

Tarea 03 - Métodos Numéricos

Luz Agüero Contreras 18.355.502-2

Profesor: Valentino Gonzales

Auxiliar: Felipe Pesce

28 Octubre 2015

1 Introducción

Parte 1

La primera parte consistió en integrar por medio del método de Runge Kutta de orden 3, la ecuación diferencial del oscilador de Van del Pol, dada por:

$$\frac{d^2x}{dt^2} = -kx - \mu(x^2 - a^2)\frac{dx}{dt}$$

Por medio del siguiente cambio de variables $x(t) = a \cdot y(s)$ y $s(t) = t\sqrt{k}$ queda:

$$\frac{d^2y}{ds^2} = -y - \mu^*(y^2 - 1)\frac{dy}{ds} \quad (1)$$

Donde $\mu^* = \frac{a^2\mu}{\sqrt{k}}$

Se integró la solución hasta $T = 20\pi$ para las siguientes condiciones iniciales:

$$\frac{dy}{ds} = 0; y = 0.1$$

$$\frac{dy}{ds} = 0; y = 4.0$$

Parte 2

Esta parte consistió en integrar por medio del metodo de Runge Kutta 4 (disponible en las librerías de python) las ecuaciones diferenciales del llamado Atractor de Lorenz:

$$\frac{dx}{dt} = \sigma(y - x) \quad (2)$$

$$\frac{dy}{dt} = x(\rho - z) - y \quad (3)$$

$$\frac{dz}{dt} = xy - \beta z \quad (4)$$

Donde $\sigma = 10$, $\beta = \frac{8}{3}$ y $\rho = 28$. Para posteriormente analizar su comportamiento con distintas condiciones iniciales a traves de gráficos 3D.

2 Procedimiento

Parte 1

Usando la formula del método de Runge Kutta 3 vista en clases:

$$y_{n+1} = y_n + \frac{k_1 + 4k_2 + k_3}{6}$$

$$k_1 = h \cdot f(x_n, y_n)$$

$$k_2 = h \cdot f(x_n + k_1/2, y_n + k_1/2)$$

$$k_3 = h \cdot f(x_n - k_1 - 2k_2, y_n - k_1 - 2k_2)$$

Donde $f(x_n, y_n)$ es la función a integrar. Todas las ecuaciones anteriores fueron definidas en funciones para facilitar su uso a lo largo del código.

Luego, la ecuación (1) puede escribirse de la siguiente manera:

$$\frac{d}{ds} \begin{pmatrix} y \\ v \end{pmatrix} = \begin{pmatrix} v \\ -y - \mu^* v(y^2 - 1) \end{pmatrix} \quad (5)$$

Con $v = \frac{dy}{ds}$, así definiendo una función con los parametros de la izquierda y que retorne el vector de la derecha. Buscando y y v con RK3 y ocupando el vector a la derecha como $f(x_n, y_n)$. Parametros usados:

$$h = \frac{20\pi}{5000} \quad (Paso)$$

$$\mu^* = 1.502 \quad (RUT)$$

Posteriormente, se calcularon los valores de y y v y se guardaron en listas vacias para su uso posterior.

Parte 2

El conjunto de ecuaciones (2), (3), (4) pueden escribirse de la siguiente manera:

$$\frac{d}{dt} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \sigma(y - x) \\ x(\rho - z) - y \\ xy - \beta z \end{pmatrix} \quad (6)$$

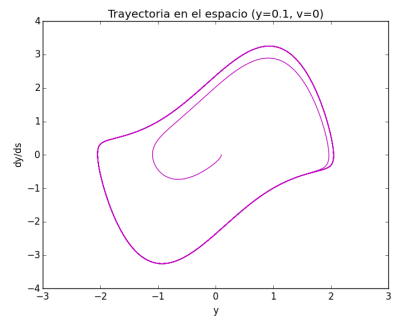
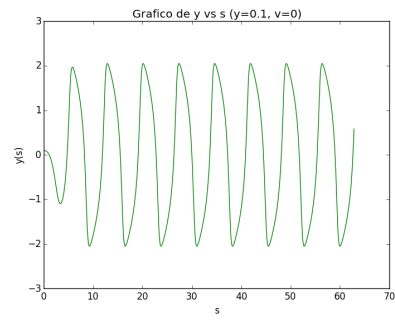
Para luego definir una función, al igual que la Parte 1, con los parametros de la izquierda y que retorne el arreglo de la derecha. Se implementó la rutina *ODE* de *scipy.integrate*, y luego *dopri5*, que es el código que contiene el método de Runge Kutta de orden 4. Se eligio un tiempo desde 0 a 100 con 50000 valores entremedio, para que resultase una curva suave. Con tiempos mayores, el gráfico resultante no presentaba una estructura definida de la función.

Posteriormente, se calcularon los valores de x , y y z y se guardaron en listas vacias para su uso posterior.

