



Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingeniería Aeronáutica y del Espacio

Máster Universitario en Sistemas Espaciales

Milestone 4: N-body problem

Ampliación de matemáticas 1

1 de diciembre de 2022

Autor:

Tomás Ley Oliver

Índice

1. Introducción	3
2. Oscilador armónico	3
2.1. Discretización	3
2.2. Resultados	3
Euler	3
Runge-Kutta de orden 4	4
Crank-Nicholson	5
Euler Inverso	5
Leap-Frog	6
3. Regiones de estabilidad absoluta	7
3.1. Introducción	7
3.2. Resultados	7
<i>Leap-Frog</i>	7
Euler	7
Euler Inverso	8
Runge-Kutta 4	9
Crank-Nicholson	9
4. Actualización de módulos	10
4.1. Main	10
4.2. Esquemas_temporales	10
4.3. Cauchy_problem	10
4.4. Physics	10
4.5. Estabilidad	11

1. Introducción

El objetivo de este hito es estudiar y resolver el problema del oscilador armónico mediante los diferentes esquemas numéricos desarrollados en anteriores hitos, y además añadiendo el esquema *Leap-Frog*.

Después se observarán las regiones de estabilidad absoluta de estos esquemas, analizando su significado con respecto a la convergencia de los esquemas.

Finalmente se comentarán las actualizaciones de los módulos del código *Python*.

2. Oscilador armónico

Un oscilador armónico se define como aquel sistema que cuando se deja en libertad fuera de su posición de equilibrio, vuelve hacia ella describiendo oscilaciones sinusoidales, o sinusoidales amortiguadas, en torno a dicha posición estable.

En el caso no amortiguado, el oscilador armónico se define como un movimiento armónico simple, que sigue la siguiente ecuación:

$$\ddot{x} + x = 0 \quad \text{con} \quad x(0) = 1, \quad \dot{x}(0) = 0$$

Este problema tiene una solución analítica que se utilizará para comparar la precisión de la integración numérica

$$x(t) = \cos(t)$$

2.1. Discretización

Para la discretización del oscilador armónico se ha utilizado un tiempo total de integración adimensional, $T = 20$, y un número de pasos de integración, $N = 2000$, con el objetivo de obtener un $\Delta t \ll 1$ y un tiempo de integración lo suficientemente largo como para poder observar las diferencias entre los distintos esquemas.

2.2. Resultados

Euler

Los resultados obtenidos mediante el esquema numérico de Euler se muestran en la figura 2.1.

Se observa como existe un error en amplitud y en fase de la oscilación que amplifica estas magnitudes con tiempo de simulación. Esto se asemeja a lo que se observó en el análisis de la órbita kepleriana, en la que se alejaba del centro de la órbita.

