Métodos Numéricos para la Ciencia y la Ingeniería: Informe Tarea 3.

Camila Castillo Pinto. 18.889.762-2

8 de Octubre, 2015.

1 Pregunta 1.

1.1 Introducción.

Esta pregunta consistió en integrar, usando el Método de Runge Kutta de orden 3, la ecuación del oscilador de Van Der Pol:

$$\frac{d^2x}{dt^2} = -kx - \mu(x^2 - a^2)\frac{dx}{dt} \tag{1}$$

donde k es la constante elástica y μ es un coeficiente de roce.

Se realizó un cambio de variable de la forma: $x(t) = ay(s)$, $s(t) = t\sqrt{k}$ y se obtuvo la siguiente ecuación que corresponde a la simplificación de la ecuación (1):

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

con lo cual ahora la ecuación sólo depende de un parámetro: $\mu^* = \dfrac{a^2\mu}{\sqrt{k}}$.

Se pidió implementar una versión propia del método de Runge Kutta e integrarlo hasta $T = 20 * \pi$ (aproximadamente 10 períodos) para dos pares de condiciones iniciales:

1) $\dfrac{dy}{ds} = 0; y = 0.1$

2) $\dfrac{dy}{ds} = 0; y = 4.0$

1.2 Procedimiento.

Se usó la fórmula de Runge Kutta orden 3 vista en clases:

$$y_{n+1} = y_n + \frac{k_1 + 4 * k_2 + k_3}{6} \tag{3}$$

donde

$$k_1 = h * f(x_n, y_n) \tag{4}$$

$$k_2 = h * f(x_n + \frac{k_1}{2}, y_n + \frac{k_1}{2}) \tag{5}$$

$$k_3 = h * f(x_n - k_1 - 2 * k_2, y_n - k_1 - 2 * k_2) \tag{6}$$

En este caso la fórmula (2) puede ser escrita de manera que se baje su orden:

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

donde $v = \frac{dy}{ds}$.

Luego tanto y como $\frac{dy}{ds}$ serán los valores que se encontrarán ocupando el método de Runge Kutta y el vector del lado derecho de la ecuación será la función f, con $x_n = v$ e $y_n = -y - \mu^*(y^2 - 1) * v$ para este caso.

Las fórmulas (3), (4), (5) y (6) se definieron como funciones en el algoritmo programado.

Se usó un paso h adecuado de $\frac{20\pi}{40000}$ donde 40000 corresponde al número de saltos que se deseaba y 20π corresponde al número hasta donde se deseaba integrar. Además se utilizó $\mu^* = 1.762$.

Luego se definieron vectores vacíos para poder recuperar los valores de y y de $\frac{dy}{ds}$.

Se iteró hasta el número de saltos que se fijó, calculando para cada iteración y y $\frac{dy}{ds}$ y guardándolos en los vectores respectivos. Luego se graficaron ambas cantidades.

Para el gráfico de y versus s, se definió un vector s desde 0 hasta 20π (intervalo donde se deseaba integrar), todo esto con el salto deseado de 40000.

Este procedimiento se realizó para cada par de condiciones iniciales.

1.3 Resultados.

Los resultados de este algoritmo se pueden apreciar en las figuras 1, 2, 3 y 4.

1.4 Conclusiones.

Se concluye que el algoritmo utilizado sirve, ya que nos entrega una trayectoria (ver Figura 1, ver Figura 2) muy similar a la del oscilador de Van der Pol.

De la figura 1 se puede intuir que esta trayectoria posee un equilibrio en torno al origen, ya que la espiral se acerca hacia este punto. Sin embargo, esto no ocurre en la figura 2, donde se ve que la trayectoria no se acerca al origen luego de completar una vuelta. Se cree que esto se debe al valor inicial que se tomó para $y(s)$.

De las figuras 3 y 4 se concluye que ambos gráficos son consistentes con las características que presenta un oscilador.

