UNIVERSIDAD DEL VALLE DE GUATEMALA

ECUACIONES DIFERENCIALES I
Sección 10

Proyecto Final

Eliazar José Pablo Canastuj Matías - 23384
Diego Alejandro Ramírez Velásquez - 23601
Melanie Carolina Maldonado Herrera – 21720

GUATEMALA, **18 de noviembre de 2025**

Resumen

En este trabajo se implementaron y evaluaron métodos numéricos para la solución de ecuaciones diferenciales ordinarias de primer y segundo orden, así como de sistemas lineales y no lineales. Se desarrollaron los métodos de Runge–Kutta de segundo y cuarto orden y se aplicaron a diversos problemas, comparando los resultados con soluciones analíticas cuando éstas estaban disponibles. El análisis incluyó tablas de error, gráficas de comparación y estudios de convergencia para diferentes tamaños de paso.

Los resultados mostraron que ambos métodos convergen adecuadamente hacia la solución exacta o de referencia, con un comportamiento consistente con su orden teórico. RK4 demostró una precisión sustancialmente mayor que RK2, especialmente en intervalos largos y en sistemas oscilatorios. Para el sistema no lineal de Lotka–Volterra, sin solución analítica cerrada, se utilizó una solución numérica refinada como referencia, observándose una buena reproducción de la dinámica del modelo y estabilidad en la integración.

El estudio confirma la relevancia de los métodos numéricos en la Ingeniería en Computación, donde la simulación de sistemas dinámicos es fundamental y donde la precisión y estabilidad de los algoritmos afectan directamente la calidad de los análisis.

Introducción

Las ecuaciones diferenciales constituyen una herramienta fundamental en la modelación matemática de fenómenos dinámicos presentes en la ciencia y la ingeniería. Su capacidad para describir cómo varían las magnitudes en el tiempo o en el espacio permite representar sistemas tales como el crecimiento poblacional, la dinámica de fluidos, el comportamiento de señales eléctricas, la evolución térmica de materiales o los circuitos electrónicos. En particular, en el ámbito de la Ingeniería en Computación, las ecuaciones diferenciales aparecen de forma natural en la modelación de sistemas de control, redes neuronales continuas, simulación física para gráficos por computadora, análisis de señales, dinámica de robots, optimización y en numerosos algoritmos que requieren representar o aproximar fenómenos continuos.

En muchos casos, las ecuaciones diferenciales que surgen en problemas reales no poseen una solución analítica cerrada, o bien dicha solución es demasiado compleja para ser utilizada directamente. Por esta razón, los métodos númericos ocupan un papel central ya que permiten aproximar la solución con la precisión deseada mediante algoritmos iterativos que pueden implementarse y ejecutarse eficientemente en una computadora. En este proyecto se estudian e implementan métodos numéricos iterativos para resolver ecuaciones diferenciales ordinarias de primer y segundo orden, así como sistemas lineales y no lineales de dos variables. A través de comparaciones con soluciones analíticas, análisis de error y estudio de convergencia, se evalúa el desempeño de los métodos implementados.

Fundamento Teórico

Los métodos numéricos implementados en este proyecto pertenecen a la familia de *métodos de Runge–Kutta (RK)*, ampliamente utilizados en ingeniería debido a su equilibrio entre precisión, estabilidad y facilidad de implementación. Estos métodos construyen aproximaciones

de la solución evaluando la función $f(t, y)$ en varios puntos dentro de cada paso temporal.

Método de Runge–Kutta de segundo orden (RK2)

El método de Heun aproxima la solución mediante dos evaluaciones de la derivada por paso. Para un tamaño de paso $h > 0$ y un valor aproximado $y_n \approx y(t_n)$, se define:

$$k_1 = f(t_n, y_n), \qquad k_2 = f(t_n + h,\, y_n + hk_1),$$
$$y_{n+1} = y_n + \frac{h}{2}\,(k_1 + k_2).$$

Alcance

- Orden global de convergencia: 2.
- Requiere bajo costo computacional.
- Adecuado para problemas suaves y sistemas moderadamente no lineales.

Limitaciones

- Menor precisión que métodos de orden superior.
- Necesita tamaños de paso pequeños para mantener estabilidad en sistemas rígidos.
- Puede introducir errores acumulados significativos en intervalos largos.

