REVISTA DE CIENCIAS, Vol. 1, No. 2 de 2012.
ISSN 2256-3830

Solución Numerica a la Ecuación Diferencial No Lineal que describe las Oscilaciones del Péndulo Simple con Python

Numerical Solution to the differential no linear equation, that describes the oscillations of the mathematical pendulum with python

José Luis Arévalo Guancha[a*].
[a]*Universidad de Nariño*
Departamento de Fisica.
[b]*E-Mail:Josearegu@gmail.com*

Aceptado Noviembre 2011; Publicado en línea Agosto 2012

Resumen

Uno de los ejemplos mas utilizados en cursos de física para describir el m.a.s es la dinámica del péndulo simple para pequeñas amplitudes, tanto la parte experimental como la parte teórica son accesibles para los estudiantes de física e ingeniería, sin embargo el estudio de grandes amplitudes de oscilación es un tema de una mayor complejidad y por tanto no suele aparecer en los temarios de asignaturas de física impartidas. A pesar de ello, el estudio del péndulo simple y en particular la resolución de las ecuaciones diferenciales que describen el movimiento del sistema, proporcionan al estudiante una visión global de un conjunto de técnicas para resolver problemas lineales como en pequeñas amplitudes de oscilación y problemas no lineales en grandes oscilaciones.

En este trabajo como propósito principal se resolverá la ecuación diferencial no lineal que describe la oscilación del péndulo simple para grandes amplitudes así como la solución de la integral que describe su período utilizando como herramienta fundamental de solución la programación en lenguaje de alto nivel con python.

Igualmente el programa realizara la simulación del péndulo simple, sus respectivas gráficas de solución y los datos serán consignados en dos archivos de texto, dentro del programa se podrá variar tanto la longitud del péndulo como el valor de la aceleración de la gravedad y las condiciones iniciales para resolver la ecuación diferencial.

Finalmente se comparara y observara el comportamiento para ángulos pequeños del programa, con resultados obtenidos analíticamente a primera aproximación.

Palabras claves: Pendulo Simple, Python.

Abstract

One of the examples but used the dynamics is by the mathematical pendulum for small amplitudes, so much the experimental part as the theoretic part are accessible for the students of physics and engineering, however in courses of physics to describe the m.a.s the study of big amplitudes of oscillation is a theme of a bigger complexity and therefore often to appear in the agendas of subjects of study of physics given. In spite of it, the study of the simple pendulum and in particular the resolution of the differential equations that describe the motion of the system, give a comprehensive view of a set of techniques to solve linear problems to the student I eat in small amplitudes of oscillation and nonlinear problems in big oscillations.

It will be solved in this work like principal purpose the differential equation nonlinear that you describe the oscillation of the mathematical pendulum for big amplitudes as well as the solution of the comprehensive that your period using like fundamental tool of solution the programming in high-level language with python describes.

* Fisico UDENAR

Equally the program accomplish the simulation of the mathematical pendulum, his respective graphs of solution and the data will be consigned in two files of text, inside the program will be able to vary him so much the length of the pendulum like the value of the acceleration of gravity and the initial conditions to solve the differential equation.
Finally compare and observe the behavior for little angles of the program, with results obtained analytically to first approximation.

Keywords: Mathematical Pendulum, Python.

1. Introducción

En la actualidad la utilización de software en física es casi una practica habitual cuando se aborda el estudio cuantitativo de muchos problemas fisicos, ya que la realidad es tan compleja que el tratamiento matemático de la mayoría de los sistemas ofrece unas dificultades muy considerables.
El caso del péndulo simple ilustra perfectamente esta situación. se trata de un sistema idealizado de un solo grado de libertad, un objeto puntual de masa m y sostenido de un hilo de masa despreciable que puede describirse expresando en función del tiempo su distancia angular a una dirección de referencia, tal como la vertical en un determinado punto, el considerar este sistema como idealizado se restringe las observaciones a excluir la fricción presente (1).