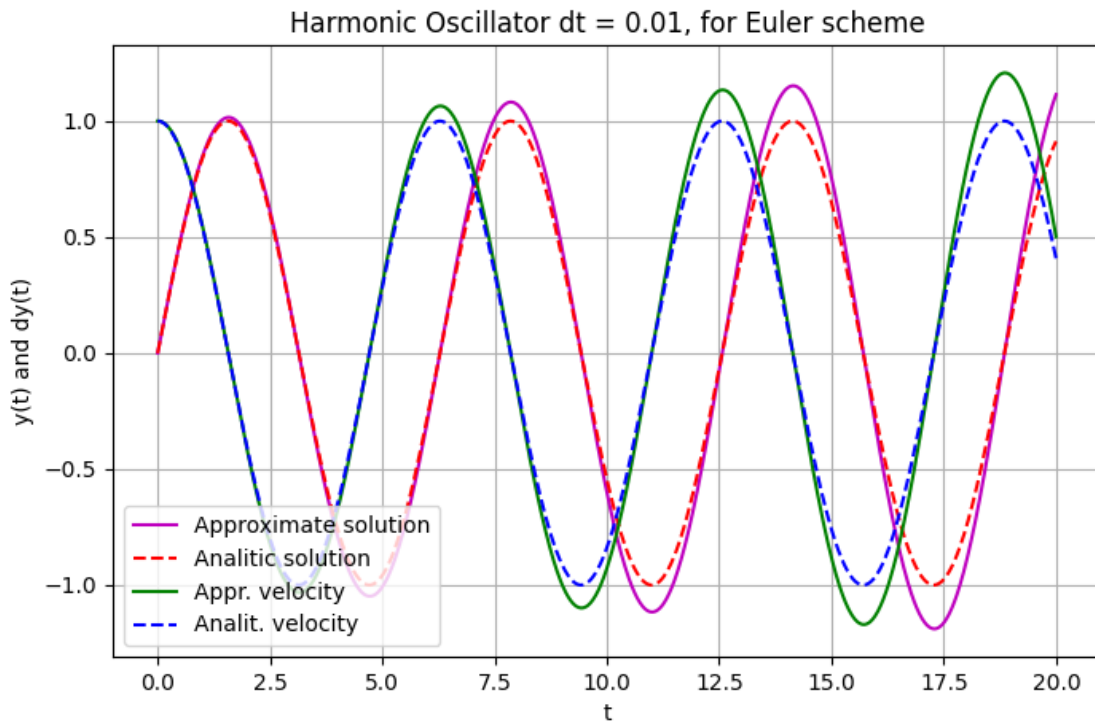


Figura 2.1: Integración del oscilador armónico mediante esquema de Euler

Runge-Kutta de orden 4

Los resultados obtenidos mediante el esquema Runge-Kutta de orden 4 se muestran en la figura 2.2.

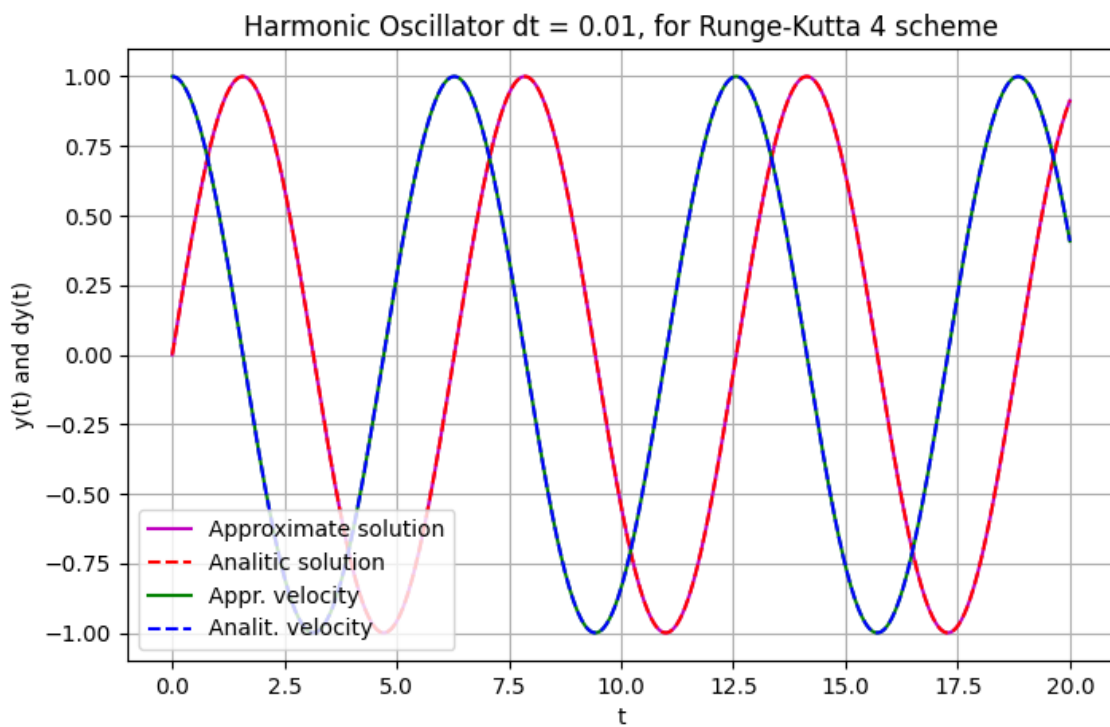


Figura 2.2: Integración del oscilador armónico mediante esquema RK4

Se observa como este esquema proporciona una buena aproximación a la solución analítica.

Crank-Nicholson

Los resultados obtenidos mediante el esquema Crank-Nicholson se muestran en la figura 2.3.

