

Universidad Central de Venezuela

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

Gráficos con jupyter

Periodo 2020-3

16/11/2020

Prof. Gilberto R. Noguera

Introducción

Se suministra el presente informe a los participantes del curso Cálculo Numérico para que sirva de guía con miras a los informes que deberán entregar como uno de los productos de las diferentes prácticas que se contemplan durante este periodo académico.

En este trabajo se consideran algunos aspectos básicos relacionados con gráficos de funciones en 2D utilizando Python. Se utilizó para la redacción Jupyter, herramienta elegida para que los estudiantes elaboren los informes de práctica y reportes de proyectos.

Marco Teórico

Las gráficas se emplean para representar funciones, ecuaciones, modelos matemáticos o situaciones que expresadas de manera gráfica ayudan a su interpretación.

Recordemos que en matemática $y = f(x)$ nos indica que se trata de una función cuyo dominio se asocia a la variable x y el rango o imágenes a la variable y . En principio estas imágenes se obtienen al realizar los cálculos especificados por la regla de asignación expresada en $f(x)$.

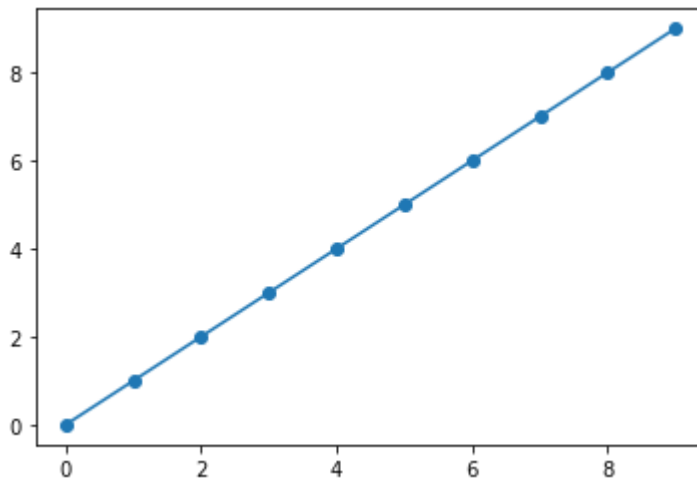
Ejemplos de funciones:

- 1. $y = f(x) = x$ tal que $x \in \mathbb{Z}$, una función discreta.*
- 2. $y = f(x) = x$ tal que $x \in \mathbb{R}$, una función continua.*

Práctica

1) Se procede a importar el módulo **pylab** y todas sus funciones.

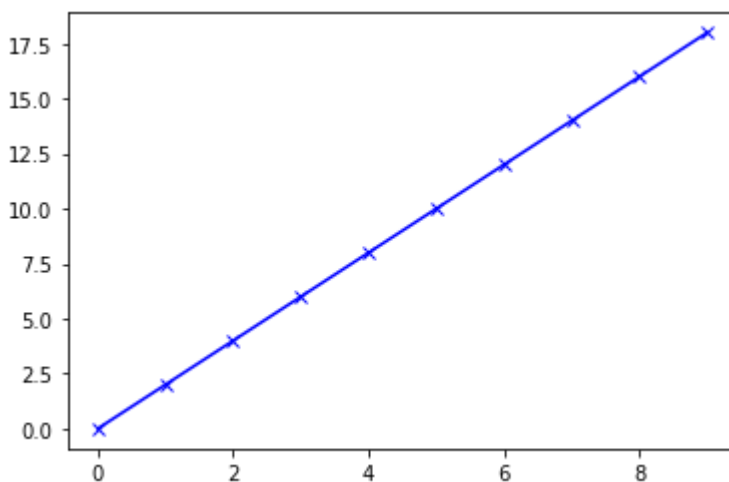
```
In [4]: from pylab import *  
x = arange(10,)  
dibujo_1, = plot(x, 'o-')
```



Análisis Se representa una función de variable discreta cuyos puntos se ha unido con una recta, La función se ha definido de manera implícita con el comando **plot**. Acá el dominio de la función es $\{x \in \mathbb{Z} : x = 1, \dots, 10\}$ y el rango es $\{y \in \mathbb{Z} : y = 1, \dots, 10\}$, es claro que la regla de asignación es $y = x$.

