Interpolação

Prof. Afonso Paiva

Departamento de Matemática Aplicada e Estatística
Instituto de Ciências Matemáticas e de Computação
USP – São Carlos

Métodos Numéricos e Computacionais II – SME0306

Motivação

A tabela abaixo mostra as temperaturas máximas (em graus Celsius) atingidas no mês de agosto a cada 5 dias:

dia	5	10	15	20	25	30
temperatura	27.8	30.1	33.8	31.8	32.1	35.4

Motivação

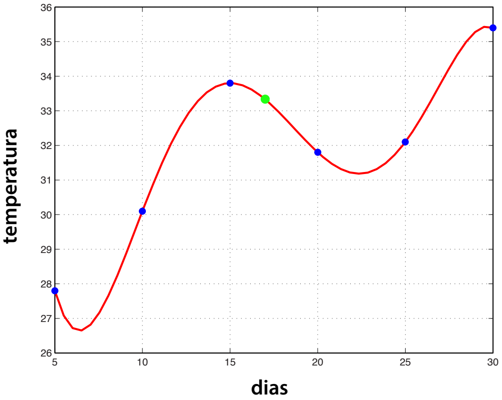
A tabela abaixo mostra as temperaturas máximas (em graus Celsius) atingidas no mês de agosto a cada 5 dias:

dia	5	10	15	20	25	30
temperatura	27.8	30.1	33.8	31.8	32.1	35.4

Como estimar a temperatura no dia 17?

Motivação

Plote uma curva suave (de classe $\mathcal{C}^k$) conectando esses pontos.



Motivação

Suponha que temos uma função complicada:

$$f(x) = \frac{\log(x)\, e^{\sqrt{x}}}{\log(1 + e^x) + x^3}$$

Motivação

Suponha que temos uma função complicada:

$$f(x) = \frac{\log(x)\, e^{\sqrt{x}}}{\log(1 + e^x) + x^3}$$

Como representar tal função de uma forma simples?

Em outras palavras, queremos avaliar, derivar, integrar $f(x)$ de uma maneira fácil e rápida.

Motivação

Suponha que temos uma função complicada:

$$f(x) = \frac{\log(x)\, e^{\sqrt{x}}}{\log(1 + e^x) + x^3}$$

Como representar tal função de uma forma simples?

Em outras palavras, queremos avaliar, derivar, integrar $f(x)$ de uma maneira fácil e rápida.

- Uma estratégia:

Motivação

Suponha que temos uma função complicada:

$$f(x) = \frac{\log(x)\, e^{\sqrt{x}}}{\log(1+e^x)+x^3}$$

Como representar tal função de uma forma simples?

Em outras palavras, queremos avaliar, derivar, integrar $f(x)$ de uma maneira fácil e rápida.

- Uma estratégia:
  1. Avalie $y = f(x)$ em alguns pontos $\{x_0, x_1, \ldots, x_n\}$;

Motivação

Suponha que temos uma função complicada:

$$f(x) = \frac{\log(x)\, e^{\sqrt{x}}}{\log(1+e^x) + x^3}$$

Como representar tal função de uma forma simples?

Em outras palavras, queremos avaliar, derivar, integrar $f(x)$ de uma maneira fácil e rápida.

- Uma estratégia:
  1. Avalie $y = f(x)$ em alguns pontos $\{x_0, x_1, \ldots, x_n\}$;
  2. Aproxime a função amostrada $y_i = f(x_i)$, $i = 0, \ldots, n$, por uma função polinomial;

Motivação

Suponha que temos uma função complicada:

$$f(x) = \frac{\log(x)\, e^{\sqrt{x}}}{\log(1+e^x)+x^3}$$

Como representar tal função de uma forma simples?

Em outras palavras, queremos avaliar, derivar, integrar $f(x)$ de uma maneira fácil e rápida.

- Uma estratégia:
  1. Avalie $y = f(x)$ em alguns pontos $\{x_0, x_1, \ldots, x_n\}$;
  2. Aproxime a função amostrada $y_i = f(x_i)$, $i = 0, \ldots, n$, por uma função polinomial;
  3. Avalie, integre ou calcule as derivadas da função polinomial.

Interpolação

- Problema Básico de Interpolação:

Interpolação

- Problema Básico de Interpolação:

  Dados $(n+1)$ pontos

  $$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n) \quad \text{com} \quad x_0 < x_1 < \cdots < x_n\,.$$

  Determine uma função $F : \mathbb{R} \to \mathbb{R}$ tal que

  $$\boxed{y_i = F(x_i), \qquad i = 0, \ldots, n\,.}$$

Interpolação

- Problema Básico de Interpolação:

  Dados $(n+1)$ pontos

  $$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n) \quad \text{com} \quad x_0 < x_1 < \cdots < x_n\,.$$

  Determine uma função $F:\ \mathbb{R} \to\ \mathbb{R}$ tal que

  $$\boxed{y_i = F(x_i), \qquad i = 0, \ldots, n\,.}$$

  - A função $F(x)$ é a função interpoladora, ou interpolante, dos pontos dados;

Interpolação

- Problema Básico de Interpolação:

  Dados $(n+1)$ pontos

  $$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n) \quad \text{com} \quad x_0 < x_1 < \cdots < x_n\,.$$

  Determine uma função $F: \mathbb{R} \to \mathbb{R}$ tal que

  $$\boxed{y_i = F(x_i), \qquad i = 0, \ldots, n\,.}$$

  - A função $F(x)$ é a função interpoladora, ou interpolante, dos pontos dados;
  - Os pontos x_i são chamados de nós da interpolação.

Interpolação Polinomial

Dados $(n+1)$ pontos

$$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n) \quad \text{com} \quad x_0 < x_1 < \cdots < x_n \,.$$

Agora queremos encontrar um polinômio $P_n(x)$ de grau $\leq n$ que satisfaz as seguintes condições:

$$\boxed{y_i = P_n(x_i), \qquad i = 0, \ldots, n \,.}$$

O polinômio $P_n(x)$ é chamado de polinômio de interpolação.

Interpolação Polinomial

Dados $(n+1)$ pontos

$$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n) \quad \text{com} \quad x_0 < x_1 < \cdots < x_n \, .$$

Agora queremos encontrar um polinômio $P_n(x)$ de grau $\leq n$ que satisfaz as seguintes condições:

$$\boxed{y_i = P_n(x_i), \qquad i = 0, \ldots, n \, .}$$

O polinômio $P_n(x)$ é chamado de polinômio de interpolação.

Teorema

Dados $(n+1)$ pontos $(x_0, y_0), \ldots, (x_n, y_n)$ com $x_0 < \cdots < x_n$, existe um único polinômio $P_n(x) \in \mathcal{P}_n$ que satisfaz as condições acima.

Interpolação Polinomial

Demonstração: para cada ponto (x_i, y_i), vamos impor a condição de interpolação ao polinômio $P_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$. Logo,

Interpolação Polinomial

Demonstração: para cada ponto (x_i, y_i), vamos impor a condição de interpolação ao polinômio $P_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$. Logo,

$$y_i = P_n(x_i) = a_0 + a_1x_i + a_2x_i^2 + \cdots + a_nx_i^n \, , \; i = 0, \ldots, n$$

Interpolação Polinomial

Demonstração: para cada ponto (x_i, y_i), vamos impor a condição de interpolação ao polinômio $P_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$. Logo,

$$\underbrace{\begin{bmatrix} 1 & x_0 & x_0^2 & & x_0^n \\ 1 & x_1 & x_1^2 & & x_1^2 \\ 1 & x_2 & x_2^2 & & x_2^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & & x_n^n \end{bmatrix}}_{\mathbf{X}} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Interpolação Polinomial

Demonstração: para cada ponto (x_i, y_i), vamos impor a condição de interpolação ao polinômio $P_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$. Logo,

$$\underbrace{\begin{bmatrix} 1 & x_0 & x_0^2 & & x_0^n \\ 1 & x_1 & x_1^2 & & x_1^2 \\ 1 & x_2 & x_2^2 & & x_2^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & & x_n^n \end{bmatrix}}_{\mathbf{X}} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Basta mostrar que o determinante da matriz de Vandermond $\mathbf{X}$ é não nulo:

$$\det(\mathbf{X}) = \prod_{i<k} (x_k - x_i) \neq 0, \quad \text{pois } x_k \neq x_i$$

Interpolação Polinomial

Demonstração: para cada ponto (x_i, y_i), vamos impor a condição de interpolação ao polinômio $P_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$. Logo,

$$\underbrace{\begin{bmatrix} 1 & x_0 & x_0^2 & & x_0^n \\ 1 & x_1 & x_1^2 & & x_1^2 \\ 1 & x_2 & x_2^2 & & x_2^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & & x_n^n \end{bmatrix}}_{\mathbf{X}} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Basta mostrar que o determinante da matriz de Vandermond $\mathbf{X}$ é não nulo:

$$\det(\mathbf{X}) = \prod_{i<k} (x_k - x_i) \neq 0, \quad \text{pois } x_k \neq x_i$$

Mas como calcular $P_n(x)$?

Forma de Lagrange

A forma de Lagrange para o polinômio de interpolação $P_n(x)$ nos pontos $(x_0, y_0), \ldots, (x_n, y_n)$ é dado por:

$$\boxed{P_n(x) = y_0\,\ell_0(x) + y_1\,\ell_1(x) + \cdots + y_n\,\ell_n(x) = \sum_{k=0}^{n} y_k\,\ell_k(x)\,,}$$

onde $\ell_k(x) \in \mathcal{P}_n$ são polinômios que dependem apenas de $x_0, \ldots, x_n$.

Forma de Lagrange

Por outro lado, $P_n(x_i) = y_0\,\ell_0(x_i) + \cdots + y_i\,\ell_i(x_i) + \cdots + y_n\,\ell_n(x_i) = y_i$.

