

# Actividad 5: Movimiento armónico simple: Péndulo

Luisa Fernanda Orci Fernandez.

18 de Febrero del 2016

## Péndulo

En esta actividad volvemos a retomar el tema de la actividad 1, el péndulo. Este es un sistema físico que puede oscilar bajo la acción de la gravedad.

Existen varios tipos de péndulos, algunos de ellos son el péndulo simple, el compuesto, el doble péndulo, el péndulo de torsión, el de Newton, entre otros. Uno de los usos que se le dan al péndulo es el de medir el tiempo, o medir la gravedad.

Matemáticamente el péndulo es muy complicado, pero se puede simplificar tomando en cuenta algunas suposiciones para facilitar su cálculo como por ejemplo, un péndulo simple con ángulos pequeños.

En esta actividad se nos solicitó un código que fuera capaz de resolver la ecuación de movimiento de un péndulo;

$$\frac{d^2\theta}{dt^2} + \frac{g}{l} \sin \theta = 0$$

, la cual es una ecuación diferencial de segundo orden.

## Actividad

Para poder llevar a cabo esta actividad, se nos proporcionó un código donde se resuelve esta ecuación para cualquier ángulo *theta*. Este lo obtuvimos de la página *SciPy.org*, el código es el siguiente:

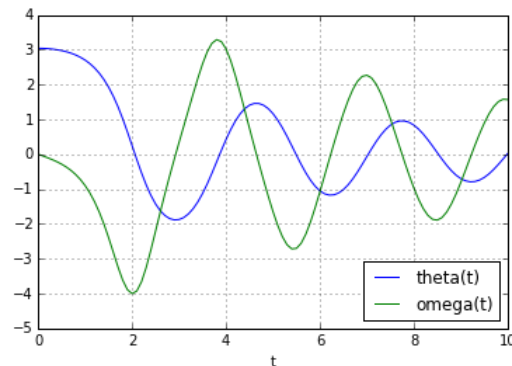
```
import numpy as np
>>> def pend(y, t, b, c):
...     theta, omega = y
...     dydt = [omega, -b*omega - c*np.sin(theta)]
...     return dydt
...
```

```

>>> b = 0.25
>>> c = 5.0
>>> y0 = [np.pi - 0.1, 0.0]
>>> t = np.linspace(0, 10, 101)
>>> from scipy.integrate import odeint
>>> sol = odeint(pend, y0, t, args=(b, c))
>>> import matplotlib.pyplot as plt
>>> plt.plot(t, sol[:, 0], 'b', label='theta(t)')
>>> plt.plot(t, sol[:, 1], 'g', label='omega(t)')
>>> plt.legend(loc='best')
>>> plt.xlabel('t')
>>> plt.grid()
>>> plt.show()

```

Y la gráfica fue la siguiente:



Este código lo modificamos de la siguiente manera:

- Primeramente se definió la ecuación:  $\ddot{\theta}(t) + b\dot{\theta}(t) + c\sin(\theta(t)) = 0$ , donde las tildes son la primera y segunda derivada.
- Después convertimos esta ecuación en un sistema de ecuaciones de primer orden, donde la velocidad angular omega es  $\dot{\theta}(t) = \omega(t)$ , después obtuvimos el sistema donde  $\dot{\omega}(t) = -b\omega(t) - c\sin(\theta(t))$  y quedó de la siguiente forma:

$$\dot{\omega}(t) = -b * \omega(t) - c * \sin(\theta(t))$$

.

- El vector  $y$  quedó como  $[\theta, \omega]$ .
- Consideramos que el péndulo estaba con posición vertical con  $y0 = [\pi - 0.1, 0.0]$
- Después se generó una solución de 101 muestras uniformemente espaciadas en un intervalo  $0 \leq t \leq 10$ .

- Para generar la solución llamamos a *odeint*, y le dimos los parámetros  $b$  y  $c$  utilizando *args*.

Quedó de la siguiente manera:

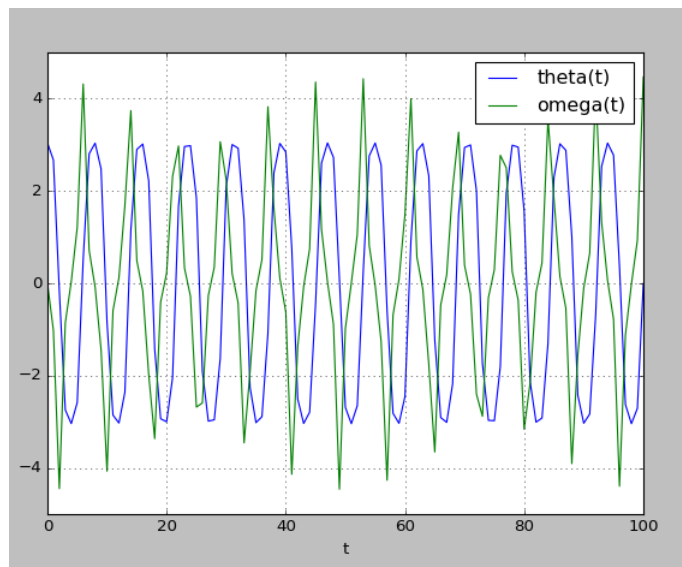
```
#Modifiacion, friccion = 0, theta = pi-0.1, c = 5
def pend(y, t, b, c):
    theta, omega = y
    dydt = [omega, -b*omega - c*np.sin(theta)]
    return dydt

b = 0.0
c = 5.0
y0 = [np.pi-0.1, 0.0]
t = np.linspace(0, 100, 101)

from scipy.integrate import odeint
sol = odeint(pend, y0, t, args=(b, c))
import matplotlib.pyplot as plt

plt.plot(t, sol[:, 0], 'b', label='theta(t)')
plt.plot(t, sol[:, 1], 'g', label='omega(t)')
plt.legend(loc='best')
plt.xlabel('t')
plt.grid()
plt.show()
```

