Interpolação de Polinômios

INF1608 – Análise Numérica

Waldemar Celes
celes@inf.puc-rio.br

Departamento de Informática, PUC-Rio

Interpolação de Polinômios

Motivação

- Funções polinomiais são avaliadas de forma eficiente
  - Fatoração natural

Interpolação de Polinômios

Motivação

- Funções polinomiais são avaliadas de forma eficiente
  - Fatoração natural
- **Compressão**
  - Aproximar conjuntos de amostras por funções polinomiais
- **Desempenho**
  - Aproximar funções complexas por funções polinomiais

Interpolação de Polinômios

Definição

- $y = f(x)$ interpola um conjunto de pontos (x_i, y_i) se, e somente se, $y_i = f(x_i) \quad \forall \quad (x_i, y_i)$.

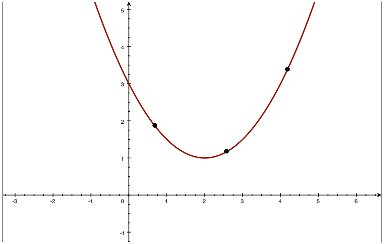
Interpolação de Polinômios

Definição

- $y = f(x)$ interpola um conjunto de pontos (x_i, y_i) se, e somente se, $y_i = f(x_i) \quad \forall \quad (x_i, y_i)$.

Exemplo

- Existe uma parábola que interpola 3 pontos não colineares



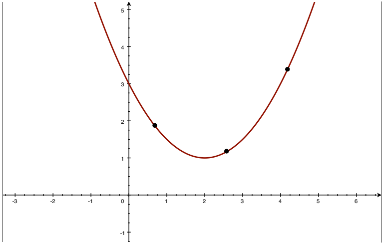
Interpolação de Polinômios

Definição

- $y = f(x)$ interpola um conjunto de pontos (x_i, y_i) se, e somente se, $y_i = f(x_i) \quad \forall \quad (x_i, y_i)$.

Exemplo

- Existe uma parábola que interpola 3 pontos não colineares



Como $f(x)$ é uma função, é necessário que x_i sejam distintos.
Se forem, o polinômio $y = P(x)$ **sempre existe**

Interpolação de Polinômios

Interpolação de Lagrange

- Dados n pontos $(x_0, y_0)...(x_{n-1}, y_{n-1})$, existe um polinômio interpolante de grau $n-1$ dado pela fórmula de Lagrange:

$$P_{n-1}(x) = y_0 L_0(x) + y_1 L_1(x) + ... + y_{n-1} L_{n-1}(x)$$

  - onde $L_k(x)$ é dado por:

$$L_k(x) = \frac{(x - x_0)...(x - x_{k-1})(x - x_{k+1})...(x - x_{n-1})}{(x_k - x_0)...(x_k - x_{k-1})(x_k - x_{k+1})...(x_k - x_{n-1})}$$

- Resulta em um polinômio de grau no máximo igual a $n-1$

Interpolação de Lagrange

Exemplo: Dados 3 pontos

$$
\begin{aligned}
P_2(x) = & y_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + \\
& y_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + \\
& y_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}
\end{aligned}
$$

Interpolação de Lagrange

Exemplo: Dados 3 pontos

$$
\begin{aligned}
P_2(x) = & y_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + \\
& y_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + \\
& y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}
\end{aligned}
$$

- Note que:

$$
\begin{aligned}
P_2(x_0) &= y_0 \\
P_2(x_1) &= y_1 \\
P_2(x_2) &= y_2
\end{aligned}
$$

Interpolação de Polinômios

Pergunta

- Dado um conjunto qualquer com n pontos com x_i distintos, o polinômio de Lagrande é **o único polinômio interpolante**?

Interpolação de Polinômios

Pergunta

- Dado um conjunto qualquer com n pontos com x_i distintos, o polinômio de Lagrande é **o único polinômio interpolante**?

Prova de unicidade

- Suponha que existe $P(x)$ e outro polinômio interpolante $Q(x)$.
- Se definirmos $H(x) = P(x) - Q(x)$
  - O grau de $H(x)$ é no máximo $n - 1$

Interpolação de Polinômios

Pergunta

- Dado um conjunto qualquer com n pontos com x_i distintos, o polinômio de Lagrande é **o único polinômio interpolante**?