Forma de Lagrange

Por outro lado, $P_n(x_i) = y_0\,\ell_0(x_i) + \cdots + y_i\,\ell_i(x_i) + \cdots + y_n\,\ell_n(x_i) = y_i$.
Assim, temos a seguinte relação:

$$\ell_k(x_i) = \delta_{ik} = \begin{cases} 1, & \text{se } i = k \\ 0, & \text{se } i \neq k \end{cases}$$

Forma de Lagrange

Por outro lado, $P_n(x_i) = y_0\,\ell_0(x_i) + \cdots + y_i\,\ell_i(x_i) + \cdots + y_n\,\ell_n(x_i) = y_i$.
Assim, temos a seguinte relação:

$$\ell_k(x_i) = \delta_{ik} = \begin{cases} 1, & \text{se } i = k \\ 0, & \text{se } i \neq k \end{cases} \implies \{x_0, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n\} \text{ raízes de } \ell_k$$

Forma de Lagrange

Por outro lado, $P_n(x_i) = y_0\,\ell_0(x_i) + \cdots + y_i\,\ell_i(x_i) + \cdots + y_n\,\ell_n(x_i) = y_i$.
Assim, temos a seguinte relação:

$$\ell_k(x_i) = \delta_{ik} = \begin{cases} 1, & \text{se } i = k \\ 0, & \text{se } i \neq k \end{cases} \implies \{x_0, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n\} \text{ raízes de } \ell_k$$

Podemos escrever ℓ_k como:

$$\ell_k(x) = a \prod_{\substack{i=0 \\ i \neq k}}^{n} (x - x_i)$$

Forma de Lagrange

Por outro lado, $P_n(x_i) = y_0\,\ell_0(x_i) + \cdots + y_i\,\ell_i(x_i) + \cdots + y_n\,\ell_n(x_i) = y_i$.
Assim, temos a seguinte relação:

$$\ell_k(x_i) = \delta_{ik} = \begin{cases} 1, & \text{se } i = k \\ 0, & \text{se } i \neq k \end{cases} \implies \{x_0, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n\} \text{ raízes de } \ell_k$$

Podemos escrever ℓ_k como:

$$\ell_k(x) = a \prod_{\substack{i=0 \\ i \neq k}}^{n} (x - x_i) \implies 1 = \ell_k(x_k) = a \prod_{\substack{i=0 \\ i \neq k}}^{n} (x_k - x_i)$$

Forma de Lagrange

Portanto, o polinômio de Lagrange $\ell_k(x)$ é dado por:

$$\boxed{\ell_k(x) = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{x - x_i}{x_k - x_i}, \quad k = 0, \ldots, n\,.}$$

Forma de Lagrange

Exemplo 1

Dada a função tabelada $y = f(x)$:

x	-2	0	3	5
y	3	-2	4	2

Calcule o polinômio de interpolação com a forma de Lagrange usando todos os pontos da tabela.

Forma de Lagrange

Solução:

x	-2	0	3	5
y	3	-2	4	2

$$P_3(x) = 3\ell_0(x) - 2\ell_1(x) + 4\ell_2(x) + 2\ell_3(x)$$

Forma de Lagrange

Solução:

x	-2	0	3	5
y	3	-2	4	2

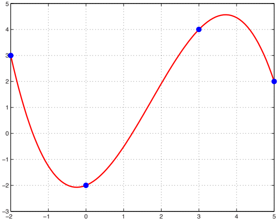
$$P_3(x) = 3\ell_0(x) - 2\ell_1(x) + 4\ell_2(x) + 2\ell_3(x)$$

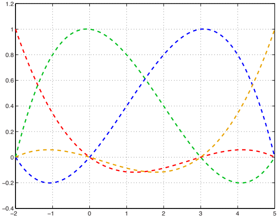
$$\ell_0(x) = \frac{x(x-3)(x-5)}{-70}$$

$$\ell_1(x) = \frac{(x+2)(x-3)(x-5)}{30}$$

$$\ell_2(x) = \frac{(x+2)x(x-5)}{-30}$$

$$\ell_3(x) = \frac{(x+2)x(x-3)}{70}$$





Polinômios de Lagrange

Forma de Lagrange

Solução:

x	-2	0	3	5
y	3	-2	4	2

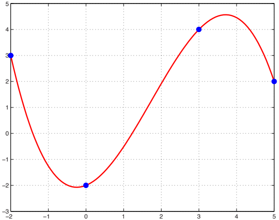
$$P_3(x) = 3\ell_0(x) - 2\ell_1(x) + 4\ell_2(x) + 2\ell_3(x)$$

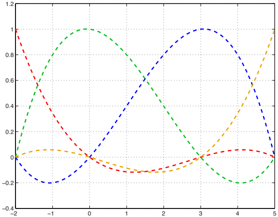
$$\ell_0(x) = \frac{x(x-3)(x-5)}{-70}$$

$$\ell_1(x) = \frac{(x+2)(x-3)(x-5)}{30}$$

$$\ell_2(x) = \frac{(x+2)x(x-5)}{-30}$$

$$\ell_3(x) = \frac{(x+2)x(x-3)}{70}$$




Polinômios de Lagrange

Atenção: se adicionarmos mais um ponto $(x_{n+1}, y_{n+1}) \Rightarrow$ todos $\ell_k(x)$ precisam ser recalculados!

MATLAB – Forma de Lagrange

```
function y = lagrange_interp(xi,yi,x)
% xi, yi, x: vetor linha ou coluna
[m,n]= size (xi);
if (n == 1) xi = xi'; yi = yi'; x = x'; n = m; end

L = ones(n,length(x));

for i =1:n
    for j =1:n
        if ( i ~= j )
            L(i,:) = L(i,:).*(x-xi(j))/(xi(i)-xi(j));
        end
    end
end

y = yi*L;
```

Forma de Newton

A forma de Newton para $P_n(x)$ é dada de maneira diferente:

$$\begin{aligned} P_n(x) &= \alpha_0 + \alpha_1(x - x_0) + \alpha_2(x - x_0)(x - x_1) + \cdots + \\ &+ \alpha_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}) \end{aligned}$$

Forma de Newton

A forma de Newton para $P_n(x)$ é dada de maneira diferente:

$$\begin{aligned} P_n(x) &= \alpha_0 + \alpha_1(x - x_0) + \alpha_2(x - x_0)(x - x_1) + \cdots + \\ &+ \alpha_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}) \end{aligned}$$

Cada coeficiente α_k é determinado por uma diferença dividida de ordem k:

$$\boxed{\alpha_k = f[x_0, x_1, \ldots, x_k]\,, \quad k = 0, 1, \ldots, n\,.}$$

Forma de Newton

Definição (diferenças divididas)

As diferenças divididas são definidas recursivamente:

$$f[x_i] := f(x_i)\,, \quad i = 0, \ldots, n\,.$$

$$f[x_i, x_{i+1}, \ldots, x_{i+k}] := \frac{f[x_{i+1}, x_{i+2}, \ldots, x_{i+k}] - f[x_i, x_{i+1}, \ldots, x_{i+k-1}]}{x_{i+k} - x_i}\,,$$

com $k = 1, \ldots, n$ *e* $i = 0, \ldots, n-k$.

Diferenças Divididas

- Diferença dividida de ordem 0:

$$f[x_i] = f(x_i)$$

Diferenças Divididas

- Diferença dividida de ordem 0:

$$f[x_i] = f(x_i)$$

- Diferença dividida de ordem 1:

$$f[x_i, x_j] = \frac{f[x_j] - f[x_i]}{x_j - x_i} = \frac{f(x_j) - f(x_i)}{x_j - x_i}$$

Diferenças Divididas

- Diferença dividida de ordem 0:

$$f[x_i] = f(x_i)$$

- Diferença dividida de ordem 1:

$$f[x_i, x_j] = \frac{f[x_j] - f[x_i]}{x_j - x_i} = \frac{f(x_j) - f(x_i)}{x_j - x_i}$$

- Diferença dividida de ordem k:

$$f[x_{i_0}, x_{i_1}, \ldots, x_{i_k}] = \frac{f[x_{i_1}, x_{i_2}, \ldots, x_{i_k}] - f[x_{i_0}, x_{i_1}, \ldots, x_{i_{k-1}}]}{x_{i_k} - x_{i_0}}$$

Diferenças Divididas

Recursivamente temos a seguinte tabela de diferenças divididas.

Diferenças Divididas

Recursivamente temos a seguinte tabela de diferenças divididas.

x	ordem 0	ordem 1	ordem 2	ordem 3	$\cdots$
x_0	$f[x_0] = \alpha_0$	$f[x_0, x_1] = \alpha_1$	$f[x_0, x_1, x_2] = \alpha_2$	$f[x_0, x_1, x_2, x_3] = \alpha_3$	$\cdots$
x_1	$f[x_1]$	$f[x_1, x_2]$	$f[x_1, x_2, x_3]$	$\vdots$	
x_2	$f[x_2]$	$f[x_2, x_3]$	$\vdots$	$\vdots$	
x_3	$f[x_3]$	$\vdots$	$\vdots$	$\vdots$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

Forma de Newton

Exemplo 2

Calcule o polinômio de interpolação com a forma de Newton usando todos os pontos da tabela do Exemplo 1.

Forma de Newton

Exemplo 2

Calcule o polinômio de interpolação com a forma de Newton usando todos os pontos da tabela do Exemplo 1.

Solução: Primeiro vamos montar a tabela de diferenças divididas

x	ordem 0	ordem 1	ordem 2	ordem 3
-2	3	$-5/2$	$9/10$	$-3/14$
0	-2	2	$-3/5$	
3	4	-1		
5	2			

Forma de Newton

Exemplo 2

Calcule o polinômio de interpolação com a forma de Newton usando todos os pontos da tabela do Exemplo 1.

Solução: Primeiro vamos montar a tabela de diferenças divididas

x	ordem 0	ordem 1	ordem 2	ordem 3
-2	3	$-5/2$	$9/10$	$-3/14$
0	-2	2	$-3/5$	
3	4	-1		
5	2			

$$P_3(x) = 3 - \frac{5}{2}(x+2) + \frac{9}{10}(x+2)x - \frac{3}{14}(x+2)x(x-3)$$

MATLAB – Forma de Newton

```
function y = newton_interp(xi,yi,x)
% xi, yi, x: vetor linha ou coluna
[m,n]= size (x);
if (n == 1) xi = xi'; yi = yi'; n = m; end
n = length(xi); ni = length(x); N = ones(n,ni);
D=zeros(n); D(:,1) = yi';

for j=1:n-1 % tabela de diferencas divididas
    for i=1:n-j
        D(i,j+1) = (D(i+1,j)-D(i,j))/(xi(i+j)-xi(i));
    end
end
for i=2:n % forma de Newton
    N(i,:) = N(i-1,:).*(x-xi(i-1));
end

y = D(1,:)*N;
```

Interpolando Derivadas

Agora vamos admitir também derivadas no nós de interpolação:

Interpolando Derivadas

Agora vamos admitir também derivadas no nós de interpolação:

$$f(0) = 2\,, \qquad f'(0) = 1\,, \qquad f(10) = 0\,.$$

Interpolando Derivadas

Agora vamos admitir também derivadas no nós de interpolação:

$$f(0) = 2\,, \qquad f'(0) = 1\,, \qquad f(10) = 0\,.$$

Dessa forma temos $\{(x_i, y_i)\} = \{(0, 2), (0, 1), (10, 0)\}$.

