Capítulo 4

Interpolación

En ciertos casos el usuario conoce el valor de una función $f(x)$ en una serie de puntos $x_1, x_2, \cdots, x_N$, pero no se conoce una expresión analítica de $f(x)$ que permita calcular el valor de la función para un punto arbitrario. Un ejemplo claro son las mediciones de laboratorio, donde se mide cada minuto un valor, pero se requiere el valor en otro punto que no ha sido medido. Otro ejemplo son mediciones de temperatura en la superficie de la Tierra, que se realizan en equipos o estaciones meteorológicas y se necesita calcular la temperatura en un punto cercano, pero distinto al punto de medida.

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. Se asume que el punto arbitrario x se encuentra dentro de los límites de los puntos de medición, en caso contrario se llamaría *extrapolación*. En este texto se dicute exclusivamente la *interpolación*, aunque la idea es similar. PRECAUCIÓN: El uso indiscriminado de extrapolación no es recomendable, siempre tratar con cuidado.

Existe un sinnúmero de métodos de interpolación, incluyendo la interpolación *lineal*, *polinomial*, y la *spline*, que se discutirán más adelante. Existen otros métodos que no serán tenidos en cuenta en este texto, pero se pueden encontrar en [**********]

En muchos casos el usuario se enfrenta a funciones para las cuales la interpolación no funciona. Por ejemplo, la función

$$f(x) = 0{,}3x^2 + \frac{1}{\pi}\ln\left[(\pi - x)^2\right] + 1 \tag{4.1}$$

tiene una singularidad cuando $x = \pi$ [ver figura 4.1]. Cualquier método de interpolación que se base en valores de $x = 3{,}14, 3{,}14, 3{,}15, 3{,}16$ muy probablemente va a generar un resultado erroneo para $x = 3{,}1415\ldots$.

En la práctica, un proceso de interpolación se realiza en dos etapas:

1. Hacer un *fit* de los datos disponibles con una función interpolante.
2. Evaluar la función interpolante en el punto de interés x.

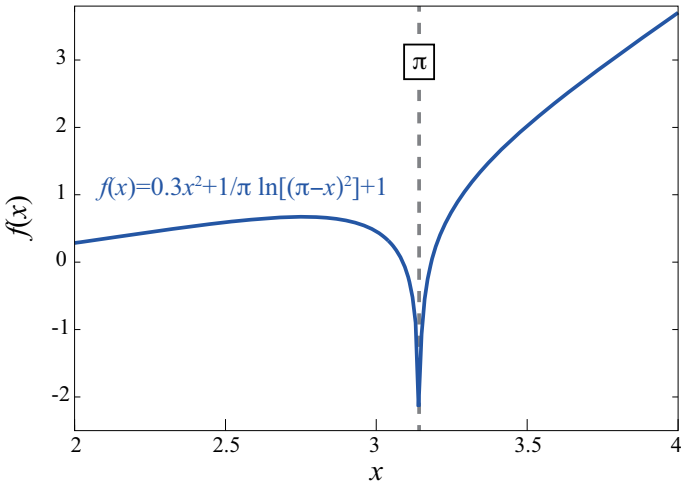

Figura 4.1: Los métodos de interpolación pueden tener problemas interpolando una función con singularidades-

Este proceso en dos etapas no es necesariamente el más eficiente. La mayoría de algoritmos comienzan con un punto cercano $f(x_i)$, y poco a poco van aplicando correciones más pequeñas a medida que la información de valores $f(x_i)$ más distantes es incorporada. El procedimiento toma aproximadamente $O(N^2)$ operaciones. Si la función tiene un comportamiento suave, la última correción será la más pequeña y puede ser utilizada para estimar un límite a rango de error.

La interpolación *local*, usando un número finito de vecinos próximos (*nearest neighbours*) genera valores interpolados $f(x)$ que, en general, no tienen continuidad es su primera o siguientes derivadas. Esto se debe a que, cuando el valor interpolado en x cruza uno de los puntos disponibles x_i, el procedimiento de interpolación **cambia** el grupo de vecinos próximos, lo cual puede generar una discontinuidad en la función de interpolación en ese punto, lo cual no es necesariamente lo que el usuario quiere.

El número de puntos (menos 1) usado en el esquema de interpolación se le conoce como el *orden* de la interpolación. Aumentar el orden de la interpolación no necesariamente aumenta su precisión, especialmente en la interpolación polinomial. Al adicionar el número de puntos vecinos al punto de interés x, el polinomio de mayor orden tiende a hacer que la función interpolante oscile excesivamente entre los puntos x_i. Esa oscilación puede no tener nada que ver con la función *verdadera*.