Prova de unicidade

- Suponha que existe $P(x)$ e outro polinômio interpolante $Q(x)$.
- Se definirmos $H(x) = P(x) - Q(x)$
  - O grau de $H(x)$ é no máximo $n - 1$

No entanto, como H se anula em todos os pontos:

$H(x_0) = 0,\ H(x_1) = 0,\ \dots\ H(x_{n-1}) = 0$

$\Rightarrow H(x)$ tem n raízes distintas

Logo: $H(x)$ é identicamente zero e $P(x) \equiv Q(x)$

Interpolação de Polinômios

- Interpolador de Lagrange gera um polinômio cuja avaliação computacional é cara, pois não favorece a fatoração

---

Interpolação de Polinômios

- Interpolador de Lagrange gera um polinômio cuja avaliação computacional é cara, pois não favorece a fatoração

---

Interpolação por Diferenças Divididas de Newton

- Interpolação linear

$$\begin{aligned} f_1(x) &= f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) \\ &= b_0 + b_1(x - x_0) \end{aligned}$$

Interpolação por Diferenças Divididas de Newton

- Interpolação quadrática

$$f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)$$

  - Pode-se mostrar que:

$$b_0 = f(x_0)$$
$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$
$$b_2 = \frac{\frac{f(x_2)-f(x_1)}{x_2-x_1} - \frac{f(x_1)-f(x_0)}{x_1-x_0}}{x_2 - x_0}$$

Interpolação por Diferenças Divididas de Newton

Fórmula geral dos Polinômios por Diferenças Divididas de Newton

- Considerando $n+1$ pontos

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

onde:

$$b_0 = f[x_0]$$

$$b_1 = f[x_0\ x_1]$$

$$b_2 = f[x_0\ x_1\ x_2]$$

$$\cdots$$

$$b_n = f[x_0 \ldots x_n]$$

- $f[x_i \ldots x_j]$: diferenças divididas de Newton de x_i a x_j
- Note que o polinômio gerado favorece fatoração

Interpolação por Diferenças Divididas de Newton

Diferenças divididas

- Ordem 0:

$$f[x_k] = f(x_k)$$

- Ordem 1:

$$f[x_k \; x_{k+1}] = \frac{f[x_{k+1} - f[x_k]}{x_{k+1} - x_k}$$

- Ordem 2:

$$f[x_k \; x_{k+1} \; x_{k+2}] = \frac{f[x_{k+1} \; x_{k+2}] - f[x_{k+1} \; x_k]}{x_{k+2} - x_k}$$

Interpolação por Diferenças Divididas de Newton

Diferenças divididas

- Ordem 0:

$$f[x_k] = f(x_k)$$

- Ordem 1:

$$f[x_k\ x_{k+1}] = \frac{f[x_{k+1} - f[x_k]}{x_{k+1} - x_k}$$

- Ordem 2:

$$f[x_k\ x_{k+1}\ x_{k+2}] = \frac{f[x_{k+1}\ x_{k+2}] - f[x_{k+1}\ x_k]}{x_{k+2} - x_k}$$

- Ordem n:

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

Interpolação por Diferenças Divididas de Newton

Algoritmo: Diferenças Divididas de Newton

Determinar $\mathbf{f[x_i \dots x_j]}$, com $i \leq j$:

if $i = j$

  return $f(x_i)$

else

  return $\dfrac{f[x_{i+1} \dots x_j] - f[x_i \dots x_{j-1}]}{x_j - x_i}$

Interpolação por Diferenças Divididas de Newton

Algoritmo: Diferenças Divididas de Newton

Determinar $\mathbf{f}[\mathbf{x_i} \ldots \mathbf{x_j}]$, com $i \leq j$:

if $i = j$

return $f(x_i)$

else

return $\dfrac{f[x_{i+1} \ldots x_j] - f[x_i \ldots x_{j-1}]}{x_j - x_i}$

- Algoritmo ineficiente
  - Múltiplas avaliações das mesmas diferenças divididas

Interpolação por Diferenças Divididas de Newton

Estrutura recursiva na avaliação das diferenças divididas

$$\begin{array}{ccccccccc}
f[x_0] & \rightarrow & f[x_0\ x_1] & \rightarrow & f[x_0 \ldots x_2] & \rightarrow & f[x_0 \ldots x_3] & \rightarrow & f[x_0 \ldots x_4] \\
 & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
 & & f[x_1] & \rightarrow & f[x_1\ x_2] & \rightarrow & f[x_1 \ldots x_3] & \rightarrow & f[x_1 \ldots x_4] \\
 & & & & \uparrow & & \uparrow & & \uparrow \\
 & & & & f[x_2] & \rightarrow & f[x_2\ x_3] & \rightarrow & f[x_2 \ldots x_4] \\
 & & & & & & \uparrow & & \uparrow \\
 & & & & & & f[x_3] & \rightarrow & f[x_3\ x_4] \\
 & & & & & & & & \uparrow \\
 & & & & & & & & f[x_4]
\end{array}$$

Interpolação por Diferenças Divididas de Newton

Estrutura recursiva na avaliação das diferenças divididas

$$
\begin{array}{ccccccccc}
f[x_0] & \rightarrow & f[x_0\ x_1] & \rightarrow & f[x_0 \ldots x_2] & \rightarrow & f[x_0 \ldots x_3] & \rightarrow & f[x_0 \ldots x_4] \\
 & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
 & & f[x_1] & \rightarrow & f[x_1\ x_2] & \rightarrow & f[x_1 \ldots x_3] & \rightarrow & f[x_1 \ldots x_4] \\
 & & & & \uparrow & & \uparrow & & \uparrow \\
 & & & & f[x_2] & \rightarrow & f[x_2\ x_3] & \rightarrow & f[x_2 \ldots x_4] \\
 & & & & & & \uparrow & & \uparrow \\
 & & & & & & f[x_3] & \rightarrow & f[x_3\ x_4] \\
 & & & & & & & & \uparrow \\
 & & & & & & & & f[x_4]
\end{array}
$$

