Evaluación comparativa de los métodos Runge-Kutta y Butcher de quinto orden

Esteban Machado Londoño

Ingeniero Industrial

Modalidad de Práctica

Inscripción Materias de posgrado

Asesor

Yony Fernando Ceballos, Doctor (PhD)

Universidad de Antioquia

Facultad de Ingeniería

Ingeniería Industrial

Medellín, Antioquia, Colombia

2026

Cita	(Audu *et al.*, 2025)
Referencia **Estilo APA 7 (2020)**	Audu, K. J., Yahaya, Y. A., & Nkereuwem, J. (2025). *Título:Comparative Numerical Evaluation of Some Runge-Kutta Methods for Solving First Order Systems of ODEs. 2024.* [Trabajo de grado profesional]. Universidad de Antioquia, Medellín, Colombia.

Tabla de contenido

Lista de tablas

Lista de figuras

Resumen

El artículo presenta una evaluación comparativa entre los métodos numéricos Runge-Kutta de cuarto orden (RK4) y Butcher Runge-Kutta de quinto orden (BRK5) para la solución de sistemas de ecuaciones diferenciales ordinarias de primer orden. A través de la implementación computacional en Python y la aplicación a cinco sistemas lineales con solución analítica conocida, se analiza la precisión, estabilidad y comportamiento del error de ambos métodos. Los resultados muestran que el método BRK5 ofrece mayor exactitud en la mayoría de los casos, aunque su desempeño depende del tamaño de paso y de la naturaleza del sistema diferencial, destacando que un mayor orden no garantiza mejores resultados bajo condiciones de inestabilidad numérica.

Palabras clave: artículo científico, ecuaciones diferenciales, métodos numéricos, sistemas lineales.

Abstract

This article presents a comparative evaluation of the fourth-order Runge-Kutta (RK4) and fifth-order Butcher Runge-Kutta (BRK5) numerical methods for solving systems of first-order ordinary differential equations. Through computational implementation in Python and application to five linear systems with known analytical solutions, the accuracy, stability, and error behavior of both methods are analyzed. The results show that the BRK5 method offers greater accuracy in most cases, although its performance depends on the step size and the nature of the differential system, highlighting that higher order does not guarantee better results under conditions of numerical instability.

Keywords: scientific article, differential equations, numerical methods, linear systems

1. Introducción

Las ecuaciones diferenciales ordinarias (EDOs) modelan fenómenos en ingeniería y ciencia. Se estudian sistemas de dinámica de poblaciones, análisis de circuitos eléctricos, sistemas mecánicos, entre otros. Estos modelos permiten describir cómo cambian las variables de estado con respecto al tiempo, capturando la esencia de los fenómenos dinámicos. Generalmente, muchos de los sistemas modelados con EDOs no tienen solución analítica simple, haciendo indispensable el uso de métodos numéricos. En este informe se replican los modelos matemáticos y resultados presentados en el artículo elaborado por (Audu et al., 2025). En este, los autores comparan la precisión y eficiencia de dos esquemas iterativos: el método Runge-Kutta de cuarto orden (RK4) y el método de Butcher de quinto orden (BRK5). Se busca evaluar la premisa de que, para los sistemas lineales de primer orden probados, el método RK4 presenta menores errores y mayor estabilidad que el método BRK5, a pesar de que la complejidad teórica de BRK5. Se propone una evaluación comparativa de la precisión, convergencia y eficiencia computacional entre los esquemas RK4 y BRK5 aplicados a sistemas de ecuaciones lineales de primer orden.

Los métodos de RK se destacan por su equilibrio entre precisión y estabilidad al no requerir el cálculo de derivadas de orden superior. El método RK4 es robusto y fácil de implementar. No obstante, la búsqueda de mayor precisión ha llevado al desarrollo de métodos de orden superior, como el método de BRK5, que ofrece un error de truncamiento local menor a cambio de un mayor costo computacional. Se busca verificar la consistencia matemática de la implementación de algoritmos de alto orden, contrastar la teoría numérica clásica con los resultados obtenidos y analizar cómo la estabilidad numérica y el tamaño de paso favorecen al método clásico RK4 bajo ciertas condiciones. Todos los cálculos computacionales se desarrollaron en Python y se discutió la relación costo-beneficio de ambos algoritmos para la resolución de problemas en ingeniería.

