante. Esta investigación coincide con [24], los autores indican que el método de Runge-Kutta 4 tiene muy buena precisión, teniendo en cuenta los variados cambios de pendiente de las curvas que representan el modelo, durante el proceso de conducción del simulador. También, [25] indica que el algoritmo para el método de Runge-Kutta de cuarto orden, o método clásico de Runge-Kutta, abreviado como RK4, es de uso extendido, y reconocido como una valiosa herramienta de cálculo, por la buena aproximación que produce.

Por último, se analiza los métodos de splitting y composición: LocallyExact, tiene un acercamiento a la solución dada por NDSolve aunque no tiene la misma precisión que los métodos Runge Kutta 4.

Como conclusión se puede decir que los resultados obtenidos en esta investigación indica 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 Euler Explícito y de LocallyExact, 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 proceso de enseñanza y aprendizaje, con el objetivo de incrementar el interés y mejorar el aprendizaje de los estudiantes en el manejo de los métodos numéricos para la solución de ecuaciones diferenciales. Se recomienda a los estudiantes y profesores utilizar manipulativos matemáticos con el propósito de verificar resultados y/o mejorar la comprensión de contenidos.

Una futura línea de investigación podría ser el estudio del esfuerzo computacional de cada uno de los métodos numéricos utilizados en este trabajo.

## REFERENCIAS

- [1] J. Barbarán y 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. 183-209, 1989.
- [3] M. Kallaher, *Revolutions in Differential Equations: Exploring ODEs with Modern Technology*. Washington: The Mathematical Association of America, 1999.
- [4] C. Rasmussen, "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", en G. Harel y E. Dubinsky (eds.), *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. 455-472, 2000.
- [8] G. Ortigoza, "Ecuaciones diferenciales ordinarias con Maxima", *Educación matemática*, vol. 21, no. 2, pp. 143-167, 2009.
- [9] L. Santos, "Potencial didáctico del software dinámico en el aprendizaje de las matemáticas", *Avance y Perspectiva*, vol. 20, pp. 247-258, 2001.
- [10] N. Blackett y 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.
- [11] K. Moore, "Trigonometry, technology, and didactic objects", In S. Swars, D. Stinson, & S. Lemons-Smith (red.), *Proceedings of 31st Meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education*, vol. 5, pp. 1480-1488, 2009.
- [12] *National Council of Teachers of Mathematics [NCTM]*, Principles and standards for school mathematics. Reston, VA: NCTM, 2000.
- [13] M. Moreno y C. Azcárate, "Concepciones y Creencias de los Profesores Universitarios de Matemáticas acerca de la Enseñanza de las Ecuaciones Diferenciales", *Enseñanza De Las Ciencias*, vol. 21, no. 2, pp. 265-280, 2003.

- [14] G. Ortigoza, "Resolviendo ecuaciones diferenciales ordinarias con Maple y Mathematica", *Revista mexicana de física E*, vol. 53, no. 2, pp. 155-167, 2007.
- [15] R. Rodríguez, "Aprendizaje y enseñanza de la modelación: el caso de las ecuaciones diferenciales", *Revista Latinoamericana de Investigación en Matemática Educativa, RELIME*, vol. 13, no. 4, pp. 191-210, 2010.
- [16] M. Sandoval, C. Ruvalcaba, J. González, Ó. Chávez y F. García, "Aproximación a la solución de ecuaciones diferenciales por métodos no convencionales y el software Mathematica", *Cultura Científica y Tecnológica*, vol. 47, 2015.
- [17] J. Molina, "Experiencia basada en la triada TICs, enseñanza por proyectos y modelado para la enseñanza de sistemas de ecuaciones diferenciales", *Uniciencia*, vol. 29, no. 2, pp. 46-61, 2015.
- [18] M. García y W. Rivera, "Nuevas tecnologías en la enseñanza de las ecuaciones diferenciales de primer orden", *Conferencia presentada en Encuentro Internacional de Matemáticas, Estadística y Educación Matemática. Duitama, Colombia*, 2015.
- [19] C. Ríos, F. Caldera, M. Ivette y V. Valenzuela, "Fidelidad en el uso de app para la resolución de ecuaciones diferenciales", *Apertura*, vol. 11, no. 1, pp. 74-89, 2019.
- [20] Mathematica Wolfram, <https://www.wolfram.com/support>.
- [21] A. Oganician, *Modelo depredador-presa de Volterra-Lotka*, Universidad de la Laguna, España, 2017.
- [22] E. Hairer, C. Lubich y G. Wanner, *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*. Springer Science & Business Media, 2006.
- [23] J. Segarra, *Resolución numérica de ecuaciones diferenciales en Wolfram Mathematica*. Universitat Jaume I, España, 2018.
- [24] G. Di Rado, G. Devincenzi y S. Presta, "Aplicación del Método de Integración Numérica de Ecuaciones Diferenciales Runge Y Kutta 4 (Rk4) a un Modelo de Simulación Longitudinal de Dinámica Vehicular Terrestre", *Education Research Journal*, vol. 30, pp. 91-112, 2011.
- [25] D. Zill, *Differential equations with modeling applications*. México: International Thomson Editores, 2002.