2) Se utiliza el código anterior y se importa la módulo (librería) **numpy** que permite realizar operaciones matemáticas

```
In [5]: from pylab import *
from numpy import *
x = arange(10,)
dibujo_2, = plot(x*2, 'bx-')
```

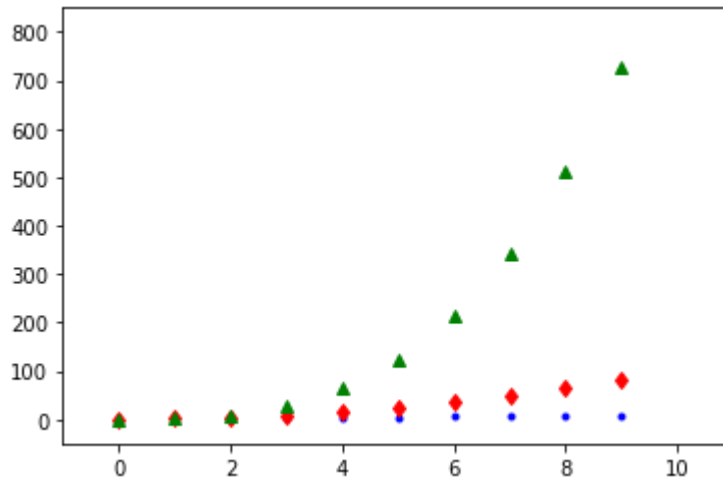


Análisis Al igual que en el numeral, 1), estamos dibujando una función discreta en la que se ha definido de manera implícita a través del comando **plot** la función $f(x) = x^2$. Los puntos resultantes se unen con una recta y se cambia el estilo de los puntos.

3) La instrucción **clf()** se utiliza para borrar de memoria todo lo relacionado con las figuras anteriores, observe que como ya se importaron y ejecutaron las librerías que nos permiten realizar gráficas y operar sobre variables, en este caso, no son llamadas. Por otra parte se establecen límites inferior y superior a los ejes de coordenadas.

```
In [6]: clf()
x2 = x**2
x3 = x**3
plot(x,x,'b.',x,x2,'rd',x,x3,'g^')
xlim(-1,11)
ylim(-50,850)
```

Out[6]: (-50.0, 850.0)

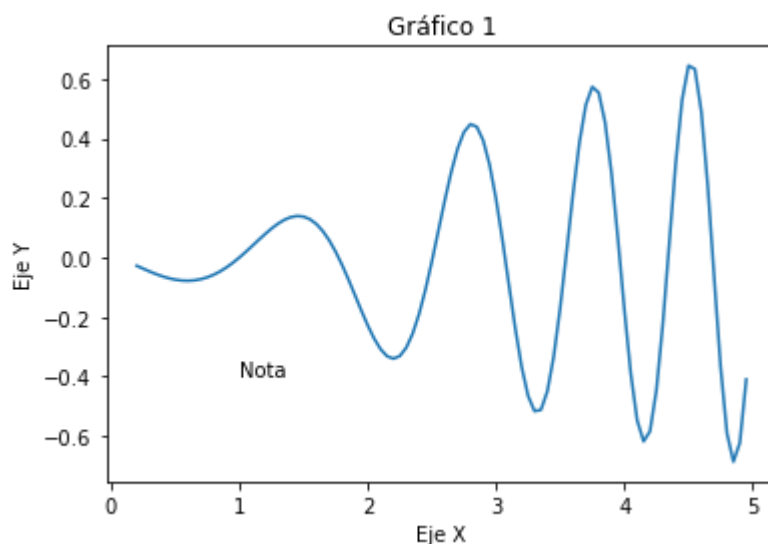


Análisis Se observa que se definen en línea dos funciones; x_2 y x_3 y de manera implícita, por decir algo en contexto, $x_1 = x$. Además se le han asignado a cada gráfica estilos de puntos: a $x_1 = x$ puntos azules, a $x_2 = x^2$ rombos rojos y a $x_3 = x^3$ triángulos verdes.

