

Tarea 4, Métodos numéricos para la ciencia e ingeniería

Farid Borbar

21 de octubre de 2015

1. Introducción

Para realizar esta tarea se estudio el tutorial de *SoftwareCarpentry* con el fin de ordenar nuestras nociones de programación, en base a este tutorial se respondieron una serie de preguntas.

La segunda parte consistió en aplicar el método de programación de clases para modelar la orbita de un planeta en torno al Sol, al mismo tiempo que aproximar su energía, todo esto implementado como funciones internas de una clase *Planeta*.

2. Metodología y Marco Teórico

Se nos entregó un potencial corregido relativisticamente para la gravitacion del Sol sobre un planeta de masa m :

$$U(r) = -\frac{GMm}{r} + \alpha \frac{GMm}{r^2} \quad (1)$$

Donde G es la constante de gravitación universal, M es la masa del Sol, r la distancia del Sol al planeta y α una constante pequeña, en este caso $-2,871$.

Para analizar la ecuación en *python* fue necesario estudiar el movimiento en coordenadas cartesianas considerando $r^2 = x^2 + y^2$, con esto se obtuvo el sistema de ecuaciones:

$$\frac{d^2X}{dt^2} = \left(\frac{-1}{x^2 + y^2} + \frac{\alpha}{(x^2 + y^2)^3} \right) * \frac{x}{x^2 + y^2} \quad (2)$$

$$\frac{d^2Y}{dt^2} = \left(\frac{-1}{x^2 + y^2} + \frac{\alpha}{(x^2 + y^2)^3} \right) * \frac{y}{x^2 + y^2} \quad (3)$$

Donde se consideró que la masa del Sol, la constante de gravitación universal y la masa del planeta tenían valor 1, para simplificar los calculos.

Basándonos en estas ecuaciones es que se graficó las trayectorias de la orbita y la energía con el paso del tiempo, para $\alpha = 0$ y $\alpha = -2,871$.

3. Resultados

3.1. Respuestas al Tutorial

Pregunta 1 Describa la idea de escribir el main driver primero y llenar los huecos luego. ¿Por qué es buena idea?

Escribir el main driver tiene la ventaja de que permite leer la estructura del programa de forma sencilla y sin las especificaciones de cada elemento, por ejemplo las definiciones de las funciones que se usan.

Pregunta 2Cuál es la idea detrás de la función mark filled? ¿Por qué es buena idea crearla en vez del código original al que reemplaza?

Esta función se utiliza para marcar características que se desea resaltar, en el caso del tutorial era que una celda específica estuviera "contaminada". Además cumplía la función de alertar sobre un error específico, el cual python alertaba pero de forma más mecánica, esto da la ventaja de poder alertar o enviar un mensaje de error en nuestras palabras, más amigable.

Pregunta 3 ¿Qué es refactoring?

Se le llama así al proceso de modificar la estructura del código para mejorar el funcionamiento de ciertas funciones internas. Sin embargo, la idea es no alterar la finalidad ni la función en sí de la parte reestructurada, es solo para optimizar su función.

Pregunta 4 ¿Por qué es importante implementar tests que sean sencillos de escribir? ¿Cuál es la estrategia usada en el tutorial?

Los test nos sirven para estudiar la funcionalidad de nuestro programa en partes específicas, la idea es que sean sencillos para que sirvan realmente como una ayuda y no representen una fuente de errores extra a las que se buscan eliminar con los mismos. En el caso del ejemplo se crearon nuevas funciones para testear las usadas en el programa, por ejemplo se implementó una función específica que permitiera generar matrices con características específicas a fin de poner a prueba la función marcaba las celdas (se quería estudiar como se comportaba frente distintos escenarios no necesariamente aleatorios). Las funciones para testear además nos dan perspectiva sobre nuestro código, y de hecho algo muy usual es que en base a las pruebas realizadas se mejore el código original que suele ser más rústico.

Pregunta 5 El tutorial habla de dos grandes ideas para optimizar programas, ¿cuáles son esas ideas? Descríbalas.

La primera idea es llamada "cambiar memoria por tiempo", consiste en guardar información de los procesos del programa si es que esta puede ser utilizada en el futuro, esto nos permite ahorrar tiempo de operaciones y cálculos internos del programa sacrificando más memoria de datos. Si se busca optimizar el tiempo de operación es una idea conveniente.