Para un péndulo simple es mas conveniente describir la posición en términos del desplazamiento angular, $\theta(t)$. La ecuación de movimiento es (2) :

$$\frac{d^2\theta}{dt^2} + \frac{g}{L}\operatorname{sen}\theta = 0 \qquad (1)$$

Donde L es longitud del hilo y g aceleración de la gravedad. En la aproximación para ángulos pequeños, $\operatorname{sen}\theta \sim \theta$, la Ec.(1) se simplifica a :

$$\frac{d^2\theta}{dt^2} + \frac{g}{L}\theta = 0 \qquad (2)$$

Lo cual conduce a una solución de la forma:

$$\theta = \theta_0 cos(\omega + \varphi) \qquad (3)$$

Donde θ_0 representa la amplitud y φ es una fase inicial que puede ser elegida a voluntad para situar la elongación en un valor conveniente al tomar el origen de tiempos. Esta consideración simplificadora recae directamente en el período de oscilación del péndulo quedando este como:

$$T = (2\pi)/\omega = (2\pi)\sqrt{\frac{L}{g}} \qquad (4)$$

Sin embargo un estudio mas cuidadoso del tema, entrando a considerar la solución analítica de la ecuación de movimiento sin aproximación para ángulos pequeños y sabiendo de la experiencia que el movimiento es todavía periódico. En efecto, es posible obtener una expresión para el período de oscilación para cualquier amplitud sin resolver explícitamente $\theta(t)$ donde éste no es propiamente independiente de la amplitud inicial de la oscilación. Su período de oscilación viene dado por (3) :

$$T = 4\sqrt{\frac{L}{2g}}\int_0^{\theta_m}\frac{d\theta}{\sqrt{(\cos\theta - \cos\theta_m)}} \qquad (5)$$

Hay diferentes maneras de resolver esta integral, pero en este trabajo el propósito principal es el desarrollo de habilidades para el uso del lenguaje python. Python es un lenguaje de programación fácil de aprender y potente. Tiene eficaces estructuras de datos de alto nivel y una solución de programación orientada a objetos simple pero eficaz y lo mejor python y la extensa biblioteca estándar están disponible libremente.

2. Metodologia

La ecuación de movimiento del péndulo simple es una ecuación diferencial no lineal de segundo orden, Ec.(1). Una ecuación diferencial como esta constituye un problema de valor inicial cuando en un punto determinado se conoce el valor de la función y de sus derivadas hasta una unidad menos que el orden de la ecuación.
Para la solución del problema, en primer lugar se reduce la ecuación de segundo orden a un sistema de dos ecuaciones de primer orden, considerando una ecuación de la forma general (4) :

$\frac{d^2y}{dx^2} = f\left(x, y, \frac{dy}{dx}\right)$ (6) Realizando el cambio de variable: $Z_1 = y;$ $Z_2 = \frac{dy}{dx} = \frac{dZ_1}{dx}$

La ecuación puede expresarse como:

$$\frac{dZ_1}{dx} = Z_2 \qquad \frac{dZ_2}{dx} = f\left(x, Z_{1,} Z_2\right) \qquad (7)$$

En el caso de la ecuación del péndulo el sistema queda : $Z_1 = \theta\,;$ $Z_2 = \frac{d\theta}{dt} = \omega$ (8)
así que:

$$\frac{d\theta}{dt} = \omega\,; \qquad \theta(t) = \theta(0) = \theta_0 \qquad (9)$$

$$\frac{d\omega}{dt} = -\frac{g}{L}\sin\theta \qquad \omega(t) = \omega(0) = 0 \qquad (10)$$

Tenemos así dos ecuaciones diferenciales ordinarias lineales para resolver y debido a que pueden existir infinitas soluciones $\omega(t)$ es necesario especificar la condición $\omega(0) = 0$.
El código que me resuelve esto es:

```
#Solución numérica con odeint
def pendu_CF(inicial, t):
    theta = inicial[0]
    omega = inicial[1]
    f=[omega, -(g/L)*sin(theta)]
    return f
t = linspace(0,50,500)
inicial = [thetan, 0]
pendu_sol = odeint(pendu_CF, inicial, t)
pylab.plot(pendu_sol)
pylab.xlabel('Tiempo (s)')
pylab.ylabel('Angulo (rad)')
legend(('Ang vs t', 'W vs t'),0)
pylab.title(u'Sol.Pendulo Simple - Ao=%g\xb0 L=%g m y g=%g m/s^2' % (angulo,L,g) )
pylab.savefig('Solución Grafica Pendulo Simple')
pylab.grid(True)
pylab.show()
```

