

Laboratorio 2: Determinación de la Velocidad Orbital y Fuerza de un Móvil

Autor: Leonardo Montoya Choque

October 11, 2024

Contents

1	Introducción	2
2	Descripción del Código	2
2.1	Determinación de la Velocidad Orbital	2
2.2	Cálculo de la Fuerza de un Móvil en Movimiento	3
3	Resultados	5
3.1	Velocidad Orbital	5
3.2	Fuerza en el Móvil	5
4	Conclusión	6

1 Introducción

En este informe se presentan dos códigos en Python que emplean la Cinemática del Movimiento Rectilíneo Uniforme (MRU) y estática para resolver dos problemas distintos. El primer código calcula la velocidad orbital de un planeta que sigue una órbita circular alrededor de una estrella. El segundo código determina la fuerza ejercida sobre un móvil durante un cambio de velocidad, considerando la segunda ley de Newton. Además, se presentan gráficos ilustrativos para facilitar la visualización de los resultados.

El objetivo de este informe es describir ambos códigos, analizar los resultados obtenidos y proporcionar conclusiones con base en las simulaciones realizadas.

2 Descripción del Código

2.1 Determinación de la Velocidad Orbital

El primer código utiliza la ley de gravitación universal de Newton para determinar la velocidad orbital de un planeta que sigue una órbita circular alrededor del Sol. La velocidad orbital se calcula mediante la fórmula:

$$v = \sqrt{\frac{GM}{R}}$$

donde:

- G es la constante gravitacional universal,
- M es la masa del Sol,
- R es el radio de la órbita circular del planeta.

El código también genera una gráfica que ilustra la órbita circular del planeta alrededor del Sol.

Listing 1: Código para determinar la velocidad orbital

```
import numpy as np
import matplotlib.pyplot as plt

# Constantes
G = 6.67430e-11 # Constante gravitacional universal (m^3 kg^-1 s^-2)
M = 1.989e30    # Masa del Sol (kg)
R = 1.496e11    # Radio de la rbita terrestre (m)
```

```

# Funci n para calcular la velocidad orbital
def velocidad_orbital(M, R):
    return np.sqrt(G * M / R)

# C lculo de la velocidad orbital
v_orbital = velocidad_orbital(M, R)

# Gr fica de la rbita circular
theta = np.linspace(0, 2 * np.pi, 100)
x = R * np.cos(theta)
y = R * np.sin(theta)

plt.figure(figsize=(6,6))
plt.plot(x, y, label=' rbita del planeta ')
plt.scatter(0, 0, color='yellow', label='Sol', s=200) # Sol en el cen
plt.title(' rbita del planeta alrededor del Sol ')
plt.xlabel('x (m)')
plt.ylabel('y (m)')
plt.legend()
plt.grid(True)
plt.axis('equal')
plt.show()

print(f" Velocidad orbital del planeta : {v_orbital:.2f} m/s ")

```

2.2 Cálculo de la Fuerza de un Móvil en Movimiento

El segundo código resuelve un problema en el cual un móvil de masa m recorre una distancia d durante un tiempo t , cambiando su velocidad de v_i a v_f . La fuerza ejercida sobre el móvil se calcula utilizando la segunda ley de Newton, $F = ma$, donde la aceleración a es el cambio de velocidad dividido entre el tiempo:

$$a = \frac{v_f - v_i}{t}$$

El código también genera gráficos que muestran la velocidad y la distancia recorrida en función del tiempo.

