

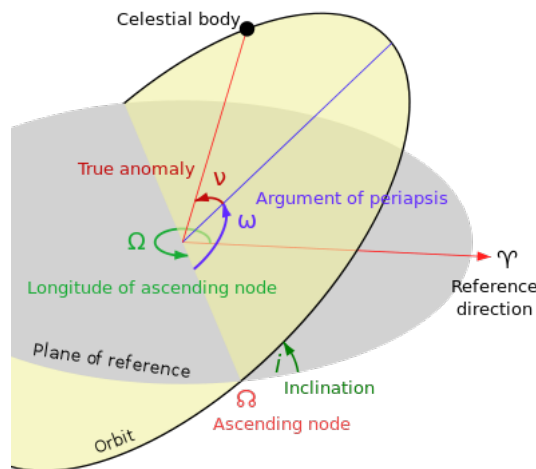
Práctica 3. Efemérides

Rosa Luz Zamora Peinado

4 de febrero de 2016

Introducción

El presente reporte corresponde a la práctica #3 del curso de Física Computacional. La práctica consistió en hacer interpolaciones básicas con el método de Splines para diferentes funciones con datos aleatorios. A continuación se presenta la motivación de la práctica y enseguida las actividades realizadas.



En la Astronomía y navegación celeste, una efemérides proporciona las posiciones de los objetos astronómicos y satélites artificiales a un determinado tiempo. Históricamente, las posiciones se han proporcionado como tablas de valores, dados en intervalos regulares de fechas y tiempos. Las efemérides modernas se obtienen de modelos matemáticos de los movimientos de los objetos astronómicos y la Tierra. Las efemérides han sido publicadas a lo largo de la historia de la humanidad, siendo las primeras efemérides en la cultura griega sobre el Cometa Halley alrededor de 466 AC, en documentos de China y Babilonia. El cometa Halley se acerca al Sol cada 76-77 años. La última aparición fue en 1986 y la próxima oportunidad de observarlo será a mediados de 2061.[1]

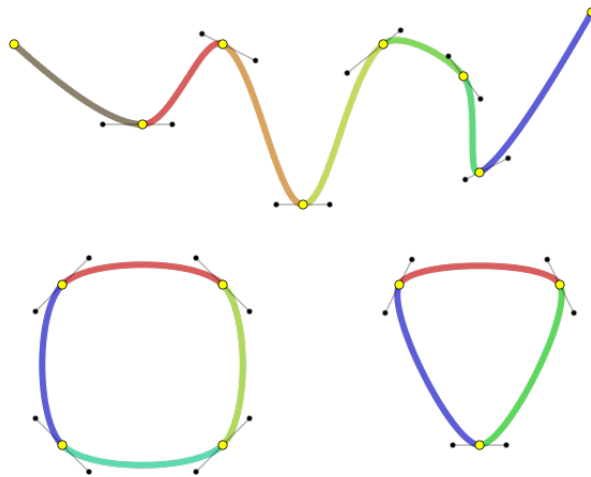


Además de las efemérides astronómicas podemos nombrar numerosos fenómenos de la naturaleza en los que a veces, observamos regularidad en la forma que se producen. Así, podemos saber qué sucedió instantes donde no se tomó dato del fenómeno, o bien encontrar un modelo matemático que "pase" por todos los puntos observados. La **Interpolación** consiste, entonces, en hallar un dato dentro de un intervalo en el que conocemos los valores en los extremos.

En esta práctica se usó el método de interpolación Splines:

En los problemas de interpolación, se utiliza a menudo la interpolación mediante splines porque da lugar a resultados similares requiriendo solamente el uso de polinomios de bajo grado, evitando así las oscilaciones, indeseables en la mayoría de las aplicaciones, encontradas al interpolar mediante polinomios de grado elevado.[2]

Es decir: encontrar funciones que pasen por 2, 3, 4,...,n-1 puntos de los observados, por intervalos o no, y luego suavizarlas y hacerlas continuas y derivables.



Actividades realizadas

En base al siguiente código ejemplo:

```
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)

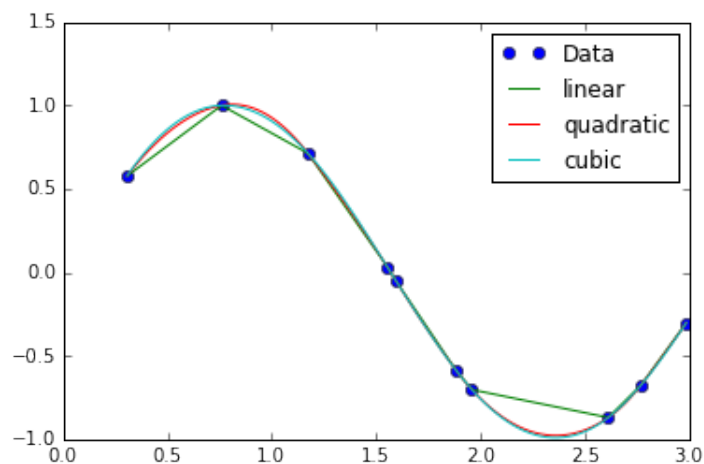
# Available options for interp1d
options = ('linear', 'nearest', 'zero', 'slinear', 'quadratic', 'cubic', 10)

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()
```