Interpolando Derivadas

Agora vamos admitir também derivadas no nós de interpolação:

$$f(0) = 2\,, \qquad f'(0) = 1\,, \qquad f(10) = 0\,.$$

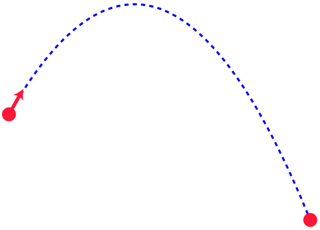
Dessa forma temos $\{(x_i, y_i)\} = \{(0, 2), (0, 1), (10, 0)\}$. Utilizando a base canônica de $\mathcal{P}_2 \Rightarrow P_2(x) = a + bx + cx^2$.

Interpolando Derivadas

Agora vamos admitir também derivadas no nós de interpolação:

$$f(0) = 2\,, \qquad f'(0) = 1\,, \qquad f(10) = 0\,.$$

Dessa forma temos $\{(x_i, y_i)\} = \{(0,2),(0,1),(10,0)\}$. Utilizando a base canônica de $\mathcal{P}_2 \Rightarrow P_2(x) = a + bx + cx^2$. Logo,

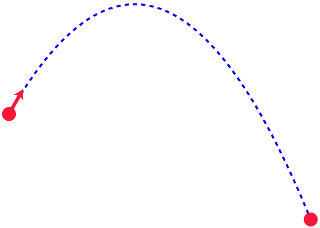


Interpolando Derivadas

Agora vamos admitir também derivadas no nós de interpolação:

$$f(0) = 2\,, \qquad f'(0) = 1\,, \qquad f(10) = 0\,.$$

Dessa forma temos $\{(x_i, y_i)\} = \{(0, 2), (0, 1), (10, 0)\}$. Utilizando a base canônica de $\mathcal{P}_2 \Rightarrow P_2(x) = a + bx + cx^2$. Logo,



$$2 = P_2(0) = a,$$

$$1 = P_2'(0) = b,$$

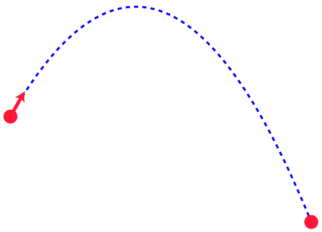
$$0 = P_2(10) = a + 10b + 100c \Rightarrow c = -\frac{12}{100}$$

Interpolando Derivadas

Agora vamos admitir também derivadas no nós de interpolação:

$$f(0) = 2\,, \qquad f'(0) = 1\,, \qquad f(10) = 0\,.$$

Dessa forma temos $\{(x_i, y_i)\} = \{(0, 2), (0, 1), (10, 0)\}$. Utilizando a base canônica de $\mathcal{P}_2 \Rightarrow P_2(x) = a + bx + cx^2$. Logo,



$$2 = P_2(0) = a,$$

$$1 = P_2'(0) = b,$$

$$0 = P_2(10) = a + 10b + 100c \Rightarrow c = -\frac{12}{100}$$

Portanto, $P_2(x) = 2 + x - \frac{12}{100}x^2$.

Interpolação de Hermite

- Interpolação de Hermite fornece um polinômio de interpolação de $f(x)$ e das derivadas $f'(x), f''(x), f^{(3)}(x)$, etc...

Interpolação de Hermite

- Interpolação de Hermite fornece um polinômio de interpolação de $f(x)$ e das derivadas $f'(x), f''(x), f^{(3)}(x)$, etc...
- As condições de interpolação em cada nó x_i são dadas por:

$$P_m^{(j)}(x_i) = f^{(j)}(x_i) = c_{ij}\,, \qquad 0 \leq j \leq k_i - 1 \text{ e } 0 \leq i \leq n$$

$$m + 1 = \kappa_0 + \kappa_1 + \cdots + \kappa_n\,,$$

onde κ_i é o número de condições em um nó x_i.

Interpolação de Hermite

- Interpolação de Hermite fornece um polinômio de interpolação de $f(x)$ e das derivadas $f'(x), f''(x), f^{(3)}(x)$, etc...
- As condições de interpolação em cada nó x_i são dadas por:

$$P_m^{(j)}(x_i) = f^{(j)}(x_i) = c_{ij}\,, \qquad 0 \leq j \leq k_i - 1 \;\text{ e }\; 0 \leq i \leq n$$

$$m + 1 = \kappa_0 + \kappa_1 + \cdots + \kappa_n\,,$$

onde κ_i é o número de condições em um nó x_i.

Teorema

Existe um único polinômio $P_m(x) \in \mathcal{P}_m$ que satisfaz as condições (de interpolação de Hermite) acima.

Forma de Newton

Considerações:

- Para isso vamos usar diferenças divididas. Seja x_k um nó de interpolação duplo. Logo,

$$f[x_k, x_k] = \lim_{h\to 0} \frac{f(x_k+h) - f(x_k)}{(x_k+h) - x_k} = \lim_{h\to 0} \frac{f(x_k+h) - f(x_k)}{h} = f'(x_k)\,.$$

Forma de Newton

Considerações:

- Para isso vamos usar diferenças divididas. Seja x_k um nó de interpolação duplo. Logo,

$$f[x_k, x_k] = \lim_{h \to 0} \frac{f(x_k + h) - f(x_k)}{(x_k + h) - x_k} = \lim_{h \to 0} \frac{f(x_k + h) - f(x_k)}{h} = f'(x_k)\,.$$

- No caso geral, podemos escrever

$$f[\underbrace{x_k, \ldots, x_k}_{m \text{ vezes}}] = \frac{1}{(m-1)!} f^{(m-1)}(x_k)$$

Forma de Newton

Considerações:

- Para isso vamos usar diferenças divididas. Seja x_k um nó de interpolação duplo. Logo,

$$f[x_k, x_k] = \lim_{h\to 0} \frac{f(x_k + h) - f(x_k)}{(x_k + h) - x_k} = \lim_{h\to 0} \frac{f(x_k + h) - f(x_k)}{h} = f'(x_k)\,.$$

- No caso geral, podemos escrever

$$f[\underbrace{x_k, \ldots, x_k}_{m \text{ vezes}}] = \frac{1}{(m-1)!} f^{(m-1)}(x_k)$$

- Perceba a relação com polinômios de Taylor!

$$P_n(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!} (x-a)^k \qquad \text{(em torno de } a\text{)}$$

Forma de Newton

Dados $(n+1)$ pontos $(x_0, y_0), \ldots, (x_n, y_n)$ com $x_0 < \cdots < x_n$.

Queremos obter $P_{2n+1}(x)$ que satisfaz as $(2n+2)$ condições:

$$P_{2n+1}(x_i) = c_{i0}\,, \qquad P'_{2n+1}(x_i) = c_{i1}\,, \qquad i = 0, \ldots, n\,.$$

Forma de Newton

Dados $(n+1)$ pontos $(x_0, y_0), \ldots, (x_n, y_n)$ com $x_0 < \cdots < x_n$.

Queremos obter $P_{2n+1}(x)$ que satisfaz as $(2n+2)$ condições:

$$P_{2n+1}(x_i) = c_{i0}\,, \qquad P'_{2n+1}(x_i) = c_{i1}\,, \qquad i = 0, \ldots, n\,.$$

Para mostrar como é o procedimento, consideremos o caso $n = 1$

x	ordem 0	ordem 1	ordem 2	ordem 3
x_0	$f[x_0] = c_{00} = \alpha_0$	$f[x_0, x_0] = c_{01} = \alpha_1$	$f[x_0, x_0, x_1] = \alpha_2$	$f[x_0, x_0, x_1, x_1] = \alpha_3$
x_0	$f[x_0]$	$f[x_0, x_1]$	$f[x_0, x_1, x_1]$	
x_1	$f[x_1] = c_{10}$	$f[x_1, x_1] = c_{11}$		
x_1	$f[x_1]$			

O polinômio de Hermite possui termos quadráticos

$$P_3 = \alpha_0 + \alpha_1(x - x_0) + \alpha_2(x - x_0)^2 + \alpha_3(x - x_0)^2(x - x_1)$$

Forma de Newton

Exemplo 3

Calcule o polinômio $p(x)$ de Hermite usando a forma de Newton com as seguintes condições:

$$p(1) = 2 \qquad p'(1) = 3 \qquad p(2) = 6 \qquad p'(2) = 7 \qquad p''(2) = 8$$

Forma de Newton

Exemplo 3

Calcule o polinômio $p(x)$ de Hermite usando a forma de Newton com as seguintes condições:

$$p(1) = 2 \qquad p'(1) = 3 \qquad p(2) = 6 \qquad p'(2) = 7 \qquad p''(2) = 8$$

Solução: Primeiro vamos montar a tabela de diferenças divididas

x	ordem 0	ordem 1	ordem 2	ordem 3	ordem 4
1	2	3	1	2	-1
1	2	4	3	1	
2	6	7	4		
2	6	7			
2	6				

Forma de Newton

Exemplo 3

Calcule o polinômio $p(x)$ de Hermite usando a forma de Newton com as seguintes condições:

$$p(1) = 2 \qquad p'(1) = 3 \qquad p(2) = 6 \qquad p'(2) = 7 \qquad p''(2) = 8$$

Solução: Primeiro vamos montar a tabela de diferenças divididas

x	ordem 0	ordem 1	ordem 2	ordem 3	ordem 4
1	2	3	1	2	-1
1	2	4	3	1	
2	6	7	4		
2	6	7			
2	6				

$$p(x) = 2 + 3(x-1) + (x-1)^2 + 2(x-1)^2(x-2) - (x-1)^2(x-2)^2$$

Erro na Interpolação

Teorema (erro para interpolação de Lagrange e Newton)

Sejam $f \in C^{n+1}([a,b])$, $a = x_0 < x_1 < \cdots < x_n = b$ *e* $P_n(x)$ *o polinômio de interpolação de* $f(x)$ *então*

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

onde $\xi = \xi(x) \in (a,b)$.