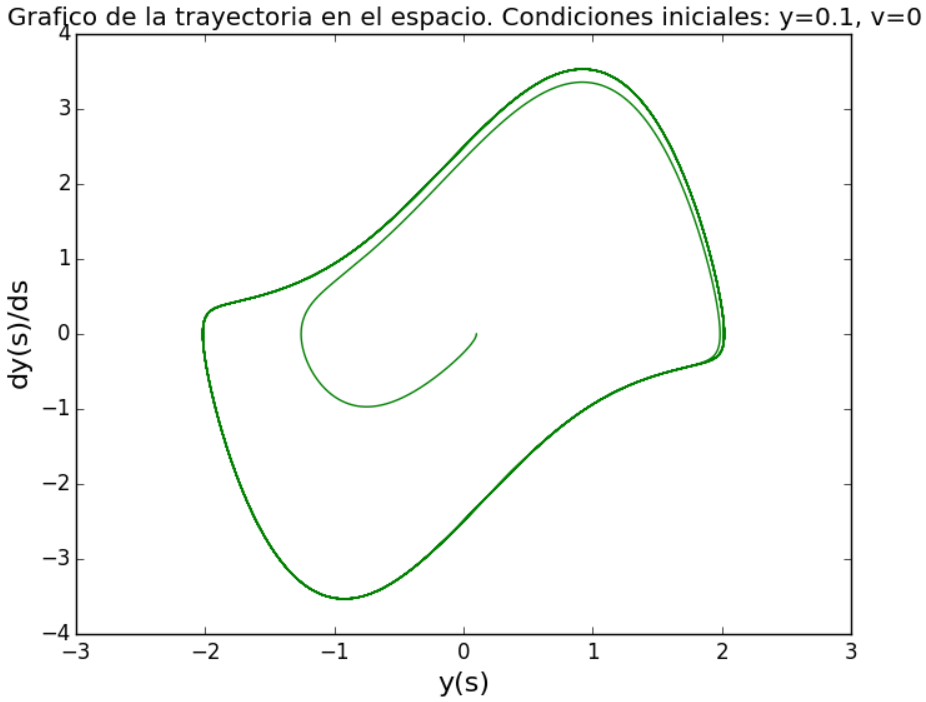


Figure 1: .

