

# Métodos matemáticos para la ciencia e Ingeniería:

## Informe Tarea 4

Fernanda PÉREZ (*rut: 18.769.232-6*)

20 de Octubre, 2015

## 1 Preguntal

### 1.1 Introducción

Se busca responder preguntas sobre un tutorial de la fundación *Software Carpentry* que tiene como objetivo enseñar a científicos e ingenieros habilidades para mejorar el diseño de software.

### 1.2 Resultados

- **¿Cuál es la idea de escribir el main driver primero y llenar los huecos luego?, ¿Por qué es buena idea?**

La idea de escribir el Main driver primero y luego ir rellenando los huecos es que el Main driver nos da la idea de todo lo que hace nuestro programa, dándonos la estructura de éste. Es una buena idea ya que el hecho de ir rellenando posteriormente, y dejando las funciones fuera del Main driver, nos permite poder optimizar el programa y encontrar errores con mayor facilidad. También gracias a esta organización es posible llamar a este script en otro y usar las funciones sin que se ejecute el programa principal en sí.

- **¿Cuál es la idea detrás de la función `mark.filled`?, ¿Por qué es buena idea crearla en vez del código original al que reemplaza?**

La función `mark.filled` sirve para testear que X e Y sean coherentes con lo que necesitamos en nuestro problema. Es una buena idea crearla por varias razones:

- Plantea, mejor que un comentario, lo que realmente se espera de X e Y.
- Cuando python falla entrega un mensaje de error genérico, mientras que con esta función es posible crear un mensaje más explicativo.

- Por la manera en que python trabaja, para ciertos valores de X e Y no habría error según python, pero nuestro programa sí estaría fallando. Con esta función podemos captar esas fallas y avisar con un mensaje significativo.

- **¿Qué es refactoring?**

Es el proceso de reorganización de la estructura del código, sin modificar su comportamiento o funcionalidad, para mejorar su calidad en algún aspecto. En el ejemplo del tutorial es ocupado para mejorar la testeabilidad.

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

Se necesitan test sencillos de escribir para que nuestro testing nos ayude a crear un programa correcto en lugar de darnos más cosas de las cuales preocuparnos (posibles bugs). En el tutorial se utiliza la estrategia de *dibujar* las cosas. Se utilizan strings, en lugar de más código, para testear. Esto hace muy fácil el hecho de ir agregando más tests.

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

- Trade memory for time: Guardar información redundante con el objetivo de ahorrar recálculos para así hacer más rápido el algoritmo.
- Trade human time for higher machine performance: Escribir programas más complicados mejorando así el desempeño del programa.

- **¿Qué es lazy evaluation?**

Es una forma común de optimizar programas. Consiste en no computar datos hasta que sean realmente necesitarlos. Es complicado de llevar a cabo, por lo cual se negocia trabajo computacional versus desempeño del programa.

- **Describe la other moral del tutorial (es una de las más importantes a la hora de escribir buen código).**

Si se quiere escribir un programa que sea rápido, hay que comenzar escribiendo un programa simple. Luego se testea y mejora por partes, reusando los test para chequear el trabajo en cada etapa.

## 2 Pregunta 2

### 2.1 Introducción

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

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

Donde  $\alpha$  es un número pequeño. Esta corrección a la ley de gravitación de Newton es una buena aproximación derivada de la teoría de la relatividad general de Einstein. 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.

Se pide implementar la clase Planeta que contenga los métodos euler explícito, runge-kutta 4 y verlet. Con esto, se busca en primer lugar integrar la ecuación de movimiento por aproximadamente 5 órbitas, utilizando los tres métodos anteriormente nombrados, con las condiciones iniciales  $x_0 = 10$ ,  $y_0 = 0$ ,  $v_x = 0$  y  $v_y$  a elección exigiendo que la energía inicial sea negativa y  $G = 1$ ,  $M = 1$  y  $\alpha=0$ , graficando las órbitas y la energía total del sistema en función del tiempo.

En segundo lugar, se busca integrar la misma ecuación de movimiento por 30 órbitas, utilizando el método de verlet considerando esta vez  $\alpha = 10^{-2.232}$ , y determinar la velocidad angular de precesión.

### 2.2 Procedimiento

Dado que:

$$F_x = -\frac{\partial U}{\partial x}, F_y = -\frac{\partial U}{\partial y}$$

Se tiene que las ecuaciones de movimiento son las siguientes:

$$\ddot{x} = -xGM\left(\frac{1}{r^3} - \frac{2\alpha}{r^4}\right) = f_x$$

$$\ddot{y} = -yGM\left(\frac{1}{r^3} - \frac{2\alpha}{r^4}\right) = f_y$$

Donde  $r = \sqrt{x^2 + y^2}$ .

A continuación se muestra la forma que toman las posiciones y velocidades luego de un paso, para cada uno de los 3 métodos a utilizar.