Erro na Interpolação

Teorema (erro para interpolação de Lagrange e Newton)

Sejam $f \in C^{n+1}([a,b])$, $a = x_0 < x_1 < \cdots < x_n = b$ e $P_n(x)$ o polinômio de interpolação de $f(x)$ então

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

onde $\xi = \xi(x) \in (a,b)$.

Demonstração: Burden & Faires, Seção 3.1.

Erro na Interpolação

Teorema (erro para interpolação de Hermite)

Sejam $f \in C^{2n+2}([a,b])$, $a = x_0 < x_1 < \cdots < x_n = b$ e $P_{2n+1}(x)$ o polinômio de interpolação de Hermite que satisfaz

$$P_{2n+1}(x_i) = f(x_i)\,, \qquad P'_{2n+1}(x_i) = f'(x_i)\,, \qquad i = 0, \ldots, n$$

então para ponto $x \in [a,b]$ há um $\xi = \xi(x) \in (a,b)$, tal que

$$R_n(x) = f(x) - P_n(x) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \prod_{i=0}^{n} (x - x_i)^2\,.$$

Erro na Interpolação

Teorema (erro para interpolação de Hermite)

Sejam $f \in \mathcal{C}^{2n+2}([a,b])$, $a = x_0 < x_1 < \cdots < x_n = b$ *e* $P_{2n+1}(x)$ *o polinômio de interpolação de Hermite que satisfaz*

$$P_{2n+1}(x_i) = f(x_i)\,, \qquad P'_{2n+1}(x_i) = f'(x_i)\,, \qquad i = 0, \ldots, n$$

então para ponto $x \in [a,b]$ *há um* $\xi = \xi(x) \in (a,b)$, *tal que*

$$R_n(x) = f(x) - P_n(x) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \prod_{i=0}^{n} (x - x_i)^2\,.$$

Demonstração: Burden & Faires, Seção 3.3.

Estimativas de Erro na Interpolação

Corolário

- $|E_n(x)| \leq \frac{1}{(n+1)!} \|f^{(n+1)}\|_\infty \prod_{i=0}^{n} |x - x_i|$
- $\|E_n\|_\infty \leq \frac{1}{(n+1)!} \|f^{(n+1)}\|_\infty (b-a)^{n+1}$
- $|R_n(x)| \leq \frac{1}{(2n+2)!} \|f^{(2n+2)}\|_\infty \prod_{i=0}^{n} (x - x_i)^2$
- $\|R_n\|_\infty \leq \frac{1}{(2n+2)!} \|f^{(2n+2)}\|_\infty (b-a)^{2n+2}$

Corolário (distribuição uniforme de nós)

Dado x_0, se $x_{i+1} = x_i + h$ para $i = 0, \ldots, n-1$ com $h = (b-a)/n$ então

$$\|E_n\|_\infty \leq \frac{h^{n+1}}{4(n+1)} \|f^{(n+1)}\|_\infty .$$

Estimativas de Erro na Interpolação

Exemplo 4

Se a função $f(x) = \cos(x)$ é aproximada por polinômio de grau 9 que interpola f em 10 pontos no intervalo $[0, 1]$, quão grande é o erro de interpolação neste intervalo?

Estimativas de Erro na Interpolação

Exemplo 4

Se a função $f(x) = \cos(x)$ é aproximada por polinômio de grau 9 que interpola f em 10 pontos no intervalo $[0, 1]$, quão grande é o erro de interpolação neste intervalo?

Solução: Vamos calcular um limitante superior para $\|E_9\|_\infty$. Logo,

$$\|f^{(10)}\|_\infty = \max_{t \in [0,1]} |f^{(10)}(t)| = 1 \quad \Longrightarrow \quad \|E_9\|_\infty \leq \frac{1}{10!} < 2.8 \times 10^{-7}.$$

Estimativas de Erro na Interpolação

Exemplo 4

Se a função $f(x) = \cos(x)$ é aproximada por polinômio de grau 9 que interpola f em 10 pontos no intervalo $[0,1]$, quão grande é o erro de interpolação neste intervalo?

Solução: Vamos calcular um limitante superior para $\|E_9\|_\infty$. Logo,

$$\|f^{(10)}\|_\infty = \max_{t\in[0,1]} |f^{(10)}(t)| = 1 \quad \Longrightarrow \quad \|E_9\|_\infty \leq \frac{1}{10!} < 2.8 \times 10^{-7}\,.$$

Será que P_n converge para f quando $n \to \infty$?

Fenômeno de Runge

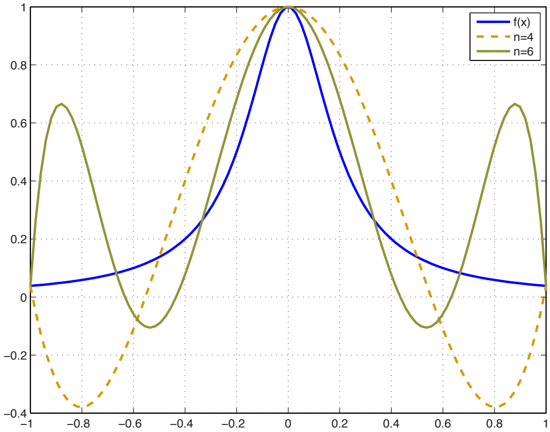
Exemplo 5

Considere a função

$$f(x) = \frac{1}{1+25x^2}, \qquad x \in [-1, 1].$$

Calcule e plote no MATLAB o polinômio de interpolação de f usando a forma de Lagrange com 5, 7 e 16 nós de interpolação igualmente espaçados no intervalo $[-1, 1]$.

Fenômeno de Runge



Fenômeno de Runge

Conclusões:

- Não há garantias que P_n converge para f quando $n \to \infty$;
- Interpolação polinomial de alta ordem é instável em uma distribuição uniforme de nós.

Fenômeno de Runge

Conclusões:

- Não há garantias que P_n converge para f quando $n \to \infty$;
- Interpolação polinomial de alta ordem é instável em uma distribuição uniforme de nós.

Soluções:

- Usar uma distribuição não uniforme de nós que minimize o erro;
- Interpolação polinomial por partes.

Nós de Chebyshev

Como escolher nós que minimizem o limitante do erro?

$$\|E_n(x)\|_\infty \leq \frac{1}{(n+1)!}\|f^{(n+1)}\|_\infty \|\omega_n\|_\infty \qquad \omega_n(x) = \prod_{i=0}^{n}(x - x_i)$$

Nós de Chebyshev

Como escolher nós que minimizem o limitante do erro?

$$\|E_n(x)\|_\infty \leq \frac{1}{(n+1)!} \|f^{(n+1)}\|_\infty \|\omega_n\|_\infty \qquad \omega_n(x) = \prod_{i=0}^{n}(x - x_i)$$

Basta minimizar $\|\omega_n\|_\infty$, isso nos leva aos nós de Chebyshev:

Nós de Chebyshev

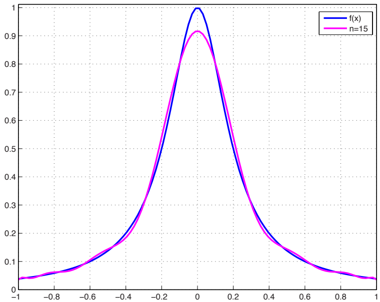
Como escolher nós que minimizem o limitante do erro?

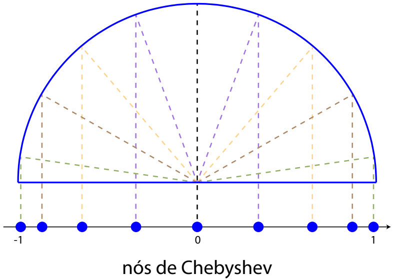
$$\|E_n(x)\|_\infty \leq \frac{1}{(n+1)!}\|f^{(n+1)}\|_\infty \|\omega_n\|_\infty \qquad \omega_n(x) = \prod_{i=0}^{n}(x - x_i)$$

Basta minimizar $\|\omega_n\|_\infty$, isso nos leva aos nós de Chebyshev:

$$\boxed{x_i = \frac{a+b}{2} + \frac{b-a}{2}\cos\left(\frac{2i+1}{2(n+1)}\pi\right), \qquad i = 0, \ldots, n}$$

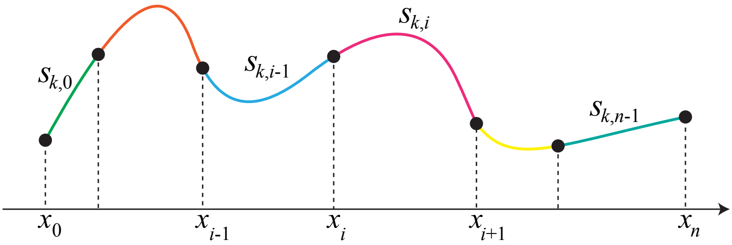
Nós de Chebyshev





$$f(x) = \frac{1}{1+25x^2}, \qquad x \in [-1, 1].$$

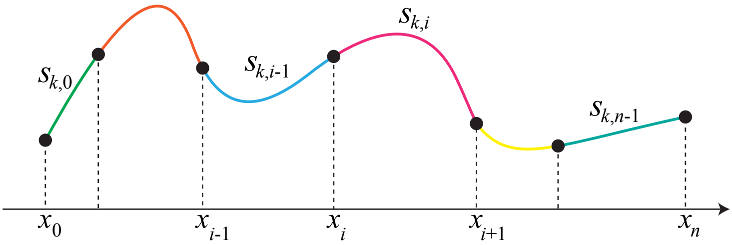
Splines



Dados $(n+1)$ pontos $(x_0, y_0), \ldots, (x_n, y_n)$ com $a = x_0 < \cdots < x_n = b$.