La segunda es llamada "divide para vencer." "búsqueda binaria", consiste en un método de búsqueda en el cual se divide toda la lista (diccionario) por la mitad y se analiza si el "objeto" buscado está en una mitad u otra, luego se procede a hacer lo mismo con la mitad con la que nos quedamos. Esto permite una ve-

locidad de búsqueda mucho mayor que analizar objeto por objeto, y de hecho puede demostrarse que es la manera de buscar más rápida.

Pregunta 6 ¿Qué es lazy evaluation?

Este método también es una manera de optimizar el funcionamiento, memoria y tiempo de operación de nuestro código, consiste en solamente crear los elementos que realmente serán utilizados a medida que sean necesarios, con esto podemos, potencialmente, ahorrar memoria y tiempo de operación que se utilizaría en recorrer código que de facto, no está siendo utilizado. Sin embargo, para esto es necesario implementar un código más complejo, es por esto que se considera que el *Lazy evaluation* cambia tiempo humano por mejor desempeño del programa.

Pregunta 7 Describa la other moral del tutorial (es una de las más importantes a la hora de escribir buen código).

La otra enseñanza a la cual hace referencia el tutorial apunta al concepto de rapidez del programa: Para hacer un programa rápido, es preferible escribir uno sencillo como primera opción, y este programa sencillo, editarlo con la ayuda de test y mecanismos de optimización para mejorar, reestructurar o modificar sus partes. Esto permite partir de un nivel de abstracción menor para luego complejizarlo en la búsqueda de su mejoramiento.

3.2. Gráficos de Órbitas y Energía

A continuación se adjuntan los gráficos obtenidos del programa con los cuales se busca representar la trayectoria que sigue el planeta en torno al Sol, y la energía del planeta a lo largo del tiempo de desplazamiento.

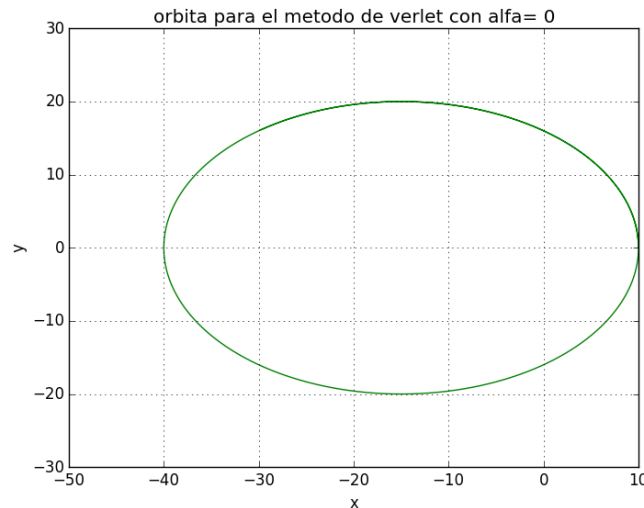


Figura 1: Gráfico de la órbita seguida por el planeta, para $\alpha = 0$.

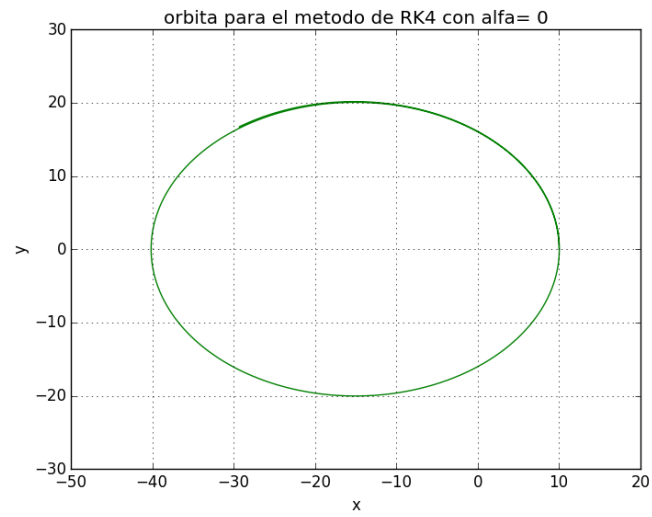


Figura 2: Gráfico de la órbita seguida por el planeta, para $\alpha = 0$.

