Regressão, Interpolação e Extrapolação Numéricas

Alexandre Rosas

Departamento de Física
Universidade Federal da Paraíba

29 de Maio de 2009

O problema

Quem é quem

- Um problema muito comum na física é o de conhecermos os valores de uma função em uma série de pontos e desejamos fazer previsões ou estimativas fora deles
- Conhecimento da forma analítica subjascente ao problema $\longrightarrow$ estimar parâmetros da função

  Regressão
- Estimar valor da função entre os valores conhecidos $\longrightarrow$ interpolação
- Estimar valor da função fora do intervalo de valores conhecidos $\longrightarrow$ extrapolação

O problema

Quem é quem

- Um problema muito comum na física é o de conhecermos os valores de uma função em uma série de pontos e desejamos fazer previsões ou estimativas fora deles
- Conhecimento da forma analítica subjascente ao problema $\longrightarrow$ estimar parâmetros da função

Regressão

- Estimar valor da função entre os valores conhecidos $\longrightarrow$ interpolação
- Estimar valor da função fora do intervalo de valores conhecidos $\longrightarrow$ extrapolação

O problema

Quem é quem

- Um problema muito comum na física é o de conhecermos os valores de uma função em uma série de pontos e desejamos fazer previsões ou estimativas fora deles
- Conhecimento da forma analítica subjascente ao problema $\longrightarrow$ estimar parâmetros da função

  Regressão
- Estimar valor da função entre os valores conhecidos $\longrightarrow$ interpolação
- Estimar valor da função fora do intervalo de valores conhecidos $\longrightarrow$ extrapolação

O problema

Quem é quem

- Um problema muito comum na física é o de conhecermos os valores de uma função em uma série de pontos e desejamos fazer previsões ou estimativas fora deles
- Conhecimento da forma analítica subjascente ao problema $\longrightarrow$ estimar parâmetros da função

  Regressão
- Estimar valor da função entre os valores conhecidos $\longrightarrow$ interpolação
- Estimar valor da função fora do intervalo de valores conhecidos $\longrightarrow$ extrapolação

O problema

Quem é quem

- Um problema muito comum na física é o de conhecermos os valores de uma função em uma série de pontos e desejamos fazer previsões ou estimativas fora deles
- Conhecimento da forma analítica subjascente ao problema $\longrightarrow$ estimar parâmetros da função

  Regressão
- Estimar valor da função entre os valores conhecidos $\longrightarrow$ interpolação
- Estimar valor da função fora do intervalo de valores conhecidos $\longrightarrow$ extrapolação

O que usar?

- Ajuste de dados é mais confiável que interpolação é mais confiável que extrapolação
- Os algoritmos para interpolação ”servem” para extrapolação
- Contudo, são muito menos confiáveis

O que usar?

- Ajuste de dados é mais confiável que interpolação é mais confiável que extrapolação
- Os algoritmos para interpolação ”servem” para extrapolação
- Contudo, são muito menos confiáveis

O que usar?

- Ajuste de dados é mais confiável que interpolação é mais confiável que extrapolação
- Os algoritmos para interpolação ”servem” para extrapolação
- Contudo, são muito menos confiáveis

Interpolação polinomial

Polinômios

- Suponha que conheçamos $y_i = f(x_i)$ em $N+1$ pontos $x_0 < x_1 < \ldots < x_N$
- Existe um único polinômio $P_N(x)$ de grau N que passa por todos eles $P_N(x_i) = y_i$

$$P_N(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \cdots + a_N(x-x_0)(x-x_1)\cdots(x-x_{N-1})$$

$$\begin{array}{ccccccccc}
a_0 & = & y_0 & & & & & & \\
a_0 & + & a_1(x_1-x_0) & = & y_1 & & & & \\
a_0 & + & a_1(x_2-x_0) & + & a_2(x_2-x_0)(x_2-x_1) & = & y_2 \\
\ldots & & \ldots & & \ldots & & \ldots
\end{array}$$