A menos que el usuario tenga evidencia sólida que la función interpolante sea similar a la función verdadera f, se recomienda ser cuidadoso con interpolación de un orden muy alto. Interpolación con 3 - 4puntos, y hasta 5 - 6 puntos es aceptable, pero rara vez se usan métodos de mayor orden, a menos que haya alguna forma de monitorear el error.

4.1. Interpolación Lineal

La interpolación lineal es el método más simple en uso hoy. Es el método usado por los programas de generación de gráficas, donde se interpola con lineas rectas entre una serie de puntos que el usuario quiere graficar.

La idea básica es conectar los 2 puntos dados en x_i, es decir (x_0, y_0) y (x_1, y_1). La función interpolante es una linea recta entre los dos puntos. Para cualquier punto entre los dos valores de x_0 y x_1 se debe seguir la ecuación de la linea

$$\frac{y - x_0}{y_1 - y_0} = \frac{x - x_0}{x_1 - x_0}, \tag{4.2}$$

que se puede derivar geométricamente.

En lo anterior, el único valor desconocido es y, que representa el valor desconocido para x, despejando queda:

$$y = y_0 + (x - x_0)\frac{y_1 - y_0}{x_1 - x_0}, \tag{4.3}$$

donde se asume que $x_0 < x < x_1$, de otra forma esto se conocería como exptrapolación.

Si se tienen más de dos puntos para la interpolación, es decir $N > 2$, con puntos $x_0, x_1, \cdots, x_N$, simplemente se concatena la interpolación lineal entre pares de puntos continuos.

4.2. Interpolación polinomial

Cuando se tienen dos puntos, éstos pueden ser unidos con una linea recta. Dos puntos cualquiera en un plano (x_0, y_0) and (x_1, y_1), donde $x_0 \neq x_1$, determinan un polinomia de primer grado en x, donde la función psa por ambos puntos (lo que se discutio en la sección anterior).

Una generalización de lo anterior sugiere que dados N puntos en un plano (x_k, y_k) con $k = 1, 2, \cdots, N$ y distintos x_k, existe un único polinomio en x de grado menor a N cuya función pasa por todos los puntos.

Este tipo de polinomio se le conoce como polinomio de interpolación ya que reproduce los datos exactamente

$$P(x_k) = y_k, \quad k = 1, \cdots, N. \tag{4.4}$$

La forma de describir este tipo de polinomios es con la forma *Lagrangiana*:

$$P(x) = \sum_k \left(\prod_{j \neq k} \frac{x - x_j}{x_k - x_j} \right) y_k \tag{4.5}$$

donde hay N términos en la suma y $N - 1$ en los productos, de tal manera que esta expresión describe un polinomio de grado hasta $N - 1$. Si $P(x)$ es evaluado en los puntos $x = x_k$, todos los productos excepto el k son ceros. Además el

producto k es igual a 1, así que la suma es igual a y_k y las condiciones de interpolación (puntos x_k exactos) son cumplidas.

Una forma más común de representar un polinomio, diferente a la Lagrangiana es de la forma

$$x^3 - 2x - 5,$$

conocida como *power form*. Esta expresión se puede generalizar para polinomios de interpolación asi:

$$P(x) = c_1 x^{n-1} + c_2 x^{n-2} + \cdots + c_{n-1} x + c_N, \tag{4.6}$$

donde c_n son los coeficientes, que deben ser *estimados* o encontrados. Este sistema de ecuaciones lineales se puede resolver con métodos de teoría de inversión y estimación paramétrica que más adelante se discutirán.

En la práctica, se pude utilizar la ecuación (4.5) para estimar el polinomio de interpolación para los datos disponibles. Existen ciertas desvantajes del uso de polinomios ya que puede tener mucha oscilación entre los puntos x_k. Esto se puede ver en la seccin *Comparación de métodos*.

Programa 4.1 Función o rutina de Matlab para interpolación polinomial.

```
(polyinterp.m)
function v = polyinterp(x,y,u,n)
% Polinomial interpolation
% Input
% x - vector of tabulated position of data
% y - vector of values at points x_i
% u - vector with x positions where interpolation is wanted
% n - degree of polinomial interpolation
%
% Output
% v - vector with values at points u_j
%
...
v = zeros(size(u));

for k = 1:n
   w = ones(size(u));
   for j = [1:k-1 k+1:n]
      w = (u-x(j))./(x(k)-x(j)).*w;
   end
   v = v + w*y(k);
end
```

4.3. Interpolación Spline Cúbico

Hasta ahora se hemos presentado dos métodos de interpolación. Uno de ellos es local (sólo usa dos puntos alrededor del punto de interés x) y el otro es no-local, usa todos los puntos para estimar el valor de $f(x)$ para cualquier x.

En ambos casos el lector puede encontrar desventajas. En la interpolación lineal l aprimera derivada presenta un *salto* en los puntos x_k. En cambio, en la interpolación polinomial este no es el caso, pero tiende a presentar *overshoot* (excederse o presentar demasiada oscilación) entre los puntos x_k.