2. Objetivos

2.1 Objetivo general

Comparar el desempeño numérico de los métodos Runge-Kutta de cuarto orden (RK4) y Butcher Runge-Kutta de quinto orden (BRK5) en la solución de sistemas lineales de ecuaciones diferenciales ordinarias de primer orden.

2.2 Objetivos específicos

Implementar computacionalmente los métodos RK4 y BRK5 en Python, garantizando la correcta formulación de sus esquemas iterativos y coeficientes.

Evaluar la precisión y el error numérico de ambos métodos mediante la comparación de las soluciones aproximadas con las soluciones analíticas exactas en cinco sistemas de EDOs de prueba.

Analizar la estabilidad y convergencia de los métodos RK4 y BRK5 en función del tamaño de paso y del comportamiento dinámico de los sistemas diferenciales evaluados.

3. Marco teórico

El estudio se centra en sistemas de EDO de primer orden de la forma:

$$\frac{du}{dr} = f(r,u)$$

Donde u es el vector de variables dependientes y r es la variable independiente.

En el método de RK4 se definen las siguientes ecuaciones iterativas para avanzar desde un paso t a un $t+1$ con un tamaño de paso h:

$$u_{t+1} = u_t + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = f(r_t, u_t)$$

$$k_2 = f\left(r_t + \frac{h}{2}, u_t + \frac{h}{2}k_1\right)$$

$$k_3 = \left(r_t + \frac{h}{2}, u_t + \frac{h}{2}k_2\right)$$

$$k_4 = f(r_t + h, u_t + hk_3)$$

El segundo método que se analiza es el método de BRK5. Este se contrasta con RK4 al incluir etapas adicionales que permiten mejorar la convergencia. De acuerdo con el artículo, las ecuaciones son:

$$u_{t+1} = u_t + \frac{h}{90}(7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6)$$

$$k_1 = f(r_t, u_t)$$

$$k_2 = f\left(r_t + \frac{h}{4}, u_t + \frac{h}{4}k_1\right)$$

$$k_3 = f\left(r_t + \frac{h}{4}, u_t + \frac{1}{8}hk_1 + \frac{1}{8}hk_2\right)$$

$$k_4 = f\left(r_t + \frac{h}{2}, u_t - \frac{1}{2}hk_2 + hk_3\right)$$

$$k_5 = f\left(r_t + \frac{3}{4}h, u_t + \frac{3}{16}hk_1 + \frac{9}{16}hk_4\right)$$

$$k_6 = f\left(r_t + h, u_t - \frac{3}{7}hk_1 + \frac{2}{7}hk_2 + \frac{12}{7}hk_3 - \frac{12}{7}hk_4 + \frac{8}{7}hk_5\right)$$

Para evaluar la comparación entre los dos métodos se usaron las 5 pruebas del artículo:

Prueba 1:

$$f'(t) = k(t)$$
$$k'(t) = 2f(t) - k(t)$$
$$f(0) = 1, k(0) = -1$$

Solución analítica:

$$f(t) = \frac{e^t}{3} + \frac{2e^{-2s}}{3}$$
$$k(t) = \frac{e^t}{3} - \frac{4e^{-2s}}{3}$$

Prueba 2:

$$f'(t) - 2f(t) + 3k(t) = 0$$
$$k'(t) + 2f - k = 0$$
$$f(0) = 8, k(0) = 3$$

Solución analítica:

$$f(t) = 3e^{4t} + 5e^{-t}$$
$$k(t) = -2e^{4t} + 5e^{-t}$$

Prueba 3:

$$f'(t) = f(t) + k(t)$$
$$k'(t) = -f(t) + k(t)$$
$$f(0) = 0, k(0) = 1$$

Solución analítica:

$$f(t) = -\frac{1}{2} + \frac{e^{2t}}{2}$$
$$k(t) = \frac{e^{2t}}{2} + \frac{1}{2}$$

