



Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingeniería Aeronáutica y del Espacio

Máster Universitario en Sistemas Espaciales

Milestone 3: Error estimation of numerical solutions

Ampliación de matemáticas 1

23 de octubre de 2022

Autor:

■ Tomás Ley Oliver

Índice

1. Introducción	3
2. Error numérico. Extrapolación de Richardson	3
2.1. Discretización	3
2.2. Resultados	3
Euler	3
Runge-Kutta de orden 4	5
Crank-Nicholson	7
Euler Inverso	9
3. Convergencia	10
3.1. Discretización	10
3.2. Resultados	10
Euler	10
Runge-Kutta 4	11
Crank-Nicholson	12
Euler Inverso	12
4. Módulos	13
4.1. Main	13
4.2. Error_Schemes	13
Extrapolación de Richardson	13
Función de convergencia	14
Función combinada de error y convergencia	14
Funciones de representación	14

1. Introducción

El objetivo de esta práctica es observar los errores de los distintos esquemas de integración numérica que se han estudiado anteriormente. Para ello se utilizará la extrapolación de Richardson para calcular los errores cometidos, y se observará la convergencia de los esquemas.

2. Error numérico. Extrapolación de Richardson

El error numérico es la diferencia entre resultado obtenido con la solución analítica del problema cometido por utilizar esquemas numéricos.

La extrapolación de Richardson aproxima este error comparando la diferencia entre los resultados del problema calculados con un número N de puntos, y con un número $2*N$ de puntos.

2.1. Discretización

Para este apartado se ha tomado un tiempo total de integración adimensional dependiente del esquema para observar las dependencias con el tiempo total de integración, excepto para el esquema de Euler Inverso, en el que se ha limitado a 15 debido a problemas de convergencia causados por el diferencial de tiempo utilizado. Se han estudiado los esquemas con 3 diferenciales de tiempo distintos: $\Delta t = \{0.1, 0.01, 0.001\}$.

2.2. Resultados

Euler

Este esquema se ha discretizado mediante un tiempo total de integración de 100.

De los resultados obtenidos se observa en las figuras 2.1, 2.2 y 2.3 que el error en los esquemas de Euler son del orden unidad, es decir, de las magnitudes del problema. Esto se debe a la imprecisión del esquema de Euler para tiempos grandes.

Se observa como el error total sufre oscilaciones de frecuencia dependiente con el diferencial de tiempo.



