Métodos Numéricos para la Ciencia e Ingeniería

Informe Tarea 3

Nelson Soto Medina RUT 18694312-0

Pregunta 1:

En esta pregunta se introduce la ecuación del oscilador de van der Pool:

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

A la que hay que aplicar el siguiente cambio de variable:

$$y = \frac{x}{a} \ , \ s = \frac{t}{\sqrt{k}}$$

Para tener finalmente:

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

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

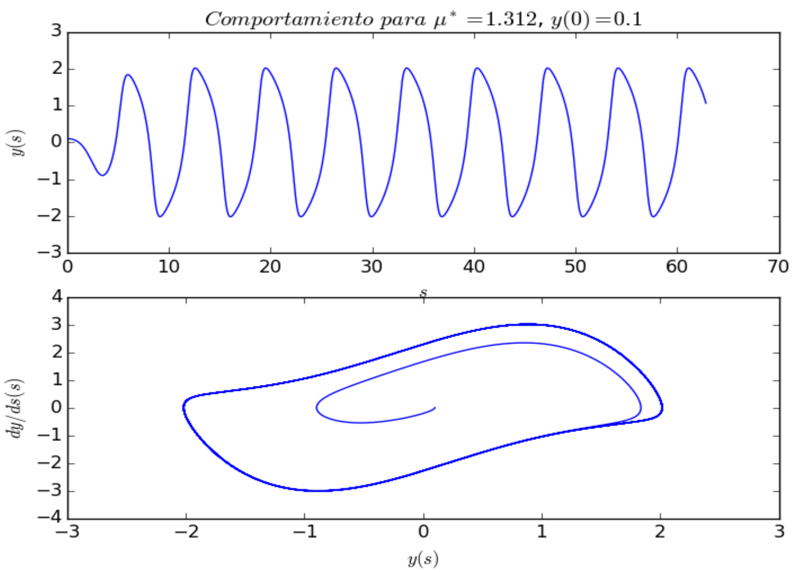
Para esta pregunta se pide integrar la función utilizando el método de Runge-Kutta de orden 3 visto en clases, utilizando un algoritmo propio y no usando uno de internet.

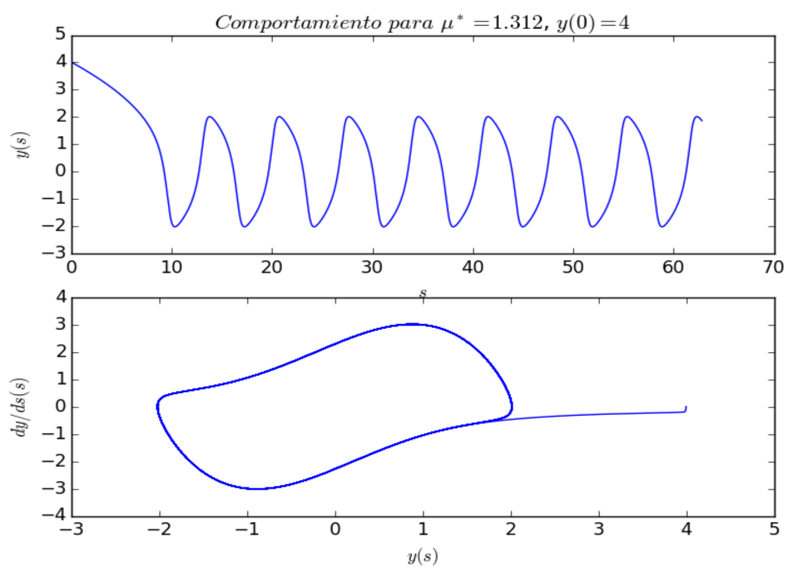
Para eso definí las funciones de van der Pool, k1, k2, k3 y finalmente la que calcula con Runge-Kutta utilizando las anteriores:

```
def VanDerPool(y, dy):
    return dy, -y-u*((y**2)-1)*dy

def k1(Xn, Yn, f):
    F=f(Xn, Yn)
    return h*F[0], h*F[1]

def k2(Xn, Yn, f):
    K1=k1(Xn, Yn, f)
    F=f(Xn+(h/2), Yn+(1/2)*K1[1])
    return h*F[0], h*F[1]

def k3(Xn, Yn, f):
    K1=k1(Xn, Yn, f)
    K2=k2(Xn, Yn, f)
    F=f(Xn+h, Yn-K1[1]-2*K2[1])
    return h*F[0], h*F[1]

def RK3(Xn, Yn, f):
    K1=k1(Xn, Yn, f)
    K2=k2(Xn, Yn, f)
    K3=k3(Xn, Yn, f)
    Xn1=Xn+(1/6)*(K1[0]+4*K2[0]+K3[0])
    Yn1=Yn+(1/6)*(K1[1]+4*K2[1]+K3[1])
    return Xn1, Yn1
```

Dado que los 3 últimos dígitos de mi RUT antes del guión son 312, μ^* toma el valor de 1.312.