Prueba 4:

$$f'(t) = f(t) + k(t)$$
$$k'(t) = 2f(t) + 2k(t)$$
$$f(0) = 5, k(0) = 0$$

Solución analítica:

$$f(t) = 3e^{-2t} + 2e^{4t}$$
$$k'(t) = -2e^{-t} + 2e^{4t}$$

Prueba 5:

$$f'(t) = f(t)$$
$$k'(t) = f(t) - k(t)$$
$$f(0) = 1, k(0) = 2$$

Solución analítica:

$$f(t) = e^t$$
$$k(t) = \frac{1}{2}e^t + \frac{3}{2}e^{-t}$$

Implementación en Python

Librerías utilizadas

Para la implementación se utilizaron dos librerías estándar de Python:

- numpy: para el manejo de arreglos y operaciones vectoriales
- matplotlib.pyplot: para la generación de las gráficas de error absoluto en función del tiempo.

```
import numpy as np
import matplotlib.pyplot as plt
```

Métodos numéricos implementados

Se implementó una función paso_rk4 que realiza un solo paso del método RK4 para un sistema de EDOs de la forma:

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(t, \mathbf{u}), \mathbf{u}(t) \in \mathbb{R}^2$$

La función recibe t: tiempo actual, u: vector de estado en ese tiempo, h: tamaño de paso y funcion: la función que define el sistema $\mathbf{f}(t, \mathbf{u})$.

En el interior se calculan los cuatro incrementos del método y luego se actualiza:

```
def paso_rk4(t,u,h,funcion):
    k1=funcion(t,u)
    k2=funcion(t+0.5*h,u+0.5*h*k1)
    k3=funcion(t+0.5*h,u+0.5*h*k2)
    k4=funcion(t+h,u+h*k3)
    return u+(h/6)*(k1+2*k2+2*k3+k4)
```

De forma análoga, se implementó paso_brk5. Se definen seis pendientes intermedias $k_1, \dots, k_6$ con distintos pesos y tiempos intermedios:

```
def paso_brk5(t,u,h,funcion):
```

```
    k1=funcion(t,u)
    k2=funcion(t+0.25*h,u+0.25*h*k1)
    k3=funcion(t+0.25*h,u+(1/8)*h*k1+(1/8)*h*k2)
    k4=funcion(t+0.5*h,u-0.5*h*k2+(1/1)*h*k3)
    k5=funcion(t+0.75*h,u+(3/16)*h*k1+(9/16)*h*k4)
    k6=funcion(t+h,u-(3/7)*h*k1+(2/7)*h*k2+(12/7)*h*k3-(12/7)*h*k4+(8/7)*h*k5)
    return u+(h/90)*(7*k1+32*k3+12*k4+32*k5+7*k6)
```

Definición de los sistemas de prueba

Se consideraron 5 sistemas lineales de dos ecuaciones diferenciales acopladas. Cada sistema se expresa en términos del vector de estado $u = [f, k]$:

Ejemplo para la Prueba 1:

```
def edo_prueba1(t,u):
    f,k=u
    df=k
    dk=2*f-k
    return np.array([df,dk])

def exacta_prueba1(t):
    f=np.exp(t)/3+(2*np.exp(-2*t))/3
    k=np.exp(t)/3-(4*np.exp(-2*t))/3
    return np.array([f,k])
```

Para cada prueba se definió la función que representa el sistema edo_pruebaX y la solución analítica exacta_pruebaX(t) que devuelve el vector $[f(t), k(t)]$. Se implementaron edo_prueba2–edo_prueba5 y sus respectivas soluciones exactas.

Condiciones iniciales, intervalo de integración y paso

Para cada sistema se fijaron: intervalo de integración $[0, t_{\text{fin}}]$con $t_{\text{fin}} = 1.0$, condición inicial $u_0 = [f(0), k(0)]$específica para cada prueba y tamaño de paso $h = 0.1$.