Figura 2.1: Error en esquema de Euler para $\Delta t = 0.1$

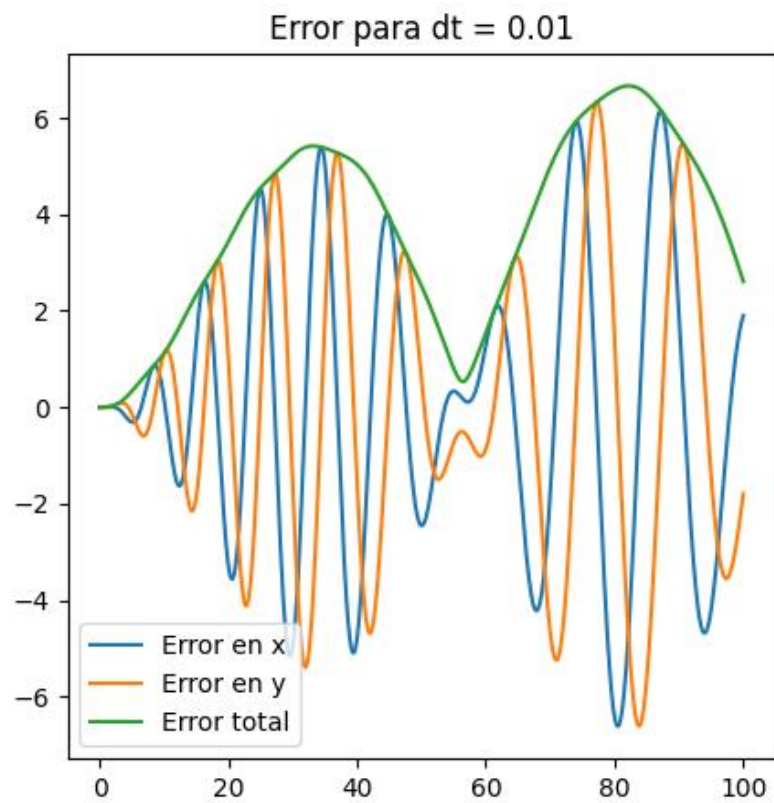


Figura 2.2: Error en esquema de Euler para $\Delta t = 0.01$

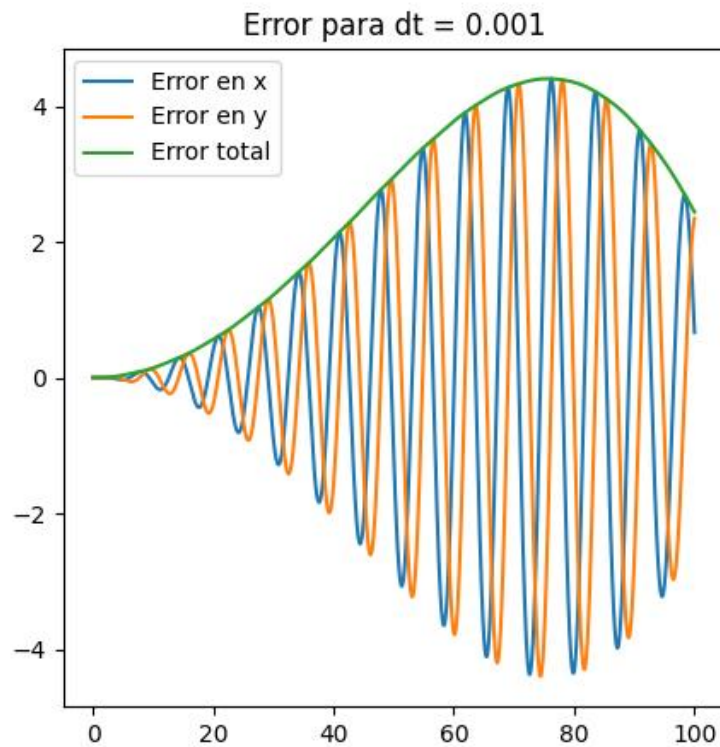


Figura 2.3: Error en esquema de Euler para $\Delta t = 0.001$

Runge-Kutta de orden 4

Se puede observar en las figuras 2.4, 2.5 y 2.6 como el error total del esquema Runge-Kutta de orden 4 crece linealmente con el tiempo, mientras que la magnitud de este error es proporcional al diferencial de tiempo utilizado. Por este motivo, se ha discretizado este esquema con un tiempo total de integración 20.