Como lo mencione es posible obtener una expresión para el período sin resolver explícitamente $\omega(t)$ asi; la energía total del péndulo es (5) :

$$E = \frac{1}{2} mL^2 \omega^2 - mgL\cos\theta \qquad (11)$$

La energía total es conservada e igual a : $E = -mgL\cos\theta_m$ donde θ_m es el angulo máximo.

De aquí se tiene :

$$\frac{1}{2} mL^2 \omega^2 - mgL\cos\theta = -mgL\cos\theta_m \qquad \text{ó} \qquad \omega^2 = \frac{2g}{L}(\cos\theta - \cos\theta_m) \qquad (12)$$

Como $\omega = \frac{d\theta}{dt}$

$$dt = \frac{d\theta}{\sqrt{\frac{2g}{L}(\cos\theta - \cos\theta_m)}} \qquad (13)$$

Como el período es el tiempo que le toma la partícula completar el ciclo, es decir :

$$T = \theta_0 \rightarrow 0 \rightarrow -\theta \rightarrow 0 \rightarrow \theta_0 \qquad T = 4(\theta_0 \rightarrow 0)$$

Así integrando ambos lados de la ecuación tengo:

$$T = 4\sqrt{\frac{L}{2g}} \int_0^{\theta_m} \frac{d\theta}{\sqrt{(\cos\theta - \cos\theta_m)}} \qquad (14)$$

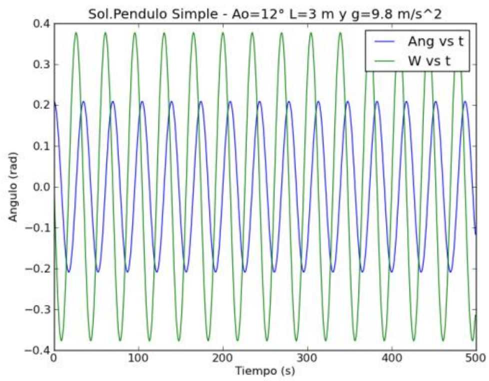
Una vez definida la ecuación integral. Para su solución se usara una de las librerías de python, Scypy. A partir de la herramienta Integrate. El código que me resuelve esto es:

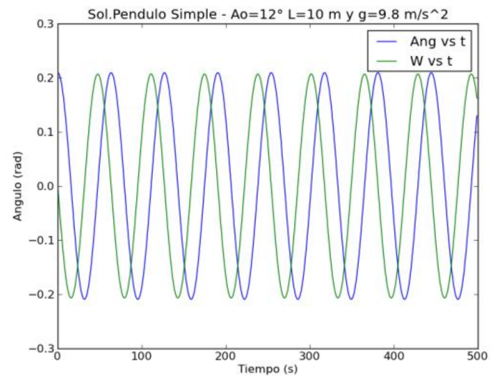
Solución al Periodo

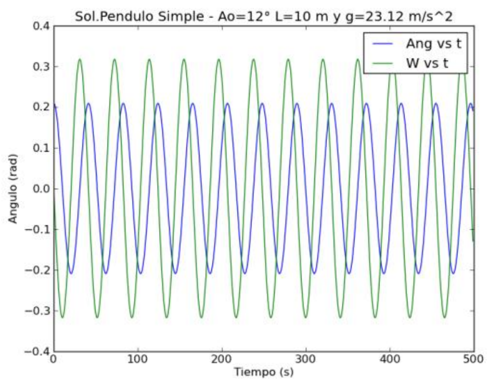
```
PA = []
aA = []
a = 0
da = 0.01
while (a < 0.999*pi) :
        a = a + da
        b=a*180/pi
        k= ((4.0/sqrt(2.0))*sqrt(L/g))
        P = integrate.quad(lambda x: (k*(1.0/sqrt(cos(x)-cos(a)))),0,a )
        print a, P[0]
        lineadedatosP='Angulo = %1.1f\xb0=%1.2f rad  PERIODO = %1.3f s \n' % (b,a,P[0])
        print >> datosP, lineadedatosP

        PA.append(P[0])
        aA.append(b)
plot(aA,PA)
pylab.xlabel('Angulo (grados)')
pylab.ylabel('Periodo (s)')
pylab.title(u'T vs Angulo - Ao=%g\xb0 L=%g m y g=%g m/s^2' % (angulo,L,g) )
pylab.savefig('Periodo Pendulo Simple')
pylab.grid(True)
```

