

Computación científica LFIS-126

Profesor: Julio C. Marín

Departamento de Meteorología, Universidad de Valparaíso

Segundo Semestre

Temas a tratar en clase

- Interpolación
- Spline lineal y spline cúbico
- Interpolación 2D
- Ejemplos y ejercicios

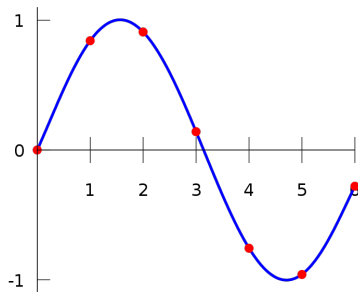
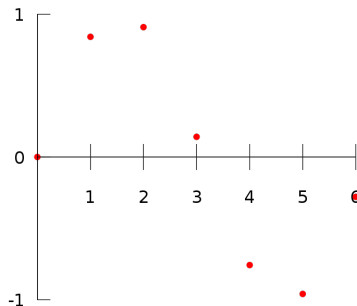
Tenemos un conjunto de datos discretos:

x_1	x_2	x_3	\dots	x_n
y_1	y_2	y_3	\dots	y_n

Fuente de datos

- Observaciones experimentales
- Simulaciones numéricas

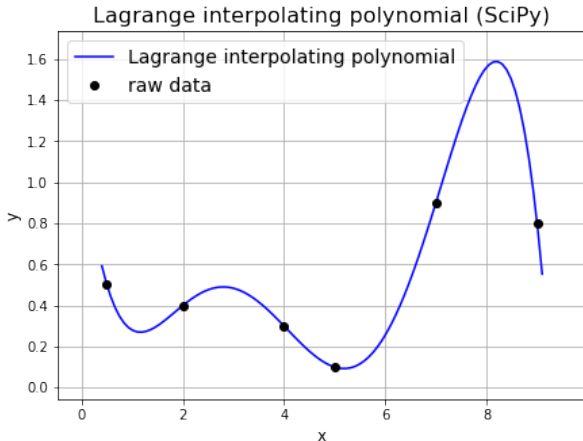
Interpolación



Método para obtener nuevos valores de $f(x)$ en el rango de datos discretos de x conocidos

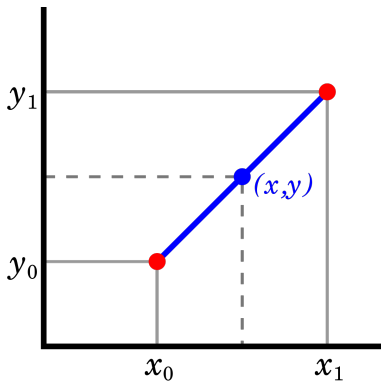
Interpolación

- Se construye una curva a través de datos discretos.
- Curva incluye los datos discretos.
- Se asume implícitamente que datos discretos son exactos y definidos



Interpolación polinomial

- Dos ptos determinan una línea recta.
- Dos ptos en plano (x_0, y_0) y (x_1, y_1) , con $x_0 \neq x_1$ determinan único polinomio de primer grado en x cuyo gráfico pasa a través de 2 ptos.



Generalización a más de 2 ptos

- Dado n ptos distintos en plano (x_k, y_k) , $k = 1, 2, \dots, n$, \exists polinomio único en x de grado $< n$ cuyo gráfico pasa por todos los ptos.
- No. ptos $n =$ No. coeficientes
- Polinomio se llama polinomio de interpolación
- Reproduce los datos:

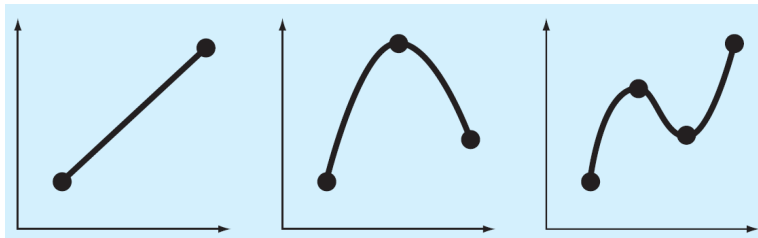
$$P(x_k) = y_k, \quad k = 1, 2, \dots, n$$

Interpolación polinomial

- Para 2 pts \exists solo una recta (polinomio primer orden) que conecta los 2 pts.
- Una parábola (polinomio segundo orden) conecta 3 pts.
- Para n pts, \exists uno y solo un polinomio de grado $n - 1$ que pasa por todos los pts.