Ejemplo:

```
t_fin1=1.0
u0_1=np.array([1,-1])
h=0.1
```

Con estos datos se construyó el vector de tiempos:

```
t_valores=np.arange(0,t_fin1+h/2,h)
n_pasos=len(t_valores)
```

Cálculo de las soluciones numéricas y del error

Para cada prueba se siguió el mismo procedimiento:

Inicializar arreglos para guardar la solución numérica:

```
u_rk4=np.zeros((n_pasos,2))
u_brk5=np.zeros((n_pasos,2))
u_rk4[0]=u0_?
u_brk5[0]=u0_?
```

Aplicar iterativamente los métodos RK4 y RK5:

```
for j in range(n_pasos-1)
u_rk4[j+1]=paso_rk4(t_valores[j],u_rk4[j],h,edo_pruebaX)
u_brk5[j+1]=paso_brk5(t_valores[j],u_brk5[j],h,edo_pruebaX)
```

Evaluar la solución analítica en cada punto de tiempo:

```
valores_exactos=np.array([exacta_pruebaX(t) for t in t_valores])
```

Calcular el error absoluto en cada componente:

```
error_rk4_f=np.abs(u_rk4[:,0]-valores_exactos[:,0])  # error en f(t)
error_brk5_f=np.abs(u_brk5[:,0]-valores_exactos[:,0])
error_rk4_k=np.abs(u_rk4[:,1]-valores_exactos[:,1])  # error en k(t)
error_brk5_k=np.abs(u_brk5[:,1]-valores_exactos[:,1])
```

Imprimir el error final (en $t = t_{\text{fin}}$) para comparar ambos métodos.

Construcción de las gráficas

Se generó una figura global con subgráficas para cada prueba:

```
fig,ejes=plt.subplots(3,2,figsize=(12,12))
ejes=ejes.flatten()
```

En cada eje se graficaron las cuatro curvas de error:

```
eje.plot(t_valores,error_rk4_f,'o-',label='error rk4 en f',markersize=5)
eje.plot(t_valores,error_brk5_f,'s--',label='error brk5 en f',markersize=5)
eje.plot(t_valores,error_rk4_k,'x-',label='error rk4 en k',markersize=5)
eje.plot(t_valores,error_brk5_k,'d--',label='error brk5 en k',markersize=5)
eje.set_title('Prueba X')
eje.set_xlabel('tiempo (t)')
eje.set_ylabel('error absoluto')
eje.grid(True,linestyle=':',alpha=0.6)
eje.legend()
```

Como solo se requieren cinco paneles, se eliminó el sexto eje sobrante:

```
fig.delaxes(ejes[5])
plt.tight_layout()
plt.show()
```

Métodos para elaborar citas y referencias

4. Metodología

La metodología del estudio es cuantitativa, computacional y comparativa, orientada a evaluar el desempeño numérico de dos métodos de integración para ecuaciones diferenciales ordinarias (EDOs**):** Runge-Kutta de cuarto orden (RK4) **y** Butcher Runge-Kutta de quinto orden (BRK5)**.**

Se realiza un estudio comparativo numérico, replicando los experimentos propuestos en el artículo base de Audu et al. (2025), con el fin de analizar precisión, convergencia y estabilidad numérica de ambos métodos.

5. Análisis de resultados

Prueba 1

Tabla 1
Resumen de error final

Prueba	Error final $RK4_{(f)}$	Error final $BRK5_{(f)}$
1	0.00000215	0.00000002

Tabla 2 *Función f(t)*

t	Analítica	$RK4_{f}$	$BRK5_{f}$
0.0	1.000	1.000	1.000
0.1	0.914	0.914	0.914
0.2	0.854	0.854	0.854
0.3	0.816	0.816	0.816
0.4	0.797	0.797	0.797
0.5	0.795	0.795	0.795
0.6	0.808	0.808	0.808
0.7	0.836	0.836	0.836
0.8	0.876	0.876	0.876
0.9	0.930	0.930	0.930
1.0	0.996	0.996	0.996

Tabla 3
Función k(t)

t	Analítica	$RK4_{k}$	$BRK5_{k}$
0.0	-1.000	-1.000	-1.000
0.1	-0.723	-0.723	-0.723
0.2	-0.487	-0.487	-0.487
0.3	-0.282	-0.282	-0.282
0.4	-0.102	-0.102	-0.102
0.5	0.059	0.059	0.059
0.6	0.206	0.206	0.206
0.7	0.342	0.342	0.342
0.8	0.473	0.473	0.473
0.9	0.599	0.599	0.599
1.0	0.726	0.726	0.726