Método de Runge–Kutta de cuarto orden (RK4)

El método RK4 es uno de los más utilizados por su alta precisión relativa. Para cada paso:

$$k_1 = f(t_n, y_n),$$
$$k_2 = f\left(t_n + \frac{h}{2},\, y_n + \frac{h}{2}k_1\right),$$
$$k_3 = f\left(t_n + \frac{h}{2},\, y_n + \frac{h}{2}k_2\right),$$
$$k_4 = f(t_n + h,\, y_n + hk_3),$$
$$y_{n+1} = y_n + \frac{h}{6}\,(k_1 + 2k_2 + 2k_3 + k_4).$$

Alcance

- Orden global de convergencia: 4, con errores notablemente menores que RK2.
- Método estable para una amplia variedad de sistemas no lineales.
- Excelente relación precisión–costo computacional.
- Muy utilizado en simulación física, gráficas computacionales, sistemas de control y análisis dinámico.

Limitaciones

- Más costoso computacionalmente.
- Como la mayoría de métodos explícitos, puede fallar o volverse ineficiente en sistemas rígidos.
- La estabilidad depende fuertemente del tamaño de paso, especialmente en ecuaciones oscilatorias.

Supuestos, variables y notación utilizada

Para el desarrollo e implementación de los métodos numéricos empleados en este proyecto se adoptan los siguientes supuestos, se definen las variables involucradas y se establece la notación estándar utilizada a lo largo del informe:

Supuestos

1. **Existencia y unicidad de la solución.**
   Se asume que la función $f(t, y)$ que define la ecuación diferencial es continua y satisface una condición de Lipschitz en y, garantizando la existencia y unicidad de la solución del problema de valor inicial.

2. **Tamaño de paso constante.**
   En todas las aproximaciones numéricas se utiliza un tamaño de paso uniforme

   $$h = \frac{t_f - t_0}{N},$$

   tal como lo requieren los métodos RK2 y RK4 en su formulación estándar.

3. **No se consideran errores de redondeo.**
   Se asume que los errores asociados al hardware (precisión de doble punto flotante) son pequeños comparados con los errores de truncamiento propios del método.

4. **Condiciones iniciales exactas.**
   Las condiciones iniciales $y(t_0) = y_0$ se consideran conocidas sin error.

Variables

- t: variable independiente (tiempo).
- $y(t)$: solución exacta del sistema en el tiempo t.
- y_n: aproximación numérica de $y(t_n)$.
- t_0, t_f: tiempos inicial y final del intervalo de integración.
- h: tamaño de paso o incremento temporal.
- N: número total de pasos, dado por $N = (t_f - t_0)/h$.
- $f(t, y)$: campo vectorial que define la ecuación diferencial ordinaria, ya sea escalar o de dimensión 2×2.
- k_i: evaluaciones intermedias del método de Runge–Kutta.

Solución analítica de las EDs

- **ED de primer orden**

$$y' = -2y + 3e^{-t}$$

$$y' + 2y = 3e^{-t}.$$

$$\mu(t) = e^{\int 2\,dt} = e^{2t}.$$

$$e^{2t}y' + 2e^{2t}y = 3e^{t}.$$

$$\frac{d}{dt}\big(e^{2t}y\big) = 3e^{t}.$$

$$e^{2t}y = 3\int e^{t}\,dt = 3e^{t} + C.$$

$$y(t) = (3e^{t} + C)e^{-2t} = 3e^{-t} + Ce^{-2t}.$$

$$\boxed{y(t) = 3e^{-t} + Ce^{-2t}}.$$

- **ED de segundo orden**

$$y'' + 4y = 0$$

$$r^2 e^{rt} + 4e^{rt} = 0 \quad \Rightarrow \quad r^2 + 4 = 0.$$

$$r^2 + 4 = 0 \quad \Rightarrow \quad r = \pm 2i.$$

$$y(t) = C_1 \cos(\omega t) + C_2 \sin(\omega t).$$

$$\boxed{y(t) = C_1 \cos(2t) + C_2 \sin(2t)}.$$

- **Sistema 2 × 2 lineal**

$$\begin{cases} y' = x + y, \\ x' = 3x - y \end{cases}$$

$$\mathbf{u} = \begin{pmatrix} x \\ y \end{pmatrix} : \qquad \mathbf{u}' = A\mathbf{u}, \qquad A = \begin{pmatrix} 3 & -1 \\ 1 & 1 \end{pmatrix}.$$

$$\det(A - \lambda I) = \begin{vmatrix} 3-\lambda & -1 \\ 1 & 1-\lambda \end{vmatrix} = (3-\lambda)(1-\lambda) + 1 = \lambda^2 - 4\lambda + 4 = (\lambda - 2)^2.$$