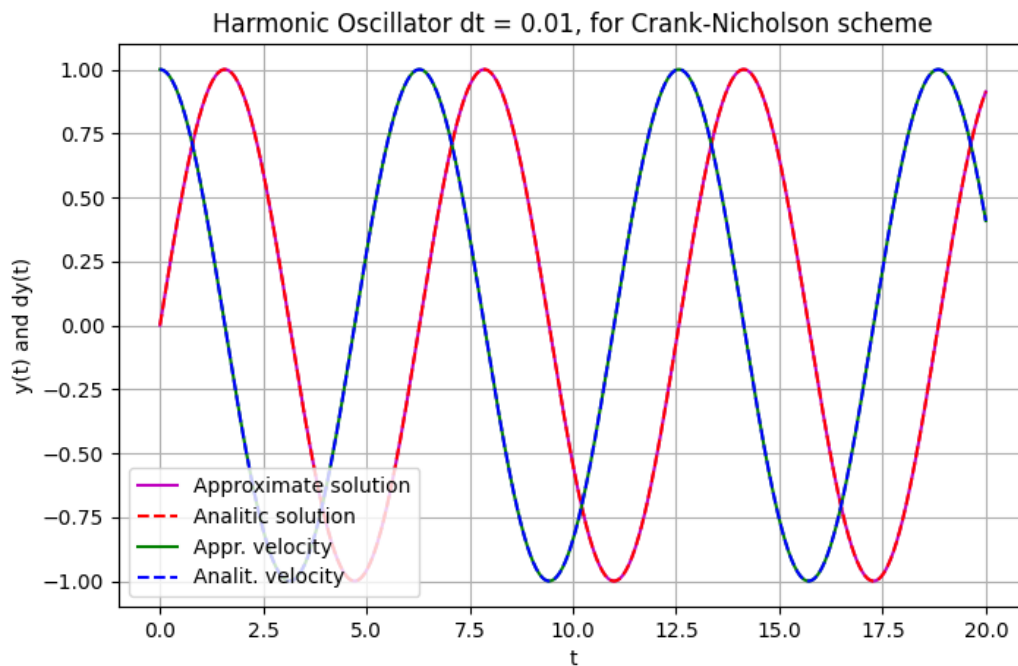


Figura 2.3: Integración del oscilador armónico mediante esquema Crank-Nicholson

De forma similar al Runge-Kutta 4, se puede observar como con este esquema se obtiene un resultado muy similar a la solución analítica

Euler Inverso

Los resultados obtenidos mediante el esquema numérico de Euler Inverso se muestran en la figura 2.4.