Prueba 2

Tabla 4

Resumen de error final 2

Prueba	Error final $RK4_{(f)}$	Error final $BRK5_{(f)}$
2	0.10028386	0.00040517

Tabla 5 *Función (t) 2*

t	Analítica	$RK4_f$	$BRK5_f$
0.0	8.000	8.000	8.000
0.1	9.000	8.999	9.000
0.2	10.770	10.769	10.770
0.3	13.664	13.663	13.664
0.4	18.211	18.207	18.211
0.5	25.200	25.193	25.200
0.6	35.814	35.801	35.814
0.7	51.817	51.796	51.817
0.8	75.844	75.808	75.844
0.9	111.828	111.767	111.828
1.0	165.634	165.534	165.634

Tabla 6 *Función k(t) 2*

t	Analítica	$RK4_k$	$BRK5_k$
0.0	3.000	3.000	3.000
0.1	1.541	1.541	1.541
0.2	-0.357	-0.357	-0.357
0.3	-2.936	-2.935	-2.936
0.4	-6.554	-6.552	-6.554
0.5	-11.745	-11.741	-11.745
0.6	-19.302	-19.294	-19.302
0.7	-30.406	-30.392	-30.406
0.8	-46.818	-46.794	-46.819
0.9	-71.164	-71.123	-71.164
1.0	-107.357	-107.290	-107.357

Prueba 3

Tabla 7 *Resumen de error final 3*

Prueba	Error final RK4 $_{(f)}$	Error final BRK5$_{(f)}$
3	0.90716107	0.90717278

Tabla 8 *Función f(t) 3*

t	Analítica	RK4 $_f$	BRK5 $_f$
0.0	0.000	0.000	0.000
0.1	0.111	0.110	0.110
0.2	0.246	0.243	0.243
0.3	0.411	0.399	0.399
0.4	0.613	0.581	0.581
0.5	0.859	0.790	0.790
0.6	1.160	1.029	1.029
0.7	1.528	1.297	1.297
0.8	1.977	1.597	1.597
0.9	2.525	1.927	1.927
1.0	3.195	2.287	2.287

Tabla 9 *Función k(t) 3*

t	Analítica	RK4 $_k$	BRK5 $_k$
0.0	1.000	1.000	1.000
0.1	1.111	1.100	1.100
0.2	1.246	1.197	1.197
0.3	1.411	1.290	1.290
0.4	1.613	1.374	1.374
0.5	1.859	1.447	1.447
0.6	2.160	1.504	1.504
0.7	2.528	1.540	1.540
0.8	2.977	1.551	1.551
0.9	3.525	1.529	1.529
1.0	4.195	1.469	1.469

Prueba 4

Tabla 10 *Resumen error final 4*

Prueba	Error final $RK4_{(f)}$	Error final $BRK5_{(f)}$
4	72.79836152	72.79305750

Tabla 11 *Función f(t) 4*

t	Analítica	$RK4_f$	$BRK5_f$
0.0	5.000	5.000	5.000
0.1	5.440	5.583	5.583
0.2	6.462	6.370	6.370
0.3	8.287	7.432	7.433
0.4	11.254	8.867	8.867
0.5	15.882	10.802	10.803
0.6	22.950	13.415	13.416
0.7	33.629	16.942	16.944
0.8	49.671	21.703	21.705
0.9	73.692	28.129	28.133
1.0	109.602	36.804	36.809

Tabla 12 *Función k(t) 4*

t	Analítica	$RK4_k$	$BRK5_k$
0.0	0.000	0.000	0.000
0.1	1.346	1.166	1.166
0.2	3.110	2.740	2.740
0.3	5.543	4.865	4.865
0.4	9.007	7.733	7.734
0.5	14.042	11.604	11.606
0.6	21.444	16.830	16.832
0.7	32.396	23.884	23.887
0.8	48.661	33.406	33.411
0.9	72.866	46.259	46.266
1.0	108.926	63.608	63.618