#### 2.2.1 Euler explícito

$$x_{n+1} = x_n \cdot dt + f_{x_n} \quad y_{n+1} = y_n \cdot dt + f_{y_n}$$

$$v_{x_{n+1}} = f_{x_n} \cdot dt + v_{x_n} \quad v_{y_{n+1}} = f_{y_n} \cdot dt + v_{y_n}$$

### 2.2.2 Runge-kutta 4

$$\begin{aligned}
K1 &= (v_{x_n}, v_{y_n}, f_{x_n}, f_{y_n}) \\
K2 &= (v_{x_n} + k_{11}/2, v_{y_n} + k_{12}/2, f_{x_n} + k_{13}/2, f_{y_n} + k_{14}/2) \\
K3 &= (v_{x_n} + k_{21}/2, v_{y_n} + k_{22}/2, f_{x_n} + k_{23}/2, f_{y_n} + k_{24}/2) \\
K4 &= (v_{x_n} + k_{31}, v_{y_n} + k_{32}, f_{x_n} + k_{33}, f_{y_n} + k_{34})
\end{aligned}$$

En donde  $k_{ij}$  corresponde a la componente número  $j$  del vector  $K_i$ . Luego se tiene:

$$\begin{aligned}
x_{n+1} &= x_n + \frac{1}{6}dt(k_{11} + 2k_{21} + k_{31} + k_{41}) & y_{n+1} &= y_n + \frac{1}{6}dt(k_{12} + 2k_{22} + k_{32} + k_{42}) \\
v_{x_{n+1}} &= v_{x_n} + \frac{1}{6}dt(k_{13} + 2k_{23} + k_{33} + k_{43}) & v_{y_{n+1}} &= v_{y_n} + \frac{1}{6}dt(k_{14} + 2k_{24} + k_{34} + k_{44})
\end{aligned}$$

### 2.2.3 Verlet

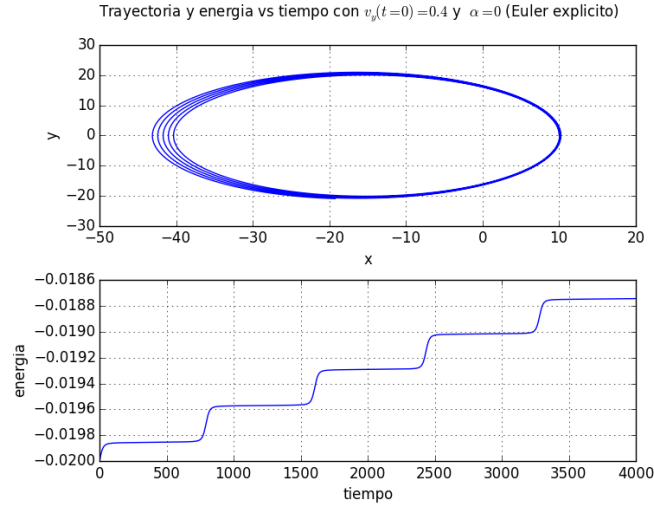
$$\begin{aligned}
x_{n+1} &= x_n + v_{x_n}dt + f_{x_n}dt^2/2 & y_{n+1} &= y_n + v_{y_n}dt + f_{y_n}dt^2/2 \\
v_{x_{n+1}} &= v_{x_n} + (f_{x_n} + f_{x_{n+1}})dt/2 & v_{y_{n+1}} &= v_{y_n} + (f_{y_n} + f_{y_{n+1}})dt/2
\end{aligned}$$

Se escoge  $v_y(t=0) = 0.4$ . Se utiliza  $dt=0.1$  en todos los casos excepto al utilizar runge-kutta 4, que se utiliza  $dt=0.5$ .

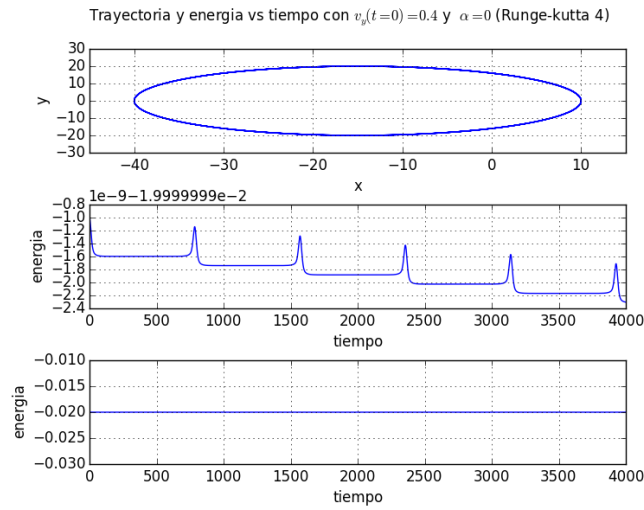
Para calcular la velocidad angular de precesión se buscan los  $r_{max}$  en cada vuelta (con  $r = \sqrt{x^2 + y^2}$ ), lo que corresponde a buscar los afelios. Cuando se encuentran, se guardan las posiciones  $x_{afelio}$  e  $y_{afelio}$ . Luego tenemos que  $\theta = \arctan(\frac{x_{afelio}}{y_{afelio}})$ . Finalmente utilizando la variación de los  $\theta$  y del tiempo es posible obtener la velocidad angular. Se toma un valor promedio de todas las velocidades obtenidas (damos 31 vueltas).

