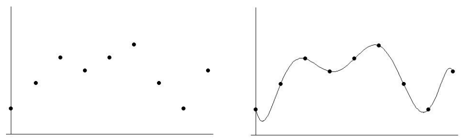
Introducción
Errores
Aproximación de raíces
Interpolación
Resolución Numérica de EDO's

Construcción del polinomio de interpolación

Interpolación

Dada una tabla de datos, se ha de encontrar una función que tome los valores requeridos en los puntos dados; en el caso que nos ocupa, la función buscada será de carácter polinómico.



Teorema: El polinomio de interpolación, si existe, es único.

Departamento de Matemática Aplicada | Cálculo Numérico

Introducción
Errores
Aproximación de raíces
Interpolación
Resolución Numérica de EDO's

Construcción del polinomio de interpolación

Para calcular el polinomio de interpolación asociado a una serie de nodos utilizaremos distintos métodos:

1. Coeficientes indeterminados
2. Polinomio de Lagrange
3. Polinomio de Newton
4. Interpolación a trozos
5. Splines cúbicos
6. Hermite

Departamento de Matemática Aplicada | Cálculo Numérico

Análisis de errores
Resolución de ecuaciones de una variable
Interpolación polinomial
Diferenciación e integración numérica

Construcción del polinomio de interpolación
Polinomio de Lagrange

Consideremos una tabla de n+1 puntos distintos (nodos):

x	x_0	x_1	…	x_n
y	y_0	y_1	…	y_n

Entonces existe un único polinomio de grado n, denominado polinomio de Lagrange, y = $P_n(x)$, tal que $P_n(x_i)=y_i$ para todo i=0,1,…,n.
La expresión explícita de dicho polinomio es la siguiente:

$$P_n(x) = \sum_{i=0,n} y_i L_i(x) = \sum_{i=0,n} y_i \prod_{\substack{j=0,n\\ j\neq i}} \frac{x-x_j}{x_i-x_j} = \sum_{i=0,n} \frac{y_i}{\prod_{\substack{j=0,n\\ j\neq i}} x_i-x_j} \prod_{\substack{j=0,n\\ j\neq i}} (x-x_j)$$

$$L_i(x) = \frac{(x-x_0)(x-x_1)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_n)}{(x_i-x_0)(x_i-x_1)\ldots(x_i-x_{i-1})(x_i-x_{i+1})\ldots(x_i-x_n)} = \prod_{\substack{j=0,n\\ j\neq i}} \frac{x-x_j}{x_i-x_j}$$

Departamento de Matemática Aplicada | Cálculo Numérico

Introducción
Errores
Aproximación de raíces
Interpolación
Resolución Numérica de EDO's

Construcción del polinomio de interpolación
Polinomio de Newton

Polinomio de Newton

La fórmula de interpolación de Newton proporciona una manera de calcular el polinomio P_{n+1} a partir del polinomio P_n añadiendo un término nuevo.

Lema: Sea $P_n(x)$ el polinomio de interpolación asociado a los puntos $x_0, x_1, \ldots, x_n$ y los valores $y_0, y_1, \ldots, y_n$, y sea $P_{n+1}(x)$ el polinomio de interpolación asociado a los puntos $x_0, x_1, \ldots, x_n, x_{n+1}$ y los valores $y_0, y_1, \ldots, y_n, y_{n+1}$. Existe una constante c_{n+1} tal que:

$$P_{n+1}(x) = P_n(x) + c_{n+1}(x-x_0)\ldots(x-x_n)$$

Departamento de Matemática Aplicada | Cálculo Numérico

Introducción
Errores
Aproximación de raíces
Interpolación
Resolución Numérica de EDO's

Construcción del polinomio de interpolación
Polinomio de Newton

Teorema (Método de Newton): En las condiciones anteriores

$$P_n(x_0) = f(x_0) + \sum_{k=1,n} c_k\,(x-x_0)\ldots(x-x_{k-1})$$

$$c_k = \sum_{j=0,k} \frac{f(x_j)}{\prod_{\substack{i\neq j\\ i=0,k}} (x_j-x_i)}$$

Departamento de Matemática Aplicada | Cálculo Numérico

Introducción
Errores
Aproximación de raíces
Interpolación
Resolución Numérica de EDO's