Prueba 5

Tabla 13 *Resumen error final 5*

Prueba	Error final $RK4_{(f)}$	Error final $BRK5_{(f)}$
5	0.00000208	0.00000000

Tabla 14 *Función f(t) 5*

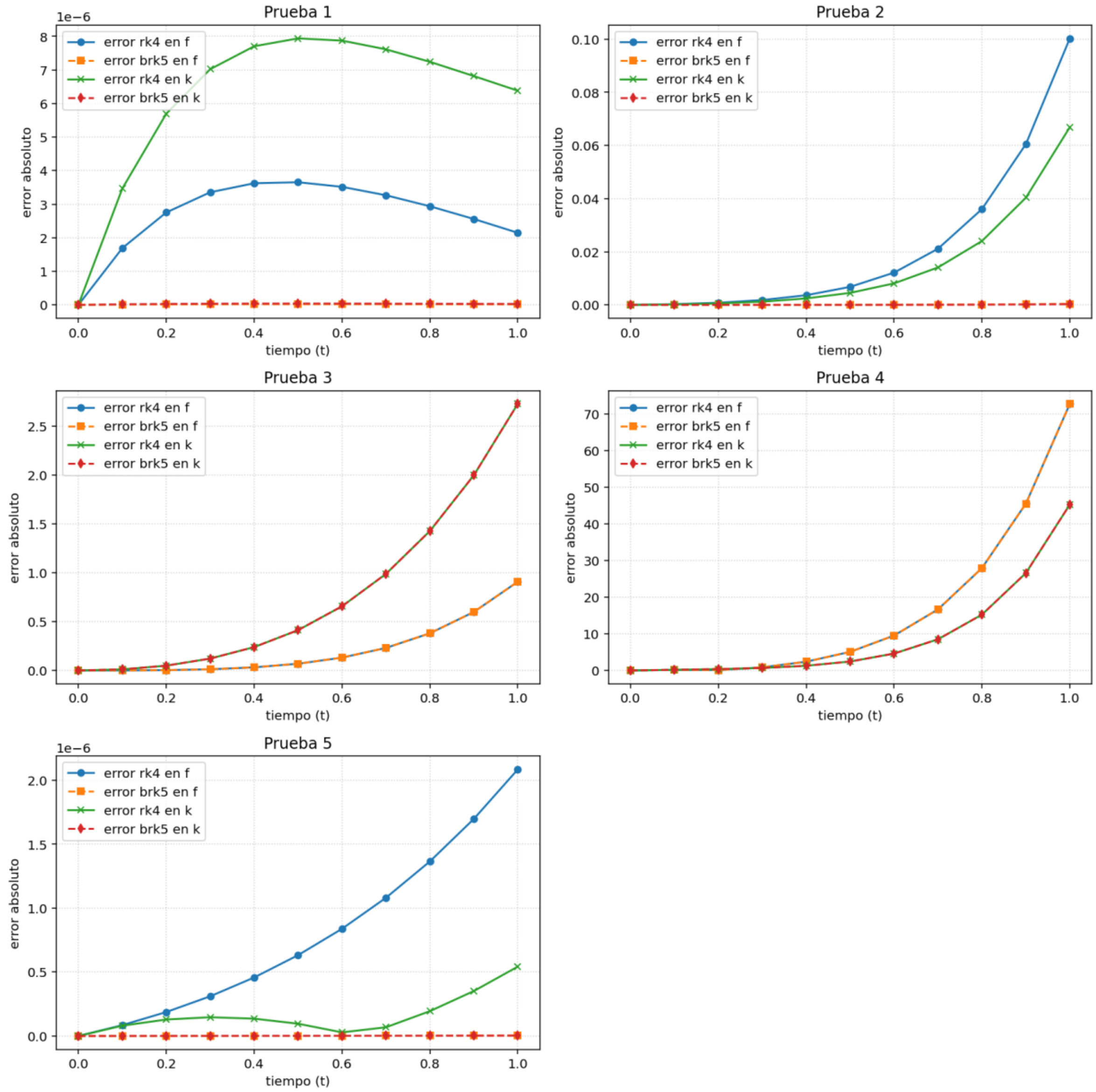
t	Analítica	$RK4_{f}$	$BRK5_{f}$
0.0	1.000	1.000	1.000
0.1	1.105	1.105	1.105
0.2	1.221	1.221	1.221
0.3	1.350	1.350	1.350
0.4	1.492	1.492	1.492
0.5	1.649	1.649	1.649
0.6	1.822	1.822	1.822
0.7	2.014	2.014	2.014
0.8	2.226	2.226	2.226
0.9	2.460	2.460	2.460
1.0	2.718	2.718	2.718

