


Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingeniería Aeronáutica y del Espacio

Máster Universitario en Sistemas Espaciales

Milestone 4

Ampliación de Matemáticas I

5 de noviembre de 2022

Autor:

- Sergio Cavia Fraile

Índice

1. Introducción

El objetivo del trabajo será integrar un oscilador lineal mediante distintos esquemas temporales y analizar tanto los resultados como las respectivas regiones de estabilidad absoluta de cada método.

2. Ecuaciones y Métodos

2.1. Oscilador lineal

El oscilador armónico lineal se define mediante el siguiente problema de Cauchy:

$$\ddot{x} + x = 0 \rightarrow \ddot{x} = -x = F(x)$$
$$x(0) = 1; \dot{x} = 0$$

Cuya solución analítica es:

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

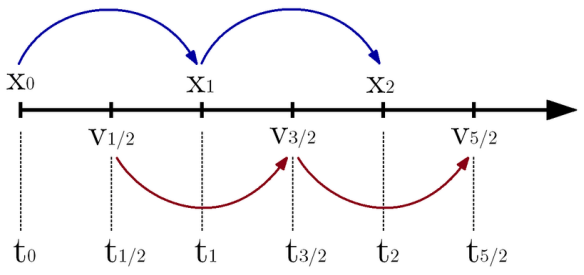
2.2. Método de Leap-Frog

Tan solo explicaremos este método ya que los otrs 4 han sido explicados previamente en otros informes.

El método de Leap-Frog se describe de la siguiente forma:

$$x_{i+1} = x_i + v_{i+\frac{1}{2}} * \Delta t$$
$$v_{i+\frac{1}{2}} = v_{i-\frac{1}{2}} + F_i * \Delta t$$

En la siguiente imagen se puede observar cómo se realizan los saltos:



Por simplicidad, nosotros no hemos trabajado con mitades del paso temporal y hemos hecho los cálculos de dos en dos pasos temporales. También veo relevante comentar que para integrar el primer paso temporal hemos utilizado un esquema de Euler para, una vez calculados dos pasos temporales, poder comenzar a usar el Leap-Frog.

2.3. Regiones de estabilidad absoluta

El primer paso para calcular las regiones de estabilidad absoluta es conocer el polinomio de estabilidad de cada método. Los respectivos a los 5 esquemas temporales que usaremos son:

Euler: $r = 1 + \omega$

Euler inverso: $r = \frac{1}{1-\omega}$

Crank Nicholson: $r = \frac{1+\frac{\omega}{2}}{1-\frac{\omega}{2}}$

Runge-Kutta Orden 4: $r = 1 + \omega + \frac{\omega^2}{2!} + \frac{\omega^3}{3!} + \frac{\omega^4}{4!}$

Leap-Frog: $r = \sqrt{1}$

Una vez conocidos dichos polinomios, para conocer la región de estabilidad absoluta tan solo debemos ver qué valores de ω nos dan un r de módulo menor que 1 ($|r| < 1$).

3. Código

3.1. Oscilador Lineal

El oscilador lineal se ha implementado mediante las siguientes líneas de código:

```
from numpy import array

"Linear Oscilator function"

def LinOsc(U, t):

    return array([U[1], -U[0]])
```

3.2. Método de Leap-Frog

El Leap-Frog se ha implementado mediante las siguientes líneas de código:

```
def LF(U2, U1, dt, t, F):

    return U1 + 2*dt*F(U2,t)
```

Aunque solo se han escrito dos líneas, para la implementación completa del método ha sido necesario modificar el código que resuelve el problema de Cauchy para que en el caso de utilizar el Leap-Frog se calcule el primer paso por el método de Euler como ya explicamos en apartados anteriores:

```
from numpy import zeros
from ODEs.SchemesCode import Euler, RK4, CN, EulerInv, LF

"Cauchy Problem"

def Cauchy(F, t, U0, Scheme):

    U = zeros((len(t),len(U0)))
    U[0,:] = U0
    dt = t[2] - t[1]

    if Scheme == LF:
        U[1,:] = U[0,:] + dt*F(U[0,:], t[0])

        for n in range(1,len(t)-1):
            U1 = U[n-1, :]
            U2 = U[n, :]
            U[n+1, :] = Scheme(U2, U1, dt, t[n], F)

    else:
        for n in range(len(t)-1):
            U[n+1,:] = Scheme(U[n, :], t[n+1] - t[n], t[n],  F)

    return U
```

