Tarea 05

fl.gomez10 at uniandes.edu.co

6 de marzo de 2019

Entrega

En un archivo comprimido .tar antes de finalizar la clase.

Ejercicio 1 (50 puntos)

En el archivo `ejercicio01.py` vamos a trabajar el problema de objetos celestes recorriendo el sistema solar. Este problema está basado en el vídeo de Programación Oientada a Objetos en Python.

Vamos calcular la órbita de la Tierra a partir de la posición y la velocidad inicial. Para simplificar el problema vamos a trabajar en dos dimensiones. No todos los cuerpos en el sistema solar se mueven en el mismo plano, pero para esta tarea no será relevante estudiar esto en detalle.

Generalmente en aplicaciones de física se trabaja con las unidades del sistema internacional (metro, kilogramo, segundo), sin embargo en este problema tenemos masas de varias veces millones de toneladas, distancias de miles de millones de kilómetros y tiempos muy largos (meses y años). Así que adoptaremos las siguientes unidades:

- Unidad astronómica de distancia (Distancia Tierra-Sol): 1a.u. $\sim 1{,}45 \times 10^8$km
- Unidad de tiempo: 1yr $\sim 3{,}15 \times 10^7$s
- Unidad de Masa (Masa Solar): $1M_\odot \sim 333000$ Masas terrestres $\sim 2 \times 10^{30}$kg

Con esto, la velocidad que tiene la Tierra en su traslación alrededor del Sol es de 6,28u.a./yr (perímetro del çírculo"sobre el periodo de órbita, unos 29,8km/s.)

La constante de gravitación G queda:

$$G = 39{,}48 \frac{\text{u.a.}^3}{\text{yr}^2 M_\odot}$$

a. Inicia atributos (5 pts)

Siguiendo el ejemplo del vídeo, vamos a crear una clase llamada CuerpoCeleste con los siguientes atributos:

- x, y: posición en unidades astronómicas
- v_x, v_y: velocidad en unidades astronómicas por año
- M: masa en masas solares
- F_x, F_y: Fuerza gravitacional que actúa sobre el objeto. En masas solares por unidad astronómica sobre año al cuadrado.

b. Calcula Fuerza (30 pts)

Implementar un método llamado "calculeFuerza" que calcule la fuerza de atracción gravitacional que experimenta el cuerpo celeste debido al Sol. Este método va a cambiar los valores de F_x y F_y.

La fuerza gravitacional que experimenta el cuerpo 1 debido al cuerpo 2 se puede calcular como:

$$\vec{F}_{1,2} = -G\frac{M_1 \times M_2}{R^2}\hat{r}_{2,1}$$

con $\hat{r}_{2,1}$ el vector unitario que va de 2 hacia 1.

Si elegimos al Sol como $M_2 = 1$ unidad de masa solar y lo colocamos en el origen de las coordenadas $\vec{r}_2 = 0$, queda:

$$\vec{F}_1 = -G\frac{M_1}{R^2}\hat{r}_1$$

Finalmente, la fuerza que experimenta el cuerpo celeste debido al Sol se puede esribir como:

$$\vec{F}_1 = (F_x, F_y)$$

c. Muevete(10 pts)

Implementar un método llamado "muevete(dt)"que dependa de un δt. Este método desplaza la partícula y calcula el cambio de velocidad según la fuerza experimentada.

d. Evoluciona e imprime (5 pts)

- Cree un cuerpo celeste llamado Tierra con $x = 1$, $y = 0$, $v_x = 0$, $v_y = 6{,}28$ y $M = 1/333000$
- Utilice un intervalo de tiempo $\delta t = 1/365$
- Haga el sistema evolucionar desde $t = 0$ hasta $t = {,}25$ (esto es la cuarta parte de un año) imprimiendo t, x e y.
- ¿Cuál es la posición final?

Ejercicio 2 (20 puntos)

Una vez tenga el ejercicio anterior completo, copielo en un nuevo archivo llamado “jercicio02.py”.

Cree un par de listas de posición $X =[\ ]$ y posición $Y =[\ ]$, en estas va a guardar la posición cada vez que ejecute ”tierra.muevete(dt)”

Después que el sistema haya terminado de evolucionar, incluya estas líneas en su código para generar una gráfica de la trayectoria.

```python
import matplotlib.pyplot as plt
fig = plt.figure(figsize=(5,5))
plt.plot(X,Y)
plt.savefig('orbita.png')
```

Listing 1: Python example

Con esto se genera un archivo de imágen con el segmento de órbita calculado.

a (10pt)

Grafique la trayectoria desde t=0 hasta t = 0.25. Modifique el nombre del arguivo de salida para que no sea reescrito en el siguiente item.

b (10 pt)

Grafique la trayectoria desde t = 0 hasta t = 0.75, Cambie el nombre del archivo de salida.

Ejercicio 03 (30 pts)

Se define en clase.

Anexo: Estructura del código.

Esta puede ser una guía de la estructura del código a implementar para los ejercicios 01 y 02.

```python

#!/usr/bin/python

class CuerpoCeleste:

    def __init__(self, x0,y0,Vx0, Vy0, M0):
        # Parte a. Inicia atributos

    def calculaFuerza(self):
        # Parte b. Calcula Fuerza (30 pts)
        G = 39.5
        M_sun = 1 # Solar Mass

        R = # Calcular R
```

```

        self.Fx =  # Calcular Fuerza en x
        self.Fy =  # Calcular Fuerza en y

    def muevete(self, dt):
        # Parte c. Muevete (10 pts)

Tierra = CuerpoCeleste( 1.0, 0.0, 6.28 , 0.0 , 1/333000)
Tierra.calculaFuerza()

t=0.0
DeltaT = 1 / 365 # one day

X = []
Y = []

while t < 0.25 :
    Tierra.calculaFuerza()
    Tierra.muevete(DeltaT)
    t += DeltaT
    # Ejercicio 2 (X.append(algo), Y.append(algo))

import matplotlib.pyplot as plt

fig = plt.figure(figsize=(5,5))
plt.plot(X,Y)
plt.savefig('orbita.png')
```

Listing 2: Estructura del ejercicio 2.

Ejercicio 3 (30 pts)

a (30 pts)

Una pelota de billar tiene una masa de 150g y un radio de 2,6cm. Asuma $g = 980\text{cm/s}^2$.

Puede modelar la fuerza de fricción como:

$$\vec{F} = -kmg\vec{v}$$

con $k = 0{,}0005$ dinas*(m/s)$^{-1}$)

Grafique la trayectoria para la pelota que está en el origen en $t = 0$, con dos velocidades distintas $v_{x0} = 10$cm/s y $v_0 = 30$cm/s (que no sea paralela a x o y)

b (20 extra pts)

La magnitud de la fuerza de colisión entre dos pelotas de billar puede modelarse como:

$$F = h \times (2R - |\vec{R_2} - \vec{R_1}|^3)$$

con $h \sim 10^5$dina/cm^3

Cree dos pelotas a = ($x = -2{,}5$, $y = 0$, $v_x = 20$, $v_y = 0$) b = (0, 0, 0, 0)

Grafique la trayectoria de las dos desde $t = 0$ hasta $t = 10$