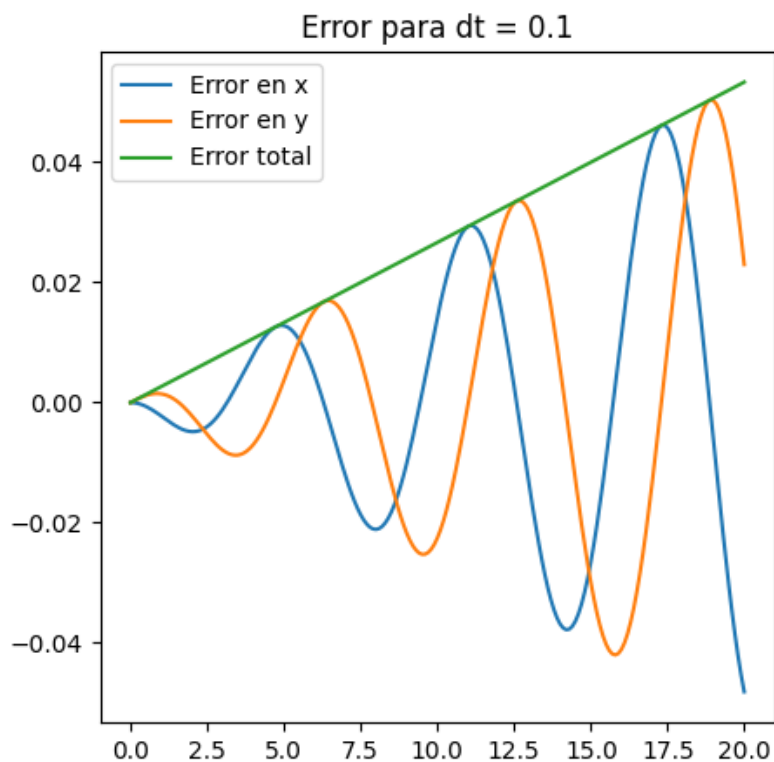


Figura 2.4: Error en esquema de RK4 para $\Delta t = 0.1$

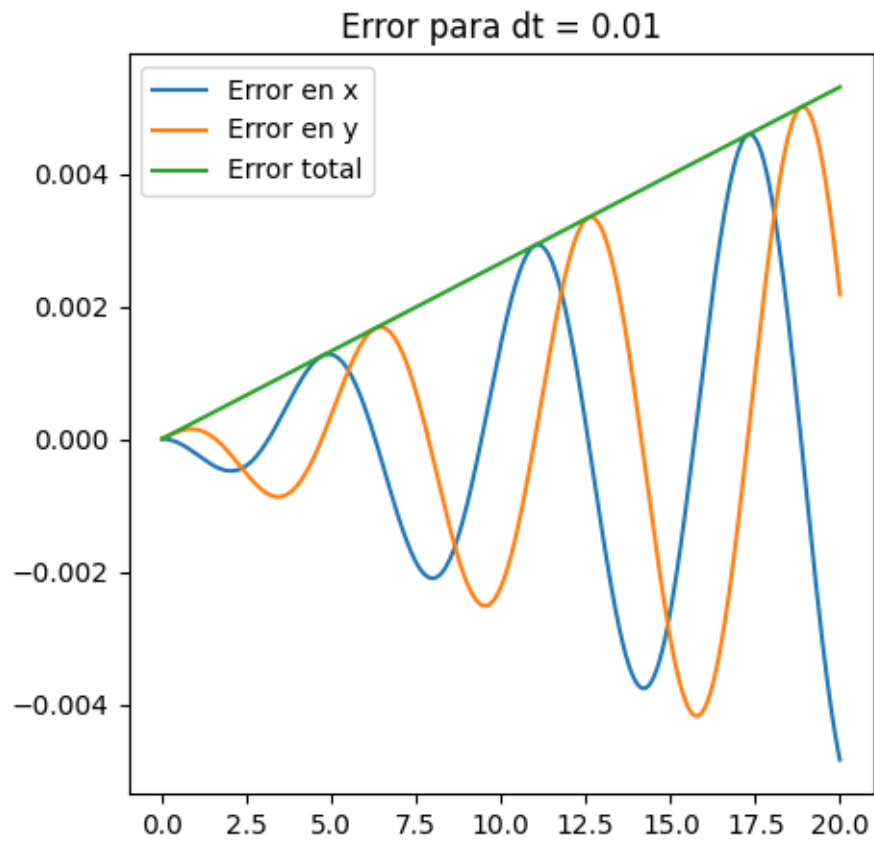


Figura 2.5: Error en esquema de RK4 para $\Delta t = 0.01$



Figura 2.6: Error en esquema de RK4 para $\Delta t = 0.001$

Crank-Nicholson

De forma similar al Runge-Kutta 4, se puede observar en las figuras 2.7, 2.8 y 2.9 como el error total del esquema Crank-Nicholson crece linealmente con el tiempo, mientras que la magnitud de este error es proporcional al diferencial de tiempo utilizado.