Interpolación polinomial

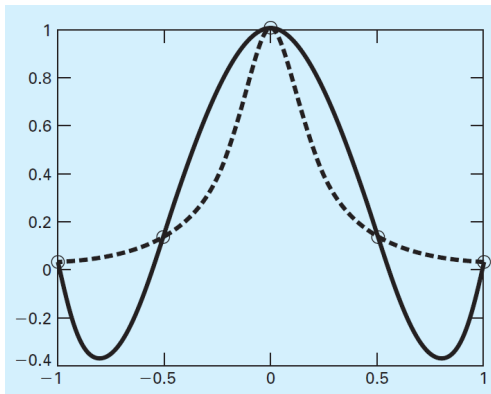
Encontrar el polinomio único de orden $n - 1$ que ajusta a n pts distintos



Interpolación polinomial - Limitaciones

- Limitaciones al usar interpolación polinomial con muchos pts.
- Son muy sensibles a errores de truncamiento

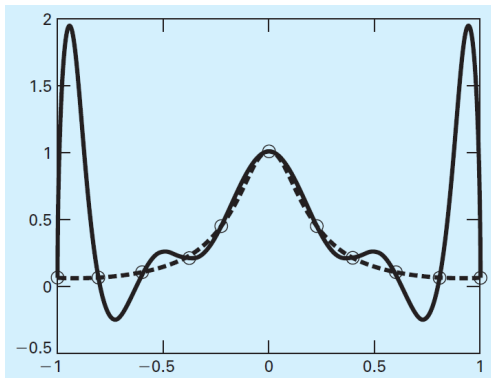
- $f(x) = \frac{1}{1+25x^2}$ Función de Runge en intervalo $[-1, 1]$
- Comparación con polinomio de 5 pts.
- Pobre representación de la función.



Interpolación polinomial - Limitaciones

- Limitaciones al usar interpolación polinomial con muchos pts.
- Son muy sensibles a errores de truncamiento

- $f(x) = \frac{1}{1+25x^2}$ Función de Runge en intervalo $[-1, 1]$
- Comparación con polinomio de 11 pts.
- Polinomio tiene tendencia a oscilar excesivamente entre pts.
- Interpolando cúbico usando 4 pts más cercanos daría mejores resultados.



Método alternativo es usar splines.

Splines

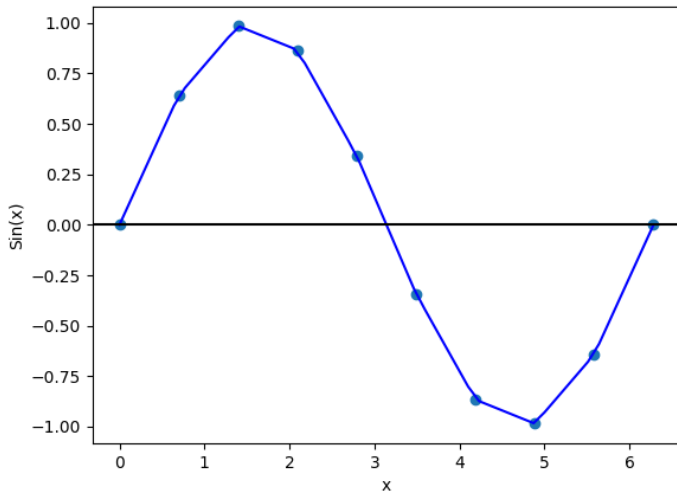
- Interpolación por partes en subgrupos de datos usando polinomios de orden pequeño.
- Si tenemos muchos datos discretos \Rightarrow el spline cúbico será casi siempre nuestro mejor interpolador
- Spline cúbico tiende menos a oscilar entre ptos.

- Numpy tiene función `interp` que realiza interpolación lineal 1-D

Ejemplo

- `import numpy as np`
- `import matplotlib.pyplot as plt`
- `x = np.linspace(0, 2*np.pi, 10); y = np.sin(x)`
- `xvals = np.linspace(0, 2*np.pi, 50)`
- `yinterp = np.interp(xvals, x, y)`
- `plt.clf(); plt.plot(x,y,'o'); plt.plot(xvals,yinterp, '-b')`
- `plt.xlabel('x'); plt.ylabel('Sin(x)')`
- `plt.axhline(y=0, xmin=0, xmax=2*np.pi, color='k')`

