

Actividad 3: Efemérides

Ana Gabriela Carretas Talamante

04 de febrero de 2016

1. Introducción

En el estudio de los cuerpos celestes, una efemérides, es una tabla de valores que da las posiciones de los objetos astronómicos en el cielo en un momento o momentos dados [1].

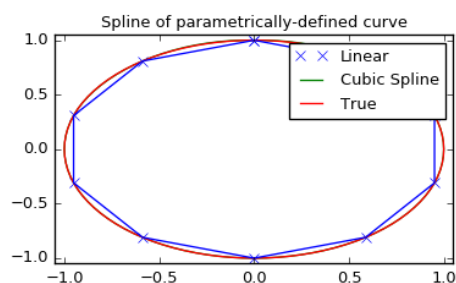


Figura 1: Ejemplo de una interpolación para una curva parametrizada [3].

Para poder tener modelos matemáticos que describan el movimiento periódico de ciertos astros, se recurre a la utilización de interpolaciones con los datos arrojados por las efemérides. Interpolación se refiere a la obtención de nuevos puntos partiendo del conocimiento de un conjunto discreto de puntos [2].

Hacer un programa que estime una función modelo para cierto fenómeno es la encomienda de la tercera actividad. Con respecto al programa base, se hicieron las modificaciones necesarias para poder obtener diferentes resultados.

Programa original

```
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()
```

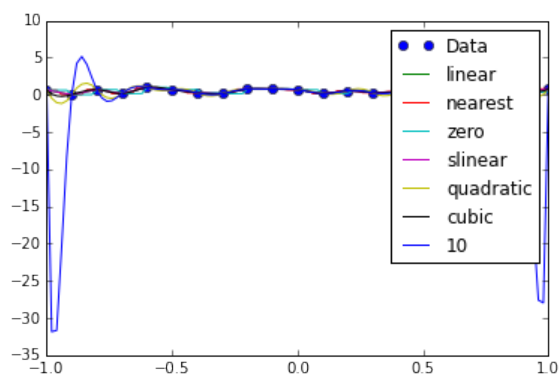


Figura 2: Gráfica generada por el programa original.

10 puntos aleatorios con $x = [0, 3]$ para la función $f(x) = \sin 2x$.

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

# 10 números aleatorios entre 0 y 3.
xr = np.random.random(10)
x = xr*3
# "Data set" original
y = np.sin(3*x)

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

# Array con los puntos entre el "data set" original para la interpolación
xp = np.linspace(min(x), max(x), 10000)
# opciones para interp1d
options = ('linear', 'quadratic', 'cubic')
for o in options:
    f = interp1d(x, y, kind=o)    # función interpoladora
    plt.plot(xp, f(xp), label=o)  # gráfica de la función interpoladora

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

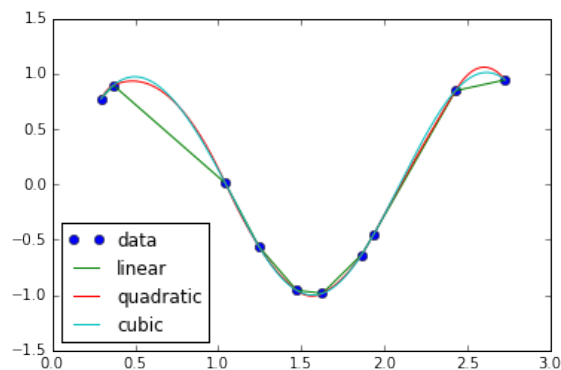


Figura 3: Gráfica generada por el programa para la función $f(x) = \sin 2x$.

20 puntos aleatorios con $x = [-10, 10]$ para la función $f(x) = \frac{\sin x}{x}$.

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

# 20 números aleatorios entre -10 y 10.
xr = np.random.random(20)
x = (xr*20)-10
# "Data set" original
y = (np.sin(x))/x

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

# Array con los puntos entre el "data set" original para la interpolación
xp = np.linspace(min(x), max(x), 10000)
# opciones para interp1d
options = ('linear', 'quadratic', 'cubic')
for o in options:
    f = interp1d(x, y, kind=o)    # función interpoladora
    plt.plot(xp, f(xp), label=o)  # gráfica de la función interpoladora

