Introducción a la Programación y Análisis Numérico

Año 2022

Práctica 3: Interpolación.

Toda función continua puede ser aproximada por un polinomio de la forma $P_n(x) = a_n x^n + a_{n-1}x^{n-1} + ... + a_1 x + a_0$, donde n es un entero no negativo y a_i con $i = 0, ...n$ son constantes reales. Dada una función cualquiera, definida y continua en un intervalo cerrado, existe un polinomio que se aproxima a la función tanto como se quiera. Dicho polinomio es único. Una de las ventajas de aproximar una función con un polinomio es, por ejemplo, que la derivada y la integral de un polinomio también son polinomios y son fáciles de determinar.

Para la interpolación, los polinomios de Taylor no son apropiados puesto que centran la información alrededor de un único punto, dando una buena aproximación en el entorno del punto pero no en general a medida que nos apartamos de él. Una buena aproximación polinómica debe ofrecer una aproximación relativamente exacta en todo el intervalo de interés. Si bien el polinomio interpolante es único, la manera de calcularlo no lo es. En esta práctica veremos algunos de los diferentes métodos para determinar a partir de conocer n+1 puntos distintos y su evaluación, el polinomio interpolante de grado a lo sumo n (el que se obtiene de considerar los n+1 puntos) que cumple con pasar por los valores considerados para su construcción. Los puntos a utilizar no requieren que estén igualmente espaciados o que los valores de la abscisa estén en orden ascendente.

Interpolación polinómica de Lagrange

Sea f una función definida en los $n+1$ puntos distintos $x_0, x_1, ..., x_n$. Existe un único polinomio $P_l(x)$ de grado a lo sumo n tal que:

$$f(x_k) = P_l(x_k) \qquad \text{para cada } k = 0, 1, ..., n.$$

Este polinomio está dado por:

$$P_l(x) = f(x_0)L_{n,0}(x) + ... + f(x_n)L_{n,n}(x) = \sum_{k=0}^{n} f(x_k)L_{n,k}(x),$$

donde $L_{n,k}$ se define para cada $k = 0, 1, ..., n$ a partir de:

$$L_{n,k}(x) = \frac{(x-x_0)(x-x_1)...(x-x_{k-1})(x-x_{k+1})...(x-x_n)}{(x_k-x_0)(x_k-x_1)...(x_k-x_{k-1})(x_k-x_{k+1})...(x_k-x_n)} = \prod_{\substack{i=0 \\ i\neq k}}^{n} \frac{(x-x_i)}{(x_k-x_i)}.$$

Interpolación de Newton en diferencias divididas

El polinomio de interpolación de Newton de grado n,

$$P_n(x) = b_0 + b_1(x-x_0) + b_2(x-x_0)(x-x_1) + b_n(x-x_0)(x-x_1)...(x-x_{n-1}),$$

es obtenido a partir de los pares $(x_0, f(x_0)), (x_1, f(x_1)), ...(x_n, f(x_n))$, donde los coeficientes $b_i, i = 0, 1, ..., n$ están dados por:

$$\begin{aligned} b_0 &= f(x_0) \\ b_1 &= f[x_1, x_0] \\ b_2 &= f[x_2, x_1, x_0] \\ &\vdots \\ b_n &= f[x_n, x_{n-1}, ..., x_1, x_0] \end{aligned}$$

A las evaluaciones de las funciones entre paréntesis se las conoce como diferencias divididas y se calculan a partir de:

$$f[x_i, x_j] = \frac{f(x_i) - f(x_j)}{x_i - x_j} \quad \text{(expresión para la primera diferencia dividida)}$$

$$f[x_i, x_j, x_k] = \frac{f[x_i - x_j] - f[x_j - x_k]}{x_i - x_k} \quad \text{(expresión para la segunda diferencia dividida)}$$

$$\vdots$$

$$f[x_n, x_{n-1}, ..., x_1, x_0] = \frac{f[x_n, x_{n-1}, ..., x_1] - f[x_{n-1}, ..., x_1, x_0]}{x_n - x_0} \quad \text{(expresión para la n-ésima diferencia dividida)}$$

Como puede notarse, las diferencias de orden superior son obtenidas a partir de las diferencias de orden inferior, es decir, son expresiones recursivas.

Matriz de Vandermonde

La matriz de Vandermonde es una matriz de progresión geométrica en cada fila:

$$\begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & ... & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & ... & \alpha_2^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha_n & \alpha_n^2 & ... & \alpha_n^{n-1} \end{bmatrix}$$