Solução eficiente

- Implementação **bottom-up**, ou
- Implementação com **cache**

Interpolação por Diferenças Divididas de Newton

Estrutura recursiva na avaliação das diferenças divididas

$$
\begin{array}{ccccccccc}
f[x_0] & \rightarrow & f[x_0\ x_1] & \rightarrow & f[x_0 \ldots x_2] & \rightarrow & f[x_0 \ldots x_3] & \rightarrow & f[x_0 \ldots x_4] \\
 & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
 & & f[x_1] & \rightarrow & f[x_1\ x_2] & \rightarrow & f[x_1 \ldots x_3] & \rightarrow & f[x_1 \ldots x_4] \\
 & & & & \uparrow & & \uparrow & & \uparrow \\
 & & & & f[x_2] & \rightarrow & f[x_2\ x_3] & \rightarrow & f[x_2 \ldots x_4] \\
 & & & & & & \uparrow & & \uparrow \\
 & & & & & & f[x_3] & \rightarrow & f[x_3\ x_4] \\
 & & & & & & & & \uparrow \\
 & & & & & & & & f[x_4]
\end{array}
$$

Solução eficiente

- Implementação **bottom-up**, ou
- Implementação com **cache**

} Exige armazenamento de ordem quadrática

Interpolação de Polinômios

Como obter polinômio na forma convencional?

$$f(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + \ldots + c_{n-1} x^{n-1}$$

Interpolação de Polinômios

Como obter polinômio na forma convencional?

$$f(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + \ldots + c_{n-1} x^{n-1}$$

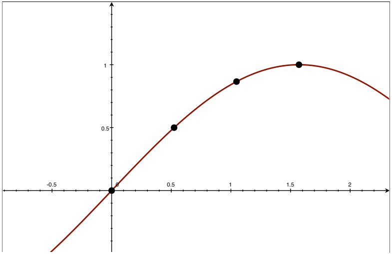
- Não existe fórmula para determinar o polinômio nesta forma
- Solução seria resolver um sistema linear $n \times n$
  - Ineficiente computacionalmente
  - Instável numericamente
    - Sistemas tendem a ser mal condicionados para n grandes
    - Perde-se o controle do erro na determinação dos coeficientes

Interpolação de Polinômios

Exemplo

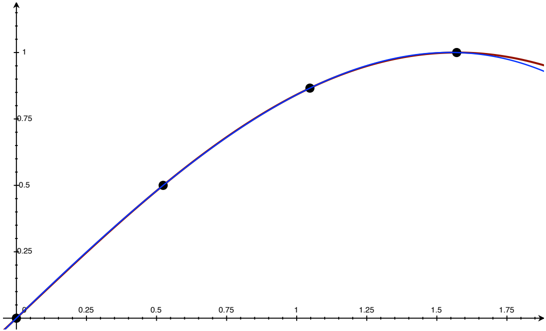
- Aproximar $f(x) = \sin x$ usando
  4 amostras igualmente espaçadas entre 0 e $\frac{\pi}{2}$

x_i	$y_i = \sin x_i$
0	0
$(\pi)/6$	0.5
$(\pi)/3$	0.866
$(\pi)/2$	1



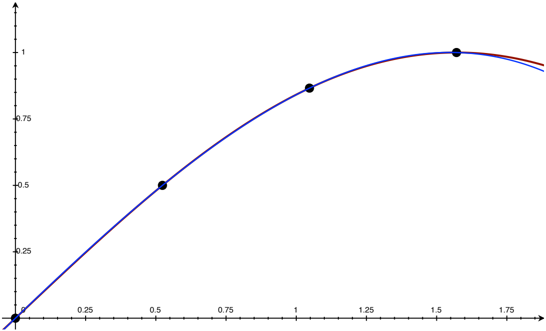
Interpolação de Polinômios

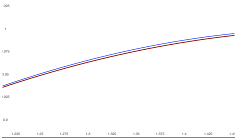
Exemplo: Interpolação de $f(x) = \sin x$



Interpolação de Polinômios

Exemplo: Interpolação de $f(x) = \sin x$





- Qual o erro da interpolação?