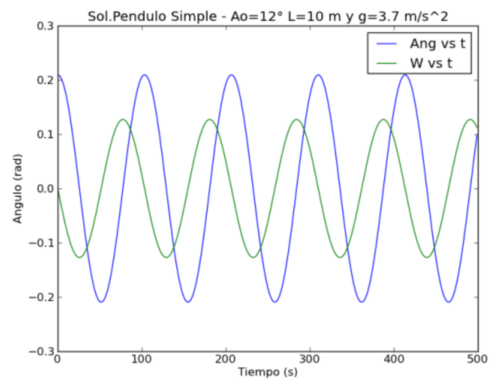
3. Resultados y Discusion

Aquí presento algunas gráficas que arroja el programa, donde Ao es el angulo inicial, L es la longitud del hilo del péndulo y g la aceleración de la gravedad. Para : Ao = 12 ° y diferentes valores de L y g.

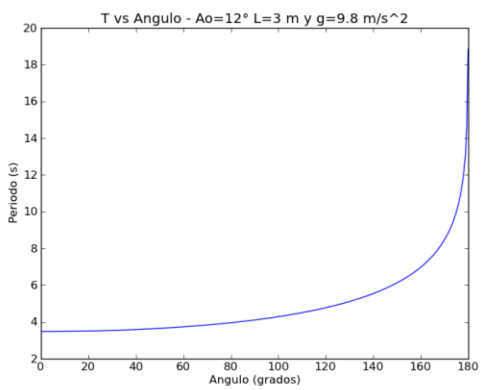


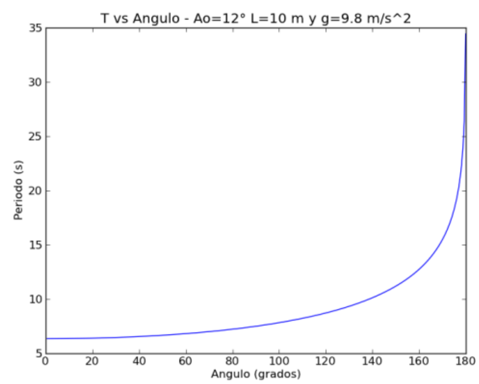


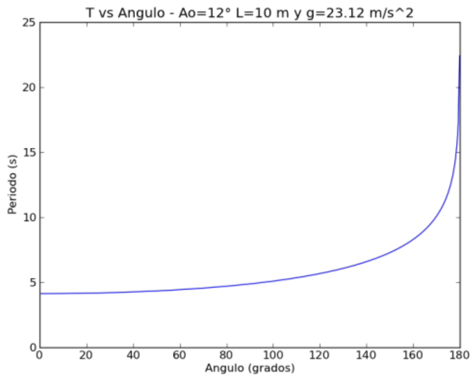


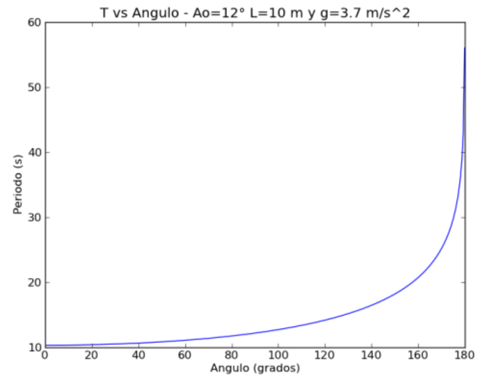


Estas son gráficas de solución para el Periodo Para : Ao = 12 ° y diferentes valores de L y g.

