Actividad 7: El Espacio Fase

Valenzuela Carrillo María Inés

14 de Marzo del 2016

1. Introducción

El espacio fase es una representación geométrica de las trayectorias de un sistema dinámico. Este se construye con diferentes condiciones iniciales, donde cada una representa una curva o punto.

Los espacios fase son una herramienta valiosa, ya que te permiten visualizar al mismo tiempo que pasaría para diferentes condiciones y poder identificar características especiales del sistema dinámico.

Para esta actividad se pide construir el retrato del espacio fase que represente la familia de soluciones del péndulo simple, tal como se muestra en la figura 1.

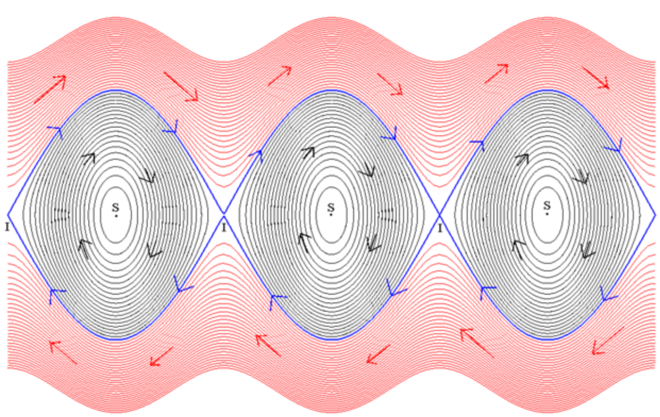

Figura 1: Espacio fase ejemplo

2. El Espacio Fase del Péndulo Simple

Para construir la figura, se utilizó la biblioteca Matplotlib de Python para graficación, y se adaptó el código del ejemplo resuelto en el tutorial de Lotka-Volterra que aparece en el SciPy CookBook, en el ual de muestra una gráfica del espacio fase.

Código

En la primera parte se define la ecuación del péndulo y los parámetros, de la misma forma que se hizo en la actividad 5.

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


#Definicion de la ecuacion
def pend(y, t, b, c):
        theta, omega = y
        dydt = (omega, -b*omega - c*np.sin(theta))
        return dydt


#Parametros
b = 0             #friccion
g= 9.8            # aceleracion gravitacional
l=1               #longitud del pendulo
c=g/l
t = np.linspace(0.0,20,500)  #intervalo de tiempo
```

Después se establecen las condiciones iniciales, son dos ya que una está dada para los movimientos periódicos y la otra para los que no se den de forma periódica, es decir, cuando el péndulo continúe avanzando de ángulo.

```
#Condiciones iniciales
X_f1 =np.array([-70.0*np.pi,20])
X_f2 =np.array([-2.0*np.pi,0.0])
```

Se definen los colores y el número para los dos tipos de trayectorias.

```
values1 =np.linspace(-1,1,70)
values2 =np.linspace(-1,1,50)
vcolors1 = plt.cm.Blues(np.linspace(1.0, 1.0, len(
    values1)))  # colores1 para cada trayectoria
vcolors2 = plt.cm.PuRd(np.linspace(0.7, 0.7, len(values2
    )))  # colores2 para cada trayectoria
```

Se resuelven las ecuaciones y se grafican.

```
plt.figure(2)


#Trayectorias superiores e inferiores
for v1, col1 in zip(values1, vcolors1):
    y1 = v1 * X_f1
    X1 = odeint(pend, y1, t, args=(b,c))
    plt.plot( X1[:,0], X1[:,1], lw=1.5*v1, color=col1 )

#Trayectorias centrales
for v2, col2 in zip(values2, vcolors2):
    y2 = v2 * X_f2
    X2 = odeint(pend, y2, t, args=(b,c))
    plt.plot( X2[:,0], X2[:,1], lw=1.5*v2, color=col2 )


#Para graficar
plt.title('Trayectorias')
plt.xlabel('Angulo')
plt.ylabel('Velocidad Angular')
plt.grid()
```

```
plt.xlim(-3.0*np.pi,3.0*np.pi)
plt.ylim(-10,10)


plt.show()
```

A continuación se muestra la gráfica del espacio fase que resulta del código:

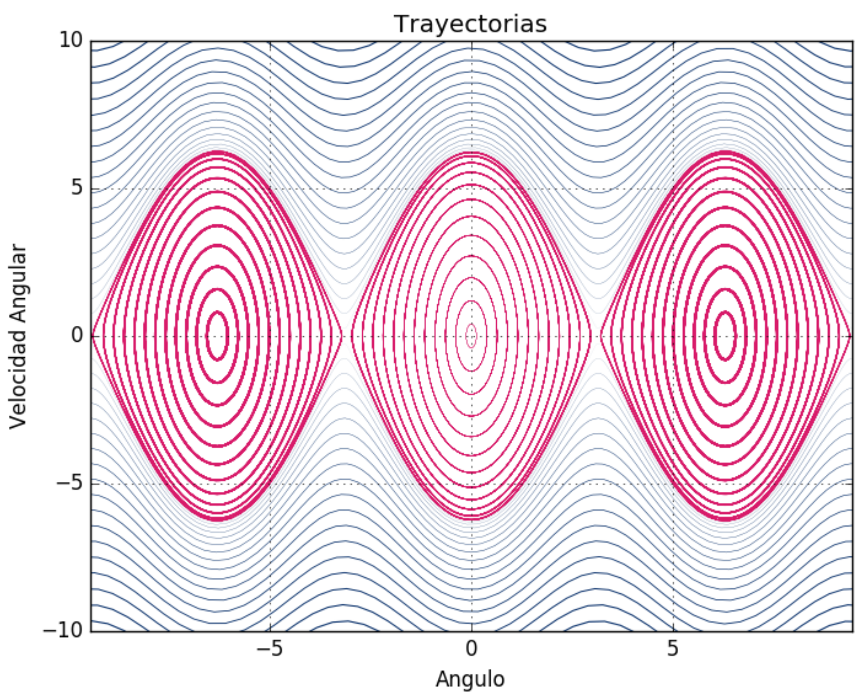

Figura 2: Espacio Fase del Péndulo Simple

La gráfica coincide con lo que se observa fisicamente.

Referencias

[1] Wikipedia, the free encyclopedia, *Phase portrait.*

[2] Actividad 7 (2016-1), *http://computacional1.pbworks.com/w/page/105676740/Actividad 7 (2016-1).*

[3] scipy-cookbook, Matplotlib: lotka volterra tutorial, *http://scipy-cookbook.readthedocs.org/items/LoktaVolterraTutorial.html.*