

Actividad #2 (Efemérides)

Instructor: Carlos Lizárraga Celaya

Student: Antonio Cota Rodríguez

Introducción

Efemérides

En el estudio de los cuerpos celestes, una efemérides, efeméride o efemeris es una tabla de valores que da las posiciones de los objetos astronómicos en el cielo en un momento o momentos dados. Aunque fue también una de las primeras aplicaciones de las computadoras mecánicas, el término efemérides continúa aplicándose generalmente a una simple tabla impresa.

Interpolación

En el subcampo matemático del análisis numérico, se denomina interpolación a la obtención de nuevos puntos partiendo del conocimiento de un conjunto discreto de puntos.

En ingeniería y algunas ciencias es frecuente disponer de un cierto número de puntos obtenidos por muestreo o a partir de un experimento y pretender construir una función que los ajuste.

Otro problema estrechamente ligado con el de la interpolación es la aproximación de una función complicada por una más simple. Si tenemos una función cuyo cálculo resulta costoso, podemos partir de un cierto número de sus valores e interpolar dichos datos construyendo una función más simple. En general, por supuesto, no obtendremos los mismos valores evaluando la función obtenida que si evaluamos la función original, si bien dependiendo de las características del problema y del método de interpolación usado la ganancia en eficiencia puede compensar el error cometido.

En todo caso, se trata de, a partir de n parejas de puntos (x_k, y_k) , obtener una función f que verifique

$$f(x_k) = y_k, \quad k = 1, \dots, n$$

Programa

En esta actividad aprenderemos un poco del proceso de encontrar una función que interpole todos los puntos, apoyados en la función `scipy.interpolate` de Python.

Con base en el siguiente código :

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

# Original "data set" --- 21 random numbers between 0 and 1.
x0 = np.linspace(-1,1,21)
y0 = np.random.random(21)

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

# Array with points in between those of the data set for interpolation.
x = np.linspace(-1,1,101)

# Available options for interp1d
options = ('linear', 'nearest', 'zero', 'slinear', 'quadratic', 'cubic', 10)
```

```

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()

```

Este programa nos da una gráfica de datos interpolados de distintas formas (cuadrática, cúbica, etc.) para números aleatorios, la gráfica es la siguiente

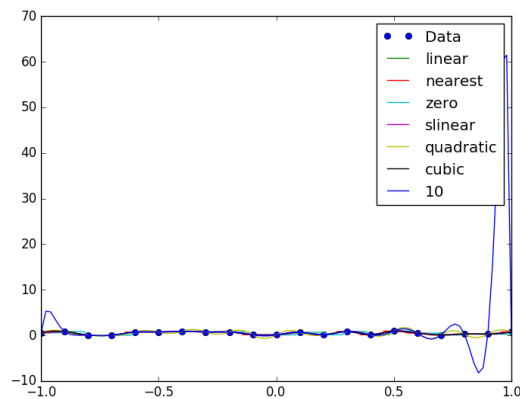


FIGURE 1 – Ejemplo

A continuación se modificará el código anterior para realizar una interpolación lineal, cuadrática y cúbica de los siguientes casos :

- Dados 10 puntos aleatorios entre $x = 0$ y $x = 3$ para la función $f(x) = \sin(2x)$.
- Dados 20 puntos aleatorios entre $x = -10$ y $x = 10$ para la función $f(x) = \sin(x)/x$.
- Dados 16 puntos aleatorios entre $x = -3$ y $x = 3$ para la función $f(x) = x^2 \sin(2x)$
- Dados 12 puntos aleatorios entre $x = -2$ y $x = 2$ para la función $f(x) = x^3 \sin(3x)$

1 $\sin(2x)$

El programa para interpolar esta función es el siguiente :

```

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

# Original "data set" --- 10 random numbers between 0 and 3.
xn = np.random.random(10)
x = xn*3
# Original "Data-set"
y = np.sin(2*x)

plt.plot(x, y, 'o', label='Data')

# Array with points in between those of the data set for interpolation.
xnew = np.linspace(min(x), max(x), 10000)
# Available options for interp1d
options = ('linear', 'quadratic', 'cubic')
for o in options:

```