Construcción del polinomio de interpolación
Diferencias divididas

Consideremos una tabla de n+1 puntos distintos (nodos):

x	x_0	x_1	…	x_n
y	y_0	y_1	…	y_n

$f[x_0]$ = coeficiente de x^0 en $P_0(x)$

$f[x_0] = f(x_0) = y_0$

$f[x_0, x_1]$ = coeficiente de x^1 en $P_1(x)$

$$f[x_0,x_1] = \frac{f(x_1)-f(x_0)}{x_1-x_0} = y_{01}$$

Departamento de Matemática Aplicada | Cálculo Numérico

$f[x_0, x_1, x_2]$ = coeficiente de x^2 en $P_2(x)$

$$f[x_0,x_1,x_2]=\frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}=y_{012}$$

Teorema: El coeficiente del término n-ésimo es:

$$f[x_0,\dots,x_n]=\frac{f[x_1,\dots,x_n]-f[x_0,\dots,x_{n-1}]}{x_n-x_0}$$

Considerando $x_0,\dots,x_n$ como variables independientes, tenemos que:

$$f[x_i,x_{i+1}\dots,x_{i+j}]=\frac{f[x_{i+1},\dots,x_{i+j}]-f[x_i,\dots,x_{i+j-1}]}{x_{i+j}-x_i}$$

Introducción
Errores
Aproximación de raíces
Interpolación
Resolución Numérica de EDO's
Construcción del polinomio de interpolación
Diferencias divididas

Así podremos escribir la tabla:

$$\begin{array}{llllll}
x_0 & f[x_0] & & & & \\
 & & f[x_0,x_1] & & & \\
x_1 & f[x_1] & & f[x_0,x_1,x_2] & & \\
 & & f[x_1,x_2] & & f[x_0,x_1,x_2,x_3] & \\
x_2 & f[x_2] & & f[x_1,x_2,x_3] & & \\
 & & f[x_2,x_3] & & & \\
x_3 & f[x_3] & & & & \\
\dots & & & & &
\end{array}$$

Departamento de Matemática Aplicada
Cálculo Numérico

Introducción
Errores
Aproximación de raíces
Interpolación
Resolución Numérica de EDO's
Construcción del polinomio de interpolación
Diferencias divididas

Diferencias divididas de orden k

Se denominan diferencias divididas de **primer orden** a los cocientes:

$$y_{i,i+1}=f\left[x_i,x_{i+1}\right]=\left[x_i,x_{i+1}\right]=\frac{y_{i+1}-y_i}{x_{i+1}-x_i}\qquad 0\le i\le n-1$$

Se llaman diferencias divididas de **segundo orden** a:

$$y_{i,i+1,i+2}=f\left[x_i,x_{i+1},x_{i+2}\right]=\left[x_i,x_{i+1},x_{i+2}\right]=\frac{y_{i+1,i+2}-y_{i,i+1}}{x_{i+2}-x_i}\qquad 0\le i\le n-2$$

Departamento de Matemática Aplicada
Cálculo Numérico

Introducción
Errores
Aproximación de raíces
Interpolación
Resolución Numérica de EDO's
Construcción del polinomio de interpolación
Diferencias divididas

Se llaman diferencias divididas de ***orden k*** a: $0\le i\le n-k$

$$y_{i,i+1,\dots,i+k}=f\left[x_i,x_{i+1},\dots x_{i+k}\right]=\left[x_i,x_{i+1},\dots,x_{i+k}\right]=\frac{y_{i+1,\dots,i+k}-y_{i,\dots,i+k-1}}{x_{i+k}-x_i}$$

$$\begin{array}{llll}
f[x_0] & & & \\
 & f[x_0,x_1] & & \\
f[x_1] & & f[x_0,x_1,x_2] & \\
 & f[x_1,x_2] & & f[x_0,x_1,x_2,x_3] \\
f[x_2] & & f[x_1,x_2,x_3] & \\
 & f[x_2,x_3] & & \\
f[x_3] & & & \dots
\end{array}$$

Departamento de Matemática Aplicada
Cálculo Numérico

Introducción
Errores
Aproximación de raíces
Interpolación
Resolución Numérica de EDO's
Construcción del polinomio de interpolación
Diferencias divididas