3.3. Regiones de estabilidad absoluta

Para calcular las regiones de estabilidad absoluta se ha escrito un código que define los polinomios de estabilidad y luego calcula el valor absoluto de r para cada punto discretizado del plano complejo:

```
from ODEs.SchemesCode import Euler, RK4, CN, EulerInv, LF
from numpy import meshgrid, array, zeros, size, absolute, sqrt

"Absolute Stability Region"

def Regions(x,y,Scheme):
    Z = zeros([size(x),size(x)],dtype=complex)
    for i in range(size(x)):
        for j in range(size(x)):
            Z[size(x)-1-j,i] = complex(x[i],y[j])

    return absolute(array(StabPoly(Scheme, Z)))


def StabPoly(Scheme, w):
    if Scheme == Euler:
        r = 1 + w
    elif Scheme == EulerInv:
        r = 1/(1-w)
    elif Scheme == CN:
        r = (1+w/2)/(1-w/2)
    elif Scheme == RK4:
        r = 1 + w + (w**2)/2 + (w**3)/6 + (w**4)/(4*3*2)
    elif Scheme == LF:
        r = sqrt(1)

    return r
```

3.4. Código final

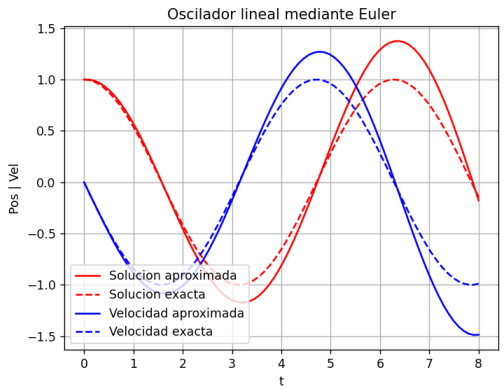
El código que realmente integra el oscilador lineal y calcula las regiones de estabilidad absoluta de los distintos métodos es el siguiente:

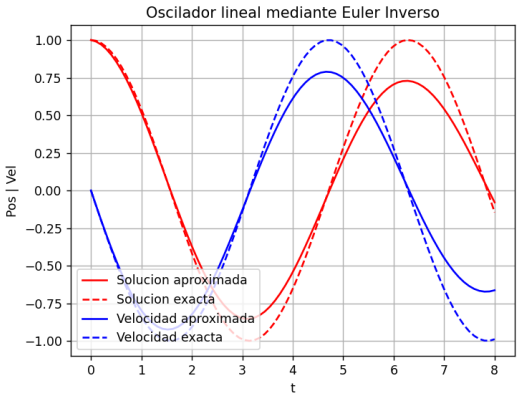
```
from numpy import array, linspace, cos, sin
import matplotlib.pyplot as plt
from ODEs.SchemesCode import Euler, RK4, CN, EulerInv, LF
from Functions.LinOscCode import LinOsc
from ODEs.CauchyCode import Cauchy
from ODEs.SchemesCode import Euler, RK4, CN, EulerInv, LF
from Stability.RegionsCode import Regions
from Richardson.ErrorCode import Error
from Richardson.ConvergenceCode import Conv

#Defino tiempo final, numero divisiones y conodiciones iniciales
tf = 8
N = 80
t = linspace(0, tf, N)
U0 = array([1, 0])


#Integro el oscilador lineal
U = Cauchy(LinOsc, t, U0, RK4)
#Lo grafico
plt.plot(t, U[:,0],"r", label = "Solucion aproximada")
plt.plot(t, cos(t),"--",color = "r", label = "Solucion exacta")
plt.plot(t, U[:,1], "b", label = "Velocidad aproximada")
plt.plot(t, -sin(t),"--", color = "b", label = "Velocidad exacta")
plt.title("Oscilador lineal dt = 0.1, Leap-Frog")
plt.xlabel("t")
plt.ylabel("y(t)|dy(t)")
plt.legend(loc = "lower left")
plt.grid()
plt.show()


#Discretizo el plano complejo
x = linspace(-5, 5, 100);
y = linspace(-5, 5, 100);

#Pasos temporales que situaré en el plano complejo
dt = array([0.01, 0.1, 0.5, 1])


#Calculo y grafico las regiones de estabilidad absoluta
stab_region = Regions(x,y,RK4)
CSF = plt.contourf(x, y, stab_region, levels = [0, 1],  colors=['#C0C0C0'] )
CS = plt.contour(x, y, stab_region, levels = [0.75, 1, 1.25] )
for t in range(len(dt)):
    plt.plot([0,0],[1*dt[t],-1*dt[t]], 'o', label = "Raíces del oscilador lineal con dt = " +str(dt[t]))
plt.clabel(CS, inline=1, fontsize=10)
plt.title('Regiones de estabilidad absoluta del Leap-Frog')
plt.xlabel("Re(|r|)")
plt.ylabel("Im(|r|)")
plt.legend(loc = "lower left")
plt.grid()
plt.show()
```