Interpolação de Polinômios

Erro da Interpolação

- Considere a interpolação da função $f(x)$ pelo polinômio $P(x)$

Interpolação de Polinômios

Erro da Interpolação

- Considere a interpolação da função $f(x)$ pelo polinômio $P(x)$
- **Teorema**

$$f(x) - P(x) = \frac{(x - x_0)(x - x_1)...(x - x_{n-1})}{n!} f^{[n]}(c)$$

  - onde:

$$c \in [\min(x, x_0), \max(x, x_{n-1})]$$

Interpolação de Polinômios

Erro da Interpolação

- Considere a interpolação da função $f(x)$ pelo polinômio $P(x)$
- **Teorema**

$$f(x) - P(x) = \frac{(x - x_0)(x - x_1)...(x - x_{n-1})}{n!} f^{[n]}(c)$$

  - onde:

$$c \in [\min(x, x_0), \max(x, x_{n-1})]$$

- Pela fórmula, concluímos que erros no meio do intervalo tendem a ser menores
  - Termos do produtório tendem a ser menores

Erro da Interpolação

No exemplo:

$$\sin x - P(x) = \frac{(x-0)(x-\frac{\pi}{6})(x-\frac{\pi}{3})(x-\frac{\pi}{2})}{4!}\ f^{[4]}(c)$$

Erro da Interpolação

No exemplo:

$$\sin x - P(x) = \frac{(x-0)(x-\frac{\pi}{6})(x-\frac{\pi}{3})(x-\frac{\pi}{2})}{4!}\, f^{[4]}(c)$$

- Como $f^{[4]}(x) = \sin x$, o máximo em $[0, \frac{\pi}{2}]$ é 1

$$|\sin - P(x)| \leq \frac{\left|(x-0)(x-\frac{\pi}{6})(x-\frac{\pi}{3})(x-\frac{\pi}{2})\right|}{24}$$

Erro da Interpolação

No exemplo:

$$\sin x - P(x) = \frac{(x-0)(x-\frac{\pi}{6})(x-\frac{\pi}{3})(x-\frac{\pi}{2})}{4!}\ f^{[4]}(c)$$

- Como $f^{[4]}(x) = \sin x$, o máximo em $[0, \frac{\pi}{2}]$ é 1

$$|\sin - P(x)| \leq \frac{\left|(x-0)(x-\frac{\pi}{6})(x-\frac{\pi}{3})(x-\frac{\pi}{2})\right|}{24}$$

  - Alguns valores

$$|\sin(1.0) - P(1.0)| \leq 0.0005348$$
$$|\sin(0.2) - P(0.2)| \leq 0.00313$$

Erro da Interpolação

Como reduzir o erro da interpolação?

Erro da Interpolação

Como reduzir o erro da interpolação?

Espaçar as amostras igualmente é o mais adequado?

Erro da Interpolação

Como reduzir o erro da interpolação?

Espaçar as amostras igualmente é o mais adequado?

Intuitivamente:

- Diminuir a taxa de amostragem no meio do intervalo
- Afastar as amostras dos extremos do intervalo

Espaçamento de Amostras

Teorema de Chebyshev: minimização do erro da interpolação

$$\frac{(x - x_0)(x - x_1)...(x - x_{n-1})}{n!} f^{[n]}(c)$$

- Considerando, inicialmente, $x \in [-1, 1]$
- Objetivo: minimizar polinômio numerador

$$N(x) = (x - x_0)(x - x_1)...(x - x_{n-1})$$

Raízes de Chebyshev considerando 9 amostras

$$\cos\frac{\pi}{18}, \cos\frac{3\pi}{18}, ..., \cos\frac{17\pi}{18}$$

Espaçamento de Amostras

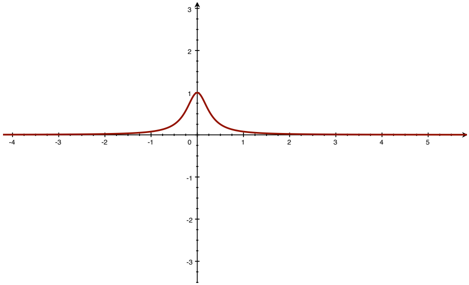
Amostras de Chebyshev

Exemplo

- Interpolação com 27 amostras, com $x \in [-2, 2]$ da função:

$$f(x) = \frac{1}{1 + 12x^2}$$

- Função original



Espaçamento de Amostras

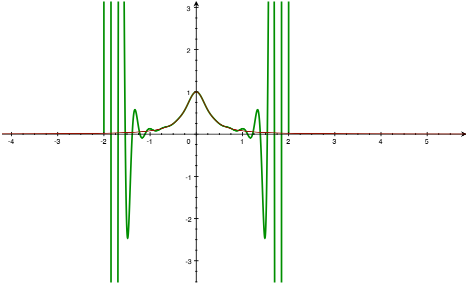
Amostras de Chebyshev

Exemplo

- Interpolação com 27 amostras, com $x \in [-2, 2]$ da função:

$$f(x) = \frac{1}{1 + 12x^2}$$

- Função original
- Amostras regulares



Espaçamento de Amostras

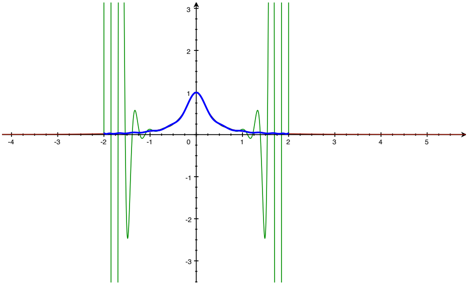
Amostras de Chebyshev

Exemplo

- Interpolação com 27 amostras, com $x \in [-2, 2]$ da função:

$$f(x) = \frac{1}{1 + 12x^2}$$

- Função original
- Amostras regulares
- Amostras de Chebbyshev

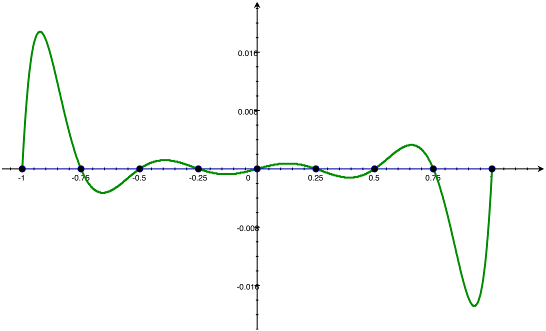


Amostras de Chebyshev

Plotagem do polinômio numerador com 9 amostras

$$N(x) = (x - x_0)(x - x_1)...(x - x_{n-1})$$

- Regulares

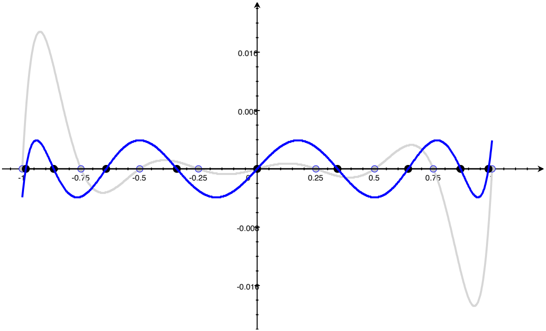


Amostras de Chebyshev

Plotagem do polinômio numerador com 9 amostras

$$N(x) = (x - x_0)(x - x_1)...(x - x_{n-1})$$

- Regulares
- Chebyshev



Teorema de Chebyshev

Fórmula geral para $x \in [-1, 1]$

$$x_i = \cos \frac{\beta\pi}{2n}, \text{ onde } \beta = 1, 3, ..., 2n-1$$

- Valor máximo do polinômio numerador

$$|N(x)|_{max} = \frac{1}{2^{n-1}}$$

Teorema de Chebyshev

Mudança de intervalo:

$$[-1, 1] \quad \longrightarrow \quad [a, b]$$

$$\cos\frac{\beta\pi}{2n} \quad \longrightarrow \quad \underbrace{\underbrace{\frac{b-a}{2}\cos\frac{\beta\pi}{2n}}_{\left[-\frac{b-a}{2},+\frac{b-a}{2}\right]} + \frac{a+b}{2}}_{\left[\frac{a+b}{2}-\frac{b-a}{2},\frac{a+b}{2}+\frac{b-a}{2}\right]}$$

$$[a, b]$$

Teorema de Chebyshev

Fórmula geral para $x \in [a, b]$

$$x_i = \frac{b-a}{2}\cos\frac{\beta\pi}{2n} + \frac{a+b}{2}, \text{ onde } \beta = 1, 3, ..., 2n-1$$

- Valor máximo do polinômio numerador

$$|N(x)|_{max} = \frac{\left(\frac{b-a}{2}\right)^n}{2^{n-1}}$$

Teorema de Chebyshev

Fórmula geral para $x \in [a, b]$

$$x_i = \frac{b-a}{2} \cos \frac{\beta\pi}{2n} + \frac{a+b}{2}, \text{ onde } \beta = 1, 3, ..., 2n-1$$

- Valor máximo do polinômio numerador

$$|N(x)|_{max} = \frac{\left(\frac{b-a}{2}\right)^n}{2^{n-1}}$$

Exemplo

- Erro máximo na aproximação de $\sin x$ com $x \in [0, \frac{\pi}{2}]$, considerando 4 amostras:

$$|\sin x - P_3(x)|_{max} \leq \frac{\left(\frac{\frac{\pi}{2} - 0}{2}\right)^4}{2^3 \;\; 4!} \; 1$$

Interpolação de Polinômios

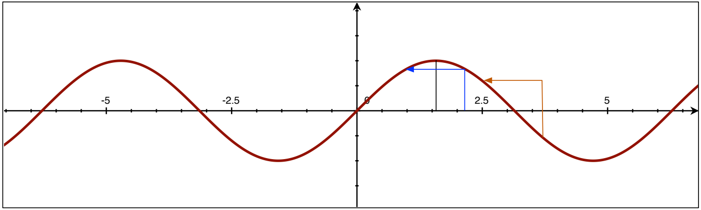
Exercício: Interpolar a função $\sin x$ com 10 dígitos de precisão

