

# Actividad 3: Interpolación

Paulina Valenzuela Coronado

February 6, 2016

## 1 Introducción

En la mayoría de los fenómenos de la naturaleza observamos una cierta regularidad en la forma de producirse, esto nos permite sacar conclusiones de la marcha de un fenómeno en situaciones que no hemos medido directamente. La **interpolación** consiste en hallar un dato dentro de un intervalo en el que conocemos los valores en los extremos.[1] La idea de la interpolación es poder estimar  $f(x)$  para un  $x$  arbitrario, a partir de la construcción de una curva o superficie que une los puntos donde se han realizado las mediciones y cuyo valor si se conoce.[2] En esta actividad se aprendió el proceso de encontrar una función que interpole todos los puntos, apoyados en la función `scipy.interpolate` de Python. El módulo `scipy.interpolate` es útil para ajustar una función a partir de datos experimental y evaluar los puntos que se requieren. El módulo se basa en *FITPACK* Fortran subroutines del proyecto *netlib*. [3]

## 2 Programas

Se realizó una serie de programas para interpolar con  $n$  números aleatorios entre un rango determinado, una función  $f(x)$ .

- Programa 1

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d

xn = np.random.random(10)
x0 = xn*3
y0 = sin(2*x0)

plt.plot(x0, y0, 'o', label='Data')

x = np.linspace(min(x0), max(x0), 101)

options = ('linear', 'quadratic', 'cubic')

for o in options:
    f = interp1d(x0, y0, kind=o)
    plt.plot(x, f(x), label=o)

plt.legend()
plt.show()
```

```
In [32]: import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d

xn = np.random.random(10)
x0 = xn*3
y0 = sin(2*x0)

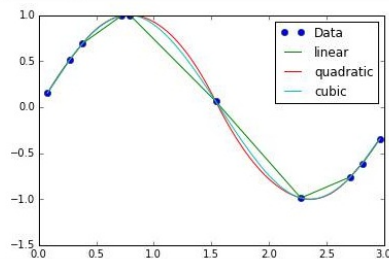
plt.plot(x0, y0, 'o', label='Data')

# Array with points in between those of the data set for interpolation.
x = np.linspace(min(x0), max(x0), 101)

# Available options for interp1d
options = ('linear', 'quadratic', 'cubic')

for o in options:
    f = interp1d(x0, y0, kind=o) # interpolation function
    plt.plot(x, f(x), label=o) # plot of interpolated data

plt.legend()
plt.show()
```



## • Programa 2

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d

xn = np.random.random(20)
x0 = (x.-(0.5))*20
y0 = (sin(x0))/x0

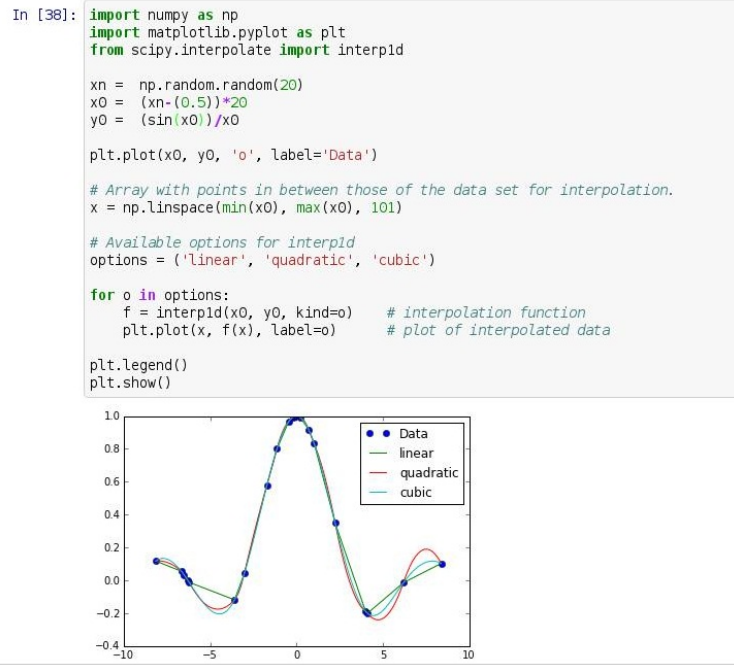
plt.plot(x0, y0, 'o', label='Data')

x = np.linspace(min(x0), max(x0), 101)

options = ('linear', 'quadratic', 'cubic')

for o in options:
    f = interp1d(x0, y0, kind=o)
    plt.plot(x, f(x), label=o)

plt.legend()
plt.show()
```