$\lambda = 2$.

$$A - 2I = \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}.$$

$$(1)x - (1)y = 0 \quad \Rightarrow \quad x = y.$$

$$v_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

$$(A - 2I)v_2 = v_1.$$

$$v_2 = \begin{pmatrix} a \\ b \end{pmatrix}, \quad \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \Rightarrow \quad a - b = 1.$$

$$v_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

$$\mathbf{u}(t) = e^{2t}\big[C_1 v_1 + C_2(t\, v_1 + v_2)\big].$$

$$\mathbf{u}(t) = e^{2t}\left[C_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} + C_2 \left(t \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right)\right].$$

$$x(t) = e^{2t}\big(C_1 + C_2(t+1)\big), \qquad y(t) = e^{2t}\big(C_1 + C_2 t\big).$$

Resultados numéricos

- **ED de primer orden**

```
============================================
PROBLEMA 1: y' = -2y + 3*exp(-t), y(0) = 1
Intervalo: [0, 5]
============================================

h       método    error_maximo
0.1      RK2       2.464884e-03
0.1      RK4       5.829608e-06
0.05     RK2       5.589963e-04
0.05     RK4       3.297318e-07
0.01     RK2       2.076273e-05
0.01     RK4       4.870788e-10
```

Figura 1: Tabla de resultados

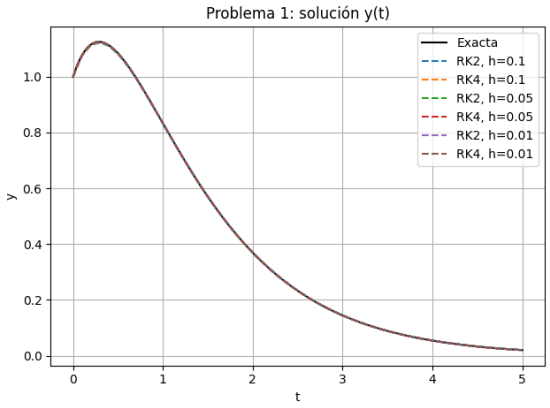


Gráfico de la solución $y(t)$.

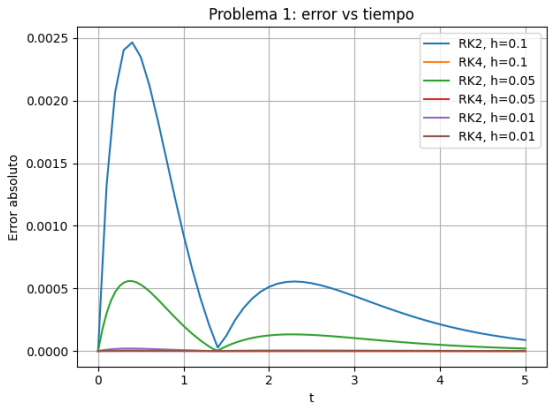


Gráfico del error vs tiempo

- **ED de segundo orden**

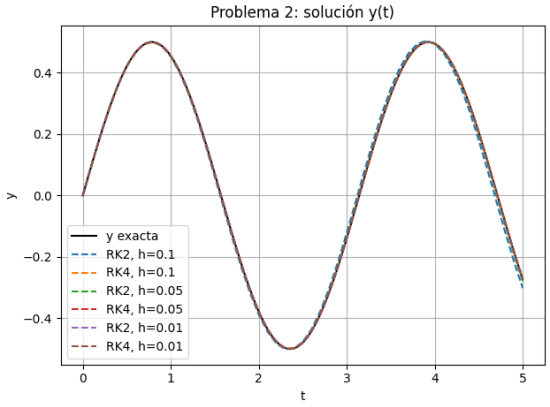


Gráfico de la solución $y(t)$.

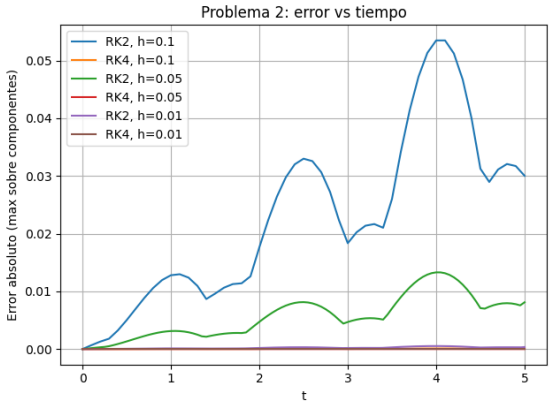


Gráfico del error vs tiempo

```
==========================================
PROBLEMA 2: y'' + 4y = 0, y(0)=0, y'(0)=1
Sistema: y1' = y2, y2' = -4*y1
Intervalo: [0, 5]
==========================================

h      método    error_maximo (sobre y e y')
0.1     RK2       5.348759e-02
0.1     RK4       1.075371e-04
0.05    RK2       1.329720e-02
0.05    RK4       6.660214e-06
0.01    RK2       5.287419e-04
0.01    RK4       1.057751e-08
```

