Actividad 9: Aproximación al cálculo del periodo del péndulo

Valenzuela Chaparro Hugo de Jesús

8 de mayo de 2016

1. Integral elíptica

Sabemos, anteriormente, que para encontrar el periodo del péndulo para oscilaciones arbitrarias debemos evaluar la integral:
$T = 4\sqrt{\frac{\ell}{2g}} \int_0^{\theta_0} \frac{1}{\sqrt{\cos\theta - \cos\theta_0}} d\theta$
Esta integral diverge conforme θ_0 tiende a la vertical (es decir, si lo soltaramos desde la parte superior, en teoría debería quedarse ahí por siempre):
$\lim_{\theta_0 \to \pi} T = \infty$
así que un péndulo con la energía necesaria para ir verticalmente nunca llegará. La integral se puede escribir en terminos de integrales elípticas conmo:
$T = 4\sqrt{\frac{\ell}{g}} K(\sin^2 \frac{\theta_0}{2})$........(Eq. 3)
donde K es la integral elíptica incompleta de primer tipo definida por:
$K(k) = F(\frac{\pi}{2}, k) = \int_0^{\frac{\pi}{2}} \frac{1}{\sqrt{1 - k^2 \sin^2 u}} du$

2. Aproximación del periodo mediante serie de potencias

Dada (Eq. 3) y la solución polinomial de Legendre para la interal elíptica:
$K(k) = \frac{\pi}{2}\{1 + (\frac{1}{2})k^2 + (\frac{1 \cdot 3}{2 \cdot 4})^2 k^2 + ... + [\frac{(2n-1)!!}{(2n)!!}]^2 k^{2n} + ...\}$,
una solución exacta para el periodo del péndulo es:
$T = 2\pi\sqrt{\frac{\ell}{g}} \cdot \sum_{n=0}^{\infty} \left[\left(\frac{(2n)!}{(2^n \cdot n!)^2} \right)^2 \cdot \sin^{2n}(\frac{\theta_0}{2}) \right]$

2.1. Código en Python

A continuación se presenta un código en Python que da como resultado una gráfica del error de la aproximación por serie de potencias a la integral elíptica para encontrar el periodo en oscilaciones arbitrarias:

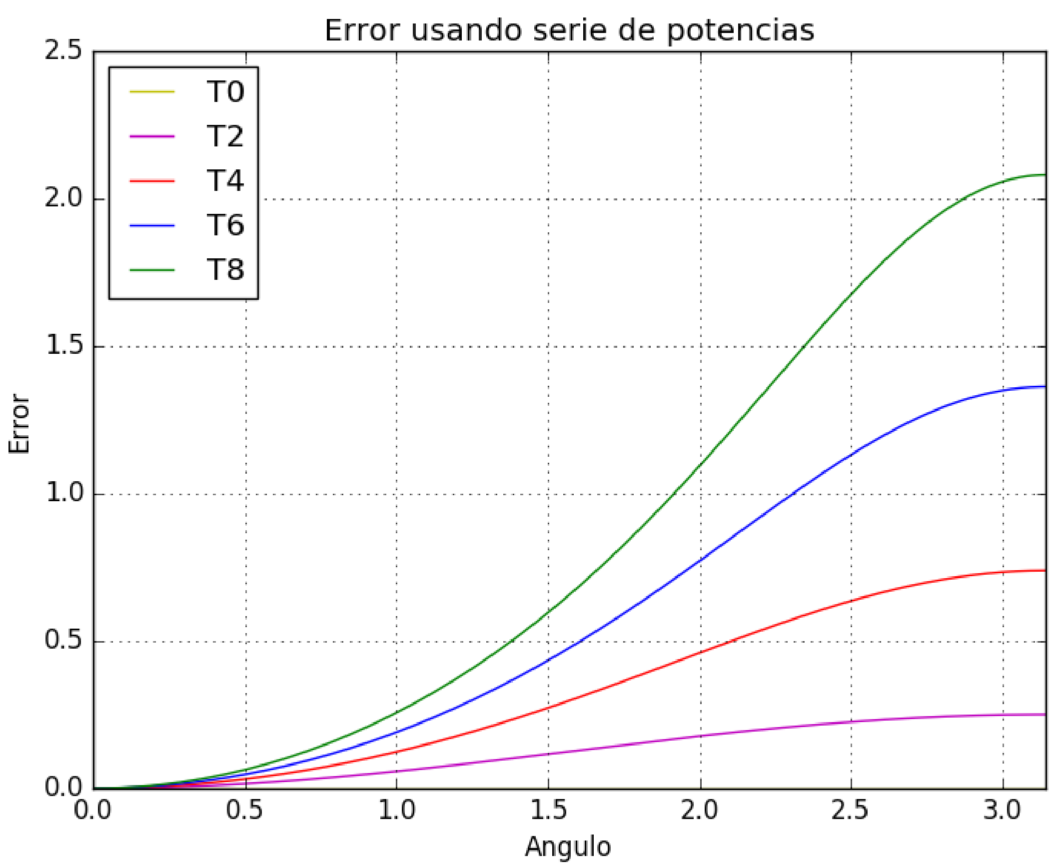
```
from scipy.integrate import quad
import numpy as np
```

```
import matplotlib.pyplot as plt
from math import factorial

g=9.8
#longitud de la cuerda
l=1

T0=2*np.pi*np.sqrt(l/g)

n=1000
e=0.0001

theta0=np.linspace(e,(np.pi)+e,n)

#Arreglos
S=[0 for i in range(n)]
TT=[0 for i in range(n)]
R=[0 for i in range(n)]
T=[0 for i in range(n)]
real0=[0 for i in range(n)]
real2=[0 for i in range(n)]
real4=[0 for i in range(n)]
real6=[0 for i in range(n)]
real8=[0 for i in range(n)]


M0=0

#loops

for i in range(M0):
    for j in range(0,n):
        F1=float(factorial(2*i))
        F2=float(((2**i)*(factorial(i)))**2)

        S[j]=np.sin(theta0[j]/2)**(2*i)
        TT[j]=((F1/F2)**2)*(S[j])
        R[j]=TT[j]+R[j]
        T[j]=R[j]*T0
        real0[j]=(T[j]/T0)
#**********************************************
M2=2
for i in range(M2):
    for j in range(0,n):
        F1=float(factorial(2*i))
        F2=float(((2**i)*(factorial(i)))**2)
```

```
        S[j]=np.sin(theta0[j]/2)**(2*i)
        TT[j]=((F1/F2)**2)*(S[j])
        R[j]=TT[j]+R[j]
        T[j]=R[j]*T0
        real2[j]=(T[j]/T0)-1
#**********************************************
M4=4
for i in range(M4):
    for j in range(0,n):
        F1=float(factorial(2*i))
        F2=float(((2**i)*(factorial(i)))**2)

        S[j]=np.sin(theta0[j]/2)**(2*i)
        TT[j]=((F1/F2)**2)*(S[j])
        R[j]=TT[j]+R[j]
        T[j]=R[j]*T0
        real4[j]=(T[j]/T0)-2
#**********************************************
M6=6
for i in range(M6):
    for j in range(0,n):
        F1=float(factorial(2*i))
        F2=float(((2**i)*(factorial(i)))**2)

        S[j]=np.sin(theta0[j]/2)**(2*i)
        TT[j]=((F1/F2)**2)*(S[j])
        R[j]=TT[j]+R[j]
        T[j]=R[j]*T0
        real6[j]=(T[j]/T0)-3
#*********************************************
M8=8
for i in range(M8):
    for j in range(0,n):
        F1=float(factorial(2*i))
        F2=float(((2**i)*(factorial(i)))**2)

        S[j]=np.sin(theta0[j]/2)**(2*i)
        TT[j]=((F1/F2)**2)*(S[j])
        R[j]=TT[j]+R[j]
        T[j]=R[j]*T0
        real8[j]=(T[j]/T0)-4

#********************************************

#Graficar
```

```
plt.plot(theta0, real0, 'y', label="T0")
plt.plot(theta0, real2, 'm', label="T2")
plt.plot(theta0, real4, 'r', label="T4")
plt.plot(theta0, real6, 'b', label="T6")
plt.plot(theta0, real8, 'g', label="T8")
plt.title('Error usando serie de potencias')
plt.grid()
plt.xlabel('Angulo')
plt.xlim(0,np.pi)
plt.ylabel('Error')
plt.legend(loc='best')
plt.show()
```

2.2. Gráfica



Se puede observar que al agregarle más teŕminos a la serie de potencias, el error va aumentando.

3. Aproximando con serie de Maclaurin

Otra solución se puede encontrar en las series de Maclaurin:

$\sin \frac{\theta_0}{2} = \frac{1}{2}\theta_0 - \frac{1}{48}\theta_0^3 + \frac{1}{3840}\theta_0^5 - \frac{1}{645120}\theta_0^7 + ...$

que se usa en la solución polinimial de Legendre. La serie de potencia que resulta entonces es:

$T = 2\pi\sqrt{\frac{\ell}{g}}\left(1 + \frac{1}{16}\theta_0^2 + \frac{11}{3072}\theta_0^4 + \frac{173}{737280}\theta_0^6 + \frac{22931}{1321205760}\theta_0^8 + \frac{1319183}{951268147200}\theta_0^{10} + \frac{233526463}{2009078326886400}\theta_0^{12} + ...\right)$