Con el tiempo igual a 10 períodos (20π), discreticé a 30000 pasos (de $20\pi/30000$ cada uno), para integrar la función utilizando las funciones mencionadas, obteniendo los siguientes gráficos para cada condición inicial pedida:





Donde se muestra y(s) y la trayectoria en el espacio (y, dy/ds).

Se puede apreciar que sin importar la condición inicial, habiendo pasado cierta cantidad de tiempo el sistema tiende a estar en un equilibrio, manteniendo una trayectoria específica y oscilando siempre en torno al mismo punto.

Pregunta 2:

En esta pregunta se presenta el sistema de Lorenz, más específicamente el llamado atractor de Lorenz, cuyo sistema de ecuaciones es:

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

Se pide integrar la función utilizando los parámetros con los que se obtiene la función más famosa, con un set de condiciones iniciales a elección y por un tiempo que se estime conveniente.

Los parámetros son σ=10, β=8/3 y ρ=28.

Primero definí la función Atractor de Lorenz:

```
def AtractorDeLorenz(t,xyz):
    x, y, z=xyz
    dx=10*(y-x)
    dy=x*(p-z)-y
    dz=x*y-b*z
    return [dx, dy, dz]
```

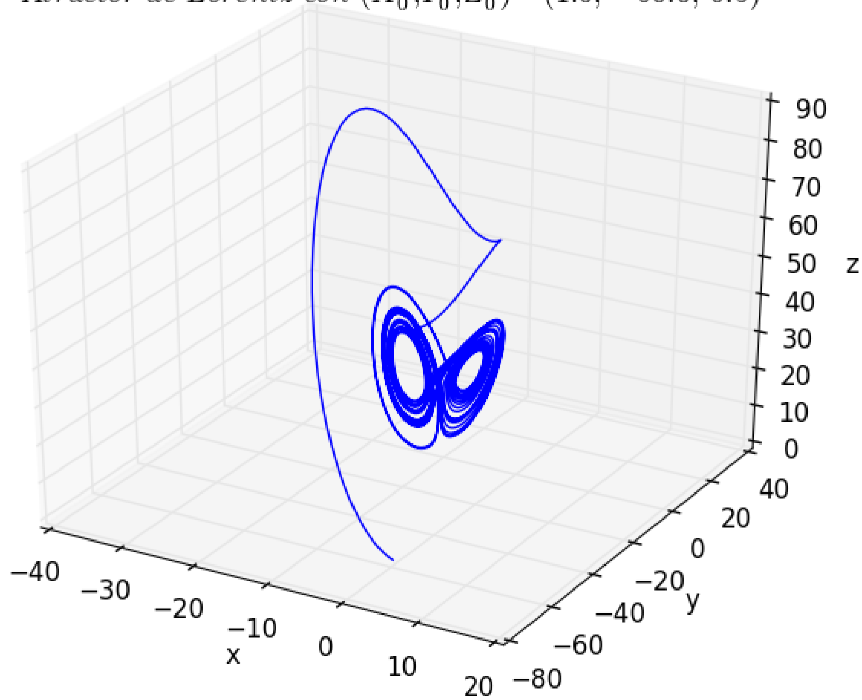
Luego utilicé el método “ode” de scipy.integrate para integrar la función.

Se elige el integrador ‘dopri5’, que corresponde al Runge-Kutta de orden (4)5, dejando como paso máximo 0.01.

