

Actividad 7: El espacio Fase

Jose Pablo Salazar Velazquez

18 de Marzo del 2016

Espacio Fase

En mecánica clásica, el espacio fásico, espacio de fases o diagrama de fases es una construcción matemática que permite representar el conjunto de posiciones y momentos conjugados de un sistema de partículas. Más técnicamente, el espacio de fases es una variedad diferenciable de dimensión par, tal que las coordenadas de cada punto representan tanto las posiciones generalizadas como sus momentos conjugados correspondientes. Es decir, cada punto del espacio fásico representa un estado del sistema físico. Ese estado físico vendrá caracterizado por la posición de cada una de las partículas y sus respectivos momentos.

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

Siguiendo la logica de las practicas anteriores, nos falta representar el espacio fase, actividad que se realizo en este trabajo.

Para realizar esta actividad, se nos proporcionó un código ejemplo, esto con fin de que nosotros lo modificáramos para representar el Espacio Fase de un péndulo simple. Posteriormente recurrimos a *Matplotlib* de *Python* para gráficar el espacio fase.

El código, que resulto de modificar el original, fue el siguiente:

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

#parametros iniciales
g = 9.81
l = 1.0
```

```

b = 0.0
c = g/l

#C.I

X_f1 = np.array([-3*np.pi,np.pi])
X_f2 = np.array([-np.pi,np.pi])
t = np.linspace(0,20,300)

def p (y, t, b, c):
    theta, omega = y
    dy_dt = [omega,-b*omega -c*np.sin(theta)]
    return dy_dt

values = np.linspace(-1.0,1.0,25) # position of X0 between X_f0 and X_f2
vcolors = plt.cm.winter_r(np.linspace(0.5, 1.0, len(values))) # colors for each trajectory

plt.figure(2)

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]))

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]))

plt.title('Trayectorias')
plt.xlabel('Angulo')
plt.ylabel('Velocidad Angular')
plt.grid()
plt.xlim(-2.0*np.pi,2.0*np.pi)

```

```
plt.ylim(-12,12)
```

```
plt.show()
```

y el espacio generado, fue el siguiente