```
f = interp1d(x, y, kind=o)    # interpolation function
plt.plot(xnew, f(xnew), label=o) # plot of interpolated data

plt.legend(['data', 'linear', 'quadratic', 'cubic'], loc='best')
plt.show()
```

Obtenemos la gráfica

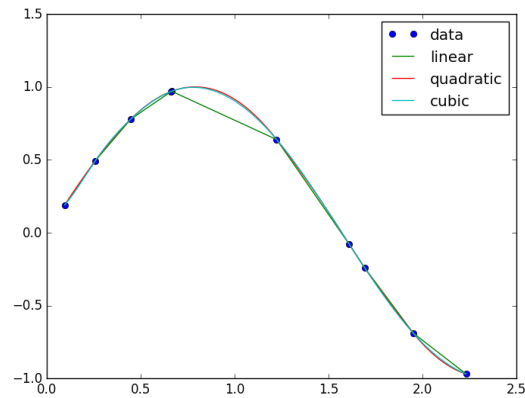


FIGURE 2 – $\sin(2x)$

2 $\sin(x)/x$

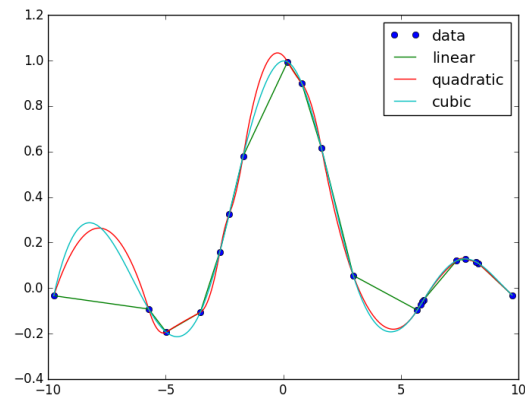
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d

# Original "data set" --- 20 random numbers between -10 and 10.
xn = np.random.random(20)
x = (xn*20)-10
# Original data set
y = np.sin(x)/x

plt.plot(x, y, 'o', label='Data')

# Array with points in between those of the data set for interpolation.
xnew = np.linspace(min(x), max(x), 10000)
# Available options for interp1d
options = ('linear', 'quadratic', 'cubic')
for o in options:
    f = interp1d(x, y, kind=o)    # interpolation function
    plt.plot(xnew, f(xnew), label=o) # plot of interpolated data

plt.legend(['data', 'linear', 'quadratic', 'cubic'], loc='best')
plt.show()
```

FIGURE 3 – $\sin(x)/x$

3 $x^2 \sin(2x)$

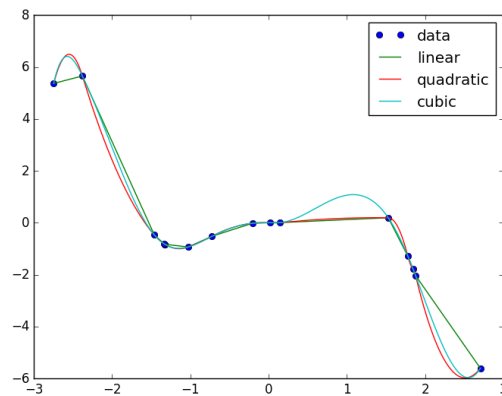
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d

# Original "data set" --- 16 random numbers between -3 and 3.
xn = np.random.random(16)
x = (xn*6)-3
# Original Data-set
y = (x**2)*np.sin(2*x)

plt.plot(x, y, 'o', label='Data')

# Array with points in between those of the data set for interpolation.
xnew = np.linspace(min(x), max(x), 10000)
# Available options for interp1d
options = ('linear', 'quadratic', 'cubic')
for o in options:
    f = interp1d(x, y, kind=o) # interpolation function
    plt.plot(xnew, f(xnew), label=o) # plot of interpolated data

plt.legend(['data', 'linear', 'quadratic', 'cubic'], loc='best')
plt.show()
```

FIGURE 4 – $x^2 \sin(2x)$

4 $x^3 \sin(3x)$

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

# Original "data set" --- 21 random numbers between 0 and 1.
x0 = np.linspace(-1,1,21)
y0 = np.random.random(21)

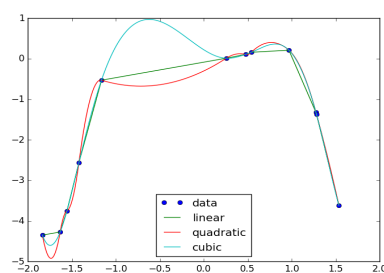
plt.plot(x0, y0, 'o', label='Data')

# Array with points in between those of the data set for interpolation.
x = np.linspace(-1,1,101)

# Available options for interp1d
options = ('linear', 'nearest', 'zero', 'slinear', 'quadratic', 'cubic', 10)

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()
```

FIGURE 5 – $x^2 \sin(2x)$