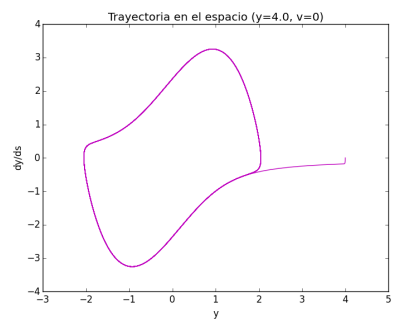
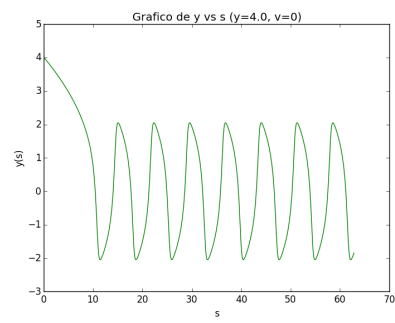
3 Resultados

Parte 1

a)



b)



Parte 2

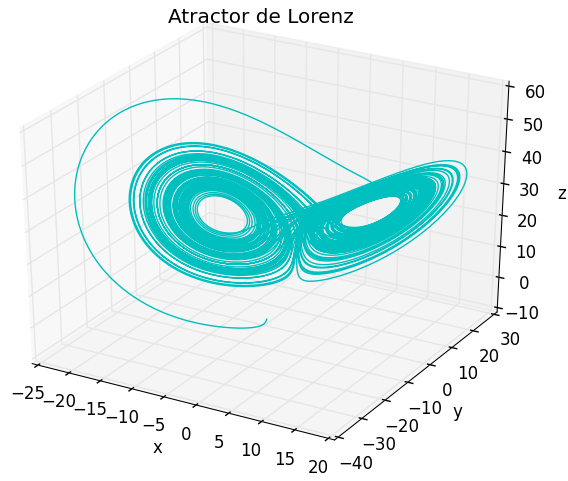


Figure 1: Gráfico para condiciones iniciales $(x_0, y_0, z_0) = (-5, 1, -10)$

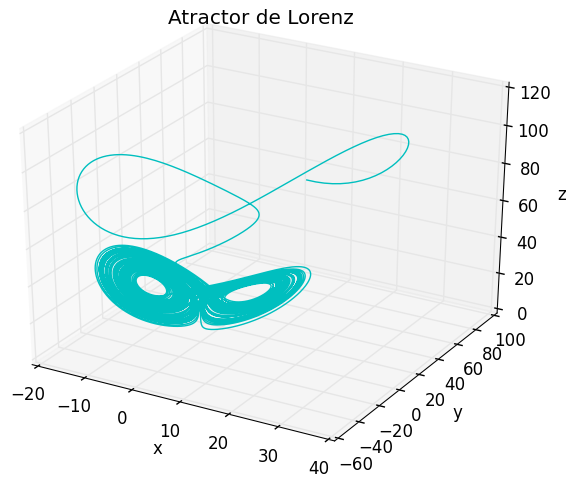


Figure 2: Gráfico para condiciones iniciales $(x_0, y_0, z_0) = (1, 100, 50)$

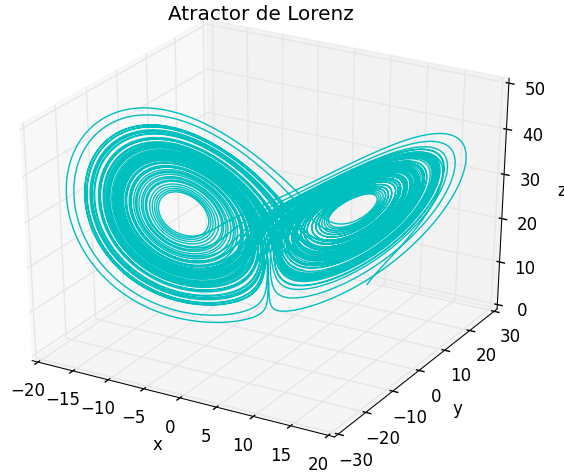


Figure 3: Gráfico para condiciones iniciales $(x_0, y_0, z_0) = (10, 10, 10)$

4 Conclusiones

Parte a

Del segundo gráfico de a), con las condiciones iniciales $y = 0.1, dy/ds = 0$, se puede apreciar la intención de entrar en equilibrio en torno a 0 por parte del sistema, caso distinto en b), donde se ve que el sistema no presenta un cambio de comportamiento en la trayectoria. Esto debido a que, dadas las distintas condiciones iniciales, el sistema se demora más o menos en variar su trayectoria, lo cual tiene sentido con cualquier sistema físico.

Parte b

Dadas las distintas condiciones iniciales el sistema se demora más o menos en encontrar su equilibrio, pero sin grandes variaciones en el resultado.

En general se puede concluir que los métodos de integración RG3 y RG4 sirven para simular sistemas físicos sin mayores problemas y de manera sencilla.