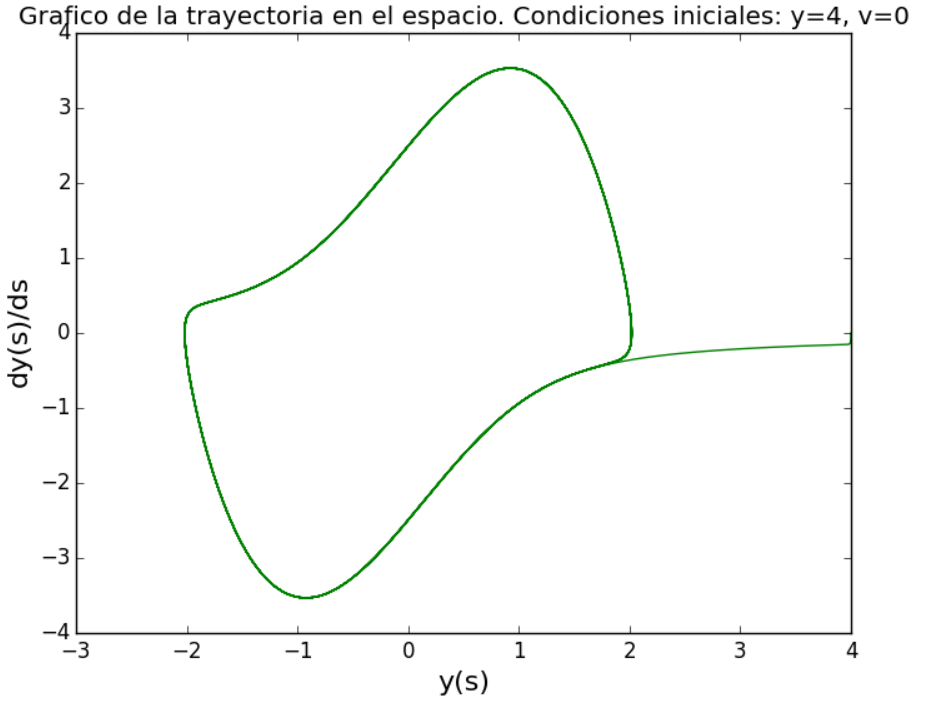
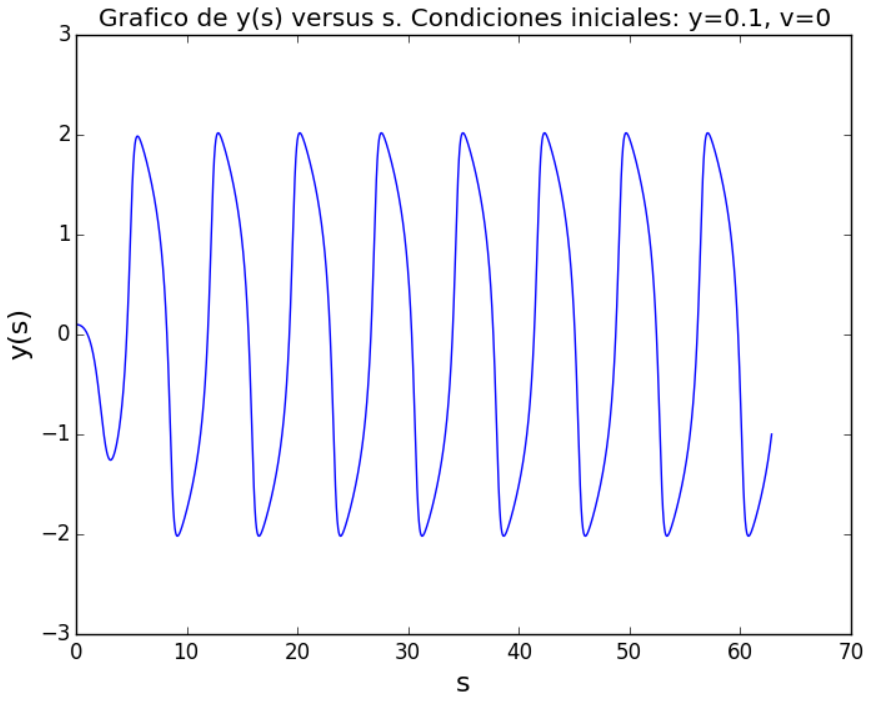


Figure 2: .

Figure 3: .



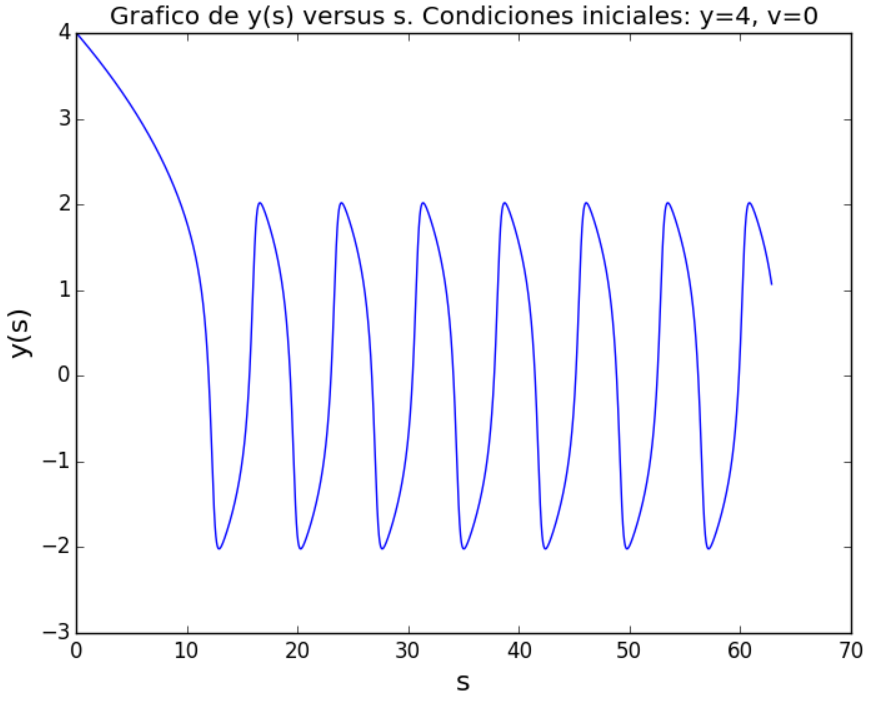


Figure 4: .