Interpolação polinomial

Polinômios

- Suponha que conheçamos $y_i = f(x_i)$ em $N+1$ pontos $x_0 < x_1 < \ldots < x_N$
- Existe um único polinômio $P_N(x)$ de grau N que passa por todos eles $P_N(x_i) = y_i$

$$P_N(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \cdots + a_N(x-x_0)(x-x_1)\cdots(x-x_{N-1})$$

$$\begin{array}{ccccccc} a_0 & = & y_0 & & & & \\ a_0 & + & a_1(x_1-x_0) & = & y_1 & & \\ a_0 & + & a_1(x_2-x_0) & + & a_2(x_2-x_0)(x_2-x_1) & = & y_2 \\ \ldots & & \ldots & & \ldots & & \ldots \end{array}$$

Interpolação polinomial

Polinômios

- Suponha que conheçamos $y_i = f(x_i)$ em $N+1$ pontos $x_0 < x_1 < \ldots < x_N$
- Existe um único polinômio $P_N(x)$ de grau N que passa por todos eles $P_N(x_i) = y_i$

$$P_N(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_N(x - x_0)(x - x_1)\cdots(x - x_{N-1})$$

$$\begin{array}{ccccccc} a_0 & = & y_0 & & & & \\ a_0 & + & a_1(x_1 - x_0) & = & y_1 & & \\ a_0 & + & a_1(x_2 - x_0) & + & a_2(x_2 - x_0)(x_2 - x_1) & = & y_2 \\ \ldots & & \ldots & & \ldots & & \ldots \end{array}$$

Interpolação polinomial

Polinômios

- Suponha que conheçamos $y_i = f(x_i)$ em $N+1$ pontos $x_0 < x_1 < \ldots < x_N$
- Existe um único polinômio $P_N(x)$ de grau N que passa por todos eles $P_N(x_i) = y_i$

$$P_N(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \cdots + a_N(x-x_0)(x-x_1)\cdots(x-x_{N-1})$$

$$\begin{array}{ccccccc}
a_0 & = & y_0 & & & & \\
a_0 & + & a_1(x_1 - x_0) & = & y_1 & & \\
a_0 & + & a_1(x_2 - x_0) & + & a_2(x_2 - x_0)(x_2 - x_1) & = & y_2 \\
\ldots & & \ldots & & \ldots & & \ldots
\end{array}$$

Interpolação polinomial

Polinômios

- Suponha que conheçamos $y_i = f(x_i)$ em $N+1$ pontos $x_0 < x_1 < \ldots < x_N$
- Existe um único polinômio $P_N(x)$ de grau N que passa por todos eles $P_N(x_i) = y_i$

$$P_N(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \cdots + a_N(x-x_0)(x-x_1)\cdots(x-x_{N-1})$$

$$\begin{array}{ccccccc} a_0 & = & y_0 & & & & \\ a_0 & + & a_1(x_1-x_0) & = & y_1 & & \\ a_0 & + & a_1(x_2-x_0) & + & a_2(x_2-x_0)(x_2-x_1) & = & y_2 \\ \ldots & & \ldots & & \ldots & & \ldots \end{array}$$

Interpolação polinomial

Polinômios

- Suponha que conheçamos $y_i = f(x_i)$ em $N+1$ pontos $x_0 < x_1 < \ldots < x_N$
- Existe um único polinômio $P_N(x)$ de grau N que passa por todos eles $P_N(x_i) = y_i$

$$P_N(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_N(x - x_0)(x - x_1)\cdots(x - x_{N-1})$$

$$\begin{array}{lllllll} a_0 & = & y_0 & & & & \\ a_0 & + & a_1(x_1 - x_0) & = & y_1 & & \\ a_0 & + & a_1(x_2 - x_0) & + & a_2(x_2 - x_0)(x_2 - x_1) & = & y_2 \\ \ldots & & \ldots & & \ldots & & \ldots \end{array}$$