A partir de la matriz de Vandermonde podemos encontrar de forma muy simple (pero no aconsejable) los coeficientes del polinomio de interpolación. Sean $x_0, x_1, ..., x_n$ los n+1 puntos donde se conoce la evaluación de la función f, $y_0, y_1, ..., y_n$. Luego, evaluar el polinomio interpolante en cada uno de los puntos nos permite escribir el siguiente sistema de ecuaciones:

$$\begin{bmatrix} 1 & x_0 & x_0^2 & ... & x_0^n \\ 1 & x_1 & x_1^2 & ... & x_1^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & ... & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \Rightarrow XA = Y$$

si el determinante de X es distinto de cero ($|X| \neq 0$), podemos encontrar los coeficientes del polinomio a partir de:

$$A = X^{-1}Y$$

Interpolación mediante trazadores - Spline

Los métodos vistos anteriormente permiten obtener polinomios interpolantes cuyo grado crece con n. Esto implica que, al aumentar la cantidad de puntos, el polinomio interpolante se vuelve más oscilante, dejando de ser una buena aproximación a f. Para solucionar este problema surge el método de trazadores o *spline*, el cual consiste en la determinación de un polinomio interpolante definido a trozos, cuyo grado podemos elegir independientemente del número de puntos a interpolar. Es decir, se busca:

$$P(x) = \begin{cases} S_0(x) & \text{si } x \in [x_0, x_1] \\ S_1(x) & \text{si } x \in [x_1, x_2] \\ & \vdots \\ S_n(x) & \text{si } x \in [x_{n-1}, x_n] \end{cases}$$

en donde el grado de cada $S_i(x)$ dependerá del tipo de trazador deseado. Por ejemplo, las curvas de tercer grado empleadas para unir cada par de datos se llaman trazadores cúbicos. Esas funciones se pueden construir de tal forma que las conexiones entre ecuaciones cúbicas adyacentes resulten visualmente suaves. Así, el trazador también une los puntos, pero como está limitado a cambios de tercer grado, las oscilaciones son mínimas. De esta manera, el trazador usualmente proporciona una mejor aproximación al comportamiento de las funciones que tienen cambios locales y abruptos.

En particular, la unión más simple entre dos puntos es una línea recta. Los trazadores de primer grado, o spline lineal, para un grupo de datos ordenados pueden definirse como un conjunto de funciones lineales:

$$
\begin{aligned}
f(x) &= f(x_0) + m_0(x - x_0) \quad x_0 < x < x_1 \\
f(x) &= f(x_1) + m_1(x - x_1) \quad x_1 < x < x_2 \\
&\vdots \\
f(x) &= f(x_{n-1}) + m_{n-1}(x - x_{n-1}) \quad x_{n-1} < x < x_n
\end{aligned}
$$

donde m_i es la pendiente de la línea recta que une los puntos:

$$m_i = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

Estas ecuaciones se pueden usar para evaluar la función en cualquier punto entre x_0 y x_n localizando primero el intervalo dentro del cual está el punto. Después se usa la ecuación adecuada para determinar el valor de la función dentro del intervalo. Este método es idéntico al de la interpolación lineal.

Ejercicios

Ej. 1: Utilice el método de diferencias divididas de Newton lineal y el método de Lagrange para interpolar el valor correspondiente a $x = 1,8$ a partir de la siguiente tabla:

X	0.5	1	3
Y	4	2	2/3

Ej. 2: A partir de los puntos representados en la siguiente tabla se busca conocer un valor interpolado en $x = 1,6$.

X	1.00	1.25	1.50	1.75	2.00	2.25	2.50
Y	-4.50	2.48	1.79	-4.47	-6.00	-0.22	14.70

a) Encuentre los polinomios de orden uno, dos y tres por el método de interpolación de Newton con diferencias hacia adelante e interpole el valor indicado.

b) Exprese los polinomios de Newton de primer, segundo y tercer orden, con diferencias hacia atrás, e interpole el valor $x = 2,3$.

c) Utilice Matlab/Octave para hallar el polinomio de mayor grado posible identificando sus coeficientes. Interpolar para $x = 1,6$. Verifique los resultados obtenidos en los incisos anteriores. Puede utilizar las funciones **polyfit** y **polyval**.

Ej. 3: Programe en OCTAVE/MATLAB los métodos de interpolación a_1) Vandermonde, a_2) Lagrange y a_3) Newton dado un conjunto de n puntos como entrada. La función `conv` puede serle de utilidad.

a) Dada la función $f(x) = |x| + x/2 - x^2$, hallar una aproximación polinómica para la misma en el intervalo $[-1, 1]$ con cada uno de los métodos anteriores. Considere 4, 10, 20 y 80 puntos equiespaciados para obtener la interpolación.

b) Grafique el polinomio de interpolación junto con la función $f(x)$ y observe qué pasa en cada caso.

c) Utilice los comandos *tic* y *toc* para un número creciente de puntos a interpolar, y observe la eficiencia de los distintos algoritmos para una misma interpolación.

Ej. 4: Aproxime la función $f(x) = \dfrac{2}{1+x^2}$ empleando algún método de interpolación obteniendo un polinomio:

(a) de segundo grado utilizando los valores de $x : \{-5; 0; 5\}$

(b) de 4º grado utilizando dos valores de x intermedios, $\{-5; -2.5; 0; 2.5; 5\}$.

(c) de 8º grado agregando más puntos intermedios.

Grafique la función $f(x)$ y los polinomios obtenidos en forma superpuesta. ¿Obtuvo una mejor aproximación al aumentar el grado del polinomio? ¿Qué polinomio aproxima mejor, en este ejemplo, a la función?

