Informe Problema 3 Cuerpos

Aaron Sarmiento
Robert Burgos
Nelson Rincon
Sergio Heredia
Laura García
Miguel Thómas

Materia:
Sistemas complejos

Universidad Sergio Arboleda

Escuela de Ciencias Exactas e Ingeniería
Ciencias de la Computación e Inteligencia Artificial

2024

Introducción

El problema de los tres cuerpos es un tema fundamental en la física, con implicaciones que van más allá de la mecánica clásica. Este problema trata de predecir el movimiento de tres cuerpos que interactúan gravitacionalmente entre sí. Aunque en un principio puede parecer un problema simple, rápidamente se revela a algo altamente complejo y no lineal. La importancia de este problema radica en la relevancia para comprender sistemas complejos. A medida que aumenta el número de cuerpos y las interacciones entre ellos, la dinámica del sistema se vuelve exponencialmente más difícil de predecir.

El objetivo de este informe es analizar y comprender el problema de los tres cuerpos desde una perspectiva computacional, utilizando el lenguaje de programación Python para simular y visualizar las interacciones entre los cuerpos celestes. Exploraremos las trayectorias y comportamientos emergentes de sistemas de dos, tres y cuatro cuerpos bajo la influencia gravitatoria mutua. Al hacerlo, esperamos obtener una comprensión más profunda de la complejidad de estos sistemas y su relevancia en diversos campos de la ciencia.

Estructura del Código:

El código implementado para la simulación del problema de los de tres cuerpos se organiza principalmente en la evolución del sistema y la visualización de las trayectorias de los cuerpos en el espacio tridimensional.

```python
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib.animation import FuncAnimation
plt.style.use('dark_background')

# Masas de los planetas
m_1 = 10
m_2 = 20
m_3 = 30

# Coordenadas iniciales de los planetas
p1_start = np.array([-10, 10, -11])
v1_start = np.array([-3, 0, 0])

p2_start = np.array([0, 0, 0])
v2_start = np.array([0, 0, 0])

p3_start = np.array([10, 10, 12])
v3_start = np.array([3, 0, 0])
```

Este bloque de código hace la definición de masas y posiciones iniciales. Donde m_1, m_2, m_3 son las masas de los tres cuerpos celestes en el sistema. En este caso, se definen valores arbitrarios para cada masa. Las variables de p1_start, p2_start, p3_start son las coordenadas iniciales de posición para los tres cuerpos en el espacio tridimensional (x, y, z). Estas coordenadas se representan como matrices de NumPy. Mientras que v1_start, v2_start, v3_start son las velocidades iniciales de los tres cuerpos en las tres dimensiones (x, y, z).

```python
def accelerations(p1, p2, p3):
    # Cálculo de las derivadas según las leyes de Newton
    planet_1_dv = -9.8 * m_2 * (p1 - p2) / (np.sqrt(((p1 - p2)**2).sum()) ** 3) - \
                   9.8 * m_3 * (p1 - p3) / (np.sqrt(((p1 - p3)**2).sum()) ** 3)

    planet_2_dv = -9.8 * m_3 * (p2 - p3) / (np.sqrt(((p2 - p3)**2).sum()) ** 3) - \
                   9.8 * m_1 * (p2 - p1) / (np.sqrt(((p2 - p1)**2).sum()) ** 3)

    planet_3_dv = -9.8 * m_1 * (p3 - p1) / (np.sqrt(((p3 - p1)**2).sum()) ** 3) - \
                   9.8 * m_2 * (p3 - p2) / (np.sqrt(((p3 - p2)**2).sum()) ** 3)

    return planet_1_dv, planet_2_dv, planet_3_dv
```

Esta parte del código define una función llamada accelerations(p1, p2, p3) que calcula las aceleraciones de los tres cuerpos celestes en el sistema, donde p1, p2, p3 representan las posiciones actuales de los tres cuerpos celestes en el espacio tridimensional (coordenadas x, y, z).

Cálculo de aceleraciones:

1. Se aplican las leyes de Newton para determinar las aceleraciones experimentadas por cada cuerpo celeste debido a su interacción gravitatoria con los otros dos. La fuerza gravitatoria entre dos objetos se establece como la masa de uno de los objetos multiplicada por la aceleración gravitacional, la cual es inversamente proporcional al cuadrado de la distancia entre ellos.

2. Para cada cuerpo celeste en el sistema (planet_1, planet_2, planet_3), se calcula la derivada de su posición con respecto al tiempo, lo que equivale a la aceleración. Este cálculo se realiza mediante la aplicación de la fórmula de la ley de gravitación universal de Newton.

3. Las derivadas se calculan para cada dimensión (x, y, z) de manera individual. Esto implica considerar la posición de cada cuerpo celeste, así como las posiciones y masas de los otros dos cuerpos en el sistema.