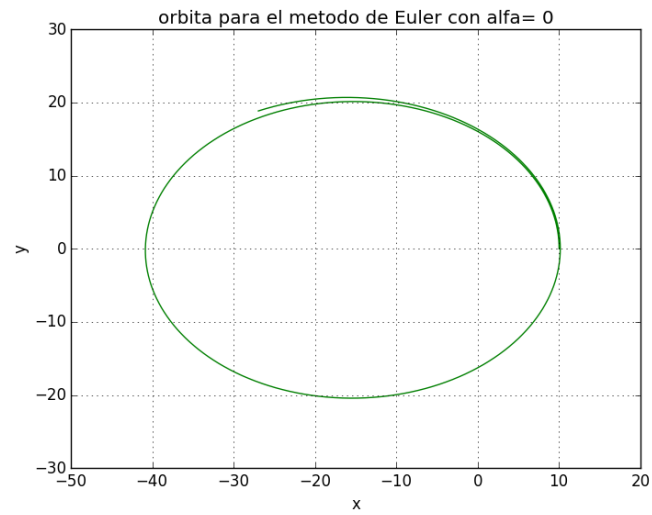


Figura 3: Gráfico de la órbita seguida por el planeta, para $\alpha = 0$.

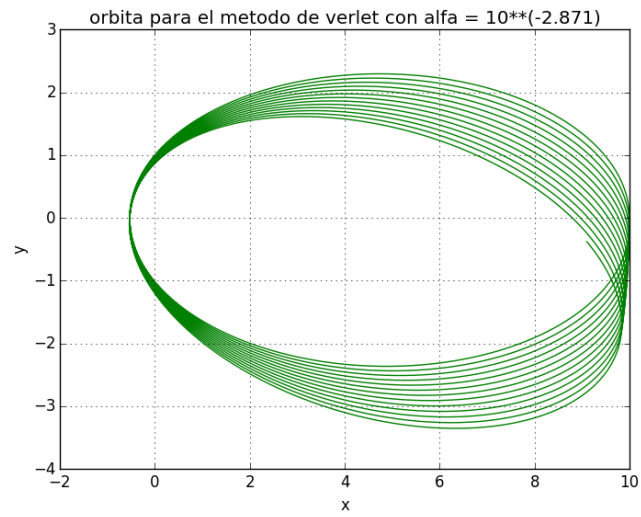


Figura 4: Gráfico de la órbita seguida por el planeta, para $\alpha = -2,871$.

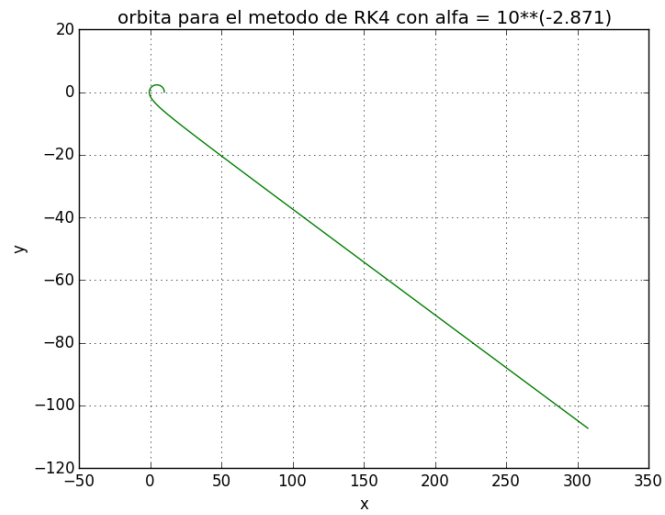


Figura 5: Gráfico de la órbita seguida por el planeta, para $\alpha = -2,871$.