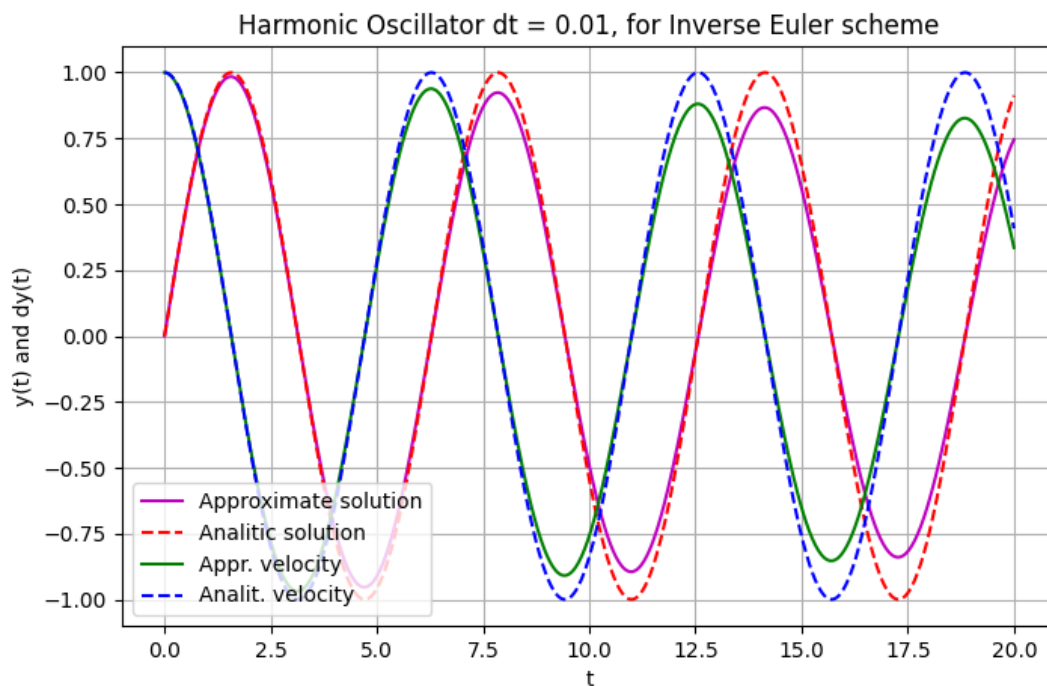


Figura 2.4: Integración del oscilador armónico mediante esquema de Euler Inverso

Se observa cómo, de manera similar al esquema de Euler, existe un error en amplitud y en fase de la oscilación que disminuye estas magnitudes con el tiempo de simulación.

Leap-Frog

El esquema *Leap-Frog*, o regla del punto medio se utiliza para resolver ecuaciones diferenciales de la forma:

$$\ddot{x} = F(x)$$

Este esquema se aplica para obtener la solución en un tiempo t_{n+1} conocida la solución en un tiempo t_{n-1} , según la fórmula:

$$U^{n+1} = U^{n-1} + 2\Delta t F^n$$

Esto da problemas en el punto inicial de integración, ya que se necesitaría saber la situación previa a las condiciones iniciales, por lo que el paso inicial U^1 se calcula mediante otro de los esquemas, en este caso el esquema de Euler.

Los resultados obtenidos mediante el esquema de *Leap-Frog* se muestran en la figura 2.5.

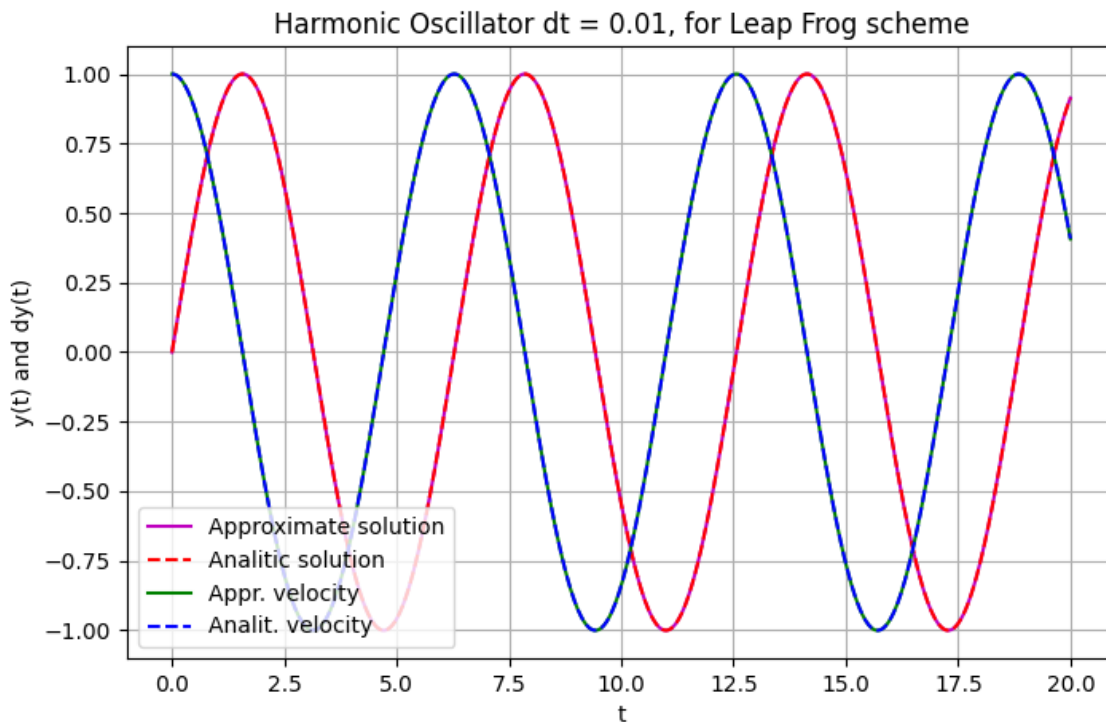


Figura 2.5: Integración del oscilador armónico mediante esquema *Leap-Frog*

3. Regiones de estabilidad absoluta

3.1. Introducción

Las regiones de estabilidad absoluta se definen como la región en el plano complejo donde las soluciones numéricas son asintóticamente estables. Esta región se calcula determinando el contorno de la región, es decir, el lugar geométrico en el que las raíces del polinomio característico del esquema, $\Pi(r, \omega)$, tienen módulo igual a la unidad. Este módulo representa la ganancia del esquema, por lo que, si es mayor a la unidad, la solución numérica aumentará con respecto a la real, y de la misma manera, disminuirá tendiendo a cero si el módulo es menor a la unidad.