4. Las aceleraciones resultantes para los tres cuerpos celestes se retornan como una tupla (planet_1_dv, planet_2_dv, planet_3_dv).

```python
# Parámetros
delta_t = 0.001
steps = 70000  # Reducido el número de pasos

# Inicializar arrays de trayectoria y velocidad
p1 = np.zeros((steps, 3))
v1 = np.zeros((steps, 3))

p2 = np.zeros((steps, 3))
v2 = np.zeros((steps, 3))

p3 = np.zeros((steps, 3))
v3 = np.zeros((steps, 3))

# Asignar puntos iniciales y velocidades
p1[0], p2[0], p3[0] = p1_start, p2_start, p3_start
v1[0], v2[0], v3[0] = v1_start, v2_start, v3_start
```

El código establece los parámetros fundamentales para la simulación del sistema de tres cuerpos. En primer lugar, se define delta_t, que representa el paso de tiempo entre cada iteración de la simulación, permitiendo ajustar la precisión del modelo. Luego, steps determina el número total de pasos de tiempo para la simulación, afectando tanto la duración como la precisión de los resultados.

Posteriormente, se inicializan matrices NumPy para registrar las posiciones (p1, p2, p3) y velocidades (v1, v2, v3) de los cuerpos celestes a lo largo de la simulación. Cada matriz tiene dimensiones (steps, 3), donde 'steps' representa la cantidad total de pasos de tiempo y '3' denota las tres dimensiones espaciales (x, y, z). Las posiciones y velocidades iniciales de los cuerpos celestes se asignan a las primeras filas de estas matrices, obtenidas de las variables previamente definidas: p1_start, p2_start, p3_start, v1_start, v2_start y v3_start.

```python
# Evolución del sistema
for i in range(steps - 1):
    dv1, dv2, dv3 = accelerations(p1[i], p2[i], p3[i])

    v1[i + 1] = v1[i] + dv1 * delta_t
    v2[i + 1] = v2[i] + dv2 * delta_t
    v3[i + 1] = v3[i] + dv3 * delta_t

    p1[i + 1] = p1[i] + v1[i] * delta_t
    p2[i + 1] = p2[i] + v2[i] * delta_t
    p3[i + 1] = p3[i] + v3[i] * delta_t
```

Con estos valores iniciales y parámetros establecidos, se procede a calcular la evolución del sistema en un bucle for, utilizando las aceleraciones previamente calculadas mediante la función accelerations(p1, p2, p3). Este bucle actualiza las posiciones y velocidades en cada paso de tiempo, registrando la trayectoria de cada cuerpo celeste.

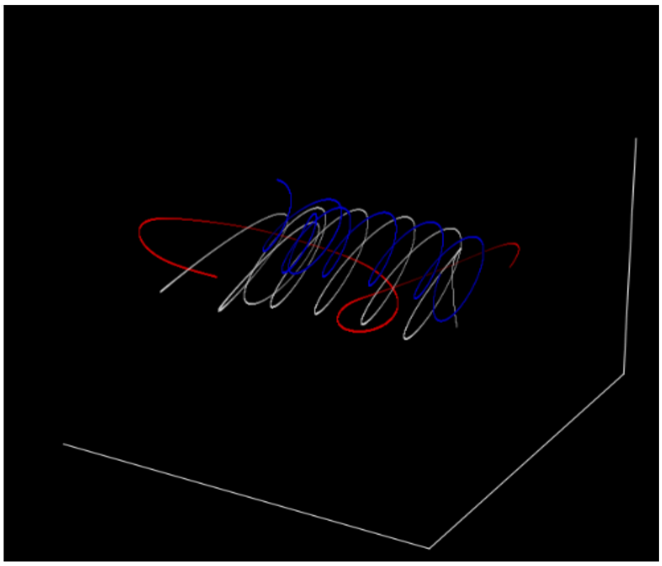
```python
# Crear la figura y el subplot
fig = plt.figure(figsize=(8, 8))
ax = fig.add_subplot(111, projection='3d')
plt.gca().patch.set_facecolor('black')

# Función para inicializar la animación
def init():
    ax.clear()
    ax.set_xticks([]), ax.set_yticks([]), ax.set_zticks([])
    ax.w_xaxis.set_pane_color((0.0, 0.0, 0.0, 1.0)), ax.w_yaxis.set_pane_color((0.0, 0.0, 0.0, 1.0)), ax.w_zaxis.set_pane_color((0.0, 0.0, 0.0, 1.0))
    return []

# Función para actualizar la animación en cada frame
def update(frame):
    ax.clear()
    ax.set_xticks([]), ax.set_yticks([]), ax.set_zticks([])
    ax.w_xaxis.set_pane_color((0.0, 0.0, 0.0, 1.0)), ax.w_yaxis.set_pane_color((0.0, 0.0, 0.0, 1.0)), ax.w_zaxis.set_pane_color((0.0, 0.0, 0.0, 1.0))
    ax.plot(p1[:frame*100, 0], p1[:frame*100, 1], p1[:frame*100, 2], '^', color='red', lw=0.05, markersize=0.01, alpha=0.5)
    ax.plot(p2[:frame*100, 0], p2[:frame*100, 1], p2[:frame*100, 2], '^', color='white', lw=0.05, markersize=0.01, alpha=0.5)
    ax.plot(p3[:frame*100, 0], p3[:frame*100, 1], p3[:frame*100, 2], '^', color='blue', lw=0.05, markersize=0.01, alpha=0.5)
    return []

# Crear la animación
ani = FuncAnimation(fig, update, frames=len(p1) // 100, init_func=init, blit=True)

# Guardar la animación como un archivo de video
ani.save('three_body_animation.mp4', fps=30, extra_args=['-vcodec', 'libx264'])

plt.show()
```