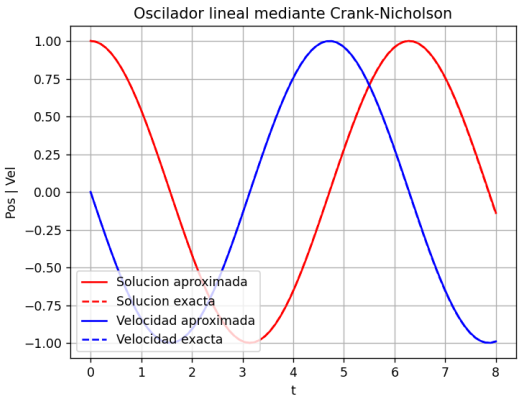
4. Resultados

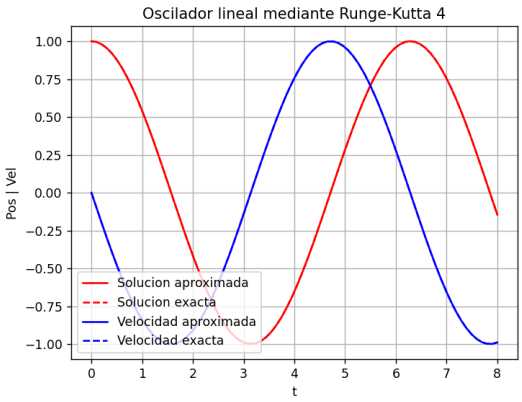
4.1. Oscilador lineal

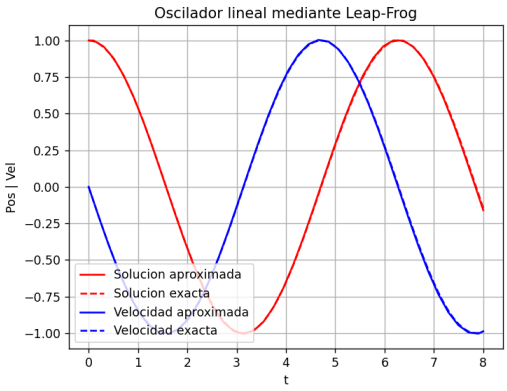
En las siguientes gráficas se pueden observar las posiciones (rojo) y velocidades (azul) del oscilador lineal para cada tiempo y para cada uno de los métodos. Se han simulado 8 unidades temporales con un paso temporal de 0.1 y se ha graficado también la solución exacta en línea discontinua para poder compararla con la solución aproximada correctamente.