Se realizaron la sgiuientes actividades:

1. Dados 10 puntos aleatorios entre $x = 0$ y $x = 3$ para la función $f(x) = \sin(2x)$



```

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

x00 = np.random.random (10)
x0 = x00*3
y0 = sin (2*x0)

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

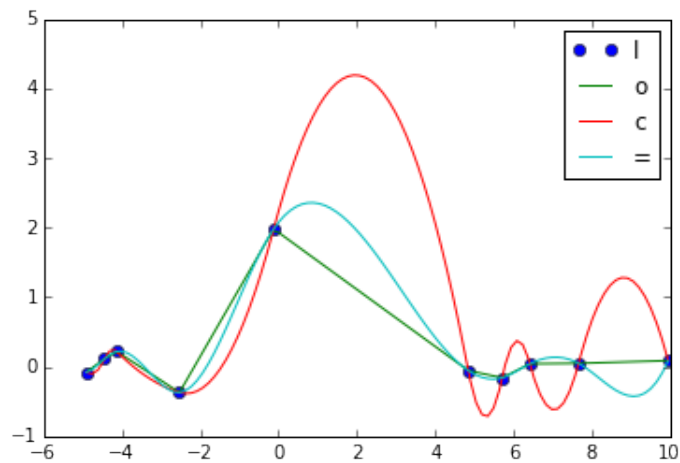
x=np.linspace(min(x0), max(x0), 101)

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

for o in options:
    f = interp1d(x0, y0, kind=o)
    plt.plot(x, f(x), label=o),
plt.legend(loc='best')
plt.show()

```

2. Dados 20 puntos aleatorios entre $x = -10$ y $x = 10$ para la función $f(x) = \frac{\sin(x)}{x}$



```

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

x00 = np.random.random (10)
x0 = ((x00*20)-10)
y0 = sin (2*x0)/x0

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

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

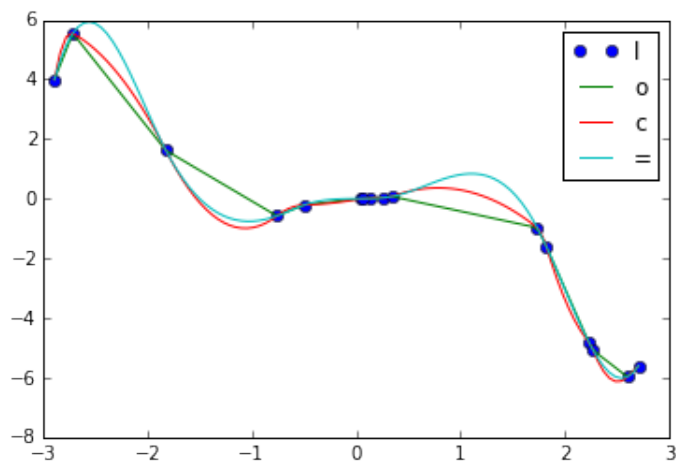
options = ('linear', 'quadratic', 'cubic')

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

plt.legend('loc=best')
plt.show()

```

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



```

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

x00 = np.random.random (16)
x0 = ((x00*6)-3)
y0 = (x0**2)*(sin(2*x0))

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

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

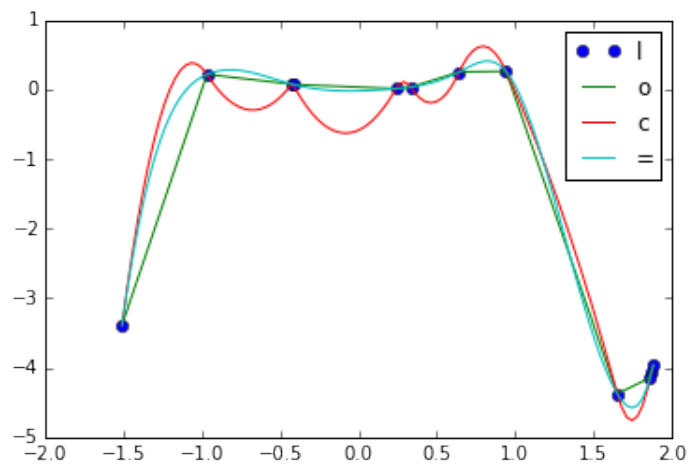
options = ('linear', 'quadratic', 'cubic')

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

plt.legend('loc=best')
plt.show()

```

4. Dados 12 puntos aleatorios entre $x = -2$ y $x = 2$ para la función $f(x) = x^3 \sin(3x)$



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

x00 = np.random.random (12)
x0 = ((x00*4)-2)
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('loc=best')
plt.show()
```

Bibliografía

- [1] Física Computacional(2016-2) *Actividad 3*. Recuperado el 4 de febrero de 2016 de [http://computacional1.pbworks.com/w/page/104792695/Actividad%203%20\(2016-1\)](http://computacional1.pbworks.com/w/page/104792695/Actividad%203%20(2016-1))
- [2] Wikipedia *Splines*. Recuperado el 4 de febrero de 2016 de <https://es.wikipedia.org/wiki/Spline>