Cálculo dos coeficientes

1. $a_0 = f(x_0)$
2. $f_0(x) = \frac{f(x)-f(x_0)}{x-x_0} = a_1 + a_2(x - x_1) + \cdots + a_N(x - x_1)\cdots(x - x_{N-1}) \Rightarrow \; a_1 = f_0(x_1)$
3. $f_1(x) = \frac{f_0(x)-f_0(x_1)}{x-x_1} = a_2 + a_3(x - x_2) + \cdots + a_N(x - x_2)\cdots(x - x_{N-1}) \Rightarrow \; a_2 = f_1(x_2)$
4. Em geral, temos

$$f_{k-1}(x) = \frac{f_{k-2}(x) - f_{k-2}(x_{k-1})}{x - x_{k-1}} =$$

$$a_k + a_{k+1}(x - x_k) + \cdots + a_N(x - x_k)\cdots(x - x_{N-1})$$

$$\Rightarrow \; a_k = f_{k-1}(x_k)$$

Cálculo dos coeficientes

1. $a_0 = f(x_0)$
2. $f_0(x) = \frac{f(x)-f(x_0)}{x-x_0} = a_1 + a_2(x - x_1) + \cdots + a_N(x - x_1)\cdots(x - x_{N-1}) \Rightarrow a_1 = f_0(x_1)$
3. $f_1(x) = \frac{f_0(x)-f_0(x_1)}{x-x_1} = a_2 + a_3(x - x_2) + \cdots + a_N(x - x_2)\cdots(x - x_{N-1}) \Rightarrow a_2 = f_1(x_2)$
4. Em geral, temos

$$f_{k-1}(x) = \frac{f_{k-2}(x) - f_{k-2}(x_{k-1})}{x - x_{k-1}} =$$

$$a_k + a_{k+1}(x - x_k) + \cdots + a_N(x - x_k)\cdots(x - x_{N-1})$$

$$\Rightarrow a_k = f_{k-1}(x_k)$$

Cálculo dos coeficientes

1. $a_0 = f(x_0)$
2. $f_0(x) = \frac{f(x)-f(x_0)}{x-x_0} = a_1 + a_2(x - x_1) + \cdots + a_N(x - x_1)\cdots(x - x_{N-1}) \Rightarrow \ a_1 = f_0(x_1)$
3. $f_1(x) = \frac{f_0(x)-f_0(x_1)}{x-x_1} = a_2 + a_3(x - x_2) + \cdots + a_N(x - x_2)\cdots(x - x_{N-1}) \Rightarrow \ a_2 = f_1(x_2)$
4. Em geral, temos

$$f_{k-1}(x) = \frac{f_{k-2}(x) - f_{k-2}(x_{k-1})}{x - x_{k-1}} =$$

$$a_k + a_{k+1}(x - x_k) + \cdots + a_N(x - x_k)\cdots(x - x_{N-1})$$

$$\Rightarrow \ a_k = f_{k-1}(x_k)$$

Cálculo dos coeficientes

1. $a_0 = f(x_0)$
2. $f_0(x) = \frac{f(x)-f(x_0)}{x-x_0} = a_1 + a_2(x - x_1) + \cdots + a_N(x - x_1)\cdots(x - x_{N-1}) \Rightarrow a_1 = f_0(x_1)$
3. $f_1(x) = \frac{f_0(x)-f_0(x_1)}{x-x_1} = a_2 + a_3(x - x_2) + \cdots + a_N(x - x_2)\cdots(x - x_{N-1}) \Rightarrow a_2 = f_1(x_2)$
4. Em geral, temos

$$f_{k-1}(x) = \frac{f_{k-2}(x) - f_{k-2}(x_{k-1})}{x - x_{k-1}} =$$
$$a_k + a_{k+1}(x - x_k) + \cdots + a_N(x - x_k)\cdots(x - x_{N-1})$$
$$\Rightarrow a_k = f_{k-1}(x_k)$$