Figura 2: Tabla de resultados

- **Sistema 2 × 2 lineal**

```
==========================================
PROBLEMA 3: sistema 2x2 lineal
   y' = x + y
   x' = 3x - y
   x(0) = 1, y(0) = 0
Intervalo: [0, 2]
==========================================

h      método    error_maximo (sobre x e y)
0.1     RK2       5.470961e+00
0.1     RK4       1.335733e-02
0.05    RK2       1.501256e+00
0.05    RK4       9.140624e-04
0.01    RK2       6.442415e-02
0.01    RK4       1.572695e-06
```

Figura 3: Tabla de resultados

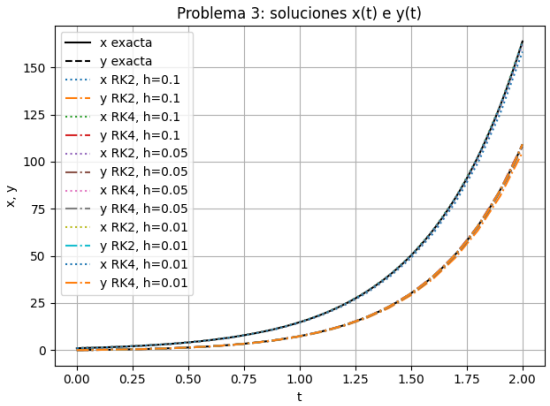


Gráfico de la solución $y(t)$.

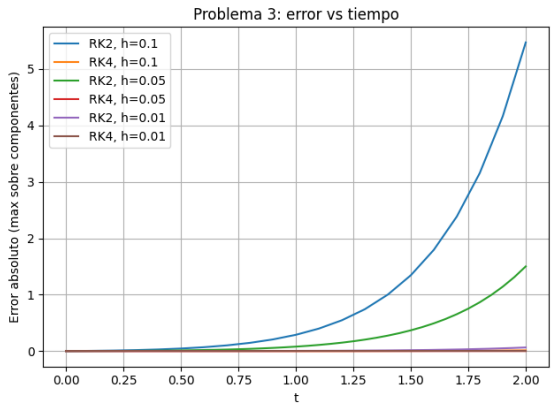


Gráfico del error vs tiempo

Solución numérica del sistema de ecuaciones no lineal

```
========================================
PROBLEMA 4: sistema no lineal de Lotka-Volterra
   dx/dt = α x - β x y
   dy/dt = δ x y - γ y
Parámetros: α=1.0, β=0.1, δ=0.075, γ=1.0
Condiciones iniciales: x(0)=10, y(0)=5
Intervalo: [0, 30]
========================================

h      método    error_maximo (vs referencia RK4, h=0.001)
0.1     RK2       3.992962e-01
0.1     RK4       2.041727e-04
0.05    RK2       1.055040e-01
0.05    RK4       1.354172e-05
0.01    RK2       4.405531e-03
0.01    RK4       2.266933e-08
```

