



TAREA 5

Fecha de entrega: 17/10/2022 22:59 hrs

Problema

El problema de interacción gravitacional de 3 cuerpos es en general caótico pero existen varias soluciones *interesantes*, algunas de ellas incluso estables. A pesar de ser un problema antiguo, varias de las soluciones *interesantes* han sido descubiertas recientemente.

En este problema, Ud. debe integrar numéricamente las trayectorias de 3 cuerpos que interactúan gravitacionalmente. Dadas las condiciones iniciales que consideraremos, las trayectorias están restringidas al plano $x - y$. Las condiciones iniciales a considerar son las siguientes:

Masa 1: $\vec{r}_0 = [-0.97000436, 0.24308753]$, $\vec{v}_0 = [0.466203685, 0.43236573]$

Masa 2: $\vec{r}_0 = [0.97000436, -0.24308753]$, $\vec{v}_0 = [0.466203685, 0.43236573]$

Masa 3: $\vec{r}_0 = [0, 0]$, $\vec{v}_0 = [-0.93240737, -0.86473146]$

Además, consideraremos: $G = m_1 = m_2 = m_3 = 1$ (esto fija las unidades de las condiciones iniciales indicadas más arriba).

Esta tarea debe ser resuelta utilizando OOP. El archivo `codigo/planetario.py` contiene el esqueleto de la clase `TresCuerpos`. Ud. debe implementar los métodos de esta clase para resolver el problema. Los `docstrings` explican en qué debe consistir cada método. Ud. tiene libertad de mejorar los `docstrings`, agregar atributos y métodos a la clase según le parezca conveniente.

En particular, debe:

1. Implementar pasos temporales utilizando ya sea el método de Verlet o el de Beeman. Utilice el esqueleto del método llamado `verlet_o_beeman` y cámbiele el nombre según corresponda. Debe implementar el método.
2. Implementar pasos temporales utilizando el método de Runge-Kutta (al menos orden 4). No es necesario que lo implemente para esta tarea. Como sugerencia, si bien puede usar una librería (por ejemplo `scipy`), probablemente sería más sencillo usar el código que ya implementó en una tarea anterior.
3. Por simplicidad, para esta tarea no usaremos paso adaptativo, puede definir un paso pequeño y usar el mismo paso para toda la integración. Debe decir qué pruebas hizo para decidir que el paso utilizado era adecuado.
4. Debe calcular las trayectorias de las 3 partículas con los 2 métodos escogidos y compararlas. Como sugerencia, cree 2 objetos de la clase `TresCuerpos` e integre uno con Verlet (o Beeman) y el otro con RK para compararlos.
5. Debe estudiar la energía total del sistema y ver cómo evoluciona en los 2 métodos de integración. Comente sobre las similitudes o diferencias en su informe.

Visualización.

La clase `TresCuerpos` contiene el método `show_animated_orbit` que muestra una animación de las órbitas de los 3 planetas. Para poder usarla, debe tener `matplotlib` instalado pero **NO ES NECESARIO USAR ESTE MÉTODO**, es sólo una adición que le puede parecer interesante. También

pueden usar el código como inspiración para escribir otros métodos de visualización de las órbitas o para generar gráficos estáticos.

Información e Instrucciones Importantes.

- Repartición de puntaje:

Código: 5 % (aprobar PEP8) + 5 % (uso de git) + 50 % resolución del problema, implementación de los algoritmos necesarios, considerando además: uso efectivo de nombres, modularidad, calidad y utilidad de los `docstrings`.

Informe 40 %: Calidad del reporte considerando: análisis de las soluciones, comparación de los métodos en términos de las órbitas y de las energías del sistema. Claridad del lenguaje, calidad de las figuras y/o tablas. Como referencia, probablemente 3 páginas son suficientes para esta tarea.

- **REVISE SU ORTOGRAFÍA.**