Cálculo dos coeficientes

1. $a_0 = f(x_0)$
2. $f_0(x) = \frac{f(x)-f(x_0)}{x-x_0} = a_1 + a_2(x-x_1) + \cdots + a_N(x-x_1)\cdots(x-x_{N-1}) \Rightarrow \; a_1 = f_0(x_1)$
3. $f_1(x) = \frac{f_0(x)-f_0(x_1)}{x-x_1} = a_2 + a_3(x-x_2) + \cdots + a_N(x-x_2)\cdots(x-x_{N-1}) \Rightarrow \; a_2 = f_1(x_2)$
4. Em geral, temos

$$f_{k-1}(x) = \frac{f_{k-2}(x) - f_{k-2}(x_{k-1})}{x - x_{k-1}} =$$

$$a_k + a_{k+1}(x - x_k) + \cdots + a_N(x - x_k)\cdots(x - x_{N-1})$$

$$\Rightarrow \; a_k = f_{k-1}(x_k)$$

Cálculo dos coeficientes

1. $a_0 = f(x_0)$
2. $f_0(x) = \frac{f(x)-f(x_0)}{x-x_0} = a_1 + a_2(x - x_1) + \cdots +$
   $a_N(x - x_1)\cdots(x - x_{N-1}) \Rightarrow \; a_1 = f_0(x_1)$
3. $f_1(x) = \frac{f_0(x)-f_0(x_1)}{x-x_1} = a_2 + a_3(x - x_2) + \cdots +$
   $a_N(x - x_2)\cdots(x - x_{N-1}) \Rightarrow \; a_2 = f_1(x_2)$
4. Em geral, temos

$$f_{k-1}(x) = \frac{f_{k-2}(x) - f_{k-2}(x_{k-1})}{x - x_{k-1}} =$$

$$a_k + a_{k+1}(x - x_k) + \cdots + a_N(x - x_k)\cdots(x - x_{N-1})$$

$$\Rightarrow \; a_k = f_{k-1}(x_k)$$

Cálculo dos coeficientes

1. $a_0 = f(x_0)$
2. $f_0(x) = \frac{f(x)-f(x_0)}{x-x_0} = a_1 + a_2(x - x_1) + \cdots + a_N(x - x_1)\cdots(x - x_{N-1}) \Rightarrow \; a_1 = f_0(x_1)$
3. $f_1(x) = \frac{f_0(x)-f_0(x_1)}{x-x_1} = a_2 + a_3(x - x_2) + \cdots + a_N(x - x_2)\cdots(x - x_{N-1}) \Rightarrow \; a_2 = f_1(x_2)$
4. Em geral, temos

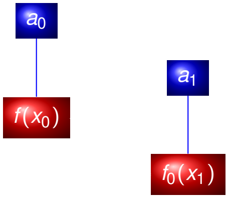
$$f_{k-1}(x) = \frac{f_{k-2}(x) - f_{k-2}(x_{k-1})}{x - x_{k-1}} =$$

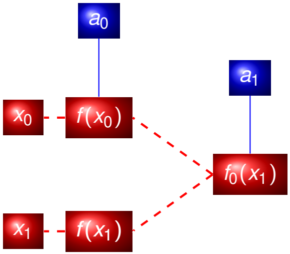
$$a_k + a_{k+1}(x - x_k) + \cdots + a_N(x - x_k)\cdots(x - x_{N-1})$$

