FI3104 Métodos Númericos para la Ciencia e Ingenieria
Tarea 3

Camila Sandivari
18934657-3
Profesor: Valentino Gonzalez
Profesor Auxiliar: Felipe Pesce
(Dated: 8 de octubre de 2015)

El presente reporte muestra la propia implementación del algoritmo del método Runge-Kutta de orden 3 para integrar el oscilador de van der Pool, que posee un amortiguamiento no lineal, por esto usaremos condiciones iniciales que nos muestran los ciclos límites. Por otro lado se muestra el uso del algoritmo implementado en ode de la librería scipy.integrate para el método Runge-Kutta de orden 4 para integrar el sistema de lorenz (sistema de ecuaciones diferenciales ordinarias) y visualizar particularmente el atractor de Lorenz que se deriva de el modelo climático de Lorenz, más especificamente de los rollos de convección en la atmósfera terrestre. Este es un fenómeno muy interesante y ejemplo clásico de un atractor caótico, es decir con gran sensibilidad a las condiciones iniciales. Por esto no podemos determinar el tiempo con gran exactitud ni a largo plazo, las pequeñas variaciones en las condiciones iniciales del sistema climático lo convierte en un misterio si intentamos mirar más allá de un par de días.
Ambos sistemas fueron importantes aportes a la teoría del Caos que a pesar de ser determinista rigurosamente hablando, nos abre las puertas a reconocer que el mundo que nos rodea es un sistema caótico, nosotros mismos somos un sistema caótico y por suerte nos es imposible conocer o repetir con exactitud una condición inicial.

Procedimiento

Parte 1 Para resolver el oscilador de van der Pool debemos integrar la ecuación que describe su movimiento

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

Para esto implementaremos el método de Runge-Kutta para orden 3 que consiste en utilizar la ecuación (2) para buscar los valores de $y(x_{n+1})$ según la deducción de la ecuación (3)

$$y(x_{n+1}) - y(x_n) = \int_{x_n}^{x_{n+1}} f(s, y(s))ds \tag{2}$$

$$\int_{x_n}^{x_{n+1}} f(s, y(s))ds \approx \frac{h}{6}(k_1 + 4k_2 + k_3) \tag{3}$$

Siguiendo con la deducción de y_{n+1} según el método

$$y_0 = y(0) \tag{4}$$
$$k_1 = f(x_n, y_n) \tag{5}$$
$$k_2 = f(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \tag{6}$$
$$k_3 = f(x_n + h, y_n - k_{1-2k_2}) \tag{7}$$
$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 4k_2 + k_3) \tag{8}$$

En el caso particular de la ecuación (1) se debe aplicar el método dos veces simultáneamente, para obtener la velocidad y a partir de la velocidad obtener la posición. Para esto se implementa una función que recibe condiciones iniciales y entrega el arreglo de posiciones y velocidades.

$$1)\frac{dy}{ds} = 0; y = 0{,}1 \tag{9}$$
$$2)\frac{dy}{ds} = 0; y = 4{,}0 \tag{10}$$

Parte 2 La segunda parte consiste en la utilización del método de Runge-Kutta de orden 4 para resolver el sistema de ecuaciones diferenciales ordinarias que componen el sistema de Lorenz

$$\frac{dx}{ds} = \sigma(y - x) \tag{11}$$
$$\frac{dy}{ds} = x(\rho - z) - y \tag{12}$$
$$\frac{dz}{ds} = xy - \beta z \tag{13}$$

Al utilizar los parámetros $\sigma = 10$,$\beta = 8/3$,$\rho = 28$,se integra con el método RK4 implemetado en la librería scipy.integrate, se obtiene como solución el atractor de Lorenz . ésto se hace mediante ode ('dotri5') que recibe la función a integrar, en este caso un sistema al que se le entregan los vectores que corresponden a las variables espaciales, dentro de un solo vector, y el tiempo. El integrador retorna el tiempo y un vector con las soluciones para los vectores entregados. Se pueden entregar condiciones iniciales y graficar en 3D , entendiendo los ejes como parte de la clase objeto.

Resultados

Parte 1 Se obtienen los gráficos para el espacio de fase en la figura 1, y la trayectoria en la figura 2, del oscilador de Van der Pool

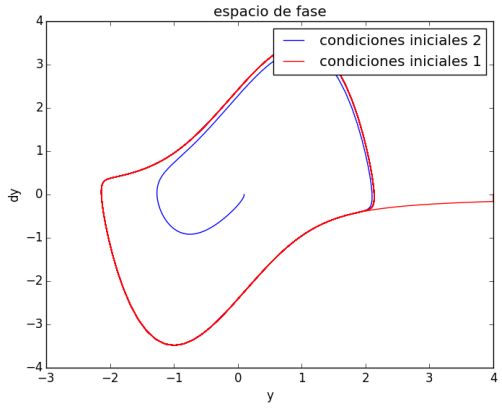

Figura 1. Espacio de fase oscilador van der Pool

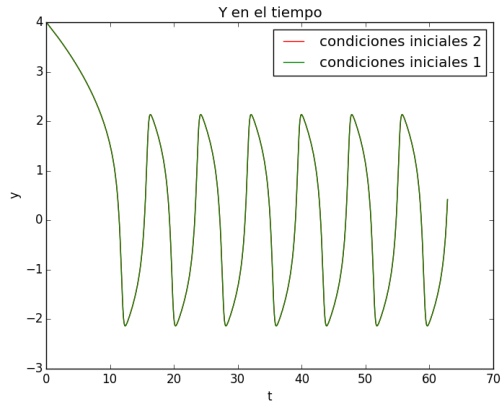

Figura 2. Trayectoria versus tiempos oscilador van der Pool

Parte 2 Se obtiene para la solución del atractor de Lorenz, un gráfico utilizando las condiciones iniciales $[1, 1, 1]$ para las posiciones $[x, y, z]$

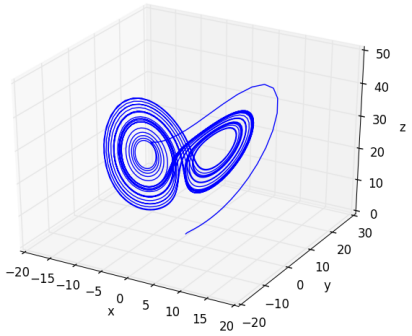

Figura 3.

Conclusiones El método Runge-Kutta tanto en la versión implementada como la existente en librerías nos permite abordar problemas de integración bastante complejos y observar o visualizar el comportamiento de sistemas en este caso caóticos que pueden ser de interés. Se obtienen resultados que coinciden con los resultados obtenidos históricamente en la resolución de estos sistemas.