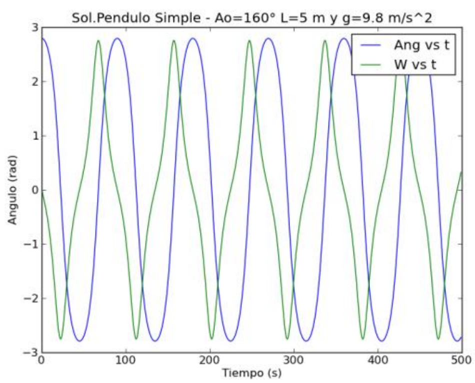


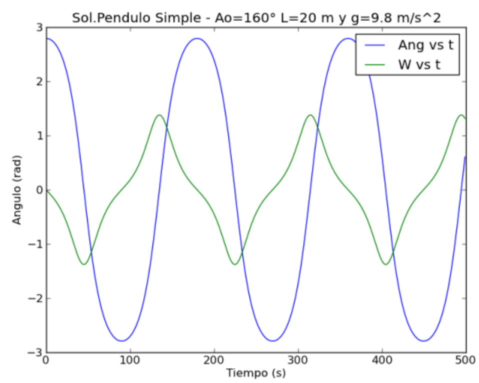


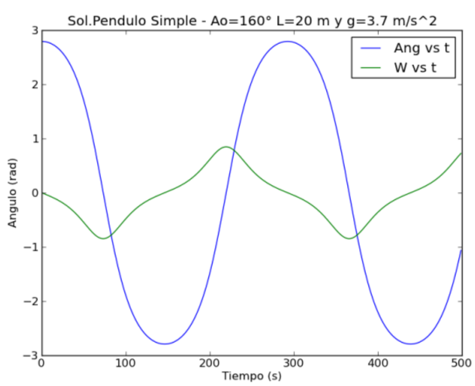


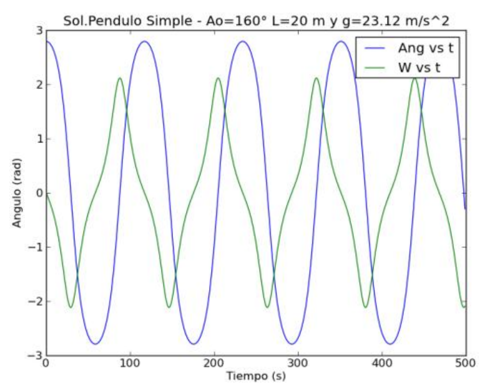


Estas son gráficas de solución de la ecuación de movimiento del péndulo simple; Para : Ao = 160 ° y diferentes valores de L y g.

