Programación

Práctica 3. Uso de expresiones

Michael González Harbour

José Javier Gutiérrez García

José Carlos Palencia Gutiérrez

José Ignacio Espeso Martínez

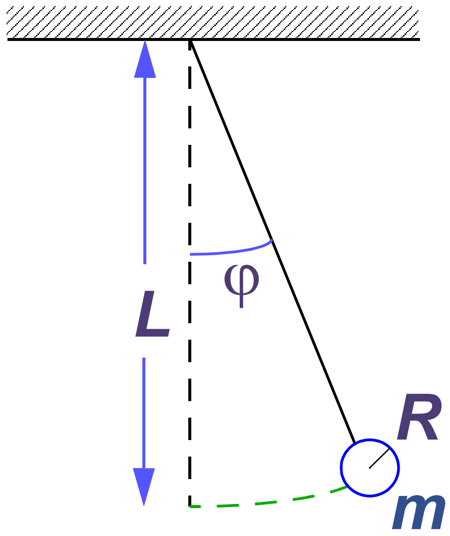
Adolfo Garandal Martín

Departamento de Ingeniería Informática y Electrónica

Práctica 3

Objetivo: Practicar con el uso de expresiones y comparar las formulaciones para un péndulo simple con pequeñas y grandes amplitudes

Descripción: Se dispone de un módulo con funciones para calcular la posición angular φ de un péndulo simple esférico de masa m, radio R, colgado de un hilo de longitud L



Usaremos unidades del sistema internacional (m, kg, s) y radianes para los ángulos

Pequeñas amplitudes

Para pequeñas amplitudes se puede aproximar el seno del ángulo por el propio ángulo y entonces la posición angular φ en función del tiempo t es (en radianes):

$$\varphi = \varphi_{max} \cos\left(\sqrt{\frac{g}{L}}t\right) \qquad (1)$$

Siendo φ_{max} la amplitud angular máxima y g la gravedad

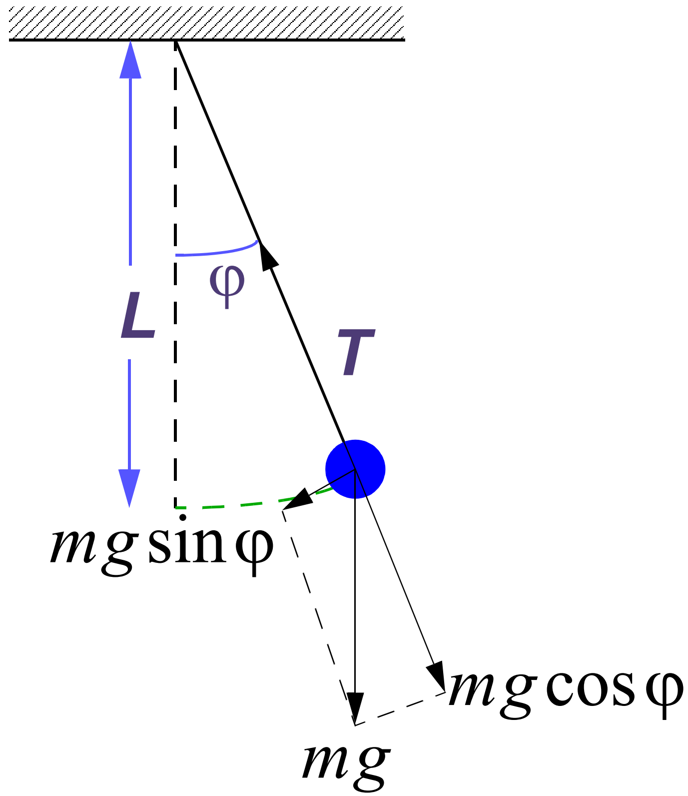
Referencia:

- https://www.acs.psu.edu/drussell/Demos/Pendulum/Pendula.html

Péndulo con amplitudes grandes

La fórmula anterior es válida para amplitudes pequeñas

- con amplitudes grandes es preciso resolver una ecuación diferencial
- o hacer una simulación, que es lo que haremos



La simulación se hace con las leyes de Newton, teniendo en cuenta las fuerzas aplicadas sobre la masa:

- el peso, con sentido hacia abajo y valor $m \cdot g$
- la tensión del hilo, T, con sentido hacia arriba en la dirección del hilo y valor igual a la componente del peso en la dirección del hilo

Ecuaciones con amplitudes grandes

La suma vectorial de las fuerzas aplicadas a la masa nos da la resultante y de ella podemos calcular la aceleración

Módulo de las aceleraciones angular (α, rad/s^2) y tangencial (a_t, m/s^2)

$$a_t = -g \sin\varphi$$

$$\alpha = \frac{a_t}{L} \qquad (2)$$

La aceleración radial es nula, pues se cancelan la tensión y la componente radial del peso