En esta sección se describe un método que tiene dos características principales

1. es no-local, para definir una interpolación suavizada (*smooth*) con ubn polinomio cúbico

2. la segunda derivada de la función de interpolación es continua

El término *interpolación spline* puede extenderse más allá de la idea 1D que se presenta acá, ya que existen métodos multi-dimensionales, splines de mayor orden (high-order), con nodos variables y *smoothing splines*. Una excelente guia para estudiar splines es el libro de Carl De Boor (de Boor, 1978). El autor es además el autor del paquete de Splines en Matlab (Spline Toolbox).

Acá se explica de forma muy básica el método de interpolación de Spline Cúbico. Si el lector quiere mayor información, se recomienda de Boor (1978). Dada una serie tabulada de una función $y_i = y(x_i)$ con $i = 1, 2, \cdots, N$, nos concentramos en un intervalo particular, por ejemplo entre x_j y x_{j+1}. La interpolación lineal daría

$$y = Ay_j + By_{j+1} \tag{4.7}$$

donde

$$A = \frac{x_{j+1} - x}{x_{j+1} - x_j} \qquad B = 1 - A = \frac{x - x_j}{x_{j+1} - x_j} \tag{4.8}$$

Note que la expresión anterior es tomada directamente de la formula Lagrangiana (4.5).

Debido a que la interpolación lineal es local, la ecuación 4.7 tiene

$$y'' = \begin{cases} 0 & \text{dentro de intervalos} \\ \infty & \text{en intervalos} \end{cases} \tag{4.9}$$

donde se muestra que la segunda derivada no esta definida ∞ justo en los puntos x_k. El objetivo de los splines cúbicos es generar una función de interpolación que tenga una primera derivada suavizada y una segunda derivada contínua, tanto dentro de los intervalos como en los puntos x_k.

Suponga (aunque no es el caso, pero sin embargo suponga) que adicional a los valores y_i, tambiés se conoce el valor de la segunda derivada y'', es decir los valores y_i''. Se puede entonces adicionar al lado derecho de la ecuación 4.7 un polinomio cúbico cuya segunda derivada cambie de manera lineal del valor y_i''

a la izquierda al valor y''_{i+1} a la derecha. De esta forma se puede entonces tener una segunda derivada que sea continua.

Adicionalmente se puede forzar que el valor del polinomio cúbico sea cero en x_i y x_{i+1} de tal forma que adicionar esto a la ecuación no previene que los datos tabulados en los puntos x_i sean ajustados por la función de manera exacta.

Este texto no pretende demostrar lo siguiente, pero si cambiamos (4.7) con:

$$y = Ay_j + By_{j+1} + Cy''_j + Dy''_{j+1} \tag{4.10}$$

y donde debemos definir

$$A = \frac{x_{j+1} - x}{x_{j+1} - x_j} \qquad B = 1 - A = \frac{x - x_j}{x_{j+1} - x_j} \tag{4.11}$$

$$C = \frac{1}{6}(A^3 - A)(x_{j+1} - x_j)^2 \qquad D = \frac{1}{6}(B^3 - B)(x_{j+1} - x_j)^2 \tag{4.12}$$

Note que la variable independiente x se encuentra relacionada de manera lineal con A y B y de forma cúbica con C y D.

Esto se puede comprobar simplemente calculando de manera explicita la segunda derivada

$$\frac{d^2y}{dx^2} = Ay''_j + By''_{j+1}. \tag{4.13}$$

Debido a que $A = 1$ en el punto x_j y $A = 0$ en x_{j+1} y que B muestra exactamente lo contrario, se demuestra que y'' es simplemente la segunda derivada tabulada y además que a segunda derivada es continua a través de intervalos contiguos (x_{j-1}, x_j) y (x_j, x_{j+1})

La forma de calcular los coeficientes no es complicada pero está más allá del interés de este texto. En general Matlab tiene rutinas para interpolación spline y es fácil entcontrar (por ejemplo en Numerical Recipes Press *et˜al.* (1989)) rutinas en $\mathbb{C}$ y $\mathbb{F}90$ para generar las interpolaciones. En la siguiente sección se discute brevemente las ventajas y desventajas de los métodos de interpolación discutidos en el texto.

4.4. Comparación de métodos

Las figuras 4.2 y 4.3 muestran el resultado del uso de los 3 métodos de interpolación discutidos en el texto [ver Programa 4.2].

La primera figura 4.2 muestra el resultado para una serie de puntos de una función suave (una función sinusoidal) y cómo en este caso la interpolación lineal presenta fuertes cambios de pendiente, mientras que la interpolación spline y polinomial hacen un buen trabajo.