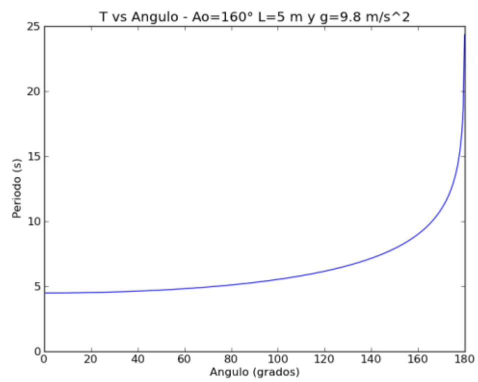


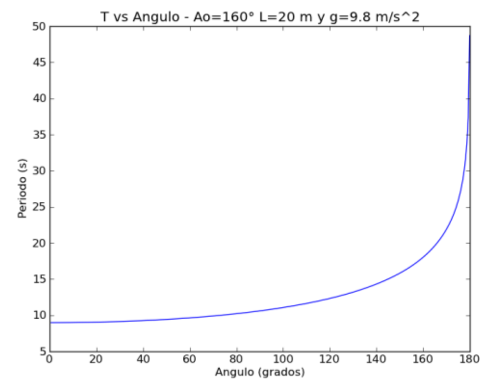


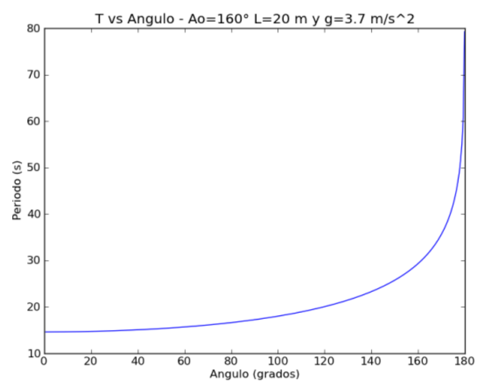


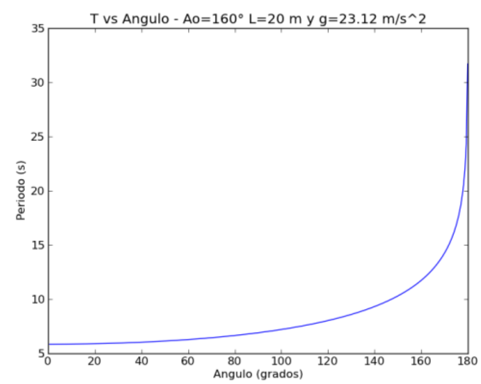


Estas son gráficas de solución para el Periodo Para : Ao = 160 ° y diferentes valores de L y g.









Al observar las gráficas de solución del péndulo simple se observa dos curvas, la curva verde es la representación de como varia $\omega_{vs}t$ y la curva azul nos representa como varia $\theta_{vs}t$.

Para mi análisis realice varias gráficas, en algunas cambiando la longitud del péndulo y manteniendo la gravedad original y en otras cambiando el valor de la gravedad manteniendo la misma longitud del péndulo para un angulo inicial inferior a 20° donde se cumple la aproximación de $\operatorname{sen}\theta \sim \theta$, en este caso fue de 16° e Igualmente para un angulo inicial de 160° donde el período no es propiamente independiente de la amplitud inicial de la oscilación.

4. Conclusiones

Cuando modificamos la longitud del péndulo simple manteniendo la aceleración de la gravedad, se observa que tanto la frecuencia, como de ω como la de θ, aumenta en forma considerable,y que la amplitud de ω aumenta en gran proporción, mientras que la de θ parece permanecer igual.
Por otro lado al cambiar el valor de la gravedad disminuirla con respecto a la gravedad de la tierra,para el caso he tomado la gravedad de marte 3.7 m/s^2 se observa una disminución en la frecuencia tanto de ω como de θ.

Estos resultados eran de esperarse, al observar el comportamiento para el angulo inicial de 12° se llega a la conclusión de que tanto ω como θ se ven afectados únicamente por los cambios en la gravedad y la longitud, así la ecuación para el período sin aproximación predice con exactitud el comportamiento para ángulos pequeños y muestra la dependencia que existe para ángulos mayores de la amplitud de oscilación.

Igualmente existen expresiones de aproximaciones que se acercan en diferente grado para resolver la ecuación del péndulo simple así como su período (6), en este ultimo su solución es expresada en términos de una integral elíptica y es resuelta numéricamente y analíticamente, en la mayoría de casos no hay solución analítica. Aquí se utiliza la programación como base para encontrar una solución al convertir una ecuación diferencial no lineal en dos ecuaciones diferenciales lineales y resolverla grafica y numericamente.

En este trabajo se ha realizado no solo la solución a la ecuación diferencial no lineal que describe las oscilaciones del péndulo simple si no que también se pretende motivar a fisicos e ingenieros en la búsqueda de nuevas alternativas que ofrece la programación para resolver problemas fisicos y matemáticos (7). Mediante el estudio de un sistema relativamente sencillo como es el péndulo simple, se ha pretendido presentar con python (8), un lenguaje prácticamente nuevo y con muy pocas lineas de código realizar simulaciones y soluciones numéricas.