El polinomio de Newton se puede redefinir de manera explícita:

$P_n(x) = y_0 + y_{01}(x-x_0) + y_{012}(x-x_0)(x-x_1)+\dots+y_{01\dots n}(x-x_0)(x-x_1)\dots(x-x_{n-}$

$$P_{n+1}(x) = P_n(x) + y_{012\dots n+1}(x - x_0)\cdot\dots\cdot(x - x_n)$$

Observaciones:

1.- De acuerdo con las definiciones y dada la unicidad del polinomio de interpolación, si $z_0,z_1,\dots,z_n$ es una permutación de los puntos $x_0, x_1,\dots,x_n$ se tiene que: $f[z_0,z_1,\dots,z_n] = f[x_0,x_1,\dots,x_n]$

2.- Se verifica que

$$y_{0,1,\dots,n}=\sum_{i=0,n}\frac{f(x_i)}{(x_i-x_0)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}$$

Departamento de Matemática Aplicada
Cálculo Numérico

Introducción
Errores
Aproximación de raíces
Interpolación
Resolución Numérica de EDO's
Construcción del polinomio de interpolación
Diferencias divididas

Comparando métodos

1.- El costo de escribir el polinomio de interpolación es máximo en el método de los coeficientes indeterminados, donde se requiere resolver un sistema lineal de n+1 ecuaciones y n+1 incógnitas. Ese costo es nulo para la forma de Lagrange e intermedio para la forma de Newton, donde hay que construir la tabla de diferencias divididas.

2.- En cuanto al costo de evaluar el polinomio en un punto, es evidente que la forma de Lagrange es la más complicada.

Departamento de Matemática Aplicada
Cálculo Numérico

3.- Si una vez calculado $P_n(x)$ se desea construir $P_{n+1}(x)$ que interpola en un nodo más, basta añadir un término al polinomio calculado si se utiliza el método de Newton y el coeficiente correspondiente en la tabla de diferencias divididas. En las formas de coeficientes indeterminados y de Lagrange no es inmediatamente posible utilizar los cálculos de la etapa anterior.

4.- En la forma de Newton, cada sumando de los que componen el polinomio tiene un significado. Son los términos que hay que añadir a un polinomio interpolador de cierto grado para transformarlo en el de un grado más que interpola en un punto más. Por eso, al ir sumando sucesivamente, es posible observar los efectos de pasar de la interpolación lineal a la cuadrática, de ésta a la cúbica, etc. En las otras dos formas, los sumandos individuales carecen de significado aprovechable.

5.- En conclusión, podemos decir que la forma de Newton es la más aconsejable. Es la que emplea Mathematica®, aunque la forma de Lagrange también se emplea.

Error en el polinomio de interpolación

Teorema. Sea f: [a,b] → R una función (n+1) veces derivable en (a,b) y sea y = P_n (x) el polinomio interpolador en los puntos:

x	$a=x_0$	x_1	…	$x_n=b$
y	$f(x_0)$	$f(x_1)$	…	$f(x_n)$

Entonces para todo x ∈ [a,b], existe un punto $c\in(a,b)$ tal que

$$f(x)-P_n(x)=\frac{f^{n+1}(c)}{(n+1)!}\prod_{i=0,n}(x-x_i)$$

Además, si $|f^{(n+1)}(x)| \le M$ para todo x ∈(a,b), se tiene la siguiente cota para el error: $|f(x)-P_n(x)|\le\frac{M}{(n+1)!}\left|\prod_{i=0,n}(x-x_i)\right|$

Observaciones:

1.- En las condiciones del enunciado, el teorema de Rolle garantiza que entre dos raíces de una función existe al menos una raíz de la función derivada.