Figura 4: Tabla de resultados

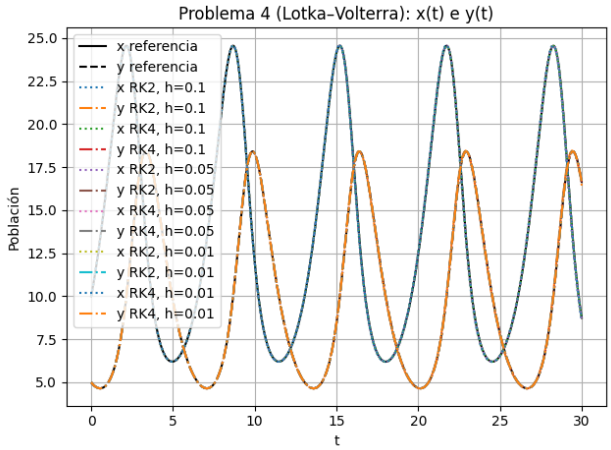
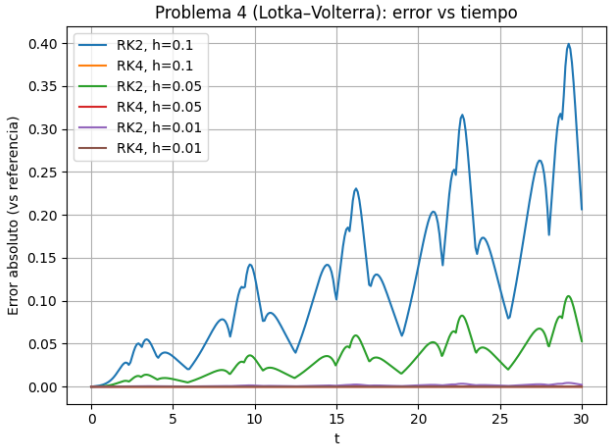
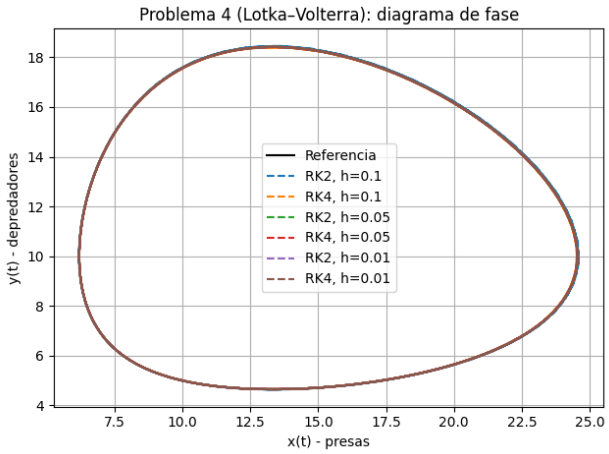


Figura 5: Gráfica de x(t) e y(t)





Discusión de resultados

Los resultados numéricos obtenidos permiten evaluar el desempeño y la precisión de los métodos de Runge–Kutta de segundo y cuarto orden (RK2 y RK4) aplicados a ecuaciones diferenciales ordinarias de diversa complejidad. En todos los problemas donde se dispone de una solución analítica, las gráficas muestran una coincidencia clara entre la solución exacta y las aproximaciones numéricas. Esto valida que las implementaciones de ambos métodos son correctas y que reproducen la dinámica teórica esperada.

En las tablas de error presentadas, se observa una tendencia sistemática: al disminuir el tamaño de paso h, el error máximo disminuye de acuerdo con el orden teórico de cada método. Para RK2, el error decrece de manera proporcional a h^2, mientras que para RK4 lo hace de forma proporcional a h^4. Esta diferencia se manifiesta en que RK4 alcanza errores considerablemente menores aun utilizando pasos más grandes. Las gráficas del error vs. tiempo refuerzan esta conclusión, evidenciando que RK4 mantiene errores muy pequeños a lo largo del intervalo y que RK2, si bien es preciso, exhibe una acumulación de error mayor en problemas no lineales o con soluciones de crecimiento exponencial.

En el caso del sistema no lineal de Lotka–Volterra, donde no existe una solución analítica cerrada, se utilizó como referencia la solución obtenida mediante RK4 con un paso muy pequeño. Los resultados muestran un comportamiento coherente con la dinámica clásica del modelo presa–depredador: oscilaciones periódicas en las poblaciones y ciclos cerrados en el plano de fases. Al comparar los métodos, RK2 produce oscilaciones más suaves y cercanas a la referencia, mientras que RK2 presenta desviaciones visibles cuando el tamaño de paso aumenta, especialmente en la aproximación de máximos y mínimos. La estabilidad de ambos métodos es adecuada para este problema, pero RK4 se muestra más robusto en modelos no lineales.

En términos de estabilidad numérica, los métodos explícitos como RK2 y RK4 pueden presentar dificultades cuando se aplican a sistemas rígidos. Aunque ninguno de los problemas analizados es rígido, en el sistema lotka–volterra se observa que RK2 tiende a amplificar pequeñas perturbaciones, especialmente en integraciones de largo plazo, lo que coincide con el carácter marginalmente estable de Lotka–Volterra. Esta tendencia es menor en RK4, lo cual coincide con su menor error y su región de estabilidad más limitada. RK4, en cambio,

presenta un desempeño notablemente más estable, manteniendo la forma del ciclo poblacional incluso con pasos relativamente grandes.