y la gráfica fue:



# 1. Resultados

## Modificación 1

En este caso la única modificación realizada fue al ángulo, se utilizó un ángulo de  $\frac{\pi}{4}$ .

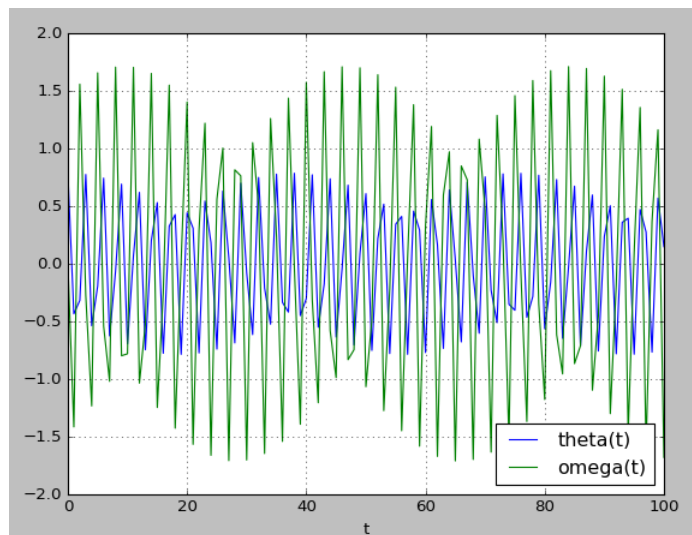
```
#Caso 1, friccion = 0, theta = pi/4, c = 5
def pend(y, t, b, c):
    theta, omega = y
    dydt = [omega, -b*omega - c*np.sin(theta)]
    return dydt
```

```
b = 0
c = 5.0
y0 = [np.pi/4, 0.0]
t = np.linspace(0, 100, 101)
```

```
from scipy.integrate import odeint
sol = odeint(pend, y0, t, args=(b, c))
import matplotlib.pyplot as plt
```

```
plt.plot(t, sol[:, 0], 'b', label='theta(t)')
plt.plot(t, sol[:, 1], 'g', label='omega(t)')
plt.legend(loc='best')
plt.xlabel('t')
plt.grid()
plt.show()
```

y la gráfica fue:



## Modificación 2

Esta vez si agregamos fricción y se tomó un ángulo de  $\frac{\pi}{4}$

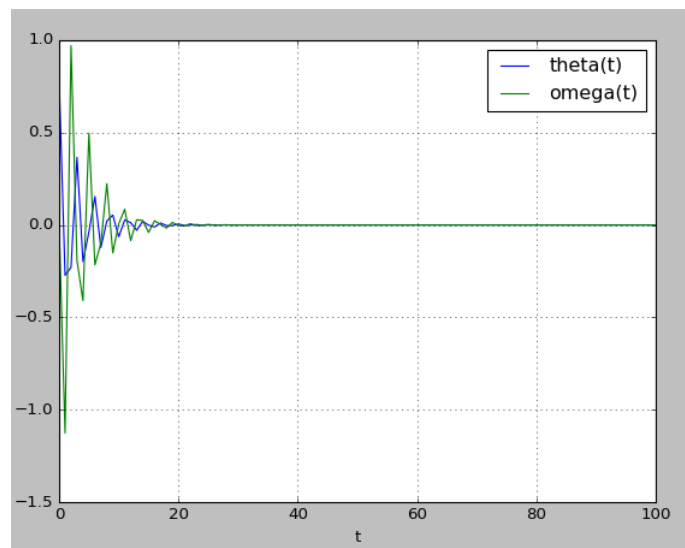
```
#Caso 2, friccion = .5, theta = pi/4, c = 5
def pend(y, t, b, c):
    theta, omega = y
    dydt = [omega, -b*omega - c*np.sin(theta)]
    return dydt

b = 0.5
c = 5.0
y0 = [np.pi/4, 0.0]
t = np.linspace(0, 100, 101)

from scipy.integrate import odeint
sol = odeint(pend, y0, t, args=(b, c))
import matplotlib.pyplot as plt

plt.plot(t, sol[:, 0], 'b', label='theta(t)')
plt.plot(t, sol[:, 1], 'g', label='omega(t)')
plt.legend(loc='best')
plt.xlabel('t')
plt.grid()
plt.show()
```

La gráfica que resultó fue:



### Modificación 3

Esta vez si agregamos fricción y se tomó un ángulo de  $\frac{\pi}{4}$  pero esta vez se modificó la longitud del péndulo.

