

Trabajo Práctico Python/Programación

Matemática III Econ., turno noche, 2019 cuat. 1.

IDEI-UNGS.

El aprendizaje no es un deporte para espectadores.

D. Blocher

Autor: G. Sebastián Pedersen — sebasped@gmail.com — Vie 17-May-2019.

Código ejemplo.

El siguiente programa en Python es para tomar como ejemplo para resolver el TP. El programa resuelve numéricamente el siguiente sistema de EDOs no lineal (como siempre $x = x(t)$ e $y = y(t)$):

$$\begin{cases} x' &= x^2 - y \\ y' &= 1 - y \end{cases}$$

sujeto a las condiciones iniciales $x(0) = -0,9$ e $y(0) = 1,1$.

El programa resuelve el sistema para $0 \leq t \leq 10$, y luego da la posibilidad de graficar t vs. x , t vs. y , o bien x vs. y . A continuación el código:

```
# -*- coding: utf-8 -*-
# Importar los paquetes necesarios
from numpy import linspace
from matplotlib import pyplot as plt
from scipy.integrate import solve_ivp

# definir la ecuación diferencial o el sistema
def f(t, M):
    #M es una lista
```

```
x=M[0] #la primera posición es la x
y=M[1] #la segunda posición es la y
```

```
# defino el sistema x' = x^2 - y
#                      y' = 1 - y
[dxdt, dydt] = [x**2-y, 1-y]
```

```
return [dxdt, dydt]
```

```
# rango de tiempo donde deseo resolver
# es tiempo inicial, tiempo final, cantidad de pasitos entre inicio y fin
tspan = linspace(0,10,100) #acá serían 100 pasos para ir de t=0 a t=10
```

```
# condiciones iniciales. Sería [x(0), y(0)]
cond_inic = [-0.9, 1.1]
```

```
#resolver la ecuación o sistema
sol = solve_ivp(lambda t, M: f(t, M), [tspan[0],tspan[-1]], cond_inic, t_eval=tspan)
```

```
# si lo prueban online y no pueden graficar
# al menos impriman el resultado (descomenten los prints)
```

```
# imprimir los valores de x(t)
#print('Los valores de x\n:')
#print(sol.y[0])
```

```
# imprimir los valores de y(t)
#print('Los valores de y\n:')
```

```
#print(sol.y[1])

#descomentar y comentar, dependiendo de qué se quiere graficar

#graficar t vs. x(t)
plt.plot(sol.t, sol.y[0], 'k--s')

#graficar t vs. y(t)
plt.plot(sol.t, sol.y[1], 'k--s')

#graficar x(t) vs. y(t)
plt.plot(sol.y[0], sol.y[1], 'k--s')

plt.show()
```

Enunciado del TP.

Resolver un sistema de EDOs no lineal, siguiendo estos lineamientos:

1. Elegir un sistema de EDOs no lineal de la práctica.
2. Justificar la elección realizada en 1. (mínima, corta y simple explicación de por qué está bueno resolver el sistema elegido).
3. Resolver el sistema a partir del código ejemplo anterior.
4. Variar las condiciones iniciales del sistema, y utilizar los gráficos del programa para analizar la estabilidad dinámica, a partir de los siguientes casos:
 - a) Las condiciones iniciales son iguales a (algún) punto de equilibrio.
 - b) Las condiciones iniciales están “cerca” de (algún) punto de equilibrio.
 - c) Las condiciones iniciales están “lejos” de cualquier punto de equilibrio.

En todos los casos, tener en cuenta que también quizás sea necesario modificar el rango de tiempo en donde se resuelve el sistema.

Pautas del TP.

El trabajo práctico deberá ser realizado a partir de los siguientes lineamientos:

1. Se deberá hacer en grupo de hasta 2 (dos) integrantes.
2. Se deberá entregar el código Python utilizado.
3. Se podrán además entregar todas las aclaraciones extras que se deseen o se consideren relevantes (por ejemplo, el sistema de EDOs elegido, algunas explicaciones sobre la resolución del punto 3. del enunciado, etc.)

4. Las entregas podrán ser mediante cualquier medio digital y se puede utilizar más de uno (pdf, documento de texto, archivo Python, foto, etc.). La responsabilidad de que el material entregado sea entendible corre por cuenta y orden de lxs integrantes del grupo.
5. Posterior a la entrega del TP, se podrá requerir a lxs integrantes del grupo (juntos o por separado), que expliquen o amplíen cualquier cosa en relación a la resolución entregada.
6. La entrega deberá ser por correo electrónico a **sebasped@gmail.com** antes del Lunes 20-May-2019 a las 23:59 hs. (hora local).