Uma função $S_k(x)$ e chamada de spline de grau k se satisfaz as seguintes condições:

Splines

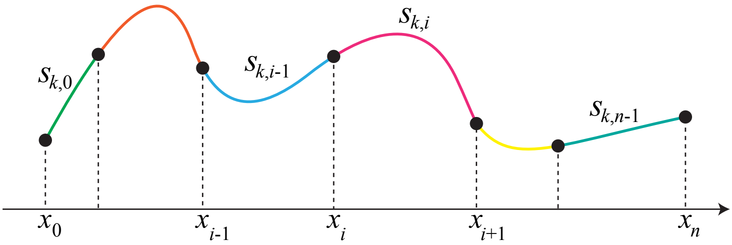


Dados $(n+1)$ pontos $(x_0, y_0), \ldots, (x_n, y_n)$ com $a = x_0 < \cdots < x_n = b$.

Uma função $S_k(x)$ e chamada de spline de grau k se satisfaz as seguintes condições:

- $S_{k,i} = S_k|_{[x_i, x_{i+1}]}$, com $i = 0, \ldots, n-1$, é um polinômio de grau k;

Splines

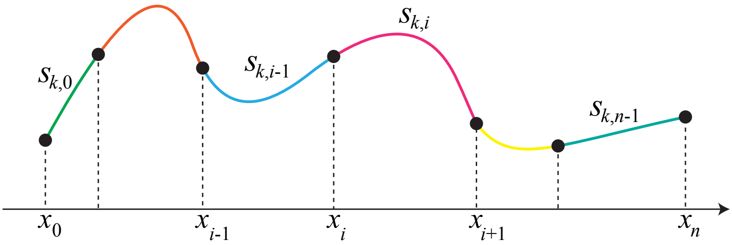


Dados $(n+1)$ pontos $(x_0, y_0), \ldots, (x_n, y_n)$ com $a = x_0 < \cdots < x_n = b$.

Uma função $S_k(x)$ e chamada de spline de grau k se satisfaz as seguintes condições:

- $S_{k,i} = S_k|_{[x_i, x_{i+1}]}$, com $i = 0, \ldots, n-1$, é um polinômio de grau k;
- $S_k \in \mathcal{C}^{k-1}([a, b])$;

Splines

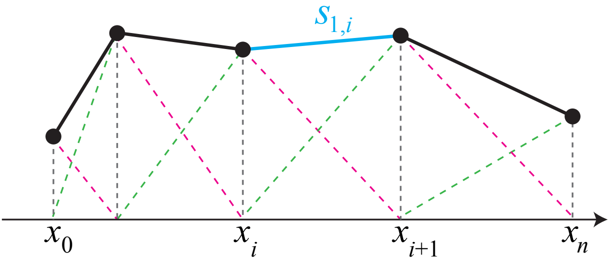


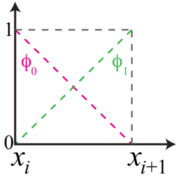
Dados $(n+1)$ pontos $(x_0, y_0), \ldots, (x_n, y_n)$ com $a = x_0 < \cdots < x_n = b$.

Uma função $S_k(x)$ e chamada de spline de grau k se satisfaz as seguintes condições:

- $S_{k,i} = S_k|_{[x_i, x_{i+1}]}$, com $i = 0, \ldots, n-1$, é um polinômio de grau k;
- $S_k \in \mathcal{C}^{k-1}([a, b])$;
- $S_k(x_i) = y_i$, com $i = 0, \ldots, n$.

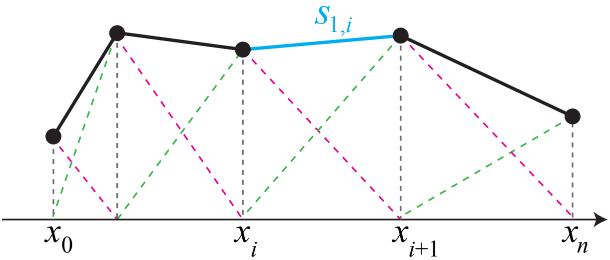
Spline Linear

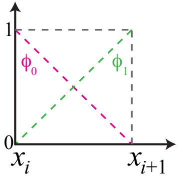




$$S_{1,i}(x) = y_i \underbrace{\frac{x_{i+1} - x}{x_{i+1} - x_i}}_{\phi_0(x)} + y_{i+1} \underbrace{\frac{x - x_i}{x_{i+1} - x_i}}_{\phi_1(x)}, \quad x \in [x_i, x_{i+1}], \quad i = 0, \ldots, n-1.$$

Spline Linear





$$S_{1,i}(x) = y_i \underbrace{\frac{x_{i+1} - x}{x_{i+1} - x_i}}_{\phi_0(x)} + y_{i+1} \underbrace{\frac{x - x_i}{x_{i+1} - x_i}}_{\phi_1(x)}, \quad x \in [x_i, x_{i+1}], \quad i = 0, \dots, n-1.$$

Note que, fazendo $t = \phi_1(x) \Rightarrow (1 - t) = \phi_0(x)$, logo

$$S_{1,i}(t) = (1 - t)\, y_i + t\, y_{i+1}\,.$$

Spline Linear

$$S_{1,i}(x) = y_i \, \frac{x_{i+1} - x}{x_{i+1} - x_i} + y_{i+1} \, \frac{x - x_i}{x_{i+1} - x_i} \, , \quad x \in [x_i, x_{i+1}] \, , \quad i = 0, \ldots, n-1.$$

Considerações:

Spline Linear

$$S_{1,i}(x) = y_i \, \frac{x_{i+1} - x}{x_{i+1} - x_i} + y_{i+1} \, \frac{x - x_i}{x_{i+1} - x_i} \, , \quad x \in [x_i, x_{i+1}] \, , \quad i = 0, \ldots, n-1.$$

Considerações:

- $S_1(x)$ é um polinômio de grau 1 em cada sub-intervalo $[x_i, x_{i+1}]$;

Spline Linear

$$S_{1,i}(x) = y_i \, \frac{x_{i+1} - x}{x_{i+1} - x_i} + y_{i+1} \, \frac{x - x_i}{x_{i+1} - x_i} \,, \quad x \in [x_i, x_{i+1}] \,, \quad i = 0, \ldots, n-1.$$

Considerações:

- $S_1(x)$ é um polinômio de grau 1 em cada sub-intervalo $[x_i, x_{i+1}]$;
- $S_{1,i-1}(x_{i-1}) = y_{i-1}$ e $S_{1,i-1}(x_i) = y_i \,, \quad i = 1, \ldots, n$;

Spline Linear

$$S_{1,i}(x) = y_i \, \frac{x_{i+1} - x}{x_{i+1} - x_i} + y_{i+1} \, \frac{x - x_i}{x_{i+1} - x_i} \,, \quad x \in [x_i, x_{i+1}] \,, \quad i = 0, \ldots, n-1.$$

Considerações:

- $S_1(x)$ é um polinômio de grau 1 em cada sub-intervalo $[x_i, x_{i+1}]$;
- $S_{1,i-1}(x_{i-1}) = y_{i-1}$ e $S_{1,i-1}(x_i) = y_i \,, \quad i = 1, \ldots, n$;
- $S_1 \in \mathcal{C}^0([a, b])$, por construção $S_{1,i}(x_{i+1}) = y_{i+1} = S_{1,i+1}(x_{i+1})$;

Spline Linear

$$S_{1,i}(x) = y_i \, \frac{x_{i+1} - x}{x_{i+1} - x_i} + y_{i+1} \, \frac{x - x_i}{x_{i+1} - x_i} \, , \quad x \in [x_i, x_{i+1}] \, , \quad i = 0, \ldots, n-1.$$

Considerações:

- $S_1(x)$ é um polinômio de grau 1 em cada sub-intervalo $[x_i, x_{i+1}]$;
- $S_{1,i-1}(x_{i-1}) = y_{i-1}$ e $S_{1,i-1}(x_i) = y_i \, , \quad i = 1, \ldots, n;$
- $S_1 \in \mathcal{C}^0([a,b])$, por construção $S_{1,i}(x_{i+1}) = y_{i+1} = S_{1,i+1}(x_{i+1})$;
- O erro é dado por

$$\|f - S_1\|_\infty \leq \frac{h^2}{8} \, \|f''\|_\infty \quad \text{com} \quad h = \max_i \{|x_i - x_{i+1}|\} \, .$$

Spline Linear

Exemplo 6

Dada a função tabelada $y = f(x)$:

x	1	2	4
y	1	3	5

Calcule a spline linear que interpola a função acima.

Spline Linear

Exemplo 6

Dada a função tabelada $y = f(x)$:

x	1	2	4
y	1	3	5

Calcule a spline linear que interpola a função acima.

Solução:

$$S_1(x) = \begin{cases} S_{1,0}(x) = (2-x) + 3(x-1) = 2x - 1 & \text{, se } x \in [1,2] \\ \\ S_{1,1}(x) = 3\frac{4-x}{2} + 5\frac{x-2}{2} = x + 1 & \text{, se } x \in [2,4] \end{cases}$$

Spline Cúbica

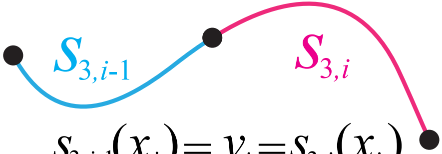
Uma função $S_3(x)$ e uma spline cúbica se satisfaz as seguintes condições:

- $S_{3,i} = S_3|_{[x_i, x_{i+1}]}$, com $i = 0, \ldots, n-1$, é um polinômio de grau 3;
- $S_3(x_i) = y_i$, com $i = 0, \ldots, n$.
- $S_3 \in \mathcal{C}^2([a, b])$;

Spline Cúbica

Uma função $S_3(x)$ e uma spline cúbica se satisfaz as seguintes condições:

- $S_{3,i} = S_3|_{[x_i, x_{i+1}]}$, com $i = 0, \ldots, n-1$, é um polinômio de grau 3;
- $S_3(x_i) = y_i$, com $i = 0, \ldots, n$.
- $S_3 \in \mathcal{C}^2([a, b])$;



$$S_{3,i-1}(x_i) = y_i = S_{3,i}(x_i)$$
$$S'_{3,i-1}(x_i) = S'_{3,i}(x_i)$$
$$S''_{3,i-1}(x_i) = S''_{3,i}(x_i)$$