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

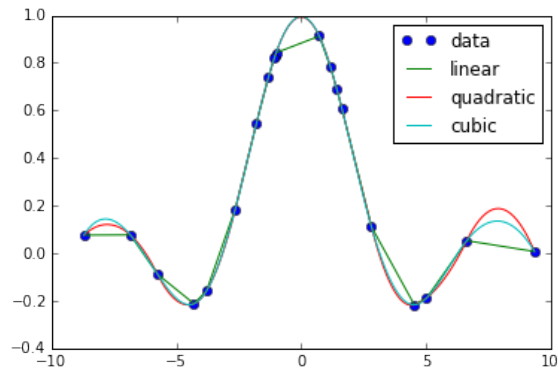


Figura 4: Gráfica generada por el programa para la función $f(x) = \frac{\sin x}{x}$.

Dados 16 puntos aleatorios $x = [-3, 3]$ para la función $f(x) = x^2 \sin 2x$.

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

# 16 números aleatorios entre -3 y 3.
xr = np.random.random(16)
x = (xr*6)-3
# "Data set" original
y = (x*x)*np.sin(2*x)

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

# Array con los puntos entre el "data set" original para la interpolación
xp = np.linspace(min(x), max(x), 10000)
# opciones para interp1d
options = ('linear', 'quadratic', 'cubic')
for o in options:
    f = interp1d(x, y, kind=o) # función interpoladora
    plt.plot(xp, f(xp), label=o) # gráfica de la función interpoladora

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

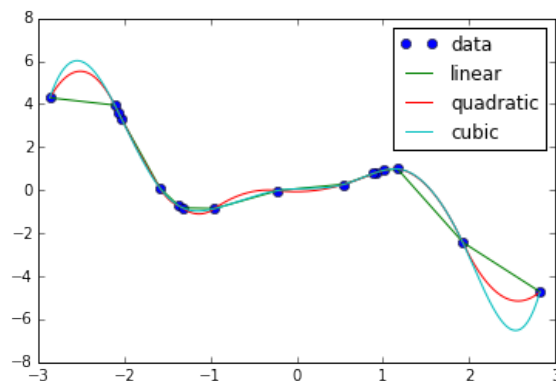


Figura 5: Gráfica generada por el programa para la función $f(x) = x^2 \sin 2x$.

Dados 12 puntos aleatorios $x = [-2, 2]$ para la función $f(x) = x^3 \sin 3x$.

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

# 12 números aleatorios entre -2 y 2.
xr = np.random.random(12)
x = (xr*4)-2
# "Data set" original
y = (x*x*x)*np.sin(3*x)

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

# Array con los puntos entre el "data set" original para la interpolación
xp = np.linspace(min(x), max(x), 10000)
# opciones para interp1d
options = ('linear', 'quadratic', 'cubic')
for o in options:
    f = interp1d(x, y, kind=o)    # función interpoladora
    plt.plot(xp, f(xp), label=o)  # gráfica de la función interpoladora

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

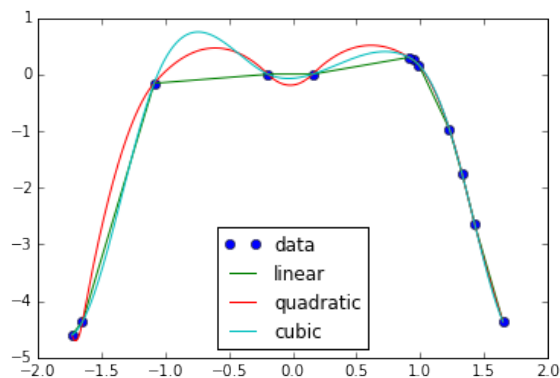


Figura 6: Gráfica generada por el programa para la función $f(x) = x^3 \sin 3x$.

Referencias

- [1] Wikipedia, *Efemérides*. Recuperado el 02 de febrero de 2016 de <https://es.wikipedia.org/wiki/Efem%C3%A9rides>
- [2] Wikipedia, *Interpolación*. Recuperado el 02 de febrero de 2016 de <https://es.wikipedia.org/wiki/Interpolaci%C3%B3n>
- [3] Wikipedia, *Interpolation*. Recuperado el 02 de febrero de 2016 de <http://docs.scipy.org/doc/scipy/reference/tutorial/interpolate.html>
- [4] Lizárraga, C. *Actividad 3 (2016-1)*. Recuperado el 02 de febrero de 2016 de <http://computacional1.pbworks.com/w/page/104792695/Actividad%203%20%282016-1%29>