5. Referencia Bibliografíca

1. **A.Serway, Raymond.** El Pendulo. *Fisica Para Ciencias e Ingenieria.* Madison : Thomson International, 2003, pág. 468.
2. **Finn, M. Alonso y E.J.** Pendulo Simple. *Fisica : Mecanica.* Caracas : Fondo Educativo Interamericano S.A, 1976, Vol. 1, pág. 367.
3. **Foundation, Wikimedia.** Wikipedia. *Pendulum (mathematics).* [En línea] Wikimedia Foundation Inc, Julio de 2011. http://en.wikipedia.org/wiki/Pendulum_%28mathematics%29.
4. **Bronson, R.** *Ecuaciones Diferenciales Modernas.* Mexico : Mc Graw-Hill, 1983.
5. **Rogan C.Jose, Muñoz G.Victor.** Aproximacion Pendulo Simple. [ed.] Facultad De Ciencias. *Programacion y Metodos Numericos.* Chile : Universidad De Chile, 2006, pág. 227.
6. *Approximation expressions for the large-angle period of a simple pendulum revisited.* **D. Amrani, P. Paradis and M. Beaudin.** 1, Mexico : Revista Mexicana De Fisica, Junio de 2008, Vol. 54.
7. **Dr.Sanders, P.David.** *Introducción a Python para el Cómputo Cientifico.* Mexico : UNAM, 2010.
8. **Hans Fangohr, Jacek Generowicks.** *Introduction to Python for Computational Science, A Beginner´s Guide.* Southampton : University de Southampton, 2011.