Con este contorno, el plano queda dividido en dos regiones, una región en la que el módulo de las raíces es mayor a la unidad, y otra en la que es menor a la unidad.

Los polinomios característicos de los esquemas estudiados son:

Esquema temporal	$\Pi(r, \omega)$
Euler	$r - 1 - \omega$
Runge-Kutta 4	$r - 1 - \omega - \frac{\omega^2}{2} - \frac{\omega^3}{6} - \frac{\omega^4}{24}$
Crank-Nicholson	$r - \frac{1 + \frac{\omega}{2}}{1 - \frac{\omega}{2}}$
Euler Inverso	$r - \frac{1}{1 - \omega}$
<i>Leap-Frog</i>	$r^2 - 1$

Las raíces del sistema del oscilador armónico quedan definidas como $\pm i\Delta t$, por lo que si se encuentran en el contorno de la región la solución será estable y similar a la analítica.

3.2. Resultados

Leap-Frog

La región de estabilidad del esquema *Leap-Frog* es un segmento del eje imaginario que une los puntos $(0, -i)$ y $(0, i)$. De esta manera, las soluciones $\Delta t \leq 1$ estarán en el contorno de la región. Para $\Delta t > 1$ el esquema será inestable y la solución numérica divergirá de la analítica.

Euler

La región de estabilidad del esquema de Euler se corresponde al interior de una circunferencia de radio unidad centrada en $(-1, 0)$. Las raíces del sistema nunca se encontrarán dentro de la región de estabilidad. Es por esto por lo que los resultados obtenidos, en este hito y en anteriores, mediante el esquema de Euler crecen alejándose de la solución real.

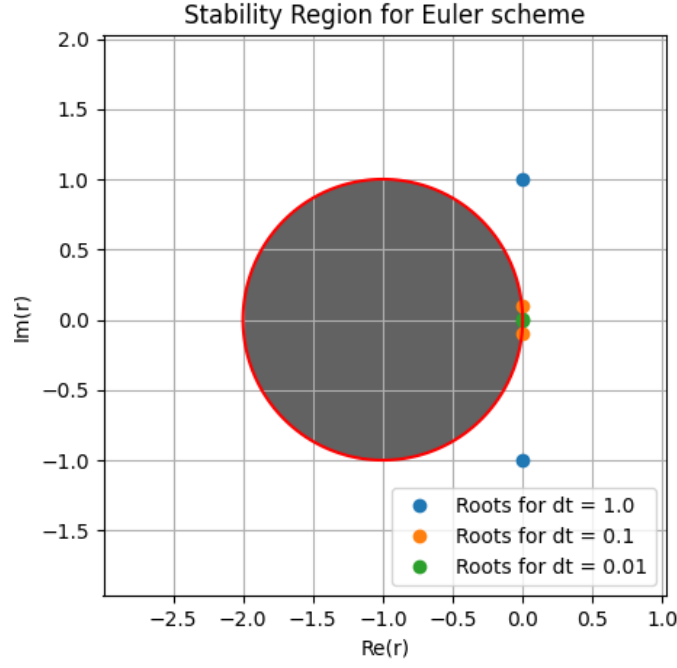


Figura 3.1: Región de estabilidad del esquema de Euler

Euler Inverso

La región de estabilidad del esquema de Euler Inverso, opuesto al Euler, se corresponde al exterior de una circunferencia de radio unidad centrada en $(1, 0)$. Las raíces del sistema siempre se encontrarán dentro de la región de estabilidad, pero no llegan al contorno. Es por esto por lo que los resultados obtenidos mediante el esquema de Euler Inverso disminuyen relativos a la solución real.