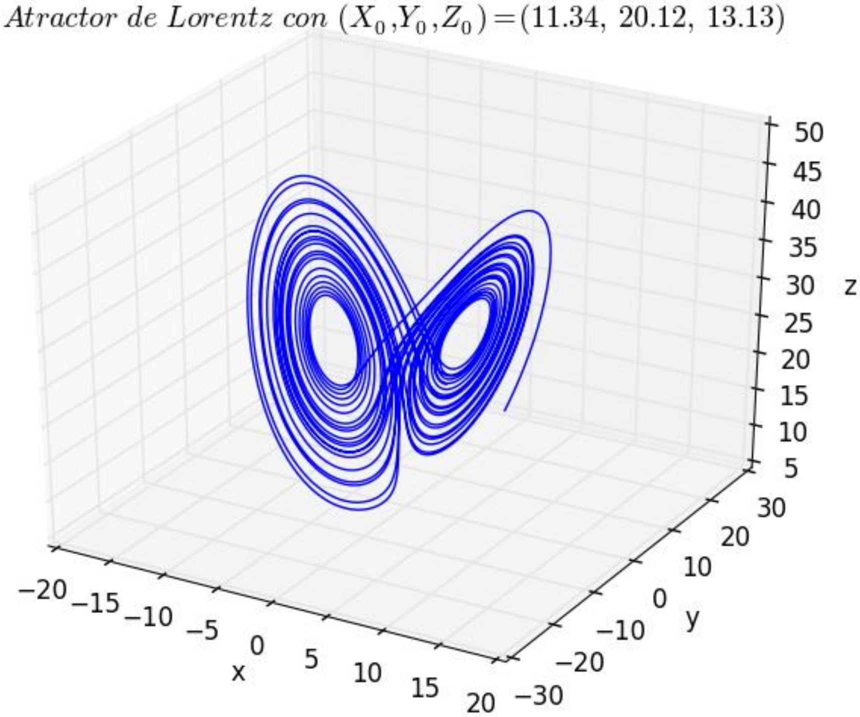
Entonces se integra con un tiempo de 0 a 30, a pasos de 30/10000.

Obteniendo, con diferentes condiciones iniciales:

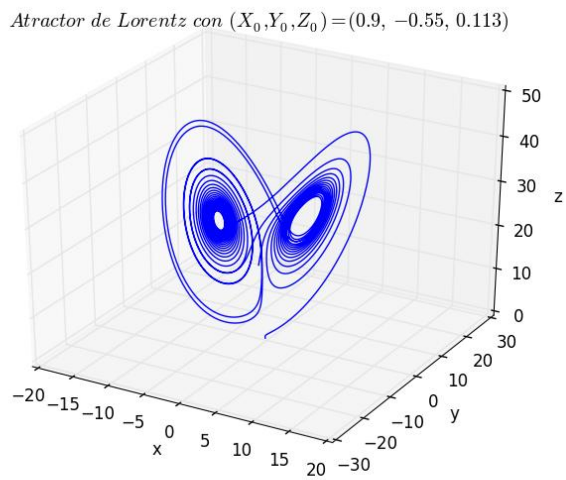
Atractor de Lorentz con $(X_0, Y_0, Z_0) = (1.0,\ -60.0,\ 0.5)$

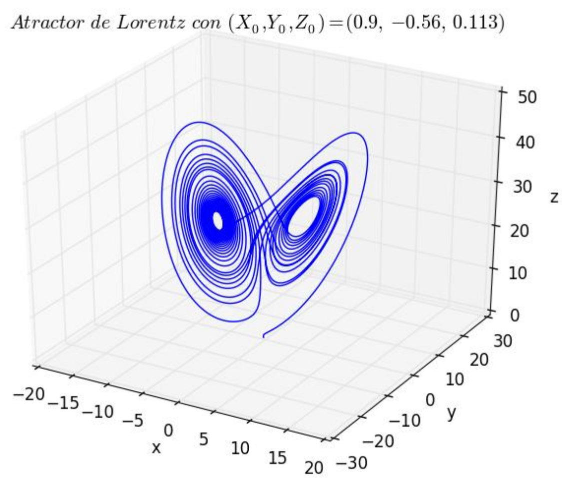


Atractor de Lorentz con $(X_0, Y_0, Z_0) = (11.34,\ 20.12,\ 13.13)$



Y estas 2 para condiciones iniciales parecidas:





Finalmente se puede apreciar que la solución "rota" en torno a dos puntos y forma algo como un ciclo y que un pequeño cambio se puede ver cómo varía.

Adjunto a este informe y el programa, además de las imágenes agregué un gif, con las condiciones iniciales $X_0 = 10$ $Z_0 = 7$ y Y_0 variando entre 5 y 5.0002, y se pueden ver algo parecido a unas pulsaciones y como se carga hacia un lado.

Es interesante como con un cambio tan pequeño se pueden apreciar tanta variación en la función.