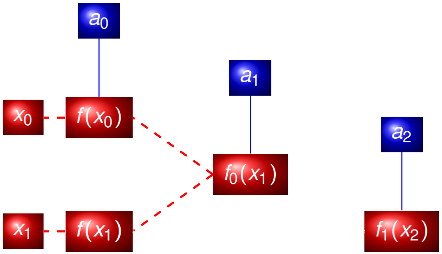
$$\Rightarrow \; a_k = f_{k-1}(x_k)$$

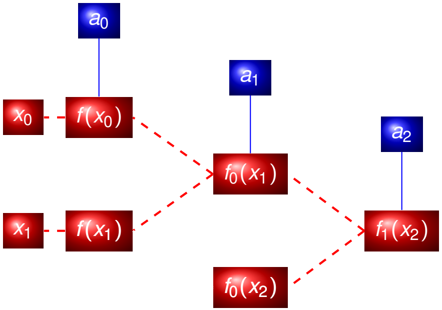
a_0

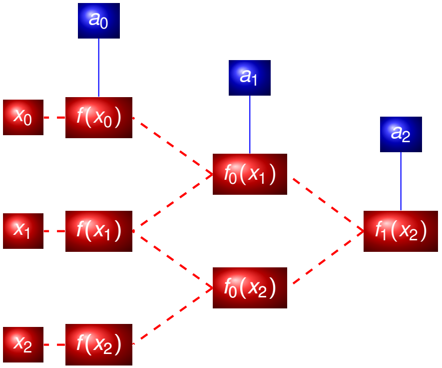
$f(x_0)$

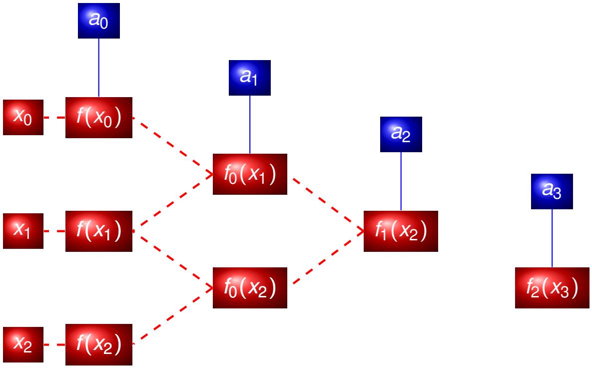

a_0
$f(x_0)$
a_1
$f_0(x_1)$

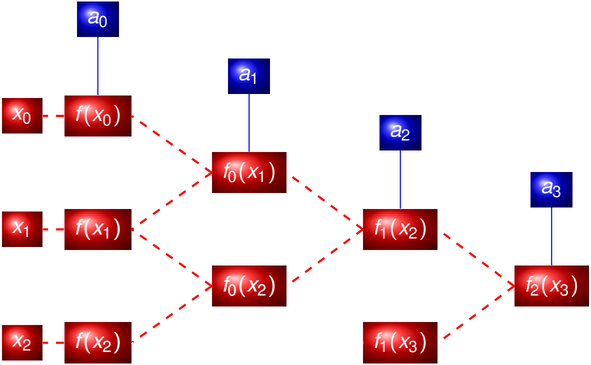

a_0
a_1
x_0
$f(x_0)$
$f_0(x_1)$
x_1
$f(x_1)$

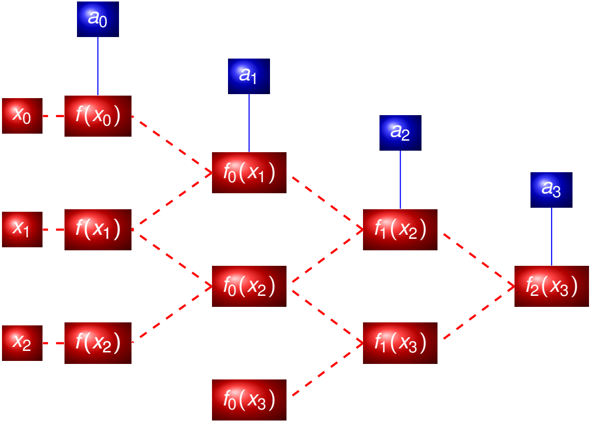

a_0
a_1
a_2
x_0
$f(x_0)$
$f_0(x_1)$
x_1
$f(x_1)$
$f_1(x_2)$

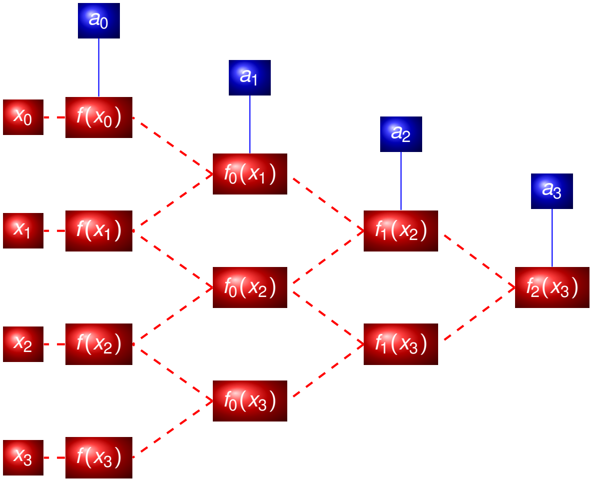

a_0
a_1
a_2
x_0
$f(x_0)$
$f_0(x_1)$
$f_1(x_2)$
x_1
$f(x_1)$
$f_0(x_2)$


a_0
a_1
a_2
x_0
$f(x_0)$
$f_0(x_1)$
x_1
$f(x_1)$
$f_1(x_2)$
$f_0(x_2)$
x_2
$f(x_2)$

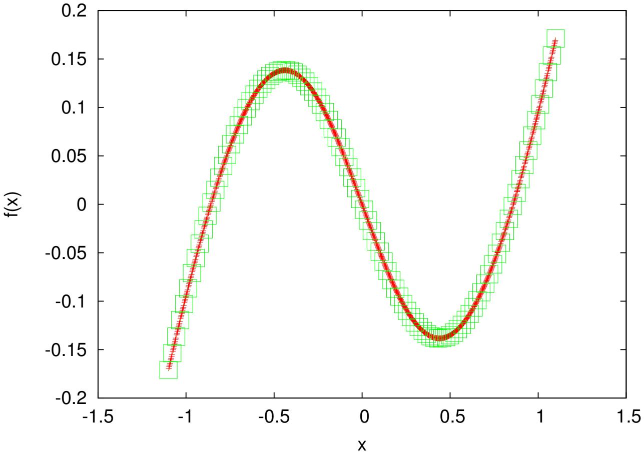

a_0
a_1
a_2
a_3
x_0
$f(x_0)$
x_1
$f(x_1)$
x_2
$f(x_2)$
$f_0(x_1)$
$f_0(x_2)$
$f_1(x_2)$
$f_2(x_3)$


a_0
a_1
a_2
a_3
x_0
$f(x_0)$
$f_0(x_1)$
x_1
$f(x_1)$
$f_1(x_2)$
$f_0(x_2)$
$f_2(x_3)$
x_2