Finalmente, se configura y genera una animación tridimensional de las trayectorias de los cuerpos celestes, la cual se guarda como un archivo de vídeo para su posterior análisis y visualización. Este proceso constituye la base de la simulación del problema de los tres cuerpos, capturando la evolución dinámica del sistema a lo largo del tiempo.

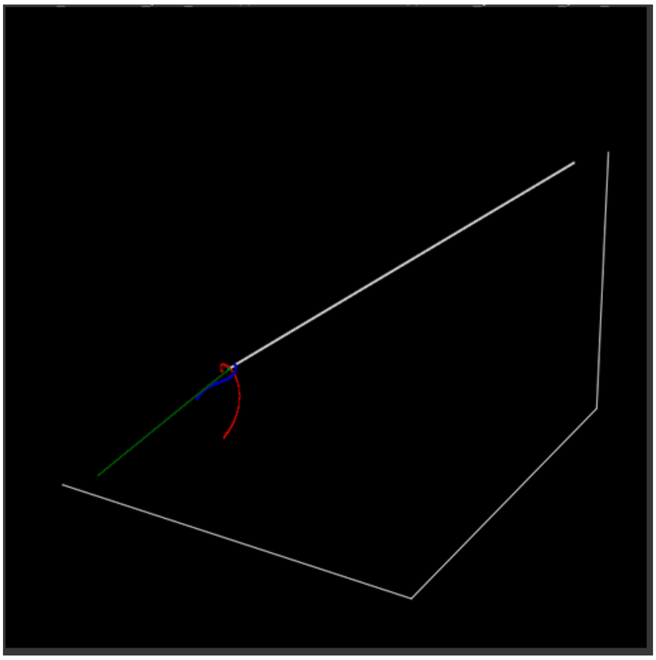
Esto mismo se trabaja con uno, dos y tres cuerpos ya que simplemente se añade el cuerpo o se quita y se calculan sus aceleraciones para así ver su comportamiento.

Visualización:

La visualización de los tres cuerpos en la simulación proporciona una representación gráfica de cómo interactúan entre sí en el espacio tridimensional. Cuando se ejecuta la simulación, se observa que los dos cuerpos blancos y azul, que representan masas más grandes, exhiben comportamientos similares en su movimiento, pero con pequeñas variaciones en sus trayectorias debido a las interacciones gravitatorias mutuas.



Por otro lado, cuando agregamos otro cuerpo , en este caso el cuerpo rojo, el cual tiene una masa menor, presenta una trayectoria diferente y más compleja en comparación con los otros dos. Este cuerpo tiende a demorarse más en completar ciertos movimientos, como dar la vuelta, lo que sugiere una interacción gravitatoria más significativa o una masa diferente que influye en su comportamiento dinámico.



Cuando se añade un cuarto cuerpo al sistema, se observan cambios significativos en el comportamiento de la simulación. Ahora, además de los cuerpos rojo, blanco y azul, hay un cuarto cuerpo verde. A medida que la simulación avanza, se percibe que los cuerpos blanco y verde siguen trayectorias más o menos rectas, mientras que los cuerpos azul y rojo comienzan a enroscarse entre sí. Esta dinámica sugiere una interacción gravitatoria más compleja y una influencia mutua entre los cuerpos. Mientras que los cuerpos blanco y verde parecen seguir trayectorias más estables, los cuerpos azul y rojo muestran un comportamiento más caótico, en el cual se enroscan y cambian su orientación en el espacio tridimensional. Este fenómeno ilustra cómo la introducción de un cuarto cuerpo introduce nuevas complejidades en el sistema, afectando las interacciones gravitatorias y generando trayectorias más variadas y dinámicas entre los cuerpos celestes.

Conclusiones:

- Se puede observar cómo las trayectorias de los cuerpos celestes cambian de manera impredecible y no periódica a lo largo del tiempo. Incluso pequeñas variaciones en las condiciones iniciales pueden conducir a cambios significativos en el movimiento de los cuerpos.

- Las masas y posiciones iniciales de los cuerpos celestes influyen en sus trayectorias y comportamientos emergentes. Las diferencias en masa y posición pueden generar dinámicas complejas y no lineales en el sistema.

- A medida que el sistema evoluciona, emergen patrones de comportamiento complejos y no lineales. Las interacciones gravitatorias entre los cuerpos generan dinámicas que no pueden preverse fácilmente, lo que demuestra la diversidad del problema de los tres cuerpos.