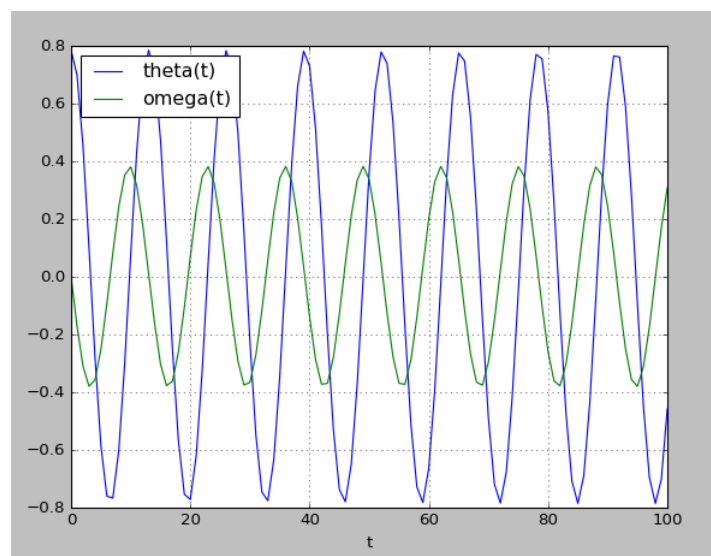
```
#Caso 3, friccion = 0, theta = pi/4, c = .25
def pend(y, t, b, c):
    theta, omega = y
    dydt = [omega, -b*omega - c*np.sin(theta)]
    return dydt
```

```
b = 0.0
c = .25
y0 = [np.pi/4, 0.0]
t = np.linspace(0, 100, 101)
```

```
from scipy.integrate import odeint
sol = odeint(pend, y0, t, args=(b, c))
import matplotlib.pyplot as plt
```

```
plt.plot(t, sol[:, 0], 'b', label='theta(t)')
plt.plot(t, sol[:, 1], 'g', label='omega(t)')
plt.legend(loc='best')
plt.xlabel('t')
plt.grid()
plt.show()
```

La gráfica que resultó fue:



## Modificación 4

Aquí hubo modificación en la fricción, el ángulo y la longitud.

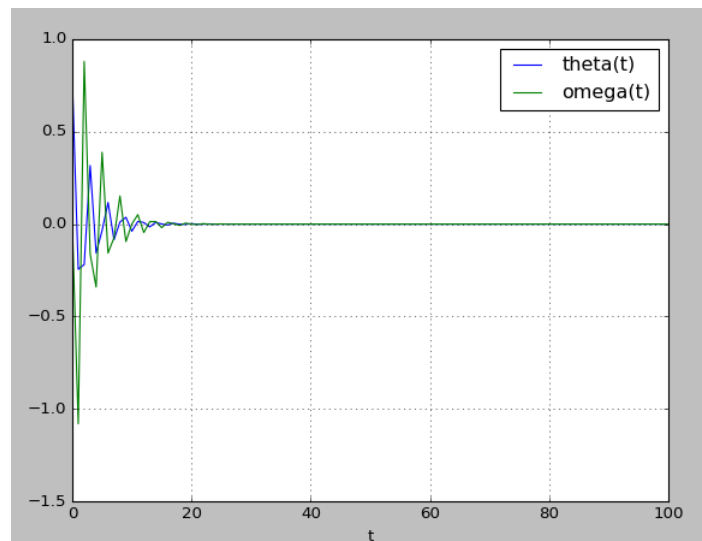
```
#Caso 4, friccion = .6, theta = pi/4, c = 5
def pend(y, t, b, c):
    theta, omega = y
    dydt = [omega, -b*omega - c*np.sin(theta)]
    return dydt

b = 0.6
c = 5.0
y0 = [np.pi/4, 0.0]
t = np.linspace(0, 100, 101)

from scipy.integrate import odeint
sol = odeint(pend, y0, t, args=(b, c))
import matplotlib.pyplot as plt

plt.plot(t, sol[:, 0], 'b', label='theta(t)')
plt.plot(t, sol[:, 1], 'g', label='omega(t)')
plt.legend(loc='best')
plt.xlabel('t')
plt.grid()
plt.show()
```

La gráfica final quedó así:



## Conclusión

Esta actividad resultó muy interesante ya que tuvimos la oportunidad de modelar un fenómeno físico y como estudiante de la Licenciatura en Física, esto resultará de mucha ayuda en un futuro. De igual manera resulta muy útil el entender como construir un programa que nos ayude con esto.

## Referencias

- [1] Wikipedia, <https://es.wikipedia.org/wiki/Pendulo>
- [2] Scipy, <http://scipy.github.io/devdocs/generated/scipy.integrate.odeint.html>