En el segundo ejemplo (Figura 4.3) la función original es una función triangular. La interpolación lineal (como era e esperarse) reproduce la función original de manera precisa, miestras que ambos métodos (spline y polinomial) presentan oscilaciones adicionales. Es importante notar que el método polinomial genera oscilaciones muy fuertes (especialmente en los bordes de la función) que el método de splines no presenta.

Programa 4.2 Programa en Matlab para comparación de métodos de interpolación. Se hace uso de rutina de Matlab `interp1`.

```
(interp_test.m)
...
% Smooth function
   xf=[0:0.05:10]; yf = sin(2*pi*xf/5);
   xp=[0:10];      yp = sin(2*pi*xp/5);

   lin=interp1(xp,yp,xf);
   spl=interp1(xp,yp,xf,'spline');
   pol=polyinterp(xp,yp,xf);
...
% Non smooth function

   t = -2:0.5:2;
   dt = 1;
   ti =-2:0.025:2;
   dti = 0.025;
   y = abs(t);

   ylin        = interp1(t,y,ti,'linear');
   yspline = interp1(t,y,ti,'spline');
   ypoly     = polyinterp(t,y,ti);
...
```

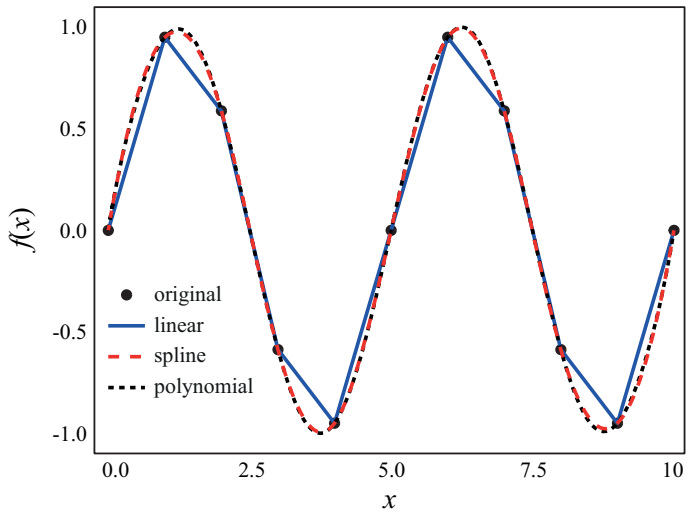

Figura 4.2: Resultados de interpolación de serie de puntos (círculos) tomados de una función sinusoidal.

El programa [4.2] muestra el uso de Matlab para la generación de las figuras que se muestrán acá. Se hace uso de l afunción intrínseca de Matlab `interp1` por medio de la cual se puede hacer interpolación lineal, spline, y varias otras. La función para interpolación polinomial es relativamente simple [ver Programa 4.1] y bsicamente muestra el uso de la función Lagrangiana.

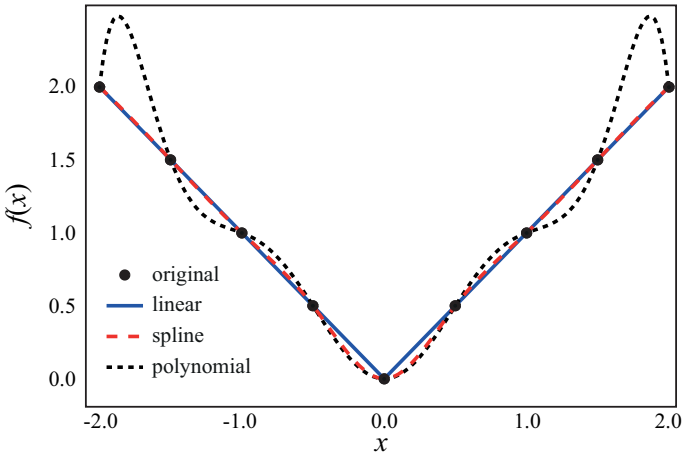

Figura 4.3: Resultados de interpolación de serie de puntos (círculos) tomados de una función triangular invertida

Preguntas

1. En su lenguaje favorito, escriba una rutina similar al Programa 4.1 que realice la interpolación lineal. Note que la función o subrutina debe aceptar como entrada los dos vectores x_i y y_i y otro vector u_j donde se define los puntos donde se quiere hacer la interpolación.

   Como ejemplo, realice la interpolación de los siguientes puntos con iun intervalo de muestreo de $dx = 0{,}05$.

   $$x = [0, 1, 2, 3, 4, 5] \qquad y = [17, 15, 12, 16, 18, 21] \tag{4.14}$$

2. En muchos temas relacionados con información geográfica (meteorología, ubicación o densidad poblacional, etc.) se tienen datos en una superficie (2D). Plantea de manera breve cómo se realizaría la interpolación lineal en 2D.