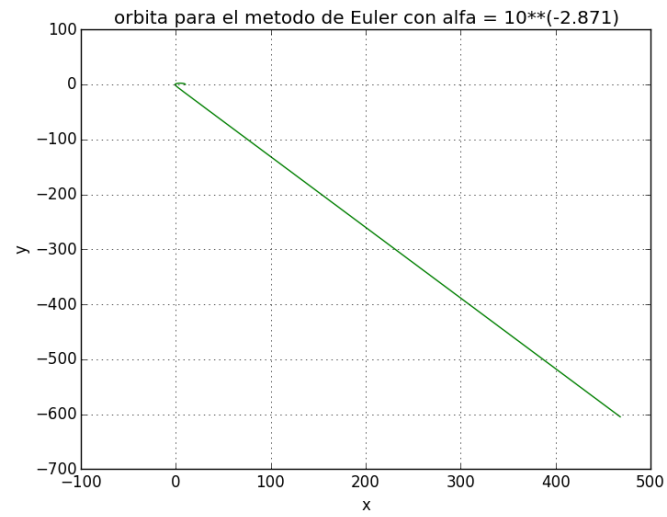


Figura 6: Gráfico de la órbita seguida por el planeta, para $\alpha = -2,871$.

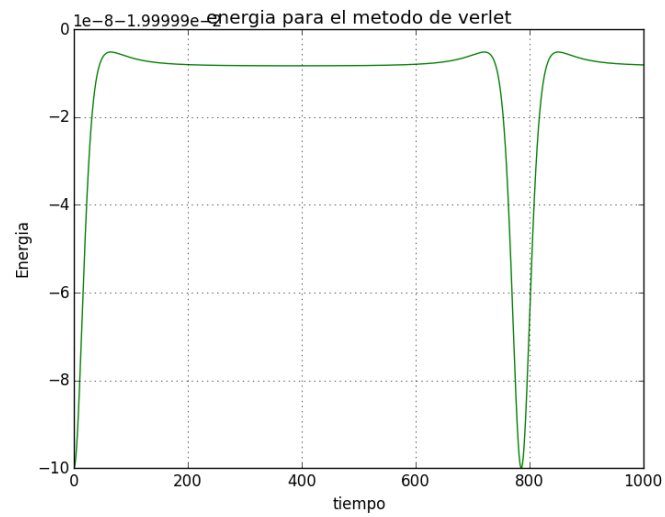


Figura 7: Gráfico de la Energía del planeta a lo largo del tiempo, para $\alpha = 0$.

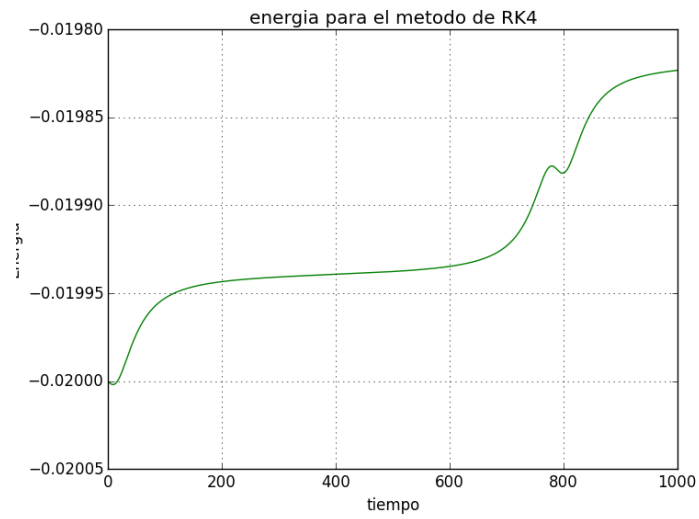


Figura 8: Gráfico de la Energía del planeta a lo largo del tiempo, para $\alpha = 0$.

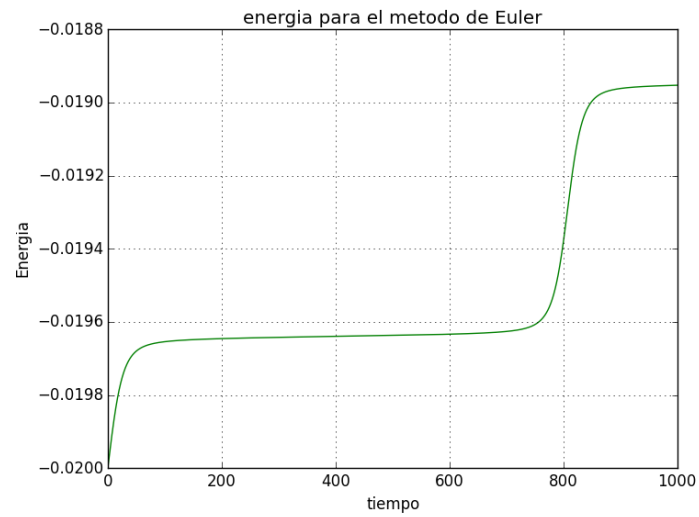


Figura 9: Gráfico de la Energía del planeta a lo largo del tiempo, para $\alpha = 0$.