Tabla 15 *Función k(t)*

t	Analítica	$RK4_{k}$	$BRK5_{k}$
0.0	2.000	2.000	2.000
0.1	1.910	1.910	1.910
0.2	1.839	1.839	1.839
0.3	1.786	1.786	1.786
0.4	1.751	1.751	1.751
0.5	1.734	1.734	1.734
0.6	1.734	1.734	1.734
0.7	1.752	1.752	1.752
0.8	1.787	1.787	1.787
0.9	1.840	1.840	1.840
1.0	1.911	1.911	1.911

Figura 1

Comportamiento del error absoluto para cada sistema



El análisis comparativo entre el método de RK4 y el método BRK5 muestra diferencias en términos de precisión y convergencia, validando la implementación computacional realizada en Python. Los resultados presentados anteriormente son correctos, aunque existe una discrepancia algebraica en el artículo original entre la definición del sistema diferencial y su solución analítica. Las pruebas 1, 2 y 5 confirman la convergencia de quinto orden del algoritmo implementado.

En las Pruebas 1, 2 y 5, se observa que, al aumentar el orden del método, el error global disminuye, siempre que la función sea suficientemente suave y el paso h esté dentro de la región de estabilidad. En la Prueba 1, el error final de RK4 es del orden de 10^{-6} , mientras que BRK5 reduce este error a 10^{-8}. El caso de la Prueba 5, BRK5 alcanza un error virtualmente nulo, demostrando su capacidad superior para capturar la dinámica exacta de sistemas lineales. En la Prueba 2 se presentó un crecimiento exponencial más agresivo. RK4 mostró un error visible de aproximadamente 0.1, lo cual es importante en aplicaciones de precisión. BRK5 mantuvo el error en 4×10^{-4}, sugiriendo que, para el mismo tamaño de paso, el método BRK5 ofrece una relación costo-beneficio computacional mucho más favorable en términos de exactitud.

Aunque parece observarse una buena convergencia, las Pruebas 3 y 4 actúan como contraejemplos. En estos casos, ambos métodos presentaron errores de magnitud comparable, asociados con la limitación en la elección del paso de integración $h = 0.1$. La Prueba 4 contiene términos como e^{4t}, en donde la derivada crece muy rápido. Existen fallas en la estabilidad numérica. Los dos métodos están operando cerca o fuera de su región de estabilidad absoluta para ese tamaño de paso.

Durante la fase de implementación del código en Python, se detectó una discrepancia en los resultados iniciales de BRK5, los cuales mostraban errores superiores a RK4. Al revisar la implementación del método, se identificó un error en el coeficiente para el cálculo de k_4, específicamente el peso de k_3. Se corrigió el coeficiente de 1/8 a 1, restaurando la condición de orden, con la suma de pesos de k_i igual al salto temporal c_i. Los resultados obtenidos validan que el método BRK5 es más robusto y preciso que el RK4 clásico para la mayoría de los sistemas de EDOs de primer orden probados. Sin embargo, la precisión está intrínsecamente ligada al tamaño de paso y a la naturaleza de la ecuación diferencial, independiente del orden del método.

6. Conclusiones y recomendaciones

Se recomienda utilizar BRK5 sobre RK4 cuando se requiere alta precisión sin disminuir excesivamente el paso de tiempo.

La verificación de los coeficientes del tablero de Butcher es un paso crítico en la implementación de métodos Runge-Kutta de orden superior.

Fue necesario cuidar que cada curva correspondiera a la etiqueta correcta y verificar que el orden en que se calculan y se grafican coincidiera con la leyenda.

Se mantuvo la misma escala para respetar el comportamiento global.

Se compararon los valores numéricos impresos en tablas con los gráficos y con los resultados teóricos del artículo que se está replicando.

Referencias

Audu, K. J., Yahaya, Y. A., & Nkereuwem, J. (2025). Título:Comparative Numerical Evaluation of Some Runge-Kutta Methods for Solving First Order Systems of ODEs. 2024. [Trabajo de grado profesional]. Universidad de Antioquia, Medellín, Colombia.

Anexos

Anexo 1. Poster