Universidad de Chile, Facultad de Ciencias Fisicas y Matematicas
Metodos Numericos para la Ciencia y Ingenieria

Tarea 3

Maria Jose Hernandez Pozo

October 8, 2015

1 Introduccion

El oscilador de van der Pool fue propuesto para describir la dinámica de algunos circuitos eléctricos. La ecuacion es la siguiente:

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

donde k es la constante elástica y μ es un coeficiente de roce. Si $|x| > a$ el roce amortigua el movimiento, pero si $|x| < a$ el roce inyecta energía. Se puede hacer un cambio de variable para convertir la ecuación a:

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

con lo cual ahora la ecuación sólo depende de un parámetro, μ^*. Indique cuál es el cambio de variable realizado.
Integre la ecuación de movimiento usando el método de Runge-Kutta de orden 3. Se pide que implementar su propia versión del algoritmo, describa la discretización usada y el paso de tiempo. Use $\mu^* = 1.376$ Integre la solución hasta $T = 20\pi$ (aproximadamente 10 períodos) para las siguientes condiciones iniciales:

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

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

Por otra parte, el sistema de Lorenz es un set de ecuaciones diferenciales ordinarias conocido por tener algunas soluciones caóticas, la más famosa el llamado atractor de Lorenz. El sistema de ecuaciones es el siguiente:

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

La solución más famosa se obtiene con los parámetros $\sigma = 10$, $\beta = 8/3$ y $\rho = 28$. Utilice esos parámetros, elija un set de condiciones iniciales (x0, y0, z0) e integre la ecuación por un tiempo que estime conveniente. Esta vez se pide que utilice un algoritmo RK4 pero no necesita implementarlo, puede usar los algoritmos disponibles en scipy.integrate de la librería de python o cualquier otro que encuentre y que sea de uso libre.
Grafique en 3D la solución (x(t), y(t), z(t)).

2 Procedimiento

Para empezar, se realiza el cambio de variable $y(s) = x(t)/a$ y $s = t\sqrt{k}$ con lo que la ecuación se modifica a lo pedido.
Para integrar la ecuación primero se requiere transformar esta a una de primer orden, esto se realiza usando vectores

$$\frac{d}{ds}\begin{pmatrix} y \\ dy(s)/ds \end{pmatrix} = \begin{pmatrix} dy(s)/ds \\ d^2y(s)/ds^2 \end{pmatrix}$$

Teniendo esto en cuenta, se puede escribir una funcion que reciba dos valores, en este caso "y" y su derivada devolviendo el vector derivado donde la segunda componente del ultimo vector será la ecuacion de onda. Para resolver dicha ecuacion se utiliza el método Runge-Kutta el cual es capaz de calcular el valor de yn+1 teniendo el valor anterior yn por medio de la fórmula:

$$y_{n+1} = y_n + (k_1 + 4k_2 + k_3)$$

Donde:

$$k_1 = f(t_n, y_n)$$
$$k_2 = f\left(t_n + h/2\,, y_n + k_1/2\right)$$
$$k_3 = f\left(t_n + h\,, y_n - k_1\, - 2k_2\right)$$

Omitimos la variable tiempo pues esta implícita en "s". Por último al definir estas funciones, es decir y_{n+1}, k_1, k_2, k_2, basta tomar un intervalo para s $[0, 20\pi]$ subdividido en 4000, con lo cual podemos definir nuestro "paso" h. Dado un

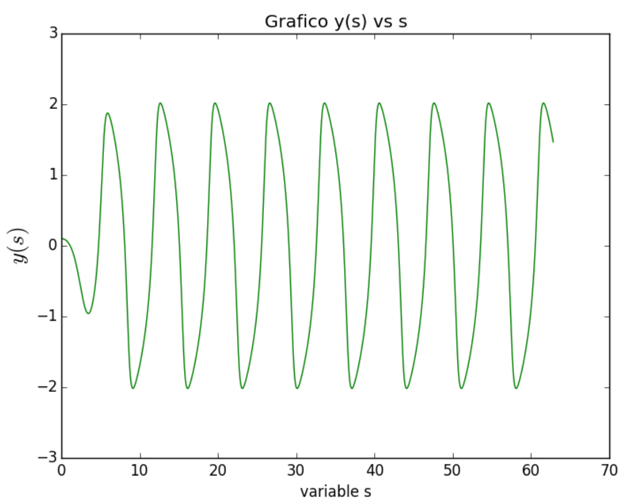
$\mu^* = 1.376$ graficamos ”y vs s” y ”dy/ds vs y” para las condiciones iniciales dadas.
Con respecto a Lorentz, primero definimos nuestro vector a derivar:

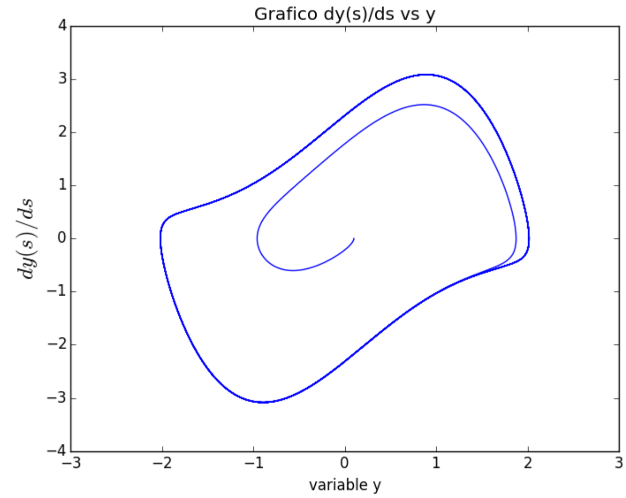
$$\frac{d}{ds}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \sigma(y-x) \\ x(\rho - z) - y \\ xy - \beta z \end{pmatrix}$$

Y por ende definimos una función que reciba x, y, z y nos retorne sus respectivas derivas, luego se definen las condiciones iniciales y los parámetros a trabajar: σ, ρ, β.
Despues construimos el sistema para resolver la ecuacion diferencial ordinaria usando las librerias proporcionadas por python y se crean listas para almacenar los datos luego de usar el el sistema descrito anteriormente. Finalmente se grafican los datos obtenidos usando la libreria mpl_toolkits.mplot3d

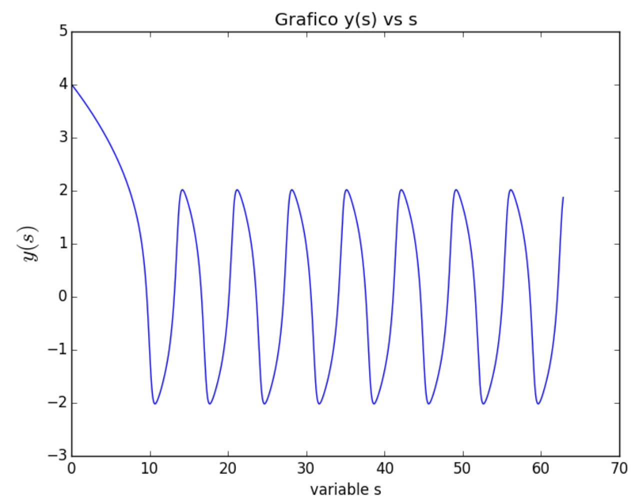
3 Resultados

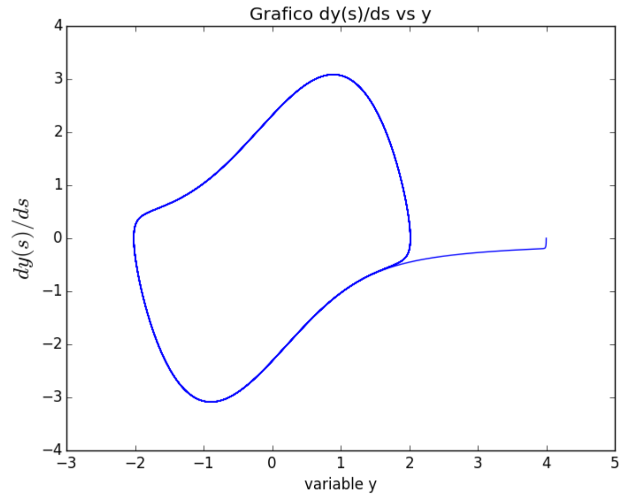
Para la ecuación de Van Der Pool con condiciones iniciales y=0.1 y dy/ds=0 se tiene:



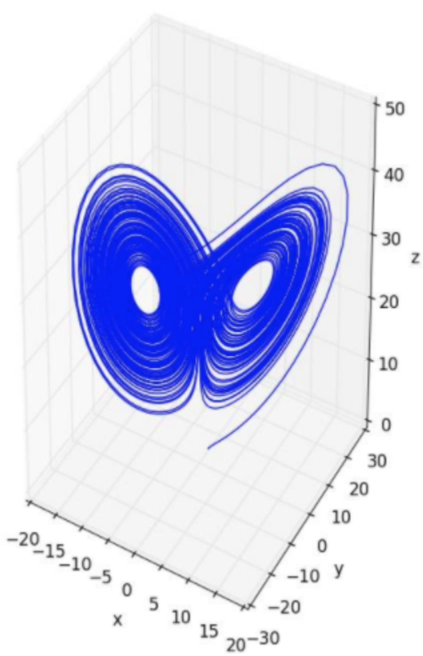


Para la ecuación de Van Der Pool con condiciones iniciales y=4 y dy/ds=0 se tiene:





Para los resultados de lorentz se tiene el siguiente gráfico:



4 Conclusiones

Se observa que la propia implementación de Runge-kutta permite resolver las ecuaciones planteadas en la introducción y que su posterior uso desde las librerías puede manejarse controlando el error y el tiempo de ejecución. Respecto a las condiciones iniciales para el osciladore de van der pool, y(s) tiende a volver al punto de equilibrio sin importar su condición inicial observandose además 4 períodos, el equilibrio también se puede observar en dy/ds vs y. Respecto al oscilador de Lorentz su resultado se graficó de forma satisfactoria, y la amplitud que tienen ambas secciones es directamente proporcional a las condiciones iniciales que se proporcionen.