Spline Cúbica

Existem n polinômios cúbicos $S_{3,i}(x)$ que satisfazem as condições:

Spline Cúbica

Existem n polinômios cúbicos $S_{3,i}(x)$ que satisfazem as condições:

1. $S_{3,i}(x_i) = y_i$ e $S_{3,i}(x_{i+1}) = y_{i+1}$, $\quad i = 0, \ldots, n-1;$

Spline Cúbica

Existem n polinômios cúbicos $S_{3,i}(x)$ que satisfazem as condições:

1. $S_{3,i}(x_i) = y_i$ e $S_{3,i}(x_{i+1}) = y_{i+1}\,, \quad i = 0, \ldots, n-1;$
2. $S_{3,i}(x_{i+1}) = S_{3,i+1}(x_{i+1})\,, \quad i = 0, \ldots, n-2;$

Spline Cúbica

Existem n polinômios cúbicos $S_{3,i}(x)$ que satisfazem as condições:

1. $S_{3,i}(x_i) = y_i$ e $S_{3,i}(x_{i+1}) = y_{i+1}\,, \quad i = 0, \ldots, n-1;$
2. $S_{3,i}(x_{i+1}) = S_{3,i+1}(x_{i+1})\,, \quad i = 0, \ldots, n-2;$
3. $S'_{3,i}(x_{i+1}) = S'_{3,i+1}(x_{i+1})\,, \quad i = 0, \ldots, n-2;$

Spline Cúbica

Existem n polinômios cúbicos $S_{3,i}(x)$ que satisfazem as condições:

1. $S_{3,i}(x_i) = y_i$ e $S_{3,i}(x_{i+1}) = y_{i+1}\,, \quad i = 0, \ldots, n-1;$
2. $S_{3,i}(x_{i+1}) = S_{3,i+1}(x_{i+1})\,, \quad i = 0, \ldots, n-2;$
3. $S'_{3,i}(x_{i+1}) = S'_{3,i+1}(x_{i+1})\,, \quad i = 0, \ldots, n-2;$
4. $S''_{3,i}(x_{i+1}) = S''_{3,i+1}(x_{i+1})\,, \quad i = 0, \ldots, n-2;$

Spline Cúbica

Existem n polinômios cúbicos $S_{3,i}(x)$ que satisfazem as condições:

1. $S_{3,i}(x_i) = y_i$ e $S_{3,i}(x_{i+1}) = y_{i+1}\,, \quad i = 0, \ldots, n-1;$
2. $S_{3,i}(x_{i+1}) = S_{3,i+1}(x_{i+1})\,, \quad i = 0, \ldots, n-2;$
3. $S'_{3,i}(x_{i+1}) = S'_{3,i+1}(x_{i+1})\,, \quad i = 0, \ldots, n-2;$
4. $S''_{3,i}(x_{i+1}) = S''_{3,i+1}(x_{i+1})\,, \quad i = 0, \ldots, n-2;$
5. Condições de contorno:
   - Naturais: $S''_3(x_0) = S''_3(x_n) = 0$
   - Fixadas: $S'_3(x_0) = f'(x_0)$ e $S'_3(x_n) = f'(x_n)$

Spline Cúbica

Existem n polinômios cúbicos $S_{3,i}(x)$ que satisfazem as condições:

1. $S_{3,i}(x_i) = y_i$ e $S_{3,i}(x_{i+1}) = y_{i+1}\,, \quad i = 0, \ldots, n-1;$
2. $S_{3,i}(x_{i+1}) = S_{3,i+1}(x_{i+1})\,, \quad i = 0, \ldots, n-2;$
3. $S'_{3,i}(x_{i+1}) = S'_{3,i+1}(x_{i+1})\,, \quad i = 0, \ldots, n-2;$
4. $S''_{3,i}(x_{i+1}) = S''_{3,i+1}(x_{i+1})\,, \quad i = 0, \ldots, n-2;$
5. Condições de contorno:
   - Naturais: $S''_3(x_0) = S''_3(x_n) = 0$
   - Fixadas: $S'_3(x_0) = f'(x_0)$ e $S'_3(x_n) = f'(x_n)$

Por simplicidade, vamos denotar $S_i(x)$ como

$$S_{3,i}(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i\,, \quad i = 0, \ldots, n-1\,.$$

Spline Cúbica

Existem n polinômios cúbicos $S_{3,i}(x)$ que satisfazem as condições:

1. $S_{3,i}(x_i) = y_i$ e $S_{3,i}(x_{i+1}) = y_{i+1}\,, \quad i = 0,\ldots,n-1;$
2. $S_{3,i}(x_{i+1}) = S_{3,i+1}(x_{i+1})\,, \quad i = 0,\ldots,n-2;$
3. $S'_{3,i}(x_{i+1}) = S'_{3,i+1}(x_{i+1})\,, \quad i = 0,\ldots,n-2;$
4. $S''_{3,i}(x_{i+1}) = S''_{3,i+1}(x_{i+1})\,, \quad i = 0,\ldots,n-2;$
5. Condições de contorno:
   - Naturais: $S''_3(x_0) = S''_3(x_n) = 0$
   - Fixadas: $S'_3(x_0) = f'(x_0)$ e $S'_3(x_n) = f'(x_n)$

Por simplicidade, vamos denotar $S_i(x)$ como

$$S_{3,i}(x) = a_i(x-x_i)^3 + b_i(x-x_i)^2 + c_i(x-x_i) + d_i\,, \quad i = 0,\ldots,n-1\,.$$

Para determinar $S_3(x)$ precisamos determinar cada $S_{3,i}(x)$, isto é:

$$\{a_0, b_0, c_0, d_0, \ldots, a_{n-1}, b_{n-1}, c_{n-1}, d_{n-1}\} \Rightarrow 4n \text{ incógnitas}\,.$$

Spline Cúbica

Aplicando a **condição (1)** obtemos:

$$\boxed{d_i = S_{3,i}(x_i) = y_i}$$

Calculando as derivadas de $S_{3,i}$, temos:

$$S'_{3,i}(x) = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i \quad \text{e} \quad S''_{3,i}(x) = 6a_i(x - x_i) + 2b_i$$

Chamando $z_i = S''_{3,i}(x_i)$, temos que:

$$\boxed{b_i = \frac{z_i}{2}}$$

Spline Cúbica

Usando a **condição (4)** e tomando $h_i = x_{i+1} - x_i$, segue que:

$$2b_{i+1} = 6a_i h_i + 2b_i$$

Isolando a_i e substituindo b_i e b_{i+1}:

$$\boxed{a_i = \frac{b_{i+1} - b_i}{3h_i} = \frac{z_{i+1} - z_i}{6h_i}}$$

Aplicando a **condição (2)** :

$$d_{i+1} = a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i$$

Isolando c_i e substituindo a_i, b_i, d_i e d_{i+1}:

$$\boxed{c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{z_{i+1} h_i}{6} - \frac{z_i h_i}{3}}$$

Spline Cúbica

Finalmente, usando a **condição (3)**:

$$c_{i+1} = 3a_i h_i^2 + 2b_i h_i + c_i$$

Trocando o índice i por $i-1$:

$$c_i = 3a_{i-1} h_{i-1}^2 + 2b_{i-1} h_{i-1} + c_{i-1}$$

Substituindo a_{i-1}, c_{i-1} e c_i :

$$\boxed{z_{i-1} h_{i-1} + z_i (2h_{i-1} + 2h_i) + z_{i+1} h_i = \frac{6(y_{i+1} - y_i)}{h_i} - \frac{6(y_i - y_{i-1})}{h_{i-1}}}$$

Spline Cúbica

Resumindo:
Para $i = 1, \ldots, n-1$, temos as equações:

$$z_{i-1}h_{i-1} + z_i(2h_{i-1} + 2h_i) + z_{i+1}h_i = \frac{6(y_{i+1} - y_i)}{h_i} - \frac{6(y_i - y_{i-1})}{h_{i-1}}$$

Além das duas equações fornecidas pelas condições de contorno.

Portanto, os valores de z_i são obtidos resolvendo um sistema linear de ordem $(n+1)$. Após encontrado os z_i, obtemos cada $S_{3,i}(x)$ através da seguinte substituição:

$$a_i = \frac{z_{i+1} - z_i}{6h_i}, \quad b_i = \frac{z_i}{2}, \quad c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i\, z_{i+1}}{6} - \frac{h_i\, z_i}{3}, \quad d_i = y_i$$

Spline Cúbica Natural

Impondo as condições de contorno naturais $z_0 = z_n = 0$, o sistema tem a seguinte forma:

$$\begin{bmatrix} 1 & & & & & & \\ h_0 & u_1 & h_1 & & & & \\ & h_1 & u_2 & h_2 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & h_{n-3} & u_{n-2} & h_{n-2} & \\ & & & & h_{n-2} & u_{n-1} & h_{n-1} \\ & & & & & & 1 \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ \vdots \\ z_{n-2} \\ z_{n-1} \\ z_n \end{bmatrix} = \begin{bmatrix} 0 \\ v_1 \\ v_2 \\ \vdots \\ v_{n-2} \\ v_{n-1} \\ 0 \end{bmatrix}$$

Spline Cúbica Natural

Melhor, resolvendo o sistema linear de ordem $(n-1)$:

$$\begin{bmatrix} u_1 & h_1 & & & \\ h_1 & u_2 & h_2 & & \\ & \ddots & \ddots & \ddots & \\ & & h_{n-3} & u_{n-2} & h_{n-2} \\ & & & h_{n-2} & u_{n-1} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-2} \\ z_{n-1} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-2} \\ v_{n-1} \end{bmatrix}$$

onde

$$u_i = 2(h_i + h_{i-1})\,, \qquad v_i = w_i - w_{i-1}\,, \qquad w_i = \frac{6}{h_i}(y_{i+1} - y_i)$$

Spline Cúbica Natural

Melhor, resolvendo o sistema linear de ordem $(n-1)$:

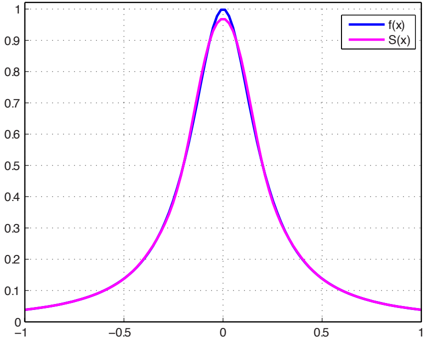
$$\begin{bmatrix} u_1 & h_1 & & & \\ h_1 & u_2 & h_2 & & \\ & \ddots & \ddots & \ddots & \\ & & h_{n-3} & u_{n-2} & h_{n-2} \\ & & & h_{n-2} & u_{n-1} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-2} \\ z_{n-1} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-2} \\ v_{n-1} \end{bmatrix}$$

onde

$$u_i = 2(h_i + h_{i-1})\,, \qquad v_i = w_i - w_{i-1}\,, \qquad w_i = \frac{6}{h_i}(y_{i+1} - y_i)$$

Consideração: as splines cúbicas naturais são obtidas resolvendo um sistema linear $n-1$ ao invés de um sistema de ordem $4n$.