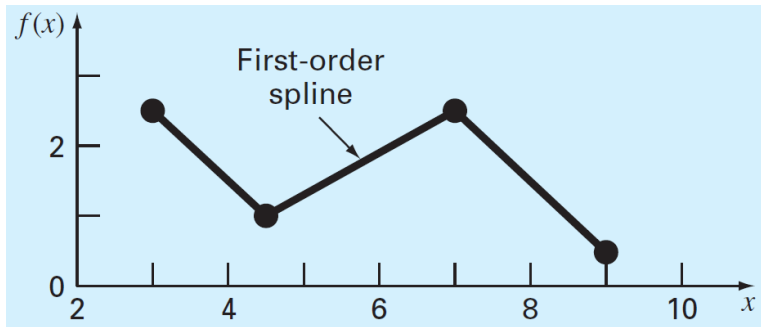
Interpolación - Spline lineal



<http://docs.scipy.org/doc/numpy/reference/generated/numpy.i>

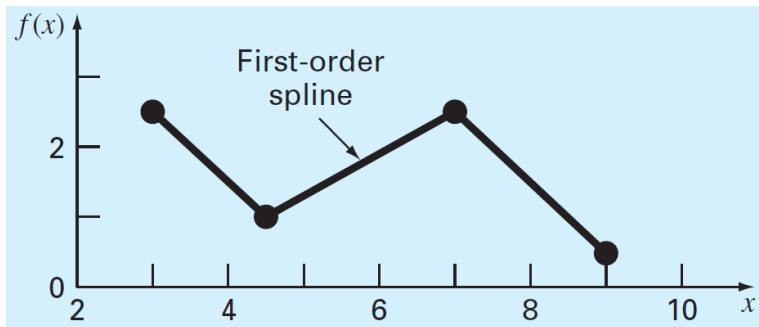
Interpolación - Spline lineal

- Principal desventaja de spline primer orden es que variaciones son bruscas.
- La pendiente cambia abruptamente en cada pto donde 2 splines se unen.
- Primera derivada es discontinua en esos ptos.



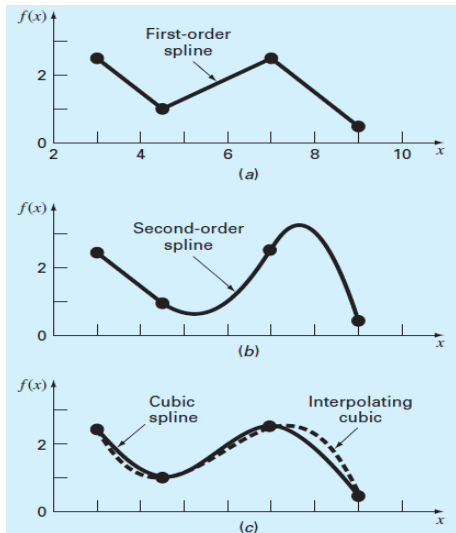
Interpolación - Spline lineal

- Deficiencia se supera usando spline polinomial de orden superior.
- Asegura suavidad en ptos igualando derivadas en ptos.



Interpolación - Spline cúbico

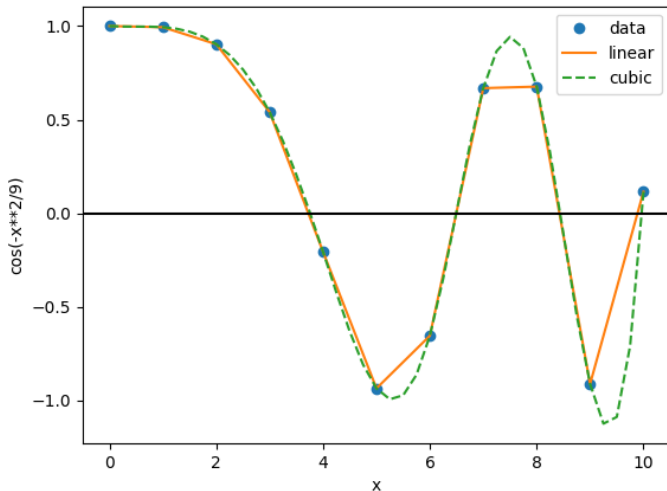
- Curvas de 3er orden usadas para conectar cada par de ptos
- Spline más frecuentemente usado en la práctica
- No presenta oscilaciones típicas de polinomios de orden mayor
- Exhibe apariencia lisa comparado con spline lineal
- Funciones pasan a través de ptos.
- 1era y 2da derivadas en ptos son iguales.



Paquete scipy tiene función interpolate

- `#### Interpolación spline lineal y cúbico #####`
- `from scipy.interpolate import interp1d`
- `import matplotlib.pyplot as plt`
- `x = np.linspace(0, 10, 11)`
- `y = np.cos(-x**2/9.0); xnew = np.linspace(0, 10, 41)`
- `f = interp1d(x, y)`
- `f2 = interp1d(x, y, kind='cubic')`
- `plt.plot(x, y, 'o', xnew, f(xnew), '-', xnew, f2(xnew), '--')`
- `plt.legend(['data', 'linear', 'cubic'], loc='best')`

Interpolación - Spline cúbico



<http://docs.scipy.org/doc/scipy/reference/tutorial/interpolate.html>

Ejercicio de Interpolación

Ejercicio: La presión atmosférica y la temperatura varían de la siguiente forma en la altura. Usando la ecuación del gas ideal $p = \rho RT$, donde $R = 287.05 \text{ J/kg/K}$ es la cte específica de los gases para el aire seco. Calcule la densidad (ρ) y realice un plot densidad vs altura. Determine el valor de la densidad a 5.3 km de altura.