$f(x_2)$
$f_1(x_3)$


a_0
a_1
a_2
a_3
x_0
$f(x_0)$
x_1
$f(x_1)$
x_2
$f(x_2)$
$f_0(x_1)$
$f_0(x_2)$
$f_0(x_3)$
$f_1(x_2)$
$f_1(x_3)$
$f_2(x_3)$


a_0
a_1
a_2
a_3
x_0
$f(x_0)$
x_1
$f(x_1)$
x_2
$f(x_2)$
x_3
$f(x_3)$
$f_0(x_1)$
$f_0(x_2)$
$f_0(x_3)$
$f_1(x_2)$
$f_1(x_3)$
$f_2(x_3)$

Interpolação polinomial



Observações finais

- A interpolação polinomial deve ser feita em cada ponto
- Em pontos diferentes, obtemos polinômios diferentes
- Erro pode ser grande próximo aos limites do intervalo.

Observações finais

- A interpolação polinomial deve ser feita em cada ponto
- Em pontos diferentes, obtemos polinômios diferentes
- Erro pode ser grande próximo aos limites do intervalo.

Observações finais

- A interpolação polinomial deve ser feita em cada ponto
- Em pontos diferentes, obtemos polinômios diferentes
- Erro pode ser grande próximo aos limites do intervalo.

