

Simulador de Órbita de un Cuerpo en un Campo Gravitatorio

Objetivo del Programa: El propósito del programa es desarrollar un simulador que modele la órbita de un satélite alrededor de un cuerpo de grandes dimensiones (por ejemplo, un planeta o una estrella) utilizando las leyes de la física gravitacional de Newton y de órbita de Kepler, el objetivo es analizar cómo varía la trayectoria del satélite según sus condiciones iniciales (distancia y velocidad) y visualizar este resultado mediante una gráfica en 2D.

Estructura y Organización del Código: El código se divide en las siguientes partes fundamentales:

1. **Entrada de Datos:** Se solicita al usuario que introduzca las distintas variables; la masa del cuerpo central, la masa del satélite, la distancia inicial del satélite y su velocidad tangencial.
2. **Inicialización de Variables:** Se definen las condiciones iniciales de posición y velocidad del satélite como unidades vectoriales.
3. **Simulación por Integración Numérica:** Utilizando el método de Euler se calcula la trayectoria del satélite a lo largo del tiempo. Esto se realiza en un bucle que itera sobre pequeños intervalos de tiempo (dt), aplicando la Ley de Gravitación Universal obtenemos la aceleración.
4. **Cálculo de Energías:** Al finalizar la simulación, se calculan las distintas energías; cinética, potencial y mecánica total del sistema, para analizar la estabilidad de la órbita.
5. **Visualización:** Finalmente, se representa mediante un gráfico la trayectoria orbital y se marca la posición del cuerpo central.

Enfoque de Desarrollo: Cada parte se desarrolla por separado para facilitar la validación de resultados parciales (por ejemplo, para verificar si la trayectoria es cerrada o si la energía total se conserva). Se usará numpy para el manejo de vectores y cálculos numéricos, y matplotlib para la visualizar los gráficos.

Retos Previstos: Uno de los principales retos es mantener la estabilidad numérica de la simulación, ya que un valor de dt inadecuado puede causar errores que se van acumulando significativos. Otro desafío es lograr que la trayectoria se asemeje a una órbita realista (elíptica), ajustando correctamente las condiciones iniciales presentadas.

Librerías Utilizadas:

- numpy para el manejo eficiente de operaciones vectoriales.
- matplotlib.pyplot para la creación de gráficos 2D.

Referencias:

- Física Universitaria (Sears & Zemansky) – Capítulo sobre gravitación.