## 2.3 Resultados

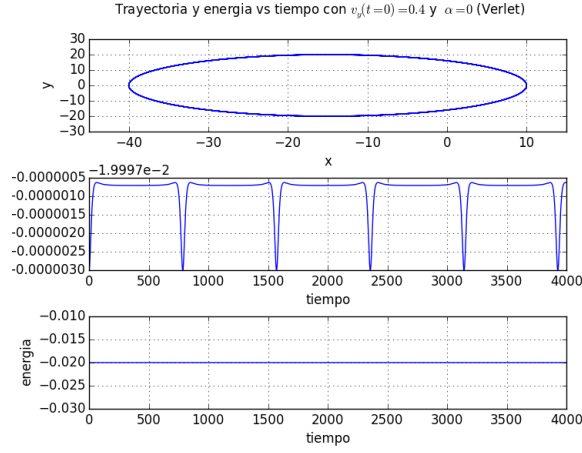
Integrando la ecuación de movimiento con los tres métodos con  $\alpha = 0$  por aproximadamente 5 órbitas y con el método de Verlet con  $\alpha = 10^{-2.232}$  por aproximadamente 30 órbitas, se obtienen las siguientes gráficas:



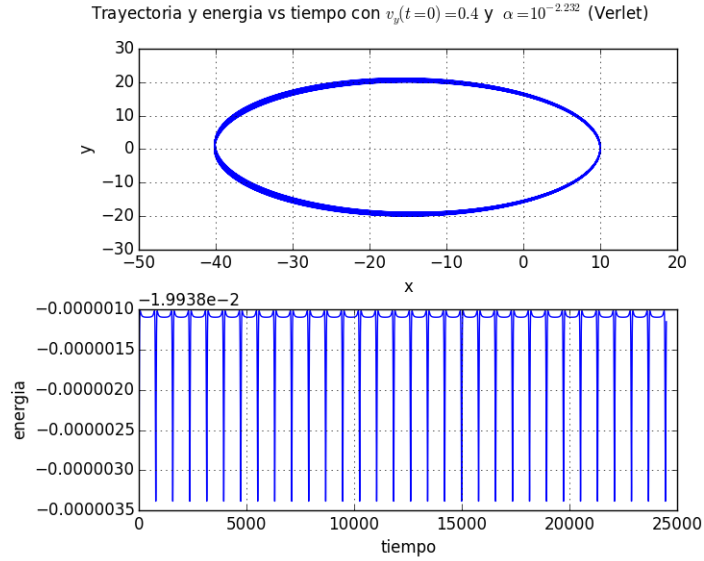
**Figura 1:** Gráfica de la trayectoria de 5 órbitas y de la energía total del sistema en función del tiempo utilizando el método de euler con  $dt=0.04$ .



**Figura 2:** Gráfica de la trayectoria de 5 órbitas y de la energía total del sistema en función del tiempo utilizando el método de runge-kutta 4 con  $dt=1$ . Se muestran dos gráficas de la energía: una cercana para ver la forma de la función y otra en la escala de la Figura 2 para comparar los métodos.



**Figura 3:** Gráfica de la trayectoria de 5 órbitas y de la energía total del sistema en función del tiempo utilizando el método de verlet con  $dt=0.5$ . Se muestran dos gráficas de la energía: una cercana para ver la forma de la función y otra en la escala de la Figura 2 para comparar los métodos.



**Figura 4:** Gráfica de la trayectoria de 31 órbitas y de la energía total del sistema en función del tiempo utilizando el método de verlet con  $dt=0.5$

Se obtiene una velocidad angular de precesión de  $-3.04e-06$ .

## 2.4 Conclusiones

Sobre la energía del sistema, es posible apreciar en las Figuras 1, 2 y 3 que es el método de Verlet el cual consigue conservarla de mejor manera, seguido por Euler-implícito. Por otro lado, para la velocidad inicial en  $y$  escogida, el método Runge-kutta 4 (ver Figura 2) resulta obtener soluciones inestables, no así el método Runge-kutta 4 (ver Figura 2) ni el de Verlet (ver Figura 3). Para el valor de  $\alpha = 10^{-2.232}$ , se genera un efecto en la trayectoria llamado precesión (ver Figura 4). Se obtiene una velocidad angular de precesión de  $-3.04 \times 10^{-6}$ .