Spline cúbico – objetivo

- Interpolar um conjunto de $n+1$ pontos $\{x_i, y_i\}$ com uma função
  $s(x) = s_i(x) \quad \text{para} \quad x_i < x \leq x_{i+1}$ n equações
- $s_i(x)$ é um polinômio de grau 3 $4n$ coeficiente
- Condições de contorno
  1. Interpolação $\Rightarrow$ $s(x_i) = y_i$ n equações
  2. Continuidade de $s(x)$ $\Rightarrow$ $s_i(x_i) = s_{i-1}(x_i) = y_i$ n equações
  3. Continuidade de $s'(x)$ $\Rightarrow$ $s_i'(x_i) = s_{i-1}'(x_i)$ $(n-1)$ equações
  4. Continuidade de $s''(x)$ $\Rightarrow$ $s_i''(x_i) = s_{i-1}''(x_i)$ $(n-1)$ equações
  5. Condições de contorno livres nos extremos
     $s''(x_0) = s''(x_n) = 0$ 2 equações

Spline cúbico – objetivo

- Interpolar um conjunto de $n+1$ pontos $\{x_i, y_i\}$ com uma função
  $s(x) = s_i(x) \quad \text{para} \quad x_i < x \leq x_{i+1}$ n equações
- $s_i(x)$ é um polinômio de grau 3 $4n$ coeficiente
- Condições de contorno
  1. Interpolação $\Rightarrow \quad s(x_i) = y_i$ n equações
  2. Continuidade de $s(x) \Rightarrow s_i(x_i) = s_{i-1}(x_i) = y_i$ n equações
  3. Continuidade de $s'(x) \Rightarrow s'_i(x_i) = s'_{i-1}(x_i)$ $(n-1)$ equações
  4. Continuidade de $s''(x) \Rightarrow s''_i(x_i) = s''_{i-1}(x_i)$ $(n-1)$ equações
  5. Condições de contorno livres nos extremos
     $s''(x_0) = s''(x_n) = 0$ 2 equações

Spline cúbico – objetivo

- Interpolar um conjunto de $n+1$ pontos $\{x_i, y_i\}$ com uma função
  $s(x) = s_i(x) \quad \text{para} \quad x_i < x \leq x_{i+1}$ n equações
- $s_i(x)$ é um polinômio de grau 3 $4n$ coeficiente
- Condições de contorno
  1. Interpolação $\Rightarrow \quad s(x_i) = y_i$ n equações
  2. Continuidade de $s(x) \Rightarrow s_i(x_i) = s_{i-1}(x_i) = y_i$ n equações
  3. Continuidade de $s'(x) \Rightarrow s'_i(x_i) = s'_{i-1}(x_i)$ $(n-1)$ equações
  4. Continuidade de $s''(x) \Rightarrow s''_i(x_i) = s''_{i-1}(x_i)$ $(n-1)$ equações
  5. Condições de contorno livres nos extremos
     $s''(x_0) = s''(x_n) = 0$ 2 equações

Spline cúbico – objetivo

- Interpolar um conjunto de $n+1$ pontos $\{x_i, y_i\}$ com uma função
  $s(x) = s_i(x) \quad \text{para} \quad x_i < x \le x_{i+1}$ n equações
- $s_i(x)$ é um polinômio de grau 3 $4n$ coeficiente
- Condições de contorno
  1. Interpolação $\Rightarrow \; s(x_i) = y_i$ n equações
  2. Continuidade de $s(x) \Rightarrow s_i(x_i) = s_{i-1}(x_i) = y_i$ n equações
  3. Continuidade de $s'(x) \Rightarrow s'_i(x_i) = s'_{i-1}(x_i)$ $(n-1)$ equações
  4. Continuidade de $s''(x) \Rightarrow s''_i(x_i) = s''_{i-1}(x_i)$ $(n-1)$ equações
  5. Condições de contorno livres nos extremos
     $s''(x_0) = s''(x_n) = 0$ 2 equações