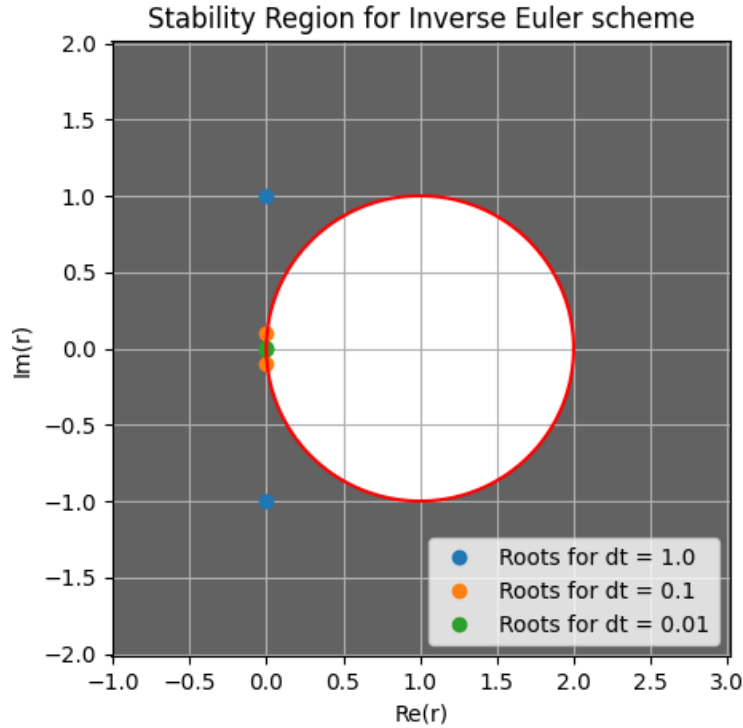


Figura 3.2: Región de estabilidad del esquema de Euler Inverso

Runge-Kutta 4

La región de estabilidad del esquema Runge-Kutta 4 tiene una forma compleja debido al orden del esquema. Las raíces del sistema se encontrarán en el contorno de la región de estabilidad para $\Delta t \in (0, 1]$. Para $\Delta t \in (1, 2.78)$ las raíces se encuentran en el interior de la región de estabilidad, y para $\Delta t > 2.78$, las raíces salen de ella.

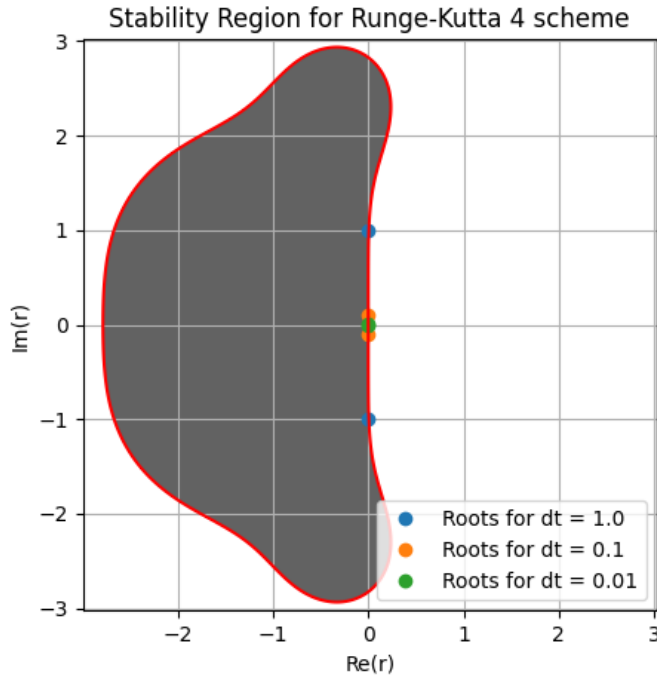


Figura 3.3: Región de estabilidad del esquema Runge-Kutta 4

Crank-Nicholson

La región de estabilidad del esquema Crank-Nicholson se corresponde al semiplano real negativo. Las raíces del sistema se encontrarán siempre en el contorno de la región de estabilidad. El paso de integración Δt que se elija afectará a la precisión de la solución, pero no a su estabilidad.