Al igual que en el Runge-Kutta 4, se ha discretizado este esquema con un tiempo total de integración 20, ya que el error escala linealmente con el tiempo



Figura 2.7: Error en esquema de Crank-Nicholson para $\Delta t = 0.1$

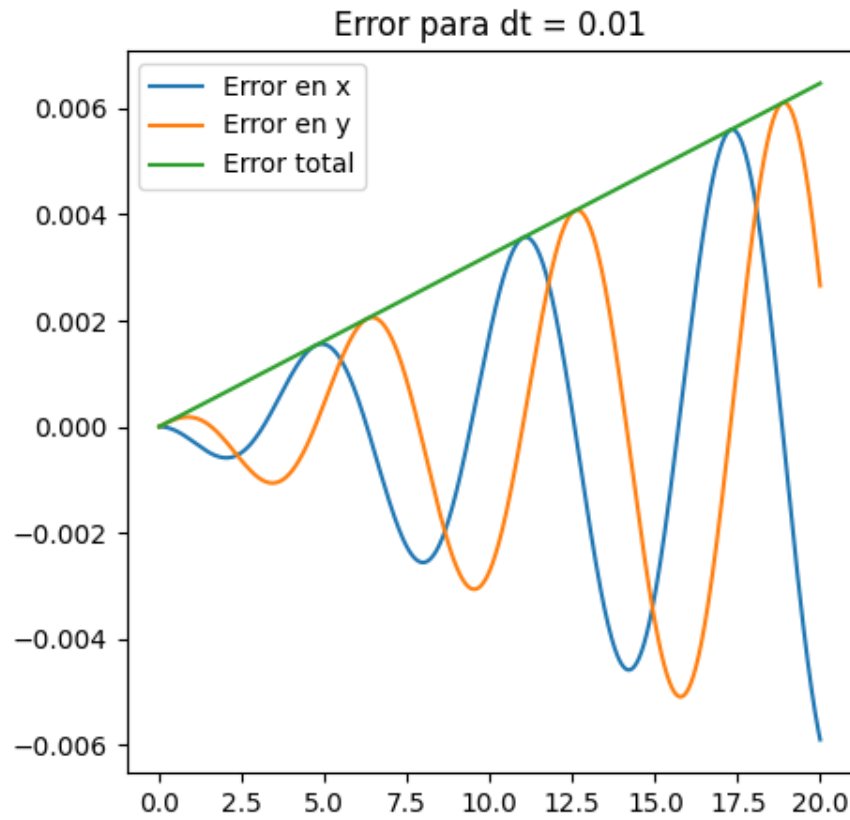


Figura 2.8: Error en esquema de Crank-Nicholson para $\Delta t = 0.01$

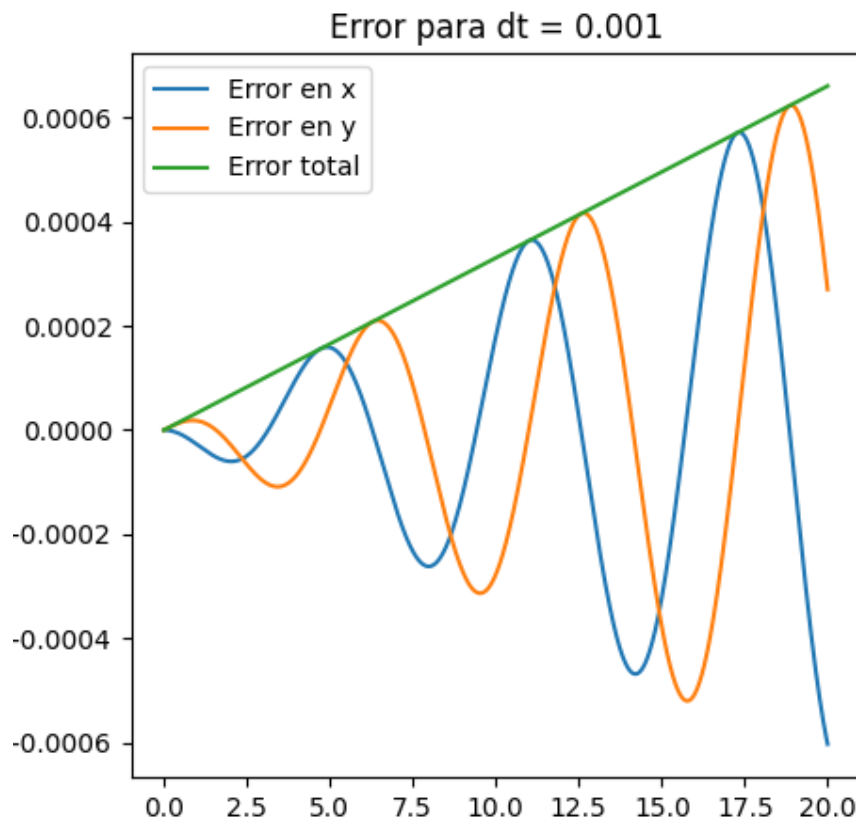


Figura 2.9: Error en esquema de Crank-Nicholson para $\Delta t = 0.001$

Euler Inverso

Debido a problemas de convergencia, solo se ha calculado el error del esquema de Euler Inverso hasta un tiempo total de 15. Se observa en la figura 2.10 como, de manera similar al esquema de Euler, el error es de orden unidad. En la figura 2.11 el error ha disminuido un orden de magnitud, debido al afinamiento del diferencial de tiempo utilizado. También se observa en la figura 2.10 una oscilación del error total.

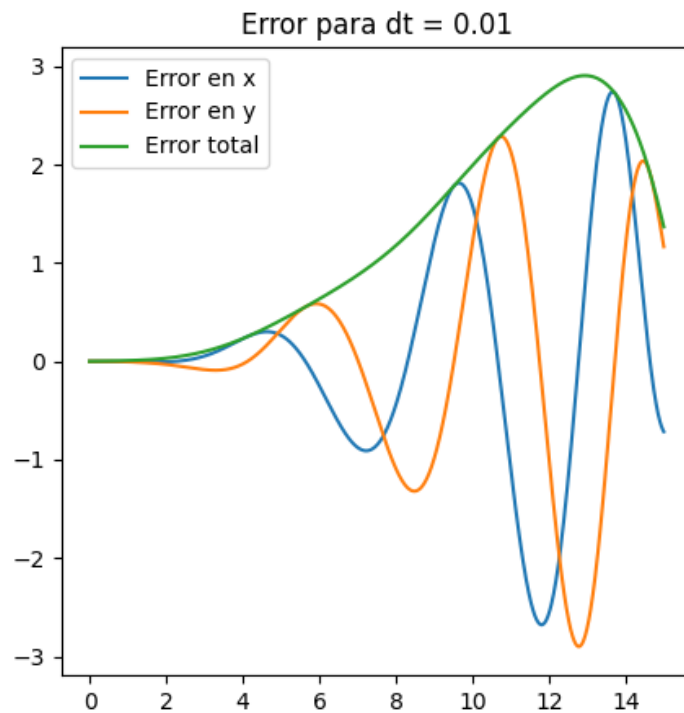


Figura 2.10: Error en esquema de Euler Inverso para $\Delta t = 0.01$



Figura 2.11: Error en esquema de Euler Inverso para $\Delta t = 0.001$

3. Convergencia

La convergencia de los esquemas temporales se estudia comparando el orden del error cometido con el número de puntos estudiado. La representación de esta relación en una escala logarítmica resulta en una curva con un tramo lineal, la pendiente del cual es equivalente al orden de convergencia del esquema temporal. Este pendiente representa la velocidad de convergencia del esquema, y ayuda a conocer el número de intervalos N a utilizar para alcanzar una determinada precisión

3.1. Discretización

Para este apartado se ha tomado un tiempo total de integración adimensional de 10, y se ha comenzado el cálculo con $N = 100$. Para el esquema de Euler Inverso se comienza desde $N = 1000$ debido a problemas de convergencia para $\Delta t \approx 0.1$

3.2. Resultados

Euler

Del esquema de Euler cabe destacar en la figura 3.1 la aparición del primer tramo de la curva de convergencia, el cual se origina para $N = 1$, dando un error de orden unidad. Este tramo ha influido en la linealización de los resultados, y se observa como la pendiente obtenida es de 0.88, aproximadamente 1, el esperado del esquema de Euler.