Interpolação de Polinômios

Exercício: Interpolar a função $\sin x$ com 10 dígitos de precisão

- Sabe-se que interpolar o intervalo $[0, \frac{\pi}{2}]$ é suficiente
  - Mapeamento da abscissa

$$
\begin{aligned}
x \in [\frac{\pi}{2}, \pi] &\longrightarrow \sin(x) = \sin(\pi - x) \\
x \in [\pi, 2\pi] &\longrightarrow \sin(x) = -\sin(2\pi - x) \\
x > 2\pi &\longrightarrow \sin(x) = \sin(x \bmod 2\pi) \\
x < 0 &\longrightarrow \sin(x) = -\sin(-x)
\end{aligned}
$$



Interpolação de Polinômios

Exercício: Quantas amostras vamos precisar

- Usando Chebyshev

$$|\sin x - P_{n-1}(x)| \leq \frac{\left(\frac{\frac{\pi}{2}-0}{2}\right)^n}{2^{n-1} \quad n!} 1$$

Interpolação de Polinômios

Exercício: Quantas amostras vamos precisar

- Usando Chebyshev

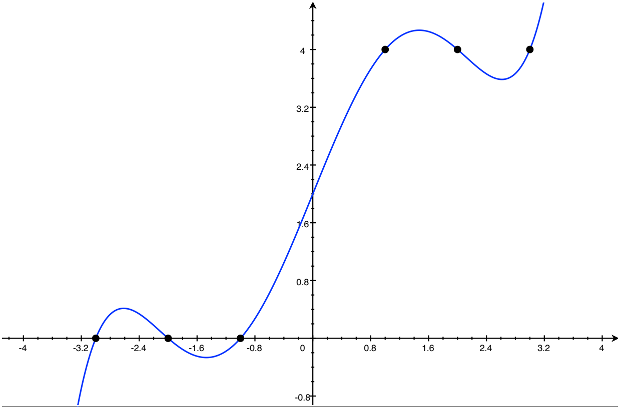
$$|\sin x - P_{n-1}(x)| \leq \frac{\left(\frac{\frac{\pi}{2}-0}{2}\right)^n}{2^{n-1}\quad n!}\ 1$$

  - Por tentativa:

$$n = 9 \longrightarrow \approx 0.1224 \times 10^{-8}$$
$$n = 10 \longrightarrow \approx 0.4807 \times 10^{-10}$$

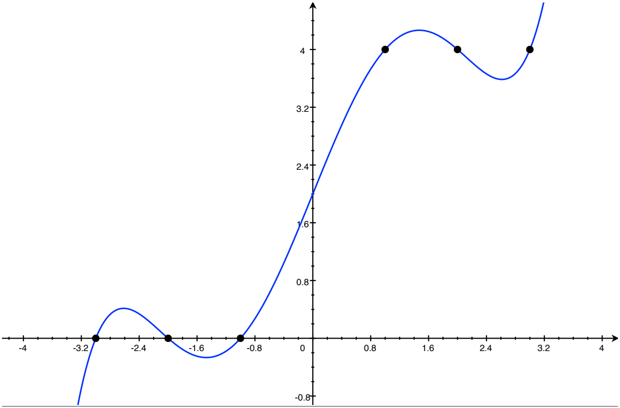
Interpolação por Partes

O polinômio interpolante "oscila" quando há discontinuidade



Interpolação por Partes

O polinômio interpolante "oscila" quando há discontinuidade



Alternativa: interpolação por **splines** (por partes)

Interpolação por Partes

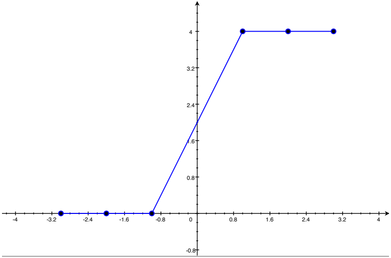
Splines lineares

- n pontos, n partes (segmentos de reta)

$$f(x) = \begin{cases} f(x_0) + m_0(x - x_0) & x_0 \leq x \leq x_1 \\ f(x_1) + m_1(x - x_1) & x_1 < x \leq x_1 \\ \dots & \\ f(x_{n-2}) + m_{n-2}(x - x_{n-2}) & x_{n-2} < x \leq x_n \end{cases}$$

- com:

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



Interpolação por Partes

Continuidade entre partes

- Continuidade $\mathbf{C^0}$: continuidade de **posição**
- Continuidade $\mathbf{C^1}$: continuidade de **tangente**
- Continuidade $\mathbf{C^2}$: continuidade de **curvatura**

Interpolação por Partes

Continuidade entre partes

- Continuidade $\mathbf{C^0}$: continuidade de **posição**
- Continuidade $\mathbf{C^1}$: continuidade de **tangente**
- Continuidade $\mathbf{C^2}$: continuidade de **curvatura**