Apendice

Programa completo personal con simulacion y soluciones, entrega resultados graficos de la solucion a la ecuacion diferencial de segundo orden y resultados de la integral del periodo, con datos consignados en dos archivos de texto respectivamente para cada caso en la misma carpeta que corres el programa, los datos se almacenan hasta que termina la simulacion.

```
#--------------------------------------------------------------------------------
#PRORAMA DE SIMULACION DEL PENDULO SIMPLE SIN APROXIMACION
#REALIZADO POR : JOSE LUIS AREVALO GUANCHA
#GRUPO DE FISICA EXPERIMENTAL DE ALTAS ENERGIAS
#UDENAR 2011
#--------------------------------------------------------------------------------
from visual.graph import *
from math import *
from scipy import *
from scipy.integrate import odeint
from scipy import integrate
from numpy import linspace
import numpy
from pylab import plot, axis, show, legend
import pylab

pic1=gdisplay(x=0, y=380, height=400, width=1300, title = 'Angulo vs Tiempo', xtitle='t(s)')
pic2=gdisplay(x=400, y=0, height=380, width=900 , title = 'Angulo de Fase')
anguloplot=gcurve(gdisplay=pic1)
faseplot=gcurve(gdisplay=pic2)
velocidadplot=gcurve(gdisplay=pic1)
scene=display(x=0,y=0, height=380, width=400, title = 'Pendulo Simple')

# Archivo de datos (t,posicion angular,velocidad angular)
datos=open("DatosSolPendulo.dat","w")
datosP=open("DatosPeriodo.dat","w")
print "\n PROGRAMA DE SIMULACION SOLUCION DEL PENDULO SIMPLE SIN APROXIMACIONES \n"
print "   Realizado por Jose Luis Arevalo Guancha \n"

#Datos de Entrada

angulo = float(raw_input("Ingrese el ANGULO Inicial valores de 0 a 180 Grados: "))
thetan=math.pi*angulo/180.0
print " Angulo Inicial = %g radianes \n " % thetan

L = float(raw_input("Ingrese la LONGITUD de la cuerda en m : "))
print " Longitud = %g m \n " % L

g = float(raw_input("Ingrese el valor de la ACELERACION gravitatoria en m/s^2 : "))
print " Aceleracion Gravitatoria = %g m/s^2 \n " % g
print "NOTA 1: Observe la Solucion Grafica y cierra la ventana grafica para ver la solucion al Periodo\n"

#Solucion numerica con odeint

def pendu_CF(inicial, t):
    theta = inicial[0]
    omega = inicial[1]
    f=[omega, -(g/L)*sin(theta)]
    return f

t = linspace(0,50,500)
inicial = [thetan, 0]
pendu_sol = odeint(pendu_CF, inicial, t)
pylab.plot(pendu_sol)
pylab.xlabel('Tiempo (s)')
pylab.ylabel('Angulo (rad)')
legend(('Ang vs t', 'W vs t'),0)
pylab.title(u'Sol.Pendulo Simple - Ao=%g\xb0 L=%g m y g=%g m/s^2' % (angulo,L,g) )
pylab.savefig('Solucion Grafica Pendulo Simple')
pylab.grid(True)
pylab.show()
```

```
#Solucion al Periodo
print '\n FAVOR ESPERAR LA SOLUCION AL PERIODO DEL PENDULO MATEMATICO\n....................'
print "NOTA 2: Observe la Solucion Grafica y cierra la ventana grafica para ver la simulacion\n"
print "Los Datos de la solucion se consignaran en DatosPendulo.dat en la misma carpeta que corres el programa\n"

PA = []
aA = []
a = 0
da = 0.01
while (a < 0.999*pi) :
	a = a + da
	b=a*180/pi
	k= ((4.0/sqrt(2.0))*sqrt(L/g))
	P = integrate.quad(lambda x: (k*(1.0/sqrt(cos(x)-cos(a)))),0,a )
	print a, P[0]
	lineadedatosP='Angulo = %1.1f\xb0=%1.2f rad  PERIODO = %1.3f s \n' % (b,a,P[0])
	print >> datosP, lineadedatosP

	PA.append(P[0])
	aA.append(b)
plot(aA,PA)
pylab.xlabel('Angulo (grados)')
pylab.ylabel('Periodo (s)')
pylab.title(u'T vs Angulo - Ao=%g\xb0 L=%g m y g=%g m/s^2' % (angulo,L,g) )
pylab.savefig('Periodo Pendulo Simple')
pylab.grid(True)
pylab.show()

# Grafico del pendulo
ball=sphere(angulo=thetan,velAng=0.0,color=color.red,radius=0.6)
base=box(pos=(0,-30,-7),size=(7,0.5,7))
hasta=box(pos=(0,-13,-2),size=(0.7,30,0.6))
pitoco=cylinder(pos=(0,0,-2),radius=0.1,axis=(0,0,4),color=color.yellow)
ball.pos=(L*math.sin(ball.angulo),-L*math.cos(ball.angulo),0)
cuerda=cylinder(pos=(0,0,0),axis=ball.pos,color=color.white,radius=0.07)

DATOS1 = label(pos=(18,20,0))
DATOS2 = label(pos=(16,27,0))
DATOS3 = label(pos=(-29,13,0), text='L = %g m' % L)
DATOS4 = label(pos=(-25,20,0), text='g = %s m/s^2' % g)
Datos5 = label(pos=(-24,27,0), text=u'Ang.Inicial = %g\xb0' % angulo)

#Animacion y Desarrollo
dt=0.05
t=0

def force(angulo,velAng,t):
    result=-g/L*sin(angulo)
    return result

while (1) :
      rate(300,300,300,300)
      t=t+dt

#Algoritmo Utilizado.

      a1=force(ball.angulo,ball.velAng,t)

      ball.angulo=ball.angulo+ball.velAng*dt+a1*0.5*dt*dt
      ball.velAng=ball.velAng+a1*dt*0.5

      if (ball.angulo>math.pi):
          ball.angulo=ball.angulo-2*math.pi
      if (ball.angulo<-math.pi):
          ball.angulo=ball.angulo+2*math.pi

      ball.pos =(-L*math.sin(ball.angulo),-L*math.cos(ball.angulo),0)
      cuerda.axis=ball.pos

      DATOS1.text='Angulo = %1.2f rad' % ball.angulo
      DATOS2.text='W= %1.2f 1/rad' % ball.velAng
      lineadedatos='t=%1.2f s  Angulo = %1.4f rad  W = %1.3f \n' % (t,ball.angulo,ball.velAng)
      print >> datos, lineadedatos

#Graficas
      anguloplot.plot(pos=(t,ball.angulo),color=color.blue)
      velocidadplot.plot(pos=(t,ball.velAng),color=color.green)
      faseplot.plot(pos=(ball.angulo,ball.velAng))
datos.close()
show()
```