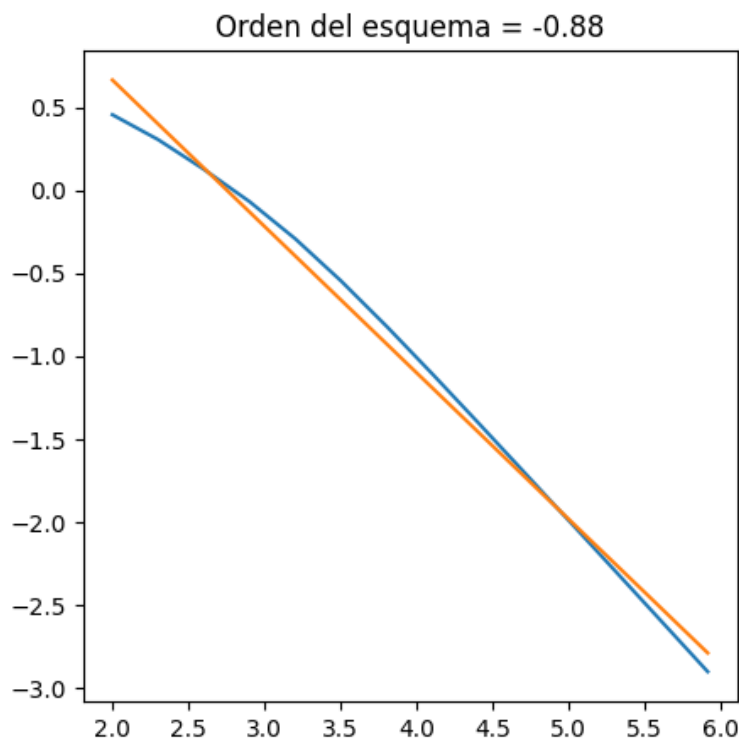


Figura 3.1: Convergencia del esquema de Euler

Runge-Kutta 4

En los resultados obtenidos del esquema RK4, destaca en la figura 3.2 la aparición del tercer tramo de la curva de convergencia, donde el error obtenido es del orden del error de cálculo del ordenador, aproximadamente 10^{-12} . Al igual que en el Euler, este tramo afecta a la linealización de los resultados.

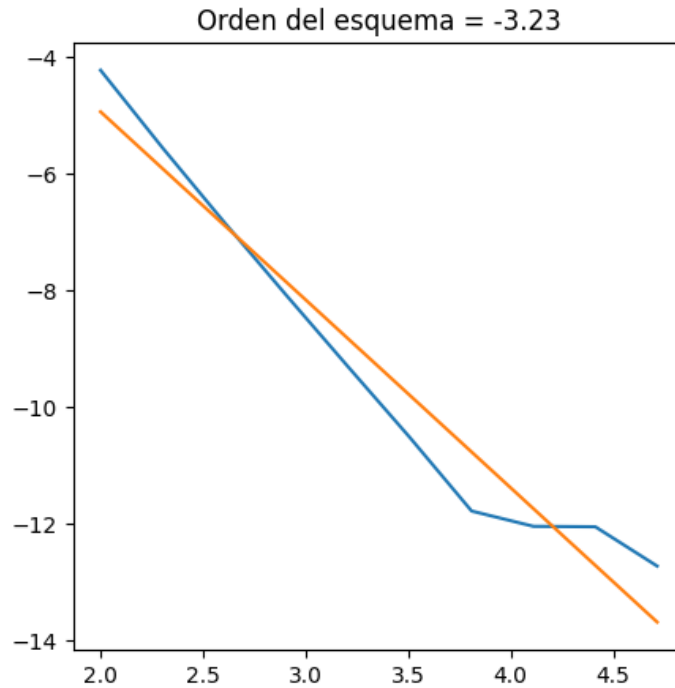


Figura 3.2: Convergencia del esquema Runge-Kutta 4 hasta $N = 10^5$

Se ha realizado un segundo cálculo hasta un N menor en el que no se alcanza el tercer tramo, representado en la figura 3.3, y se ha obtenido una pendiente de 4.16, aproximadamente 4, la esperada para el esquema Runge-Kutta de orden 4.