2.- La fórmula del error de interpolación para un punto $x\neq x_0,\dots,x_n$ puede escribirse como:

$$f(x)-P_n(x)=f[x_0,\dots,x_n,x]\prod_{i=0,n}(x-x_i)$$

3.- Si f es derivable con continuidad en [a,b] hasta el orden n+1 y $x_0, \dots, x_n, x_{n+1}$ son puntos distintos de [a,b], existe $\xi\in(a,b)$ tal que

$$f[x_0,\dots,x_n,x]=\frac{f^{n+1}(\xi)}{(n+1)!}$$

Comentario:

Dada una función continua f sobre un intervalo [a,b], si se construyen polinomios de interpolación asociados a la función f con nodos equiespaciados y de grados cada vez mayores, podríamos esperar que la sucesión de polinomios (P_n) convergiera (uniformemente) a la función f en el intervalo [a,b]. Ello no sucede en general.

Los teoremas involucrados en este sentido quedan fuera de este nivel. Basta ilustrar la situación con un ejemplo debido a Runge en 1900.

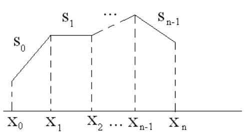
Si se interpola la función $f(x)=(1/(1+x^2))$ (que es diferenciable) sobre el intervalo [-5,5] en n+1 puntos equiespaciados se obtiene que $P_n(x)$ no converge al valor de $f(x)$ si |x|>3.6. En el Notebook de Mathematica puede observarse los grandes errores a que da origen P_{14}. Esto, por supuesto, no demuestra la divergencia de la sucesión $P_n(x)$.
Es decir, en general no es cierto que al aumentar el número de puntos sobre los que se construye el polinomio interpolador, n, el error disminuya. Existen funciones para las cuales el error aumenta al aumentar n.

Interpolación a trozos

La naturaleza oscilatoria de los polinomios de grado alto y su alta sensibilidad a pequeñas modificaciones limita su utilización.

Además hay que resolver el problema de la convergencia cuando el número de nodos es grande, la solución más frecuente es la interpolación polinómica a trozos.

Interpolación lineal a trozos



$$s(x)=\begin{cases} s_0(x) & si \quad x\in[x_0,x_1] \\ s_1(x) & si \quad x\in[x_1,x_2] \\ s_{n-1}(x) & si \quad x\in[x_{n-1},x_n] \end{cases}$$

$$s_i(x)=f[x_i]+f[x_{i,i+1}](x-x_i)$$

Sea $M_0^1(P)$ el conjunto de las funciones continuas en [a,b] que restringidas a cada subintervalo $[x_i, x_{i+1}]$ de la partición coincide con un polinomio de grado ≤1.

$M_0^1(P)$ es un espacio vectorial con las operaciones habituales.

Si $s \in M_0^1(P)$ decimos que es una función lineal a trozos en la partición P.

Sea $L_i(x)$ la función lineal a trozos de $M_0^1(P)$ que vale 1 en el nodo x_i y cero en los restantes

Error en la Interpolación lineal a trozos.

$$|f(x)-P(x)|=\frac{|f^{n+1}(x)|}{(n+1)!}\prod_{i=0,n}|x-x_i|$$

$$|f(x)-s(x)|\le\frac{K_2}{8}(x_i-x_{i+1})^2 \qquad K_2=\max_{x\in[a,b]}|f''(x)|$$

$$|f(x)-s(x)|\le\frac{K_2}{8}h^2 \qquad x\in[x_i,x_{i+1}]$$

1. Convergencia cuadrática: $\lim_{n\to\infty} s_n(x)=f(x)$

2. No es diferenciable en los extremos de los intervalos (x_i)

Comparación con el polinomio de interpolación de grado n:
1. Si n es grande, el costo de evaluar el polinomio P_n es grande. Sin embargo, el coste de evaluar la función s de interpolación lineal a trozos no crece con n.
2. No está garantizada la convergencia de P_n a f, ni aún suponiendo que f sea indefinidamente derivable. Sin embargo con sólo suponer que f tiene derivada segunda acotada, al refinar la partición sabemos que los interpolantes lineas a trozos convergen (cuadráticamente) a f.
3. El polinomio de interpolación P_n es indefinidamente derivable, mientras que la función s de interpolación lineal a trozos no es derivable, en general, en los nodos. Esta falta de regularidad hace que, a veces, no se pueda aplicar esta interpolación.

Interpolación por polinomios a trozos de grado 2.

$M_0^2(P)$ el conjunto de todas las funciones continuas en [a,b] que restringidas a cada subintervalo $[x_i, x_{i+1}]$ de la partición P coinciden con polinomios de grado ≤2.