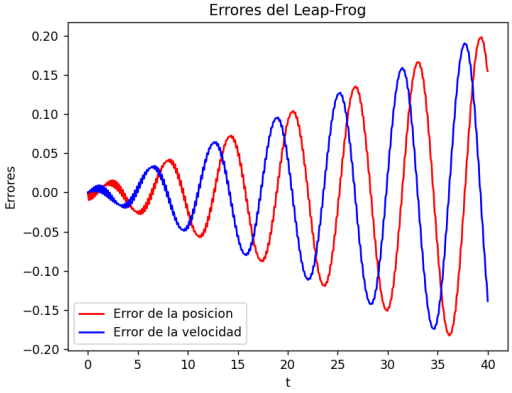


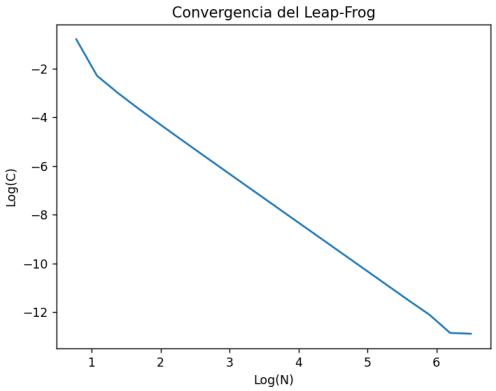




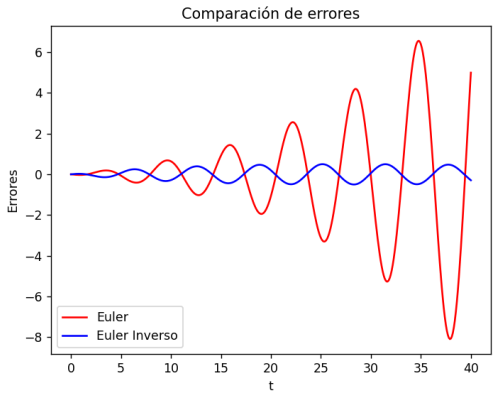


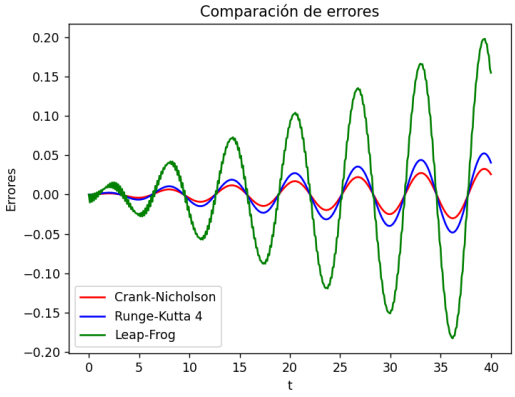
Al ser el Leap-Frog un método que aún no hemos estudiado en profundidad, he decidido analizar su error (con un paso temporal de 0.1) y su ratio de convergencia como ya hicimos en el anterior informe:





En las gráficas de las soluciones podemos ver como los dos métodos de Euler difieren rápidamente de la solución exacta mientras que los otros tres métodos parecen tener un error menos, para una mejor visualización se han graficado los errores de los distintos métodos en las dos siguientes gráficas:

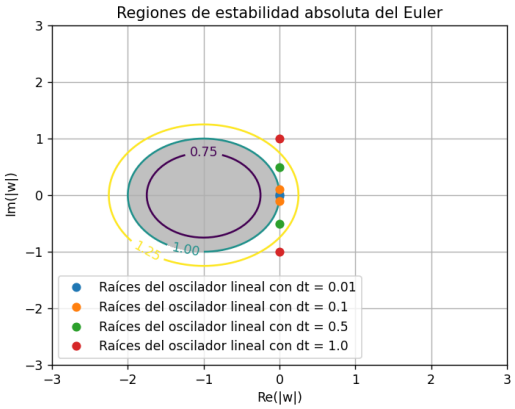


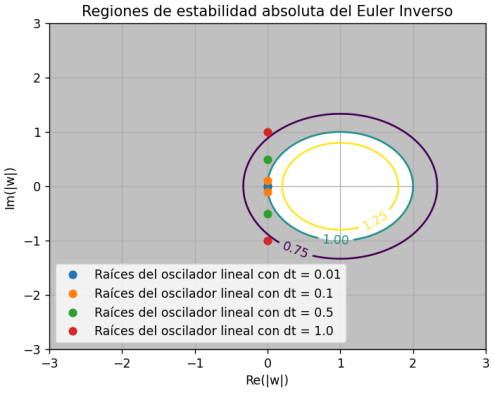


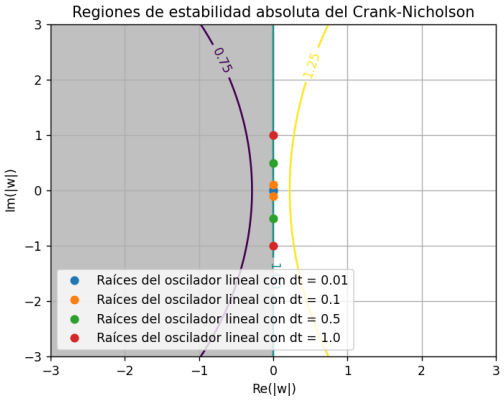
En ellas se puede ver que el método con mayor error es el de Euler, el Euler Inverso tiene un error menor pero que sigue siendo muy alto, por otro lado, el método con menor error es el Crank-Nicholson seguido de cerca por el Runge-Kutta de orden 4, el Leap-Frog se encuentra en medio de esos dos grupos.

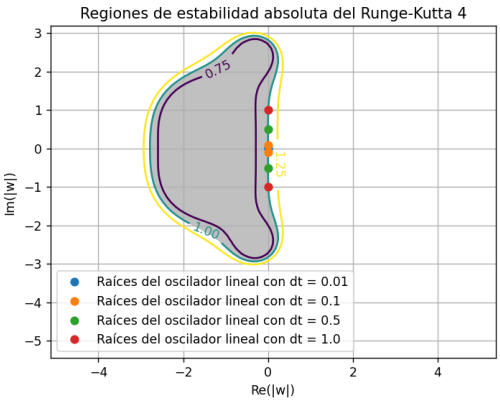
4.2. Regiones de estabilidad absoluta

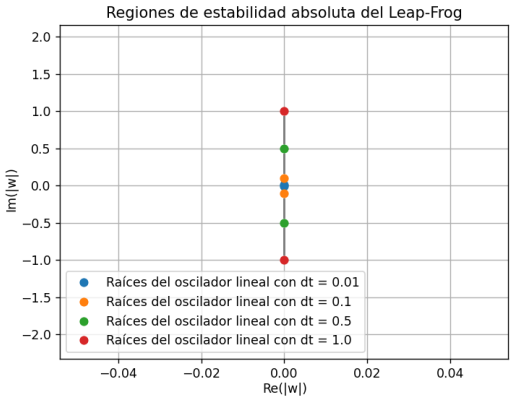
En las siguientes figuras se muestran las regiones de estabilidad de cada uno de los métodos estudiados:











El método de Euler da raíces que siempre se encuentran fuera de la región de estabilidad absoluta, solo se tendería a estar en esta cuando el paso tendiera a cero, por lo que a menor paso temporal menor será el error, pero nunca estará acotado.

En el método de Euler Inverso todos los pasos temporales caen dentro de la región de estabilidad, por lo que están acotado, acercándose más a $|r| = 1$ cuanto menor sea el paso temporal.

En el método de Crank-Nicholson da igual el paso temporal que se escoja ya que siempre nos encontraremos en la frontera de la región de estabilidad absoluta (donde $|r| = 1$).

En el Runge-Kutta de orden 4 tenemos tres casos distintos. Para un paso temporal menor que la unidad el esquema se comporta como el Crank-Nicholson y nos encontraremos en la frontera de la región de estabilidad absoluta (donde $|r| = 1$), para pasos temporales entre la unidad y 2.8 nos encontramos dentro de la región de estabilidad absoluta, y si nuestro paso temporal es superior a 2.8 nos encontraremos fuera de dicha región y el error dejará de esta acotado.

Por último, en el Leap-Frog la región de estabilidad se encuentra en el intervalo (-1,1) del eje imaginario de ω, por lo que cualquier paso temporal menor a la unidad nos dará un error acotado.