### • Programa 3

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d

xn = np.random.random(16)
x0 = (xn-(0.5))*6
y0 = ((x0)**2)*(sin(2*x0))

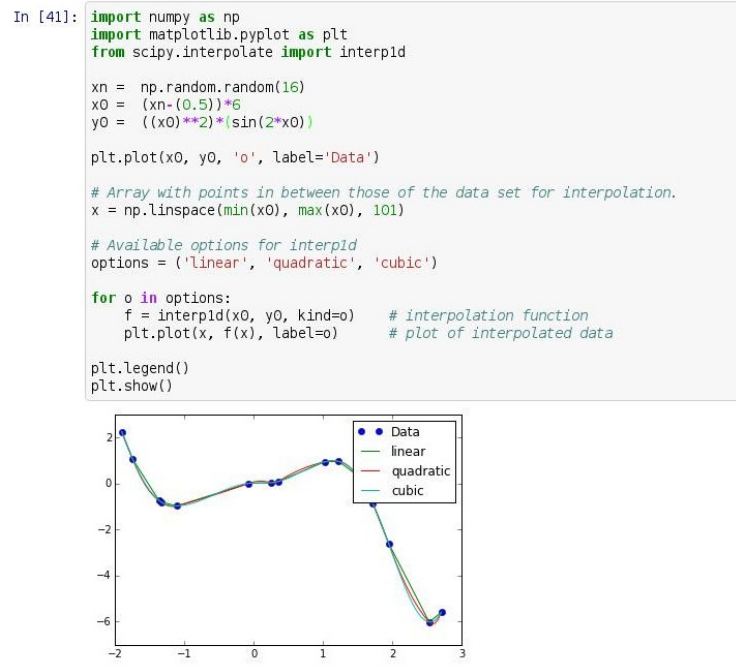
plt.plot(x0, y0, 'o', label='Data')

x = np.linspace(min(x0), max(x0), 101)

options = ('linear', 'quadratic', 'cubic')

for o in options:
    f = interp1d(x0, y0, kind=o)
    plt.plot(x, f(x), label=o)

plt.legend()
plt.show()
```



#### • Programa 4

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d

xn = np.random.random(12)
x0 = (xn-(0.5))*4
y0 = ((x0)**3)*(sin(3*x0))

plt.plot(x0, y0, 'o', label='Data')

x = np.linspace(min(x0), max(x0), 101)

options = ('linear', 'quadratic', 'cubic')

for o in options:
    f = interp1d(x0, y0, kind=o)
    plt.plot(x, f(x), label=o)

plt.legend()
plt.show()

```

```
In [43]: import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d

xn = np.random.random(12)
x0 = (xn-(0.5))*4
y0 = ((x0)**3)*(sin(3*x0))

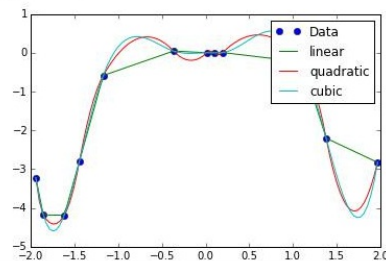
plt.plot(x0, y0, 'o', label='Data')

# Array with points in between those of the data set for interpolation.
x = np.linspace(min(x0), max(x0), 101)

# Available options for interp1d
options = ('linear', 'quadratic', 'cubic')

for o in options:
    f = interp1d(x0, y0, kind=o) # interpolation function
    plt.plot(x, f(x), label=o) # plot of interpolated data

plt.legend()
plt.show()
```



## References

- [1] INTERPOLACIÓN <http://wwwprof.uniandes.edu.co/~gprieto/classes/compufis/interpolacion.pdf>
- [2] INTERPOLACIÓN <http://carmesimatematic.webcindario.com/interpolacion%20lineal.htm>
- [3] INTERPOLACIÓN EN SPICY <http://claudiovz.github.io/scipy-lecture-notes-ES/intro/scipy.html#interpolacion-scipy-interpolate>