



## TAREA 3

Fecha de entrega: 7/11/2020 21:59 hrs

### Problema

Para planetas que orbitan cerca del Sol, como Mercurio, el potencial gravitacional se puede escribir como:

$$U(r) = -\frac{GM_{\odot}m}{r} + \alpha \frac{GM_{\odot}m}{r^2}$$

donde  $G$  es la constante de gravitación universal,  $M_{\odot}$  es la masa del Sol,  $m$  es la masa del planeta,  $r$  es la distancia entre el planeta y el Sol, y  $\alpha$  es un número pequeño. Esta corrección a la ley de gravitación de Newton se debe a los efectos de los otros planetas del sistema solar (particularmente Júpiter) y a efectos relacionados con relatividad general.

Bajo este potencial, las órbitas siguen siendo planas pero ya no son cerradas sino que precesan, es decir, el afelio (punto más lejano de la órbita) gira alrededor del Sol.

En este problema exploraremos este tipo de órbitas y compararemos distintos métodos de integración de la ecuación diferencial. También utilizaremos programación orientada al objeto como ejercicio práctico para familiarizarnos con este paradigma de programación.

1. El archivo llamado `codigos/planeta.py` contiene el esqueleto de la clase `Planeta`. Ud. debe implementar los métodos de esa clase. Los docstrings explican en qué debe consistir cada método. Ud. tiene libertad de mejorar los docstrings, y agregar atributos y métodos a la clase según le parezca conveniente para resolver el problema descrito a continuación.

El archivo llamado `codigos/solucion_usando_planeta.py` muestra cómo incluir la clase `Planeta` en un script separado. Ud. también puede resolver todo dentro del mismo archivo, en cuyo caso puede descartar `solucion_usando_planeta.py`.

2. Parta por estudiar el caso  $\alpha = 0$  y considere las siguientes condiciones iniciales:

$$x_0 = 10$$

$$y_0 = 0$$

$$v_x = 0$$

Además, utilice unidades tales que  $GM_{\odot}m = 1$  y escoja  $v_y$  según le parezca (pero asegúrese de que la energía total sea negativa).

Integre la ecuación de movimiento por aproximadamente 5 órbitas usando los métodos de RK, Verlet, y Beeman. Grafique las órbitas y la energía total del sistema como función del tiempo en los 3 casos. Comente los resultados.

3. Ahora considere el caso  $\alpha = 10^{-2.XXX}$  (donde XXX son los 3 últimos dígitos de su RUT, antes del dígito verificador). Integre la ecuación de movimiento usando el método de Beeman por al menos 30 órbitas. Determine la velocidad angular de precesión. ¿Cómo lo hizo? En particular, ¿cómo determinó la posición del afelio? Grafique la órbita y la energía como función del tiempo.

### Comentarios.

1. Esta tarea pide explícitamente que utilice OOP (Object Oriented Programming) para su desarrollo. Es un ejercicio útil para aprender esta técnica.
2. En esta tarea debe implementar Verlet y Beeman pero puede utilizar RK4 (u otro RK de orden superior) de alguna librería o de alguna tarea anterior. Debe indicar en su informe la procedencia del código utilizado para RK.

### Instrucciones Importantes.

- Evaluaremos su uso correcto de `python`. Si define una función relativamente larga o con muchos parámetros, recuerde escribir el *docstring* que describa los parámetros que recibe la función, el output, y el detalle de qué es lo que hace la función. Recuerde que generalmente es mejor usar varias funciones cortas (que hagan una sola cosa bien) que una muy larga (que lo haga todo). Utilice nombres explicativos tanto para las funciones como para las variables de su código. El mejor nombre es aquel que permite entender qué hace la función sin tener que leer su implementación ni su *docstring*.
- Su código debe aprobar la guía sintáctica de estilo (PEP8). En [esta página](#) puede chequear si su código aprueba PEP8.
- Utilice `git` durante el desarrollo de la tarea para mantener un historial de los cambios realizados. La siguiente [cheat sheet](#) le puede ser útil. **Revisaremos el uso apropiado de la herramienta y asignaremos una fracción del puntaje a este ítem.** Realice cambios pequeños y guarde su progreso (a través de *commits*) regularmente. No guarde código que no corre o compila (si lo hace por algún motivo deje un mensaje claro que lo indique). Escriba mensajes claros que permitan hacerse una idea de lo que se agregó y/o cambió de un *commit* al siguiente.
- Para hacer un informe completo, además de los gráficos que se piden explícitamente, Ud. debe decidir qué es interesante y agregar las figuras correspondientes. No olvide anotar los ejes e incluir una *caption* que describa el contenido de cada figura. Tampoco olvide las unidades asociadas a las cantidades mostradas en los diferentes gráficos.
- La tarea se entrega subiendo su trabajo a github. Clone este repositorio (el que está en su propia cuenta privada y debiese tener una url como `https://github.com/uchileFI3104B-2020B/03-tarea-nombredeusuario`), trabaje en el código y en el informe y cuando haya terminado asegúrese de hacer un último *commit* y luego un *push* para subir todo su trabajo a github.
- El informe debe ser entregado en formato pdf, este debe ser claro sin información de más ni de menos. **Esto es muy importante, no escriba de más, esto no mejorará su nota sino que al contrario.** La presente tarea probablemente no requiere informes de más de 4 páginas en total (dependiendo de cuántas figuras incluya; esto no es una regla estricta, sólo una referencia útil). Asegúrese de utilizar figuras efectivas y tablas para resumir sus resultados.
- **Revise su ortografía.**

- Repartición de puntaje: 45 % implementación y resolución del problema (más o menos independientemente de la calidad de su código); 40 % calidad del reporte entregado: demuestra comprensión del problema y su solución, claridad del lenguaje, calidad de las figuras utilizadas; 5 % aprueba o no PEP8; 10 % diseño del código: uso efectivo de nombres de variables y funciones, docstrings, uso de git, etc.