Continuidade entre partes de splines

- Spline **linear**: $\mathbf{C^0}$
- Spline **quadrática**: $\mathbf{C^0}$ e $\mathbf{C^1}$
- Spline **cúbica**: $\mathbf{C^0}$, $\mathbf{C^1}$ e $\mathbf{C^2}$

Interpolação por Partes

Splines cúbica

$$s_0(x) = y_0 + b_0(x - x_0) + c_0(x - x_0)^2 + d_0(x - x_0)^3, \quad x \in [x_0, x_1]$$

$$s_1(x) = y_1 + b_1(x - x_1) + c_1(x - x_1)^2 + d_1(x - x_1)^3, \quad x \in [x_1, x_2]$$

$$\ldots$$

$$s_{n-2}(x) = y_{n-2} + b_{n-2}(x - x_{n-2}) + c_{n-2}(x - x_{n-2})^2 + d_{n-2}(x - x_{n-2})^3, \quad x \in [x_{n-2}, x_{n-1}]$$

Interpolação por Partes

Splines cúbica

$$s_0(x) = y_0 + b_0(x - x_0) + c_0(x - x_0)^2 + d_0(x - x_0)^3, \quad x \in [x_0, x_1]$$
$$s_1(x) = y_1 + b_1(x - x_1) + c_1(x - x_1)^2 + d_1(x - x_1)^3, \quad x \in [x_1, x_2]$$
$$\ldots$$
$$s_{n-2}(x) = y_{n-2} + b_{n-2}(x - x_{n-2}) + c_{n-2}(x - x_{n-2})^2 + d_{n-2}(x - x_{n-2})^3, \quad x \in [x_{n-2}, x_{n-1}]$$

Achar spline interpolante

- Determinar b_i, c_i e $d_i \longrightarrow 3(n-1)$ incógnitas

Splines Cúbicas

Spline com n pontos, $n - 1$ partes

- Determinar $3(n-1) = 3n - 3$ incógnitas

Splines Cúbicas

Spline com n pontos, $n-1$ partes

- Determinar $3(n-1) = 3n-3$ incógnitas

Equações das propriedades

- Continuidade C^0
  - $s_i(x_i) = y_i$: já satisfeitas pelas expressões de $s_i(x)$
  - $s_i(x_{i+1}) = y_{i+1}, i = 0, ..., n-2$: $n-1$ equações

Splines Cúbicas

Spline com n pontos, $n-1$ partes

- Determinar $3(n-1) = 3n-3$ incógnitas

Equações das propriedades

- Continuidade C^0
  - $s_i(x_i) = y_i$: já satisfeitas pelas expressões de $s_i(x)$
  - $s_i(x_{i+1}) = y_{i+1}, i = 0, ..., n-2$: $n-1$ equações
- Continuidade C^1
  - $s'_{i-1}(x_i) = s'_i(x_i), i = 1, ..., n-2$: $n-2$ equações

Splines Cúbicas

Spline com n pontos, $n-1$ partes

- Determinar $3(n-1) = 3n-3$ incógnitas

Equações das propriedades

- Continuidade C^0
  - $s_i(x_i) = y_i$: já satisfeitas pelas expressões de $s_i(x)$
  - $s_i(x_{i+1}) = y_{i+1}, i = 0, ..., n-2$: $n-1$ equações
- Continuidade C^1
  - $s'_{i-1}(x_i) = s'_i(x_i), i = 1, ..., n-2$: $n-2$ equações
- Continuidade C^2
  - $s''_{i-1}(x_i) = s''_i(x_i), i = 1, ..., n-2$: $n-2$ equações

Splines Cúbicas

Spline com n pontos, $n-1$ partes

- Determinar $3(n-1) = 3n-3$ incógnitas

Equações das propriedades

- Continuidade C^0
  - $s_i(x_i) = y_i$: já satisfeitas pelas expressões de $s_i(x)$
  - $s_i(x_{i+1}) = y_{i+1}, i = 0, ..., n-2 : n-1$ equações
- Continuidade C^1
  - $s'_{i-1}(x_i) = s'_i(x_i), i = 1, ..., n-2 : n-2$ equações
- Continuidade C^2
  - $s''_{i-1}(x_i) = s''_i(x_i), i = 1, ..., n-2 : n-2$ equações

Total de equações: $3n-5$

- Faltam duas equações

Splines Cúbicas

Spline natural

- Impõe curvatura nula nas extremidades

$$s_0''(x_0) = 0$$
$$s_{n-2}''(x_{n-1}) = 0$$

- Existem outras possibilidades, quase sempre envolvendo imposições nos extremos

Splines Cúbicas

Montagem do sistema

- Determinar

$$s_i(x) = y_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \quad x \in [x_i, x_{i+1}]$$