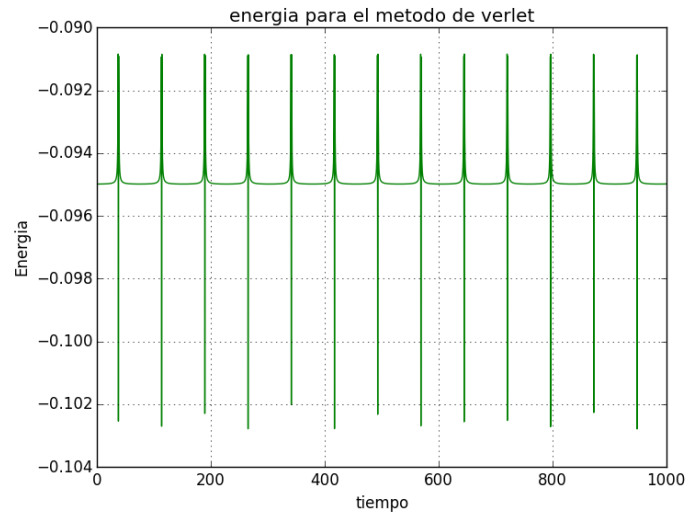


Figura 10: Gráfico de la Energía del planeta a lo largo del tiempo, para $\alpha = -2,871$.

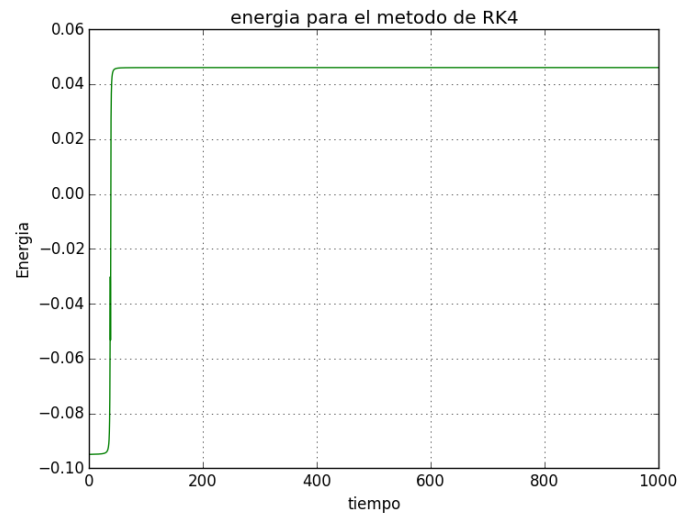


Figura 11: Gráfico de la Energía del planeta a lo largo del tiempo, para $\alpha = -2,871$.

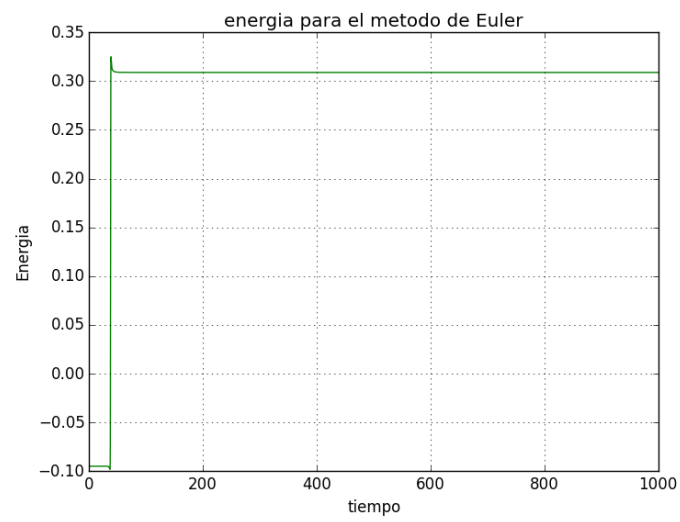


Figura 12: Gráfico de la Energía del planeta a lo largo del tiempo, para $\alpha = -2,871$.