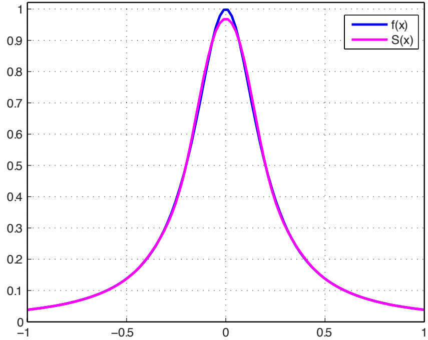
Spline Cúbica Natural



16 nós igualmente espaçados

$$f(x) = \frac{1}{1+25x^2}\,, \qquad x \in [-1,1]\,.$$

Spline Cúbica Natural



16 nós igualmente espaçados

$$f(x) = \frac{1}{1+25x^2}\,, \qquad x \in [-1, 1]\,.$$

O erro é dado por

$$\|f - S_3\|_\infty \leq \frac{5\,h^4}{384}\,\|f^{(4)}\|_\infty \quad \text{com} \quad h = \max_i\{|x_i - x_{i+1}|\}\,.$$

Spline Cúbica Natural

Exemplo 7

Dada a função tabelada $y = f(x)$:

x	0	1	2	3
y	3	1	3	2

Calcule uma aproximação de $f(0.5)$ usando uma spline cúbica natural que interpola a função acima.

Spline Cúbica Natural

Solução: Sabemos que $h_0 = h_1 = h_2 = 1$ e $z_0 = z_3 = 0$. Para determinar z_1 e z_2 temos que resolver o sistema abaixo:

Spline Cúbica Natural

Solução: Sabemos que $h_0 = h_1 = h_2 = 1$ e $z_0 = z_3 = 0$. Para determinar z_1 e z_2 temos que resolver o sistema abaixo:

$$\begin{bmatrix} u_1 & h_1 \\ h_2 & u_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

Spline Cúbica Natural

Solução: Sabemos que $h_0 = h_1 = h_2 = 1$ e $z_0 = z_3 = 0$. Para determinar z_1 e z_2 temos que resolver o sistema abaixo:

$$\begin{bmatrix} u_1 & h_1 \\ h_2 & u_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \Rightarrow \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 24 \\ -18 \end{bmatrix}$$

Spline Cúbica Natural

Solução: Sabemos que $h_0 = h_1 = h_2 = 1$ e $z_0 = z_3 = 0$. Para determinar z_1 e z_2 temos que resolver o sistema abaixo:

$$\begin{bmatrix} u_1 & h_1 \\ h_2 & u_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \Rightarrow \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 24 \\ -18 \end{bmatrix}$$

Cuja a solução é $z_1 = 7.6$ e $z_2 = -6.4$. Logo,

Spline Cúbica Natural

Solução: Sabemos que $h_0 = h_1 = h_2 = 1$ e $z_0 = z_3 = 0$. Para determinar z_1 e z_2 temos que resolver o sistema abaixo:

$$\begin{bmatrix} u_1 & h_1 \\ h_2 & u_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \Rightarrow \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 24 \\ -18 \end{bmatrix}$$

Cuja a solução é $z_1 = 7.6$ e $z_2 = -6.4$. Logo,

$$S_{3,0}(x) = a_0(x - x_0)^3 + b_0(x - x_0)^2 + c_0(x - x_0) + d_0 \quad \text{com}$$

$a_0 = (z_1 - z_0)/(6h_0) = 7.6/6 = 1.2667$
$b_0 = z_0/2 = 0$
$c_0 = -(h_0\, z_1)/6 - (h_0\, z_0)/3 + (y_1 - y_0)/h_0 = -1.2667 + 1 - 3 = -0.7333$
$d_0 = y_0 = 3$

Spline Cúbica Natural

Solução: Sabemos que $h_0 = h_1 = h_2 = 1$ e $z_0 = z_3 = 0$. Para determinar z_1 e z_2 temos que resolver o sistema abaixo:

$$\begin{bmatrix} u_1 & h_1 \\ h_2 & u_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \Rightarrow \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 24 \\ -18 \end{bmatrix}$$

Cuja a solução é $z_1 = 7.6$ e $z_2 = -6.4$. Logo,

$$S_{3,0}(x) = a_0(x - x_0)^3 + b_0(x - x_0)^2 + c_0(x - x_0) + d_0 \quad \text{com}$$

$a_0 = (z_1 - z_0)/(6h_0) = 7.6/6 = 1.2667$
$b_0 = z_0/2 = 0$
$c_0 = -(h_0\, z_1)/6 - (h_0\, z_0)/3 + (y_1 - y_0)/h_0 = -1.2667 + 1 - 3 = -0.7333$
$d_0 = y_0 = 3$

Portanto, $f(0.5) \approx S_{3,0}(0.5) = 2.7917$

Spline Cúbica Fixada

Agora vamos impor as condições de contorno fixadas:

$$S_3'(x_0) = f'(x_0) \quad \text{e} \quad S_3'(x_n) = f'(x_n)$$

Logo, $f'(x_0) = S_{3,0}'(x_0) = c_0 = \frac{y_1 - y_0}{h_0} - \frac{h_0 z_1}{6} - \frac{h_0 z_0}{3}$. Portanto,

$$\boxed{2h_0 z_0 + h_0 z_1 = 6\left[\frac{y_1 - y_0}{h_0} - f'(x_0)\right]}$$

Analogamente, $f'(x_n) = S_{3,n-1}'(x_n) = 3a_{n-1}h_{n-1}^2 + 2b_{n-1}h_{n-1} + c_{n-1}$. Assim,

$$\boxed{h_{n-1}z_{n-1} + 2h_{n-1}z_n = 6\left[f'(x_n) - \frac{y_n - y_{n-1}}{h_{n-1}}\right]}$$

Spline Cúbica Fixada

O sistema linear é dado por:

$$\begin{bmatrix} 2h_0 & h_0 & & & & & \\ h_0 & u_1 & h_1 & & & & \\ & h_1 & u_2 & h_2 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & h_{n-3} & u_{n-2} & h_{n-2} & \\ & & & & h_{n-2} & u_{n-1} & h_{n-1} \\ & & & & & h_{n-1} & 2h_{n-1} \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ \vdots \\ z_{n-2} \\ z_{n-1} \\ z_n \end{bmatrix} = \begin{bmatrix} w_0 - 6f'(x_0) \\ v_1 \\ v_2 \\ \vdots \\ v_{n-2} \\ v_{n-1} \\ 6f'(x_n) - w_{n-1} \end{bmatrix}$$

onde

$$u_i = 2(h_i + h_{i-1})\,, \qquad v_i = w_i - w_{i-1}\,, \qquad w_i = \frac{6}{h_i}(y_{i+1} - y_i)$$

Spline Cúbica Fixada

O sistema linear é dado por:

$$\begin{bmatrix} 2h_0 & h_0 & & & & & \\ h_0 & u_1 & h_1 & & & & \\ & h_1 & u_2 & h_2 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & h_{n-3} & u_{n-2} & h_{n-2} & \\ & & & & h_{n-2} & u_{n-1} & h_{n-1} \\ & & & & & h_{n-1} & 2h_{n-1} \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ \vdots \\ z_{n-2} \\ z_{n-1} \\ z_n \end{bmatrix} = \begin{bmatrix} w_0 - 6f'(x_0) \\ v_1 \\ v_2 \\ \vdots \\ v_{n-2} \\ v_{n-1} \\ 6f'(x_n) - w_{n-1} \end{bmatrix}$$

onde

$$u_i = 2(h_i + h_{i-1})\,, \qquad v_i = w_i - w_{i-1}\,, \qquad w_i = \frac{6}{h_i}(y_{i+1} - y_i)$$

Consideração: as matrizes relacionadas as splines cúbicas natural e fixada são diagonais dominantes, logo são não-singulares. Portanto, $S_3(x)$ existe e é única!

Spline Cúbica Natural

MATLAB

Primeiro vamos resolver o sistema linear $A\mathbf{z} = \mathbf{v}$:

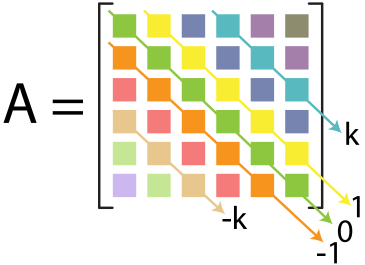
$$A = \begin{bmatrix} u_1 & h_1 & & & \\ h_1 & u_2 & h_2 & & \\ & \ddots & \ddots & \ddots & \\ & & h_{n-3} & u_{n-2} & h_{n-2} \\ & & & h_{n-2} & u_{n-1} \end{bmatrix} \qquad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-2} \\ v_{n-1} \end{bmatrix}$$

onde

$$h_i = x_{i+1} - x_i\,, \quad u_i = 2(h_i + h_{i-1})\,, \quad v_i = w_i - w_{i-1}\,, \quad w_i = \frac{6}{h_i}(y_{i+1} - y_i)$$

Matrizes Diagonais no MATLAB

- Uma matriz $m \times n$ possui $(m + n - 1)$ diagonais.