- Sabendo-se que:

$$s_i'(x) = b_i + 2c_i(x - x_i) + 3d_i(x - x_i)^2$$

  - Como:

$$s_{i-1}'(x_i) = s_i'(x_i)$$

  - Temos:

$$b_{i-1} - b_i + 2c_{i-1}(x - x_i) + 3d_{i-1}(x - x_i)^2 = 0$$

Splines Cúbicas

- Sabendo-se ainda que:

$$s_i''(x) = 2c_i + 6d_i(x - x_i)$$

  - Como:

$$s_{i-1}''(x_i) = s_i''(x_i)$$

  - Temos:

$$2c_{i-1} - 2c_i + 6d_{i-1}(x - x_i) = 0$$

Splines Cúbicas

- Sabendo-se ainda que:

$$s_i''(x) = 2c_i + 6d_i(x - x_i)$$

  - Como:

$$s_{i-1}''(x_i) = s_i''(x_i)$$

  - Temos:

$$2c_{i-1} - 2c_i + 6d_{i-1}(x - x_i) = 0$$

Para resolver o sistema, introduz-se:

$$c_{n-1} = \frac{s_{n-2}''(x_{n-1})}{2}$$

- que no caso da spline natural é zero

Splines Cúbicas

Expressa-se b_i e d_i em função de c_i:

$$b_i = \frac{\Delta_i}{\delta_i} - \frac{\delta_i}{3}(2c_i + c_{i+1}), \quad i = 0, \cdots, n-2$$

$$d_i = \frac{c_{i+1} - c_i}{3\delta_i}, \quad i = 0, \cdots, n-2$$

- onde:

$$\delta_i = x_{i+1} - x_i, \quad i = 0, \cdots, n-2$$

$$\Delta_i = y_{i+1} - y_i, \quad i = 0, \cdots, n-2$$

Splines Cúbicas

Resolve-se o sistema em c_i:

$$\begin{bmatrix} 1 & 0 & 0 & & & & \\ \delta_0 & 2\delta_0 + 2\delta_1 & \delta_1 & & & & \\ & \delta_1 & 2\delta_1 + 2\delta_2 & \delta_2 & & & \\ \cdots & & & & & & \\ & & & & \delta_{n-3} & 2\delta_{n-3} + 2\delta_{n-2} & \delta_{n-2} \\ & & & & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \cdots \\ c_{n-2} \\ c_{n-1} \end{bmatrix} =$$

$$= \begin{bmatrix} 0 \\ 3\frac{\Delta_1}{\delta_1} - 3\frac{\Delta_0}{\delta_0} \\ 3\frac{\Delta_2}{\delta_2} - 3\frac{\Delta_1}{\delta_1} \\ \cdots \\ 3\frac{\Delta_{n-2}}{\delta_{n-2}} - 3\frac{\Delta_{n-3}}{\delta_{n-3}} \\ 0 \end{bmatrix}$$

Splines Cúbicas

Algoritmo: Spline natural

- Dados
  - $x = x_0, x_1, ..., x_{n-1}$ e $y = y_0, y_1, ..., y_{n-1}$

$$
\begin{aligned}
&\textbf{for } i = 0 \textbf{ to } n-2\\
&\qquad \delta_i = x_{i+1} - x_i\\
&\qquad \Delta_i = y_{i+1} - y_i\\
&\textbf{Solve System} \text{ for } c_0 ... c_{n-1}\\
&\textbf{for } i = 0 \textbf{ to } n-2\\
&\qquad b_i = \frac{\Delta_i}{\delta_i} - \frac{\delta_i}{3}(2c_i + c_{i+1})\\
&\qquad d_i = \frac{c_{i+1} - c_i}{3\delta_i}
\end{aligned}
$$

Splines Cúbicas

Algoritmo: Spline natural

- Dados
  - $x = x_0, x_1, ..., x_{n-1}$ e $y = y_0, y_1, ..., y_{n-1}$

$$
\begin{aligned}
&\textbf{for } i = 0 \textbf{ to } n-2\\
&\qquad \delta_i = x_{i+1} - x_i\\
&\qquad \Delta_i = y_{i+1} - y_i\\
&\textbf{Solve System } \text{for } c_0 ... c_{n-1}\\
&\textbf{for } i = 0 \textbf{ to } n-2\\
&\qquad b_i = \frac{\Delta_i}{\delta_i} - \frac{\delta_i}{3}(2c_i + c_{i+1})\\
&\qquad d_i = \frac{c_{i+1} - c_i}{3\delta_i}
\end{aligned}
$$

Spline natural:

$$s_i(x) = y_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

Splines Cúbicas

Tempo esperado

- Dominado pela resolução do sistema

Splines Cúbicas

Tempo esperado

- Dominado pela resolução do sistema
  - Sistema tridiagonal

$$\mathbf{O}(\mathbf{n})$$

Splines Cúbicas

Tempo esperado

- Dominado pela resolução do sistema
  - Sistema tridiagonal

$$\mathbf{O}(\mathbf{n})$$