4) En este caso se representa la función $y = f(x) = \log_{10} \sin(x^2)$ y se agrgan a la gráfica: etiquetas para los ejes, título para la figura y una nota.

```
In [8]: x = arange(0.2,5,0.05)
p, = plot(x,log10(x)*sin(x**2))
xlabel('Eje X')
ylabel('Eje Y')
title('Gráfico 1')
text(1,-0.4,'Nota')
```

Out[8]: Text(1, -0.4, 'Nota')



Análisis Aquí se representa una función continua definida, particularmente, en el intervalo $[0, 5] \in \mathbb{R}$. Para esto se toma una muestra de 100 puntos, en efecto $\frac{5-0}{0.05} = 100$. Se

etiquetan los ejes de coordenadas, se asina un título a la gráfica y se coloca una nota dentro de la figura.

5) Importamos las librerías *math* que provee las funciones matemáticas para trabajar en el campo de los números reales y *numpy* que da soporte para crear vectores y matrices. Y de *matplotlib* importamos *pyplot* que es un conjunto de funciones que nos permiten trabajar las gráficas siguiendo el estilo del software *Matlab*.

```
In [9]: from math import *
from numpy import *
import matplotlib.pyplot as plt
t = arange(0.1, 20, 0.1)

y1 = sin(t)/t
y2 = sin(t)*exp(-t)
p1, p2 = plot(t, y1, t, y2 )

text1 = text(2,0.6,r'$f_1(t)=\frac{\sin(t)}{t}$', fontsize = 14)
text2 = text(13,0.2,r'$f_2(t)=\sin(t) \cdot e^{-t}$', fontsize = 14)

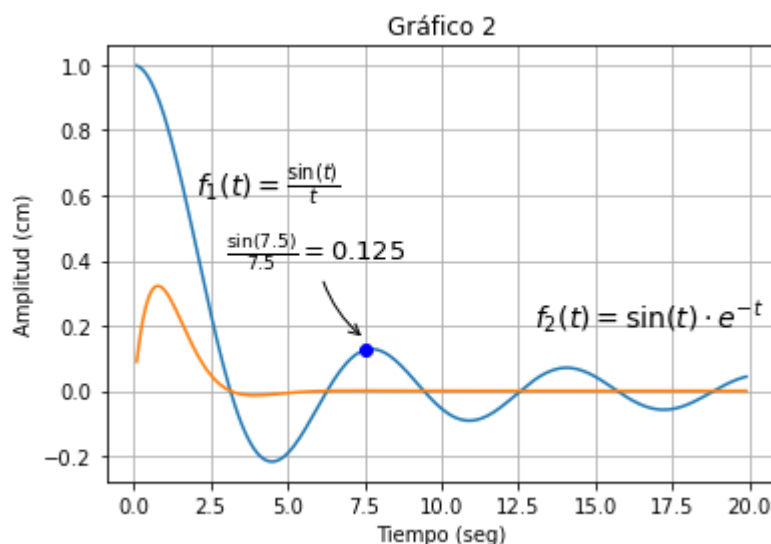
grid()

title('Gráfico 2')
xlabel('Tiempo (seg)')
ylabel('Amplitud (cm)')

px = 7.5
py = sin(px)/px

Pto = plot([px], [py], 'bo')

nota = plt.annotate(r'$\frac{\sin(7.5)}{7.5} = 0.125$',
                    xytext = (3, 0.4),
                    textcoords = 'data',
                    xy = (7.5,0.16),
                    xycoords = 'data',
                    fontsize = 13,
                    arrowprops = dict(arrowstyle = '->',
                                    connectionstyle = "arc3, rad=0.2"))
```



Análisis Se definen en línea dos funciones $y_1(t)$ y $y_2(t)$ y se grafican tomando en cuenta sus instancias. Se han agregado dos textos, usando *L^AT_EX*, en ellos se especifican en

lenguaje matemático las dos funciones $f_1(t)$ y $f_2(t)$. Por otra parte se evalúa $f_1(7.5)$ y se dibuja el punto sobre la gráfica correspondiente y se inserta como una nota usando la función **annotate** dibujando un arco para señalar el punto en cuestión, además se establece el tamaño de las letras con **fontsize**. Se etiquetan los ejes de coordenadas y se adiciona un título. Se coloca también una cuadrícula con la función **grid()**.