Los resultados numéricos corroboran los fundamentos teóricos del método, ambos esquemas convergen hacia la solución exacta al reducir el paso, pero RK4 proporciona una precisión significativamente mayor y un error mucho menor en la simulación de sistemas dinámicos. Esto justifica su uso predominante en aplicaciones de ingeniería, donde la estabilidad y la fidelidad de los resultados son esenciales.

Conclusiones

1. Los métodos de Runge–Kutta implementados resultaron correctos y confiables, ya que en todos los problemas con solución analítica las soluciones numéricas coincidieron estrechamente con la solución exacta. El análisis de error y las gráficas comparativas demostraron que ambos métodos convergen conforme disminuye el tamaño de paso, validando su adecuada implementación.

2. El método RK4 presentó un desempeño significativamente superior al de RK2, alcanzando errores varios órdenes de magnitud menores, incluso con tamaños de paso más grandes. Esto es consistente con su orden teórico y lo convierte en la opción preferida para aplicaciones que requieren alta precisión, estabilidad y eficiencia, especialmente en sistemas oscilatorios o de dinámica compleja.

3. En el sistema no lineal de Lotka–Volterra, los métodos numéricos capturaron correctamente la dinámica del modelo, reproduciendo las oscilaciones periódicas y el ciclo cerrado en el plano de fase. Ambos métodos fueron estables para los tamaños de paso analizados, aunque RK4 mostró una mayor fidelidad respecto a la solución de referencia. Esto confirma la utilidad de los métodos numéricos para resolver sistemas sin solución analítica y su importancia dentro de la Ingeniería en Computación, donde la simulación es fundamental para el estudio de sistemas dinámicos.

Referencias

- GeeksforGeeks. (2025, 12 julio). RungeKutta 2nd order method to solve Differential equations //. GeeksforGeeks. https://www.geeksforgeeks.org/dsa/runge-kutta-2nd-order-method-to-solve-differential-equations/
- JoshuaSimon. (s. f.). GitHub - JoshuaSimon/Classic-Fourth-Order-Runge-Kutta-Method-RK4: Python implementation of the classic fourth-order Runge Kutta method (RK4). GitHub. https://github.com/JoshuaSimon/Classic-Fourth-Order-Runge-Kutta-Method-RK4
- NPTEL-NOC IITM. (2020, 20 marzo). Improved Euler (RK2) and RK4 Methods for solving ODEs [Vídeo]. YouTube. https://www.youtube.com/watch?v=KE6Brb5wjxU
- Admin. (2022, 7 abril). Runge-Kutta RK4 Method. BYJUS. https://byjus.com/maths/runge-kutta-rk4-method/

- Zill, D. G. (2019). Ecuaciones diferenciales con aplicaciones de modelado (11.ª ed.). Cengage Learning.

Anexos

El código fuente completo de las implementaciones y experimentos se encuentra disponible en el siguiente repositorio:

`github.com/can23384/ProyectoEDs`

Video explicativo

`Link del video explicativo en YouTube`

Evidencia de uso de IA:

`Prompt de ChatGPT`

Prompt 1:

- **Fecha y hora:** 16/11/2025, 20:29.
- **Tarea y objetivo:** Pedir que realice un codigo en python para implementar los metodos numericos.
- **Prompt utilizado:**Necesito que me escribas un programa COMPLETO en Python, totalmente modular, documentado en español, y SIN usar ningún método simbólico (no usar SymPy). El programa debe implementar únicamente métodos numéricos y trabajar con mis tres ecuaciones diferenciales. REQUISITOS DEL PROGRAMA: Lenguaje: Python. Librerías permitidas: numpy, matplotlib. NO usar SymPy. Los métodos numéricos a implementar son: - Runge–Kutta de orden 2 (RK2 / método de Heun) - Runge–Kutta de orden 4 (RK4)
- **Decisión tomada:** Si se probo el codigo y fue aceptado.
- **Resultado o impacto:** Se tomo como base y luego se fueron alpicando cambios al codigo.

Prompt 2:

- **Fecha y hora:** 16/11/2025, 22:42.
- **Tarea y objetivo:** Hacer cambios para implementar las EDs.
- **Prompt utilizado:** ECUACIONES A RESOLVER (estas ya están definidas y NO deben resolverse de forma simbólica en el programa, solo numéricamente):

- **Decisión tomada:** Si se probo el codigo y fue aceptado.
- **Resultado o impacto:** Se probaron las ecuaciones y fueron coroboradas que si fueran correctas.