Si $s\in M_0^2(P)$ decimos que s (interpolante cuadrático a trozos) es una función cuadrática a trozos en la partición P. En los puntos $x_i \in P$ las funciones $s\in M_0^2(P)$ presentan saltos en las derivadas primera y segunda.

$$s(x)=\begin{cases} s(x_i)=f(x_i) \\ s(x_i^*)=f(x_i^*) \\ s(x_{i+1})=f(x_{i+1}) \end{cases} \qquad x_i^*=\frac{x_i+x_{i+1}}{2}$$

Error en la Interpolación cuadrática a trozos.

$$|f(x)-P(x)| = \frac{|f^{n+1}(x)|}{(n+1)!} \prod_{i=0,n} |x-x_i|$$

$$K_3 = \max_{x\in[a,b]} |f'''(x)|$$

$$|f(x)-s(x)| \le \frac{\sqrt{3}}{216} K_3 h^3 \qquad x \in [x_i, x_{i+1}]$$

1. Convergencia cúbica $\quad \lim_{n\to\infty} s_n(x) = f(x)$

2. No es diferenciable en los extremos de los intervalos (x_i)

Interpolación cúbica a trozos. Interpolación SPLINE

1. Polinomio de grado 3 en cada subintervalo.
2. De clase C^2 en el intervalo [a, b].
3. Interpolatorio.
4. Condiciones frontera:
   1. Libre: $s''(x_0) = s''(x_n)=0$
   2. Fija: $s'(x_0) = f'(x_0)$ y $s'(x_n) = f'(x_n)$

$$a_{i+1} = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3, \qquad i=0,1,\dots,n-2$$

$$b_{i+1} = b_i + 2c_i h_i + 3d_i h_i^2, \qquad i=0,1,\dots,n-2$$

$$c_{i+1} = c_i + 3d_i h_i\,, \qquad i=0,1,\dots,n-2$$

$$a_i = y_i, \qquad i=0,1,\dots,n-1$$

$$c_0 = 0,$$

$$2c_{n-1} + 6d_{n-1}h_{n-1} = 0,$$

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & \vdots & \vdots & \vdots & 0 \\ 0 & h_1 & 2(h_1+h_2) & h_2 & \cdots & \cdots & \\ \vdots & \vdots & \vdots & & & & 0 \\ & & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & & h_{n-1} \\ 0 & \cdots & \cdots & 0 & 0 & & 1 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{3}{h_1}(a_2-a_1) - \frac{3}{h_0}(a_1-a_0) \\ \frac{3}{h_2}(a_3-a_2) - \frac{3}{h_1}(a_2-a_1) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n-a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1}-a_{n-2}) \\ 0 \end{pmatrix}$$

Interpolación de Hermite

El término interpolación de Hermite se refiere a la interpolación de una función y de alguna de sus derivadas en un conjunto de nodos. La interpolación que hemos estudiado hasta ahora se llama, por diferencia de esta, interpolación de Lagrange.

Teorema: Existe un único polinomio *P*(x) de grado ≤m que satisface las condiciones de interpolación de Hermite:

$$P^{(j)}(x_i) = c_{ij}, \qquad j \le k_i - 1,\ 0 \le i \le n$$

Interpolación cúbica de Hermite a trozos

Dada una función derivable $f: [a, b]\to\Re$ y n+1 puntos $x_0=a < x_1 < \dots < x_{n-1} < x_n= b$, existe una única función derivable $g:[a,b]\to\Re$, de modo que en cada intervalo $[x_i, x_{i+1}]$, $y = g(x)$ está definida por un polinomio de tercer grado, $y = P_i(x)$, tal que:

$$P_i(x_i) = f(x_i) \qquad P_i'(x_i) = f'(x_i)$$

$$P_i(x_{i+1}) = f(x_{i+1}) \qquad P_i'(x_{i+1}) = f'(x_{i+1})$$

$$g(x) = \begin{cases} P_0(x) & si \quad x\in[x_0,x_1] \\ \dots & \\ P_{n-1}(x) & si \quad x\in[x_{n-1},x_n] \end{cases}$$

Con $\quad 0 \le i \le n\text{-}1$