6) Se definen tres funciones $f_1(x)$, $f_2(x)$ y $f_3(x)$ que codificadas en un script pueden ser llamadas por cualquier programa que las requiera para un fin determinado. Las tres funciones tienen como argumento de entrada un número real y devuelve un número real en la variable y . El dominio de las funciones está dado por la instrucción **linspace** en el intervalo $[0, 10\pi] \in \mathbb{R}$.

```
In [10]: def f1(x):
          y = sin(x)
          return y

          def f2(x):
            y = sin(x) + sin(5.0*x)
            return y

          def f3(x):
            y = sin(x) * exp(-x/10.)
            return y

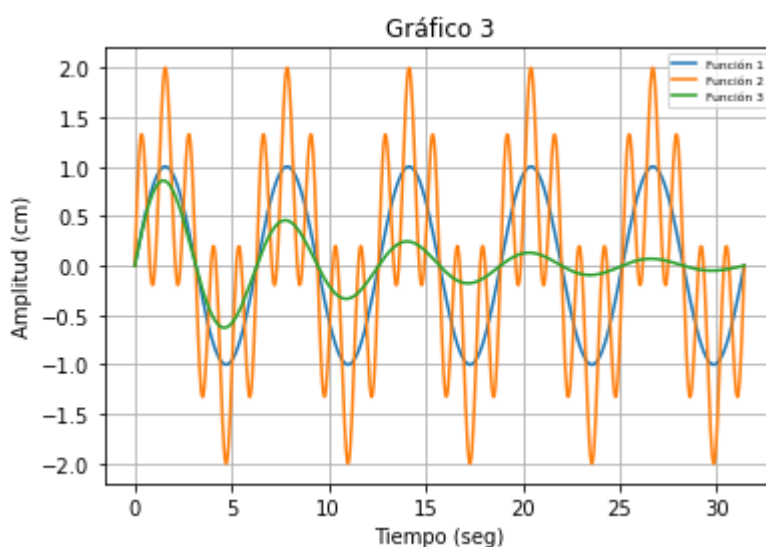
          x = linspace(0, 10*pi, 800)

          p1, p2, p3 = plot(x, f1(x), x, f2(x), x, f3(x))

          legend(['Función 1', 'Función 2', 'Función 3'],
                prop = {'size':6}, loc = 1)

          title('Gráfico 3')
          xlabel('Tiempo (seg)')
          ylabel('Amplitud (cm)')

          grid()
```



Análisis La gráfica muestra tres funciones reales de variable real. Se han etiquetado los ejes de coordenadas, adicionado un título, agregando una leyenda en la esquina superior derecha y una cuadrícula.

Conclusiones

Es importante resaltar que este trabajo sólo es un ejemplo para ilustrar la estructura de los documentos que en el futuro deberán entregar los participantes como informes de las diferentes prácticas a realizar en el periodo 2020-3, de manera que al no ser las gráficas resultado de un experimento en particular lo que se refleja en los análisis es más bien una descripción, aspecto que cambia radicalmente al tratarse de prácticas formales con experimentos bien definidos.

Por otra parte se muestra el poder del IDE (interactive development environment) que permite combinar código de Python con texto enriquecido, entre otras bondades.

*Se debe mencionar la posibilidad de exportar los **notebook's** a otros formatos como PDF, etcétera.*

Referencias

Acá se colocan citas a libros, páginas web, etc...

Reconocimientos

En este apartado se debe agradecer a las personas (ingenieros, matemáticos, compañeros coparticipes o no del curso) y/o instituciones que les ayuden en la elaboración de su trabajo.

Anexos

*Se muestra una captura de pantalla que muestra la organización se crea una carpeta **CN** y dentro de ella las carpetas de las diferentes prácticas.*

Dentro de cada carpeta de práctica se colocan: el notebook, script e imágenes.

