Actividad 5: Movimiento armónico simple: Péndulo

Martin Alejandro Paredes Sosa

Marzo, 2016

1. Introducción

La matemática de un péndulo simple es, en general, compleja. Hacer suposiciones que simplifican la descripción nos permite resolver analíticamente las ecuaciones de movimiento para las oscilaciones de ángulos pequeños.
El péndulo simple, es una idealización se un péndulo real, pero en un sistema aislado donde se asume:

- La cuerda tiene una masa despreciable, es rígida y se mantiene tensa.
- El péndulo se maneja como una masa puntual.
- El movimiento es en dos dimensiones trazando un arco.
- No pierde energía por fricción o resistencia al aire.
- El campo gravitacional es uniforme.
- El soporte no se mueve.

La ecuación que nos permite encontrar el periodo cuando se utilizan ángulos pequeños es:

$$T_o = 2\pi\sqrt{\frac{\ell}{g}} \tag{1}$$

Mientras que para cualquier angulo es:

$$T = 4\pi \int_0^{\theta_o} \frac{1}{\sqrt{cos\theta - \cos\theta_o}} d\theta \tag{2}$$

donde g es la aceleración de la gravedad, ℓ es la longitud del péndulo y θ_o es la condición inicial [1].

2. Función *integrate.quad*

Para poder resolver la integral que se tiene en la ecuacion (2) se utilizaron herramientas computacionales. En nuestro caso, para resolverla se utiliza la función *integrate.quad* de la librería *scipy* de Python [2].

Esta función utiliza los siguientes parámetros:

- func:*function*. Es la función definida anteriormente para integrar.
- a: *float*. Limite inferior de integración
- b: *float*. Limite superior de integración
- args:*tuple, opcional*. Argumentos extra para la función.

3. Ejercicio y Resultados

Esta actividad consistió en realizar un código en python que nos permitiera encontrar el periodo de un péndulo para ángulos arbitrarios. Se hizo uso de la librería *scipy.integrate* haciendo uso de la función *quad*. Una vez que se tenia el periodo se gráfico el error relativo T/T_o y se demostró como es que diverge el periodo a medida que el angulo inicial tiende a π.

El siguiente código fue el que se utilizo:

```
#LIBRERIAS
import numpy as np
from scipy.integrate import quad
import matplotlib.pyplot as plt
#Constantes del problema.
l = 1.0     #Longitud Del Pendulo
g = 9.81  #Gravedad
n = 500
eps= 0.001
theta = np.linspace(0.0,np.pi,n)
theta_0 =np.linspace(eps, np.pi-eps, n)
res = [0 for i in range(n)]
err = [0 for i in range(n)]
T = [0 for i in range(n)]

#Caso de oscilaciones pequenas
T_o = 2.0 * np.pi*np.sqrt(l/g)
#Para Oscilaciones grandes
#Calculo de la integral
inte = lambda x, k : 1.0 /(np.sqrt(np.cos(x)-np.cos(k)))
for i in range(n):
    theta_00 = theta_0[i]
    res[i] , err [i] = quad(inte, 0, theta_00, args=(theta_00))

```

```
#Calculo del periodo
    T[i] = 4*np.sqrt(l/(2*g)) * res[i]
#Calculo del error
ErrorR = (T/T_o)
x = (theta * 180.0)/np.pi #Transformacion a Grados
x1 = (theta * 180.0)/np.pi

#Grafica del Error Relativo
plt.figure(1)
plt.plot(x1,ErrorR)
plt.xlabel("Angulo")
plt.ylabel("Error Relativo T/To")
plt.xlim(0.0,90.0)
plt.ylim(1.0,1.2)
plt.grid()
#Grafica 2 Divergencia de la integral
plt.figure(2)
plt.plot(x1,T)
plt.xlabel("Angulo")
plt.ylabel("Periodo")
plt.grid()
plt.show()
```

Listing 1: Programa Periodo.py

Las gráficas que resultaron son las siguientes:

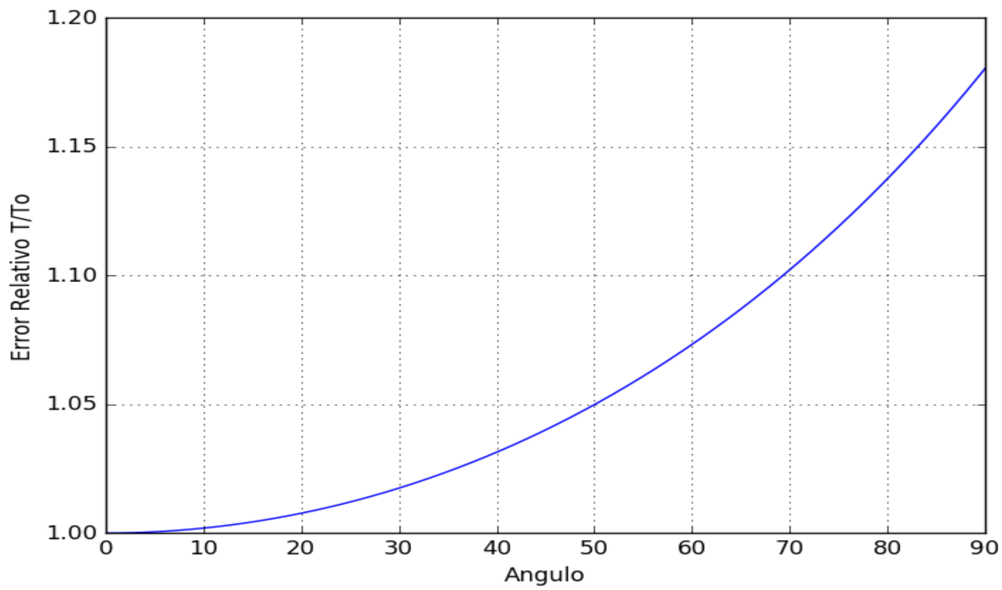

Figura 1: Angulo vs Error Relativo T/T_o

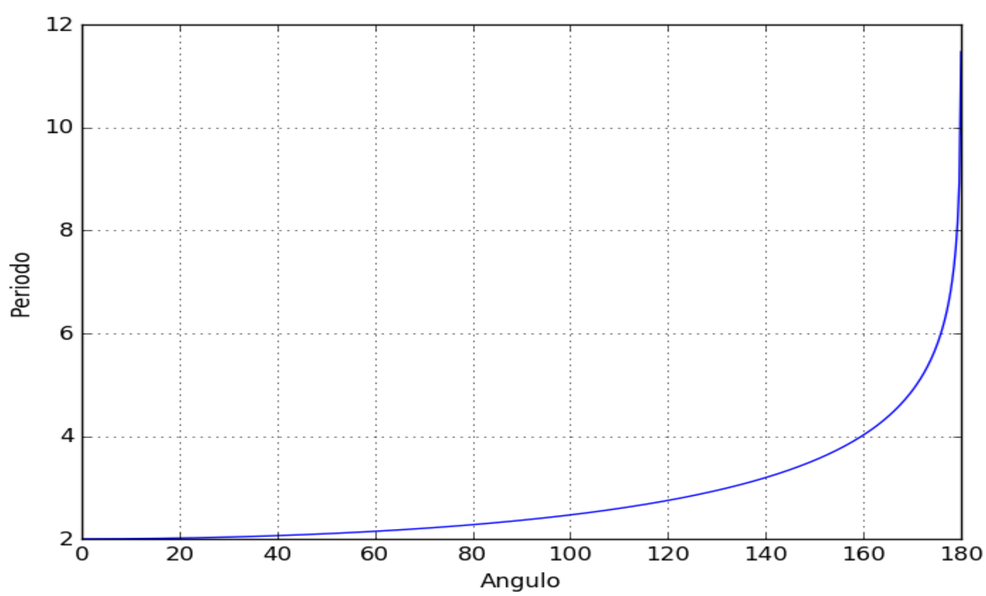

Figura 2: Divergencia del Periodo

Como Se puede observar, error aumenta conforme el ángulo crece, ademas el periodo tiende a infinito cuando el ángulo tiende a π.

Referencias

[1] Wikipedia,(2016) *Pendulum (mathematics)*. Recuperado de https://en.wikipedia.org/wiki/Pendulum_%28mathematics%29

[2] Scipy.org (2016) *Integration and ODEs*. Recuperado de http://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.quad.html #scipy.integrate.quad

[3] Lizárraga, C. (2016) *Actividad 6 (2016-1)*. Recuperado de http://computacional1.pbworks.com/w/page/105233358/Actividad%205%20 (2016-1)