

Actividad 7: El espacio Fase

Luisa Fernanda Orci Fernandez.

10 de Marzo del 2016

Espacio Fase

El espacio fase, también conocido como espacio fásico o diagrama de fases, cada punto de este espacio representa un estado del sistema físico y este espacio se caracteriza por cada uno de los momentos respectivos de cada "partícula".

El espacio fase es una herramienta de gran ayuda para estudiar sistemas dinámicos, tal es el caso del péndulo simple, un oscilador armónico simple o uno de Van de Pol, entre otros.

Actividad

En esta actividad, como en las anteriores, seguimos trabajando con un péndulo pero la diferencia es que esta vez construiremos el Espacio fase del péndulo, este espacio representará cada una de las soluciones de un péndulo simple.

Para realizar esta actividad, se nos proporcionó un código ejemplo de *Lotka – Volterra*, (este se encuentra en *SciPy CookBook*), el cual describía el comportamiento de una población de zorros y conejos, esto con fin de que nosotros lo modificáramos para representar el Espacio Fase de un péndulo simple con condiciones iniciales. Posteriormente recurrimos a *Matplotlib* de *Python* para graficar el espacio fase.

El código final fue el siguiente:

```
#PAQUETES
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
```

```
#CONSTANTES
```

```
g = 9.8
l = 1.0
b = 0.0
```

```

c = g/l

#CONDICIONES INICIALES
X_f1 = gatito.array([-4.0*gatito.pi,4.0*gatito.pi])
X_f2 = gatito.array([-2.0*gatito.pi,-0.0*gatito.pi])
t = gatito.linspace(-0.0*gatito.pi,5.0*gatito.pi,500)
#Definimos la ecuacion diferencial del pendulo
def p (y, t, b, c):
    theta, omega = y
    dy_dt = [omega,-b*omega -c*gatito.sin(theta)]
    return dy_dt

#COLOR Y NUMERO DE PUNTOS

values = gatito.linspace(-1.0,1.0,100) #Numero de puntos
vcolors = plt.cm.winter_r(gatito.linspace(0.5, 1.0, len(values)))#Color
plt.figure(2)

# TRAYECTORIA 1
for v, col in zip(values, vcolors):
    y0 = v * X_f1

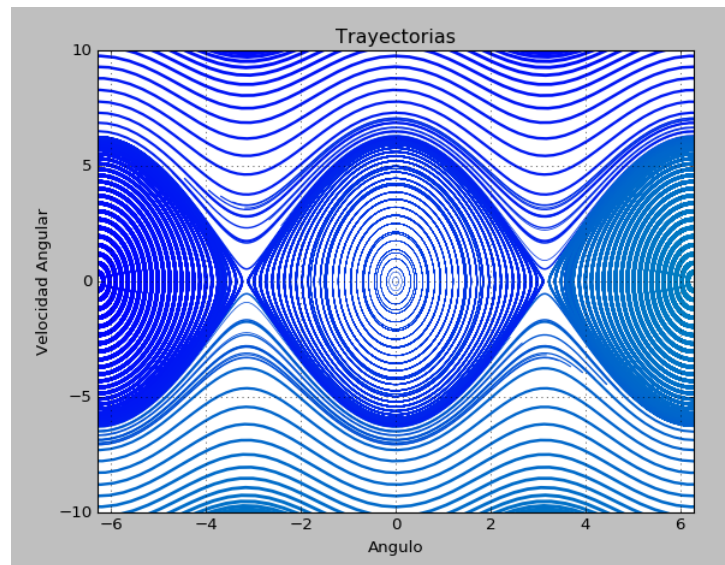
    X = odeint(p, y0, t, args=(b,c))
    plt.plot( X[:,0], X[:,1], lw=3.5*v, color=col,
    label='X0=(%.f, %.f)' % ( y0[0], y0[1]) )

#Definimos la trayectoria 1
for v, col in zip(values, vcolors):
    y1 = v * X_f2
    X1 = odeint(p, y1, t, args=(b,c))
    plt.plot( X1[:,0], X1[:,1], lw=3.5*v, color=col,
    label='X0=(%.f, %.f)' % ( y1[0], y1[1]) )

#Grafica
plt.title('Trayectorias')
plt.xlabel('Angulo')
plt.ylabel('Velocidad Angular')
plt.grid()
plt.xlim(-2.0*gatito.pi,2.0*gatito.pi)
plt.ylim(-10,10)
plt.show()

```

y la gráfica resultante fue la siguiente:



Conclusiones

Esta actividad, como todas las anteriores, se me hizo de gran interes, no solo por que ahora sabemos como graficar espacios fase en *Python*, si no por que en un futuro será de gran ayuda.

Referencias

- [1] Wikipedia, *Espacio Fásico*, (2016, 10 de Marzo).Desde: https://es.wikipedia.org/wiki/Espacio_f%C3%A1sico