2 Pregunta 2.

2.1 Introducción.

Esta pregunta consistía en integrar el sistema de ecuaciones:

$$\begin{aligned}\frac{dx}{ds} &= \sigma(y-x)\\ \frac{dy}{ds} &= x(\rho-z)-y\\ \frac{dz}{ds} &= xy-\beta z\end{aligned}$$

que es equivalente a:

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

conocido como el sistema de Lorenz, cuya solución más conocida es el Atractor de Lorenz y se obtiene con los parámetros $\sigma = 10, \beta = 8/3 y \rho = 28$. Se pediía entonces integrarla usando condiciones iniciales y tiempo a gusto personal, y mediante el método de Runge Kutta de orden 4 que están disponibles en python.

Además se pedía el gráfico de la solución en 3D.

2.2 Procedimiento.

Para esta pregunta se decidió utilizar el algoritmo ODE de *scipy.integrate* y se utilizó la estructura vista en clases.

Se fijaron las condiciones iniciales como $t_0 = 0, x_0 = y_0 = z_0 = 5$ y se definió la función como el vector del lado derecho del sistema de ecuaciones.

Luego se crea el resolvedor (ODE) y se utiliza *dopri5* que es el método que se basa en Runge Kutta.

Se crean vectores para recuperar los valores de x, y, z y luego se inicia la iteración de 0 hasta 100 (con 5000 valores entre ellos, intervalo de tiempo conveniente), aplicando el algoritmo Runge Kutta en cada iteración y guardando los resultados en los vectores correspondientes.

Finalmente se grafica en 3D, siguiendo la estructura que se muestra en el archivo *3D.py*.

2.3 Resultados.

El gráfico de la solución en 3D se muestra en la Figura 5.

2.4 Conclusiones.

De la Figura 5 es posible concluir que cualquier, ya que se probaron diferentes valores para las condiciones iniciales, la diferencia en su elección cambia de forma dramtica los resultados.

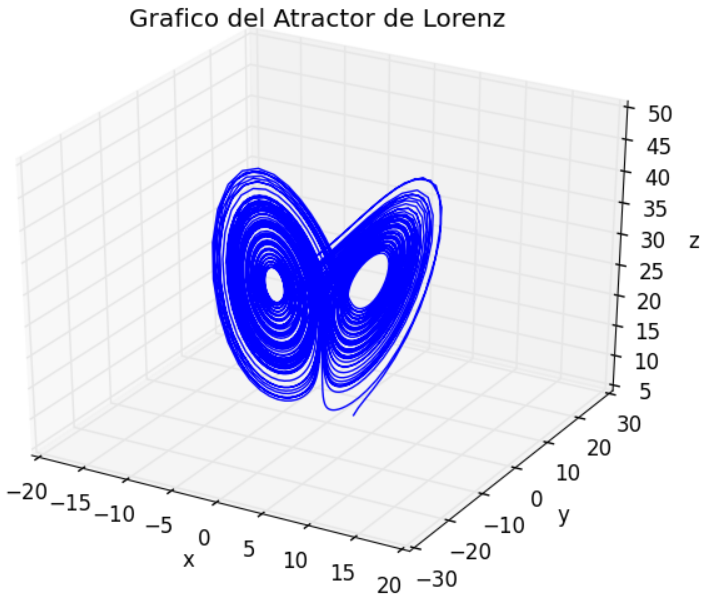


Figure 5: .

Sin embargo, este sistema presenta una tendencia a evolucionar en una zona en particular del espacio en forma de espiral.

Además se puede asegurar que el algoritmo utilizado funciona de manera correcta.