Ecuaciones del movimiento circular uniformemente acelerado

En intervalos muy pequeños podemos usar las fórmulas del movimiento uniformemente acelerado, ya que la aceleración apenas cambia

La posición angular (φ, rad) y la velocidad angular (ω, rad/s) son:

$$\begin{aligned} \varphi &= \varphi_{actual} + \omega_{actual} t + \frac{1}{2}\alpha t^2 \\ \omega &= \omega_{actual} + \alpha t \end{aligned} \qquad (3)$$

siendo ω_{actual} y φ_{actual} la velocidad angular y el ángulo al comienzo del intervalo

Módulo `pendulo.py`

El módulo `pendulo.py` dispone de las siguientes constantes:

- `L`: longitud del hilo, igual a 1.2m
- `PHI_MAX`: amplitud máxima, igual a 0.5 rad
- `G`: la gravedad terrestre

Dispone de la siguiente función para pequeñas amplitudes

- `phi_simple(tiempo)`: retorna la posición angular (rad) en función del tiempo que se pasa como parámetro
  - Se usa la ecuación (1)

pendulo
+L: float +PHI_MAX: float +G : float
+phi_simple(tiempo: float): float +avanza_simulacion(intervalo: float, phi_actual: float, w_actual: float): (float, float) +simula() +main()

Módulo `pendulo.py (cont.)`

Y la siguiente función para grandes amplitudes

- `avanza_simulacion(intervalo, phi_actual, w_actual)`: retorna la nueva posición y velocidad angular transcurrido un tiempo igual al `intervalo`
  - se usa la ecuación (2) para la aceleración, y la ecuación (3) para la posición y velocidad angulares (calcularlas en ese orden)

Por último, se dispone de la siguiente función

- `simula()`: realiza la simulación comparando gráficamente los resultados de pequeñas amplitudes con los de grandes amplitudes
  - esta función se da ya hecha

Se pide:

a) Crear las funciones `phi_simple()` y `avanza_simulacion()`

b) Crear una función `main()` que haga lo siguiente:

- Crea las variables de la posición y velocidad angular para la simulación con grandes amplitudes, con valores iniciales iguales a `PHI_MAX` y `0`, respectivamente
- Muestra en pantalla el valor del ángulo del péndulo en t=0 y al cabo de 0.01s, 0.02s, y 0.03s con las fórmulas para amplitudes pequeñas y para amplitudes grandes, para poder comparar visualmente los valores
- Invoca al método `simula()` para ver la gráfica que compara ambas formulaciones

Parte Avanzada: `pendulo_avanzado.py`

Hacer estos cambios en el módulo `pendulo.py`:

- Crear una nueva función similar a `avanza_simulacion()` que, además de la aceleración tangencial del peso, considere el efecto del rozamiento con el aire
  - Este efecto consiste en añadir una nueva aceleración tangencial, en m/s² cuyo módulo para bajas velocidades como la del péndulo está dado por la fórmula de Stokes

$$a_{rozam} = \left|\frac{6\pi\eta R v}{m}\right| \qquad (4)$$

  - siendo η la viscosidad del fluido
  - esta aceleración tiene signo contrario a v, velocidad lineal
  - Observar las relaciones entre aceleraciones y velocidades lineales y angulares

Parte avanzada (cont.)

- Crear las siguientes constantes
  - `RADIO`, que representa el valor R de la fórmula que es el radio del péndulo, de valor 0.1 m
  - `MASA`, es la masa de la esfera, de valor 0.2 kg
  - `ETA`, que representa el coeficiente de rozamiento η, de valor 0.002 en unidades kg/(s*m), y que está exagerado para ver mejor el efecto del rozamiento
- Modificar la función `simula()` para añadir una tercera gráfica que use la nueva función y permita comparar en la misma gráfica las ecuaciones para amplitudes simples, las de grandes amplitudes, y las de grandes amplitudes con rozamiento

Entrega

Entregar dos ficheros con:

- el código fuente del módulo conteniendo la parte obligatoria y la parte avanzada (si se ha hecho)
- un informe con:
  - Resultados de la ejecución del main()
  - Gráfica producida por el simulador
  - Conclusión sobre la comparación entre ambas formulaciones
  - Parte avanzada: Gráfica producida por el simulador
  - Parte avanzada: Conclusión sobre la comparación entre las tres formulaciones utilizadas

Apéndice

La constante π se puede obtener con

```
math.pi
```

El seno y el coseno de un ángulo `x` en radianes se puede obtener con

```
math.sin(x)
math.cos(x)
```

Parte avanzada: para copiar el signo de una variable `y` se puede usar esta función que retorna el valor de `x` pero con el signo de `y`:

```
math.copysign(x,y)
```