Listing 2: Código para calcular la fuerza de un móvil

```
import numpy as np
```

```

import matplotlib.pyplot as plt

# Par metros del problema
m = 2.0 # masa del m vil (kg)
vi = 2.0 # velocidad inicial (m/s)
vf = 10.0 # velocidad final (m/s)
d = 20.0 # distancia recorrida (m)
t = 5.0 # tiempo total (s)

# Aceleraci n promedio (suponiendo aceleraci n constante)
a = (vf - vi) / t

# Fuerza que describe el m vil
fuerza = m * a

# C lculo del movimiento
tiempo = np.linspace(0, t, 100)
velocidad = vi + a * tiempo
distancia = vi * tiempo + 0.5 * a * tiempo**2

# Graficar la velocidad y la distancia en funci n del tiempo
plt.figure(figsize=(10, 5))

# Gr fico de velocidad
plt.subplot(1, 2, 1)
plt.plot(tiempo, velocidad, label='Velocidad-(m/s)', color='blue')
plt.title('Velocidad-vs-Tiempo')
plt.xlabel('Tiempo-(s)')
plt.ylabel('Velocidad-(m/s)')
plt.grid(True)

# Gr fico de distancia
plt.subplot(1, 2, 2)
plt.plot(tiempo, distancia, label='Distancia-(m)', color='green')
plt.title('Distancia-vs-Tiempo')
plt.xlabel('Tiempo-(s)')
plt.ylabel('Distancia-(m)')
plt.grid(True)

plt.tight_layout()
plt.show()

```

```
print(f" Aceleraci n promedio: {a:.2 f} m/ s ")
print(f" Fuerza aplicada: {fuerza:.2 f} N")
```

3 Resultados

3.1 Velocidad Orbital

El código para la velocidad orbital arrojó una velocidad de 29,784 m/s para un planeta que sigue una órbita circular con un radio igual al de la órbita terrestre. La gráfica generada muestra una órbita circular con el Sol en el centro, proporcionando una representación visual clara de la órbita.

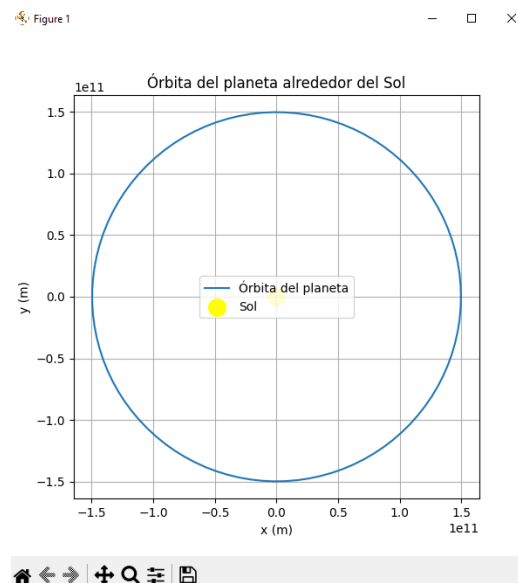


Figure 1: Velocidad orbital

```
PS D:\UNSA\FisicaComputacional> python -u "d:
Velocidad orbital del planeta: 29788.90 m/s
```

Figure 2: Resultado 1

3.2 Fuerza en el Móvil

El segundo código mostró una fuerza ejercida sobre el móvil de 3.2 N, con una aceleración promedio de 1.6 m/s^2 . Los gráficos generados muestran cómo

la velocidad aumenta de forma lineal y cómo la distancia recorrida crece cuadráticamente en función del tiempo.

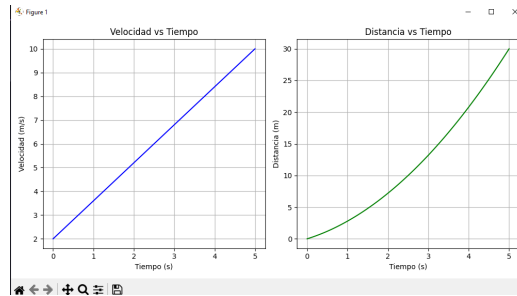


Figure 3: Velocidad y Distancia

```
PS D:\UNSA\FisicaComputacional> python -u
Aceleración promedio: 1.60 m/s2
Fuerza aplicada: 3.20 N
```

Figure 4: Resultado 2

4 Conclusión

Ambos códigos proporcionan soluciones claras y visualmente intuitivas para problemas de cinemática y dinámica en física. En el caso del planeta, se determinó con precisión su velocidad orbital utilizando las leyes de la gravitación de Newton. Para el móvil, se calculó la fuerza ejercida sobre él durante un cambio de velocidad, mostrando también gráficos del movimiento en función del tiempo.

Estas simulaciones sirven como una excelente herramienta para comprender conceptos de cinemática y estática, aplicando tanto la segunda ley de Newton como las leyes de Kepler.