Ej. 5: Hemos visto tres métodos o algoritmos distintos para encontrar el polinomio que interpola un conjunto de $n+1$ puntos $\{(x_0, y_0), (x_1, y_1), ...(x_n, y_n)\}$.

a) Analice si las siguientes afirmaciones son o no verdaderas y en cada caso explique por qué:

- Los tres métodos conducen al mismo polinomio interpolante.
- Por los distintos métodos de interpolación debemos llegar a los mismos resultados numéricos.
- Desde un punto de vista operacional algunos algoritmos resultan menos ventajosos que otros.

b) Para cada uno de los métodos mencionados, analizar el costo computacional (número de operaciones de punto flotante necesarias) para: el cálculo del polinomio interpolante $P_n(x)$; para su evaluación en un valor arbitrario $P_n(x_k)$; para agregar un punto más a la interpolación.

c) El polinomio de interpolación de Newton es de la forma

$$P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + ... + a_n(x - x_0)(x - x_1)...(x - x_{n-1}),$$

donde $a_i = f[x_0, x_1, ..., x_i]$. Vea que esta expresión puede ser reordenada y escribirse de la forma

$$P_n(x) = (...((a_n(x - x_{n-1}) + a_{n-1})(x - x_{n-2}) + a_{n-2})(x - x_{n-3}) + ... + a_1)(x - x_0) + a_0.$$

Compare la eficiencia computacional de una u otra expresión al evaluar el polinomio, $P_n(x_k)$.

Ej. 6: Spline lineal

Ajuste los datos de la tabla con trazadores de primer grado. Evalúe la función en x = 5.

x	3.0	4.5	7.0	9.0
$f(x)$	2.5	1.0	2.5	0.5

Ej. 7: Spline cúbico

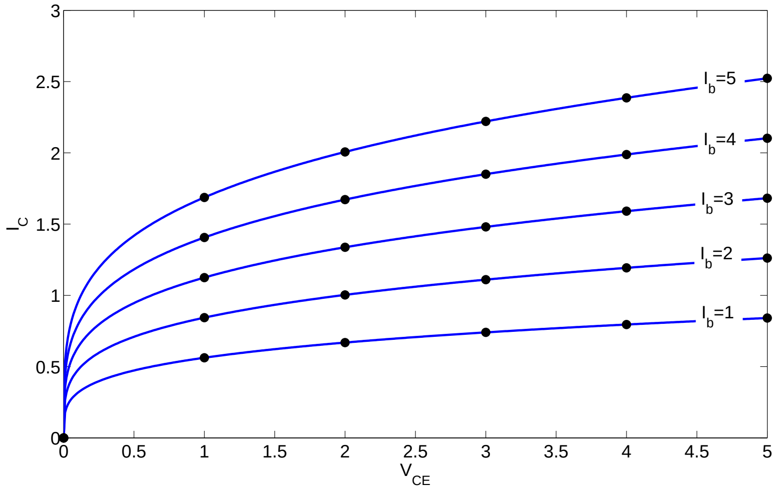
Ajuste los datos de la tabla con trazadores de tercer grado. Para ello utilice las funciones de MATLAB/OCTAVE **spline** y **ppval**. Evalúe la función en x = 2.25. Grafique los datos y el polinomio determinado.

x	1.6	2.0	2.5	3.2	4.0	4.5
$f(x)$	2.0	8.0	14.0	15.0	8.0	2.0

Ej. 8: Ejercicio libre

En cada uno de los siguientes incisos proponga el método de interpolación que crea más adecuado para la situación propuesta.

a) Un transistor posee las curvas de corriente de colector (I_c), en función de tensión Colector-Emisor (V_{ce}), para diferentes valores de corriente de base (I_b), como se muestra en la figura. Si se pretende interpolar las diferentes curvas, ¿qué método de interpolación cree más conveniente? ¿Por qué? Desarrolle el método propuesto si se quiere obtener una aproximación de I_c para $V_{ce} = 2.5$ considerando $I_b = 5$.



V_{ce}	1	2	3	4	5
$I_b = 1$	0.5623	0.6687	0.7401	0.7953	0.8409
$I_b = 2$	0.8435	1.0031	1.1101	1.1929	1.2613
$I_b = 3$	1.1247	1.3375	1.4802	1.5905	1.6818
$I_b = 4$	1.4059	1.6719	1.8502	1.9882	2.1022
$I_b = 5$	1.6870	2.0062	2.2202	2.3858	2.5227

b) Se pretende obtener un polinomio que interpole los puntos de la figura mostrada de modo tal que represente de la mejor manera el contorno superior de la misma. ¿Qué método propondría? Justifique adecuadamente. Resuelva eligiendo distintos grados para el polinomio de interpolación y grafíquelo junto con los puntos de la figura. Analice los resultados. <u>*Nota*</u>: Los puntos se encuentran en `puntos_pato.txt`. Utilice las funciones `load` o `importdata` para leer los datos.