`D = diag(v,k)`: cria uma matriz diagonal dado um vetor v;
`v = diag(A,k)`: retorna a diagonal da matriz A;
`%` k: índice da diagonal entre $(-m + 1)$ e $(n - 1)$;

Spline Cúbica Natural

MATLAB

Vamos montar a matriz tridiagonal:

$$A = \begin{bmatrix} u_1 & h_1 & & & \\ h_1 & u_2 & h_2 & & \\ & \ddots & \ddots & \ddots & \\ & & h_{n-3} & u_{n-2} & h_{n-2} \\ & & & h_{n-2} & u_{n-1} \end{bmatrix}$$

onde $u_i = 2(h_i + h_{i-1})$

Spline Cúbica Natural

MATLAB

Vamos montar a matriz tridiagonal:

$$A = \begin{bmatrix} u_1 & h_1 & & & \\ h_1 & u_2 & h_2 & & \\ & \ddots & \ddots & \ddots & \\ & & h_{n-3} & u_{n-2} & h_{n-2} \\ & & & h_{n-2} & u_{n-1} \end{bmatrix}$$

onde $u_i = 2(h_i + h_{i-1})$

```
h = xi(2:end) - xi(1:end-1);
u = 2*(h(1:end-1)+h(2:end));
A = diag(h(2:end-1),-1) + diag(u) + diag(h(2:end-1),1);
```

Spline Cúbica Natural

MATLAB

Vamos montar a matriz tridiagonal:

$$A = \begin{bmatrix} u_1 & h_1 & & & \\ h_1 & u_2 & h_2 & & \\ & \ddots & \ddots & \ddots & \\ & & h_{n-3} & u_{n-2} & h_{n-2} \\ & & & h_{n-2} & u_{n-1} \end{bmatrix}$$

onde $u_i = 2(h_i + h_{i-1})$

```
h = xi(2:end) - xi(1:end-1);
u = 2*(h(1:end-1)+h(2:end));
A = diag(h(2:end-1),-1) + diag(u) + diag(h(2:end-1),1);
```

A possui muitos zeros (matriz esparsa) $\Rightarrow$ gasto de memória!

Matrizes Esparsas no MATLAB

`S = sparse(A)`: converte uma matriz cheia para esparsa;

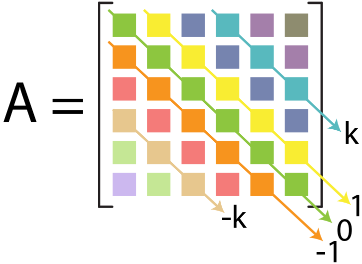
`A = full(S)`: converte uma matriz esparsa para cheia;

`S = sparse(m,n)`: cria uma esparsa $m \times n$;

`S = sparse(i,j,val)`: cria uma esparsa com $S(i(k), j(k)) = val(k)$;

`[i,j,val] = find(S)`: encontra índices e coeficientes não-nulos;

Matrizes Diagonais Esparsas no MATLAB



`A = spdiags(B,d,m,n)`: cria uma matriz diagonal esparsa $m \times n$;
`% B`: matriz cujas colunas são as diagonais de A;
`% d`: índices das diagonais entre $(-m+1)$ e $(n-1)$;

Matrizes Diagonais Esparsas no MATLAB

colunas com zeros extras acima

$$A = \begin{bmatrix} 1 & 6 & 0 & 7 & 0 \\ 8 & 1 & 7 & 0 & 4 \\ 4 & 7 & 1 & 8 & 0 \\ 0 & 3 & 6 & 1 & 9 \\ 0 & 0 & 2 & 5 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 4 & 8 & 1 & 0 & 0 \\ 3 & 7 & 1 & 6 & 0 \\ 2 & 6 & 1 & 7 & 0 \\ 0 & 5 & 1 & 8 & 7 \\ 0 & 0 & 1 & 9 & 4 \end{bmatrix}$$

colunas com zeros extras embaixo

Matrizes Diagonais Esparsas no MATLAB

colunas com zeros extras acima

$$A = \begin{bmatrix} 1 & 6 & 0 & 7 & 0 \\ 8 & 1 & 7 & 0 & 4 \\ 4 & 7 & 1 & 8 & 0 \\ 0 & 3 & 6 & 1 & 9 \\ 0 & 0 & 2 & 5 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 4 & 8 & 1 & 0 & 0 \\ 3 & 7 & 1 & 6 & 0 \\ 2 & 6 & 1 & 7 & 0 \\ 0 & 5 & 1 & 8 & 7 \\ 0 & 0 & 1 & 9 & 4 \end{bmatrix}$$

colunas com zeros extras embaixo

```
n = length(xi);
B = [ [h(2:end-1); 0] u [0; h(2:end-1)] ];
A = spdiags(B,-1:1,n-2,n-2);
```

Spline Cúbica Natural

MATLAB

Determinado o vetor solução z=A\v, finalmente vem a montagem das splines cúbicas:

$$S_{3,i}(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \qquad \text{com}$$

$$a_i = \frac{z_{i+1} - z_i}{6h_i}, \quad b_i = \frac{z_i}{2}, \quad c_i = -\frac{h_i\, z_{i+1}}{6} - \frac{h_i\, z_i}{3} + \frac{y_{i+1} - y_i}{h_i}, \quad d_i = y_i$$

Spline Cúbica Natural

MATLAB

Determinado o vetor solução z=A\v, finalmente vem a montagem das splines cúbicas:

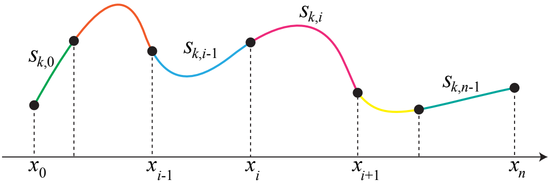
$$S_{3,i}(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \qquad \text{com}$$

$$a_i = \frac{z_{i+1} - z_i}{6h_i}, \quad b_i = \frac{z_i}{2}, \quad c_i = -\frac{h_i\, z_{i+1}}{6} - \frac{h_i\, z_i}{3} + \frac{y_{i+1} - y_i}{h_i}, \quad d_i = y_i$$

```
a = (z(2:end)-z(1:end-1))./(6*h);
b = z(1:end-1)/2;
c = -(h/6).*(z(2:end)+2*z(1:end-1)) + (yi(2:end)-yi(1:end-1))./h;
d = yi(1:end-1);
```

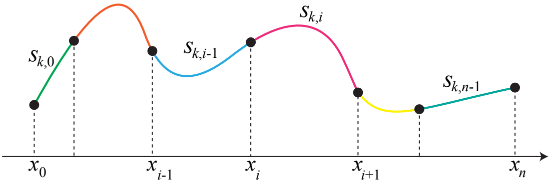
Polinômio por Partes no MATLAB

Relembrando que um polinômio $P_n(x) = a_n x^n + \cdots + a_2 x^2 + a_1 x + a_0$ é representado no MATLAB por um vetor da forma $p = [a_n, \ldots, a_2, a_1, a_0]$



Polinômio por Partes no MATLAB

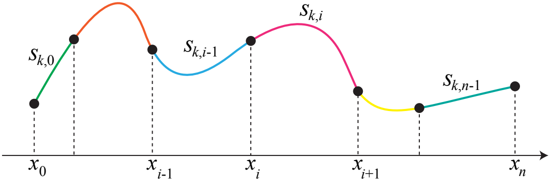
Relembrando que um polinômio $P_n(x) = a_n x^n + \cdots + a_2 x^2 + a_1 x + a_0$ é representado no MATLAB por um vetor da forma $p = [a_n, \ldots, a_2, a_1, a_0]$



`pp = mkpp(xi,coefs)`: cria polinômio por partes;
`% xi`: nós do polinômio;
`% coefs`: matriz dos coeficientes do polinômio;

Polinômio por Partes no MATLAB

Relembrando que um polinômio $P_n(x) = a_n x^n + \cdots + a_2 x^2 + a_1 x + a_0$ é representado no MATLAB por um vetor da forma $p = [a_n, \ldots, a_2, a_1, a_0]$



`pp = mkpp(xi,coefs)`: cria polinômio por partes;
`% xi`: nós do polinômio;
`% coefs`: matriz dos coeficientes do polinômio;

`s = ppval(pp,x)`: avalia um polinômio por partes;
`% pp`: polinômio por partes;
`% x`: pontos a serem avaliados;

Spline Cúbica Natural

MATLAB

```
function s = cubic_spline(xi,yi,x)

[n,m]= size (xi);
if (n == 1) xi = xi'; yi = yi'; n = m; end

h = xi(2:end) - xi(1:end-1);
u =  2*(h(1:end-1)+h(2:end));
A = spdiags([ [h(2:end-1);0] u [0;h(2:end-1)]],-1:1,n-2,n-2);
w = 6*(yi(2:end)-yi(1:end-1))./h;
v = w(2:end)-w(1:end-1);
z = A\v; z = [0;z;0];

a = (z(2:end)-z(1:end-1))./(6*h);
b = z(1:end-1)/2;
c =-(h/6).*(z(2:end)+2*z(1:end-1))+(yi(2:end)-yi(1:end-1))./h;
d = yi(1:end-1);

pp = mkpp (xi,[a b c d]);
s = ppval(pp ,x);
```

Splines no MATLAB

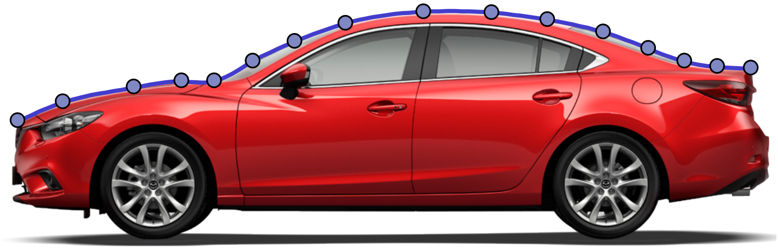
O MATLAB implementa a sua própria spline cúbica através da função:

`s = spline(xi,yi,x)`: avalia uma spline cúbica;
`% xi,yi`: pontos de interpolação;
`% x`: pontos a serem avaliados;

Spline Cúbica Natural

Aplicação

CAD em Engenharia



x_i	18.5	73.5	160	218	258	305	356	418	513	596	664	732	787	831	871	912
y_i	157.5	108.5	198.5	206	206	230	254	276	290	289	280	265	245.5	230	223.5	221.5