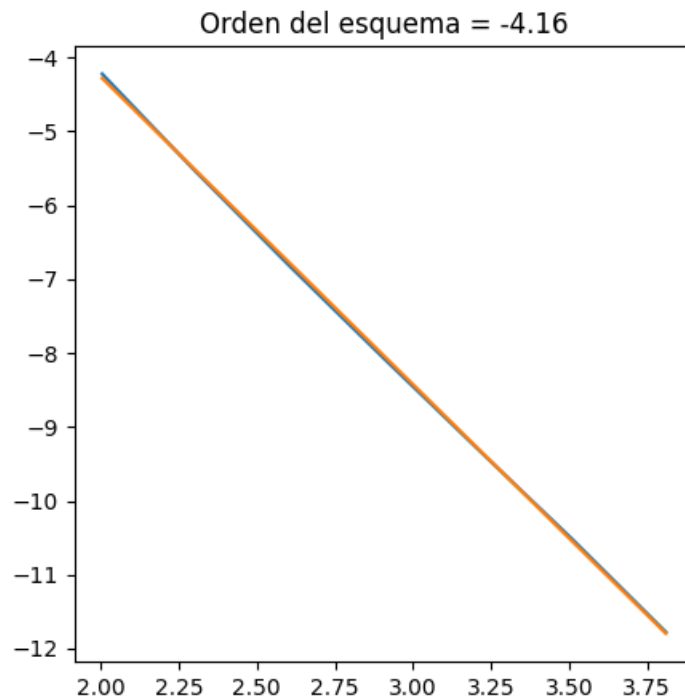


Figura 3.3: Convergencia del esquema de Runge-Kutta 4 hasta $N = 10^4$

Crank-Nicholson

La pendiente obtenida del tramo recto de la curva de convergencia es de 1.97, aproximadamente 2, el esperado del esquema de Crank-Nicholson

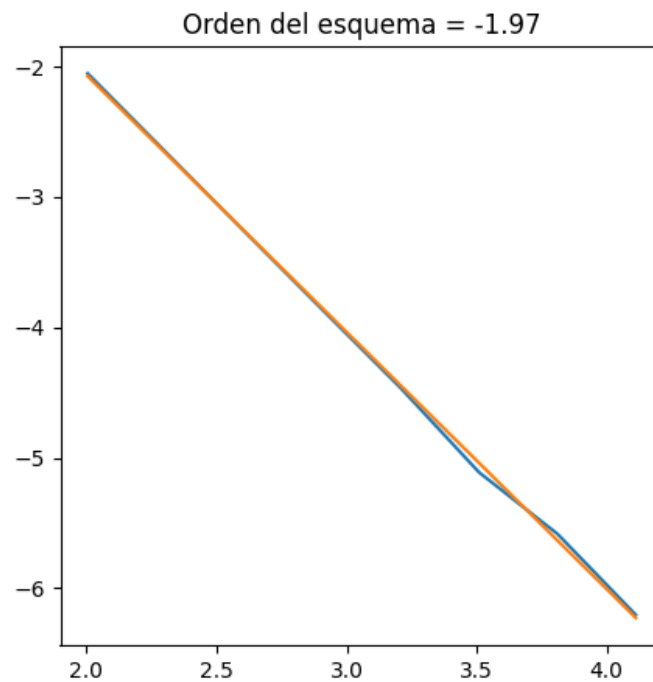


Figura 3.4: Convergencia del esquema de Crank-Nicholson

Euler Inverso

La pendiente obtenida del tramo recto de la curva de convergencia es de 1.08, aproximadamente 1, el esperado del esquema de Euler Inverso.