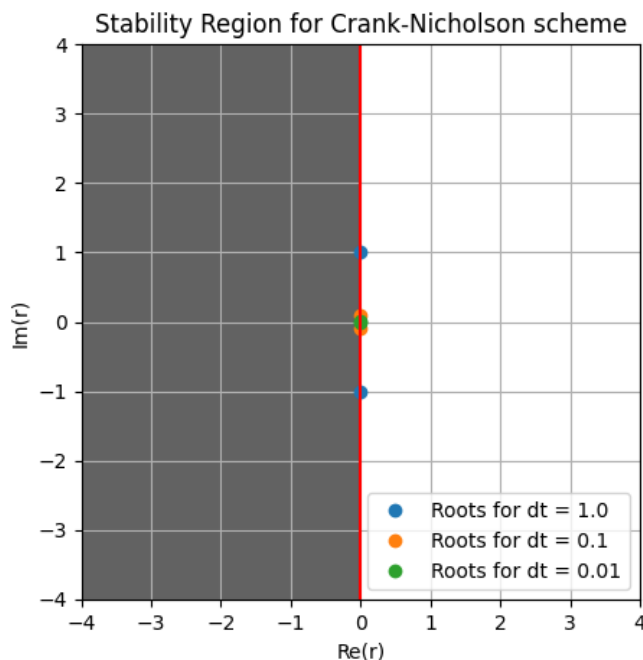


Figura 3.4: Región de estabilidad del esquema Crank-Nicholson

4. Actualización de módulos

Para resolver esta práctica se ha utilizado un nuevo módulo, *Estabilidad*, además de los utilizados en los anteriores hitos. Se ha añadido el esquema *Leap-Frog* y la función del oscilador armónico

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.

Además, incluye la representación de los resultados mediante la librería *Matplotlib*.

4.2. Esquemas temporales

Se ha añadido el esquema temporal *Leap-Frog*, donde $U1$ es el vector de estado en t_{n-1} y $U2$ es el vector de estado en t_n .

```
def LF(U1, U2, t, dt, F):  
    LF.__name__ = "Leap Frog"  
    return U1 + 2*dt*F(U2,t)
```

Además, se ha alojado el nombre de las funciones dentro de la propia definición de las funciones:

```
scheme.__name__ = "Scheme Name"
```

4.3. Cauchy_problem

Se ha actualizado la definición del problema de Cauchy para el esquema *Leap-Frog*, para que el primer paso de integración lo realice con un esquema Euler

```
if Esq_temp == LF:  
    U[:,1] = U[:,0] + dt*Phy(U[:,0], t[0])  
    for n in range(1,N):  
        U1 = U[:, n-1]  
        U2 = U[:, n]  
        U[:, n+1] = LF(U1, U2, t[n], dt, Phy)
```

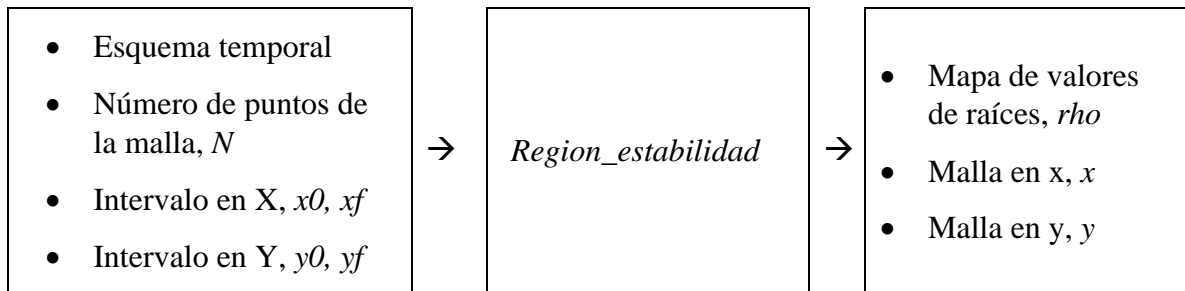
4.4. Physics

Se ha añadido la función del oscilador armónico

```
def Oscilator(U,t):  
    return array([U[1], -U[0]])
```

4.5. Estabilidad

Se ha definido este nuevo módulo que incluye el cálculo de la región de estabilidad.



```
def region_estabilidad(Temporal_Scheme, N, x0, xf, y0, yf):  
  
    x, y = linspace(x0, xf, N), linspace(y0, yf, N)  
    rho = zeros( (N, N))  
  
    for i in range(N):  
        for j in range(N):  
  
            w = complex(x[i], y[j])  
            r = Temporal_Scheme( 1., 1., 0., lambda u, t: w*u )  
            rho[i, j] = abs(r)  
  
    return rho, x, y
```