h(km)	0	2	4	6	7	8
T(K)	288.5	280.2	277.4	273.2	269.3	260.4
P(hPa)	1026.24	803	605	400	350	300
9	10	11				
257.3	250.2	245.1				
250	200	150				

Ejercicio de Interpolación

- `from scipy.interpolate import interp1d`
- `import matplotlib.pyplot as plt`
- `H = np.array([0,2,4,6,7,8,9, 10,11])`
- `T = np.array([288.5, 280.2, 277.4, 273.2, 269.3, 260.4, 257.3, 250.2, 245.1])`
- `P = np.array([1026.24, 803, 605, 400, 350, 300, 250, 200, 150])`
- `densidad = P*100/(287.05*T)`
- `HH = np.arange(0, 11.1, 0.1)`
- `fden = interp1d(H, densidad, kind='cubic')`
- `dens_HH = fden(HH); dens_53 = fden(5.3)`
- `print("El valor de densidad en 5.3 km = ", dens_53, 'kg/m3')`
- `plt.clf(); plt.plot(dens_HH, HH, '-k', dens_53, 5.3, '*r')`
- `plt.xlabel('Densidad [kg/m3]'); plt.ylabel('Altura [km]')`

- Ejemplo: Tenemos función:

$$z(x, y) = \sin\left(\frac{\pi x}{2}\right) \exp(y/2)$$

en los ptos (x, y) que no están equiespaciados en y . Usaremos función `interp2d` para interpolar estos valores en una rejilla más fina equiespaciada

- `import numpy as np`
- `from scipy.interpolate import interp2d`
- `import matplotlib.pyplot as plt`
- `x = np.linspace(0, 4, 13)`
- `y = np.array([0, 2, 3, 3.5, 3.75, 3.875, 3.9375, 4])`
- `X, Y = np.meshgrid(x, y)`
- `Z = np.sin(np.pi*X/2)*np.exp(Y/2)`
- `x2 = np.linspace(0, 4, 65); y2 = np.linspace(0, 4, 65)`
- `X2, Y2 = np.meshgrid(x2, y2)`
- `f = interp2d(x, y, Z, kind='cubic')`
- `Z2 = f(x2, y2)`

Interpolación en 2d

- `cont1 = np.arange(-7.5, 8.0, 0.5)`
- `plt.clf()`
- `plt.subplot(1,2,1)`
- `CS = plt.contourf(X, Y, Z, cont1, cmap=plt.cm.jet)`
- `plt.colorbar(CS, orientation='horizontal')`
- `plt.title('Datos originales'); plt.xlabel('X')`
- `plt.ylabel('Y') plt.tight_layout()`
- `plt.subplot(1,2,2)`
- `CS = plt.contourf(X2, Y2, Z2, cont1, cmap=plt.cm.jet)`
- `plt.colorbar(CS, orientation='horizontal')`
- `plt.title('Datos interpolados') plt.xlabel('X-Int')`
- `plt.ylabel('Y-Int') plt.tight_layout()`

Ejercicio 5.4 del libro

La difracción de la luz en los telescopios limita la calidad de las observaciones astronómicas. Cuando esa luz, con λ , pasa a través de apertura circular del telescopio (asumiendo radio unitario) y se enfoca en un plano focal, se introduce un patrón de difracción circular cuya intensidad está dada por:

$$I(r) = \left(\frac{J_1(kr)}{kr} \right)^2$$

r : distancia en plano focal desde centro del patrón de difracción, $k = 2\pi/\lambda$, y $J_1(x)$ es una función de Bessel. Las funciones de Bessel están dadas por:

$$J_m(x) = \frac{1}{\pi} \int_0^\pi \cos(m\theta - x \sin \theta) d\theta$$

donde m es un entero positivo y $x \geq 0$.

a) Escriba una función de Python $J(m, x)$ que calcule el valor de $J_m(x)$ usando el método de Simpson con $N = 1000$ puntos. Use la función en un programa que plotee las funciones de Bessel J_0 , J_1 y J_2 en un mismo gráfico como función de x entre $x = 0$ y $x = 20$.

Ejercicio de Interpolación

Los datos $y = 0.7, -2.925, -7.8, -13.25, -18.3, -21.675, -21.8, -16.8, -4.5, 17.575$ se tomaron para valores de $x = 1., 1.5, 2., 2.5, 3., 3.5, 4., 4.5, 5., 5.5$. Interpolar estos datos usando un spline lineal, un spline cuadrático y un spline cúbico. Imprima en pantalla el valor de y para el valor de $x = 4.1$ usando las tres funciones de interpolación. Plotee los datos originales y los datos interpolados en un mismo plot.