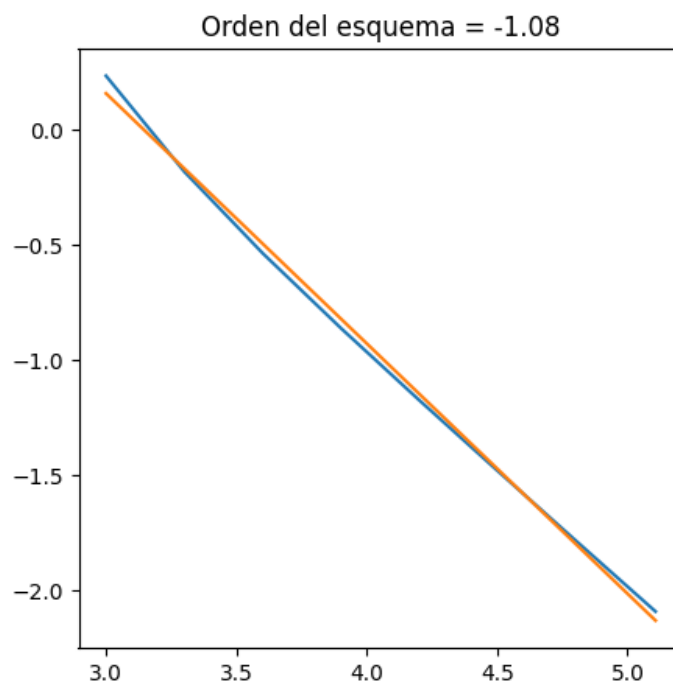


Figura 3.5: Convergencia del esquema de Euler Inverso

4. Módulos

Para resolver esta práctica se han utilizado un nuevo módulo, *Error_Schemes*, además de los utilizados en los anteriores hitos.

4.1. Main

Este archivo incluye las condiciones iniciales del problema, la discretización temporal y la definición del problema. Esto ya se ha visto en anteriores hitos, por lo que no se entrará en detalles

4.2. Error_Schemes

Este módulo define los métodos para: analizar el error según Richardson, y la convergencia de los esquemas temporales, además de unas funciones de representación de resultados. Las entradas que utilizan son:

- *Problem*: El problema que se va a estudiar, en este caso el problema de Cauchy
- *Phy*: La función que define el problema físico, en este caso la órbita de Kepler
- *U0*: las condiciones iniciales del problema,
- *Esq_temp*: El esquema temporal que se va a analizar.
- *T*: Los valores temporales en los que se estudiará el error
- *Q_order*: Orden de convergencia de los esquemas. Es necesario para analizar la función de Richardson.
- *Mpoints*: Número de puntos para el gráfico de Richardson

Extrapolación de Richardson

Esta función resuelve el problema dos veces, con un número N de puntos, y con un número $2*N$ de puntos, y compara la diferencia entre ambas para observar el error cometido. Esto se hace para distintos Δt como se puede observar en el apartado 2.

```
t1 = linspace(t[0], t[-1], N)
t2 = linspace(t[0], t[-1], 2*N)

Un = problema(Phy, t1, U0, Esq_temp)
U2n = problema(Phy, t2, U0, Esq_temp)

for j in range(N):

    Err[:,j] = (Un[:,j]-U2n[:,2*j])/(1-1/(2**q_order))
```

Para poder analizar el error de la extrapolación de Richardson es necesario conocer el orden de convergencia del esquema utilizado, por lo tanto, se introduce este orden como una entrada.

Función de convergencia

Esta función calcula el error cometido de forma similar a la extrapolación de Richardson, y lo compara con el número de puntos N utilizado.

```
Un = problema(Phy, t1, U0, Esq_temp)
U2n = problema(Phy, t2, U0, Esq_temp)

logErr[i] = log10(norm(Un[:,N-1]-U2n[:,2*N-1]))
logN[i] = log10(N)

q = linregress(logN, logErr)
```

El número de puntos que se quiera representar se introduce con la variable *mpoints*. El tiempo de cálculo de la función aumenta considerablemente con *mpoints*, teniendo tiempos de cálculo de 1 minutos para *mpoints* = 8, y de hasta 10 minutos para *mpoints* = 10.

Función combinada de error y convergencia

El propósito de esta función es disminuir la carga computacional en el caso de que se quiera calcular la extrapolación de Richardson y la convergencia consecutivamente, ya que si se utilizasen ambas funciones por separado se calcularían dos veces las mismas soluciones del problema.

```
Un = problema(Phy, t1, U0, Esq_temp)
U2n = problema(Phy, t2, U0, Esq_temp)

for j in range(N):

    Err[:,j] = (Un[:,j]-U2n[:,2*j])/(1-1/(2**q_order))

logErr[i] = log10(norm(Un[:,N-1]-U2n[:,2*N-1]))
logN[i] = log10(N)
```

Funciones de representación

Se han elaborado estas funciones adicionales para representar los resultados obtenidos. Esto sirve para simplificar el código de las funciones y facilitar su depuración.