Spline cúbico – objetivo

- Interpolar um conjunto de $n+1$ pontos $\{x_i, y_i\}$ com uma função
  $s(x) = s_i(x) \quad \text{para} \quad x_i < x \leq x_{i+1}$ n equações
- $s_i(x)$ é um polinômio de grau 3 $4n$ coeficiente
- Condições de contorno
  1. Interpolação $\Rightarrow \; s(x_i) = y_i$ n equações
  2. Continuidade de $s(x) \;\Rightarrow\; s_i(x_i) = s_{i-1}(x_i) = y_i$ n equações
  3. Continuidade de $s'(x) \;\Rightarrow\; s_i'(x_i) = s_{i-1}'(x_i)$ $(n-1)$ equações
  4. Continuidade de $s''(x) \;\Rightarrow\; s_i''(x_i) = s_{i-1}''(x_i)$ $(n-1)$ equações
  5. Condições de contorno livres nos extremos
     $s''(x_0) = s''(x_n) = 0$ 2 equações

Spline cúbico – objetivo

- Interpolar um conjunto de $n+1$ pontos $\{x_i, y_i\}$ com uma função
  $s(x) = s_i(x) \quad \text{para} \quad x_i < x \leq x_{i+1}$ n equações
- $s_i(x)$ é um polinômio de grau 3 $4n$ coeficiente
- Condições de contorno
  1. Interpolação $\Rightarrow \; s(x_i) = y_i$ n equações
  2. Continuidade de $s(x) \; \Rightarrow \; s_i(x_i) = s_{i-1}(x_i) = y_i$ n equações
  3. Continuidade de $s'(x) \; \Rightarrow \; s_i'(x_i) = s_{i-1}'(x_i)$ $(n-1)$ equações
  4. Continuidade de $s''(x) \; \Rightarrow \; s_i''(x_i) = s_{i-1}''(x_i)$ $(n-1)$ equações
  5. Condições de contorno livres nos extremos
     $s''(x_0) = s''(x_n) = 0$ 2 equações

Spline cúbico – objetivo

- Interpolar um conjunto de $n+1$ pontos $\{x_i, y_i\}$ com uma função
  $s(x) = s_i(x) \quad \text{para} \quad x_i < x \leq x_{i+1}$ n equações
- $s_i(x)$ é um polinômio de grau 3 $4n$ coeficiente
- Condições de contorno
  1. Interpolação $\Rightarrow$ $s(x_i) = y_i$ n equações
  2. Continuidade de $s(x)$ $\Rightarrow$ $s_i(x_i) = s_{i-1}(x_i) = y_i$ n equações
  3. Continuidade de $s'(x)$ $\Rightarrow$ $s_i'(x_i) = s_{i-1}'(x_i)$ $(n-1)$ equações
  4. Continuidade de $s''(x)$ $\Rightarrow$ $s_i''(x_i) = s_{i-1}''(x_i)$ $(n-1)$ equações
  5. Condições de contorno livres nos extremos
     $s''(x_0) = s''(x_n) = 0$ 2 equações

Spline cúbico – objetivo

- Interpolar um conjunto de $n+1$ pontos $\{x_i, y_i\}$ com uma função
  $s(x) = s_i(x) \quad \text{para} \quad x_i < x \leq x_{i+1}$ n equações
- $s_i(x)$ é um polinômio de grau 3 $4n$ coeficiente
- Condições de contorno
  1. Interpolação $\Rightarrow$ $s(x_i) = y_i$ n equações
  2. Continuidade de $s(x)$ $\Rightarrow$ $s_i(x_i) = s_{i-1}(x_i) = y_i$ n equações
  3. Continuidade de $s'(x)$ $\Rightarrow$ $s_i'(x_i) = s_{i-1}'(x_i)$ $(n-1)$ equações
  4. Continuidade de $s''(x)$ $\Rightarrow$ $s_i''(x_i) = s_{i-1}''(x_i)$ $(n-1)$ equações
  5. Condições de contorno livres nos extremos
     $s''(x_0) = s''(x_n) = 0$ 2 equações

Spline cúbico – objetivo

- Interpolar um conjunto de $n + 1$ pontos $\{x_i, y_i\}$ com uma função
  $s(x) = s_i(x) \quad \text{para} \quad x_i < x \leq x_{i+1}$ n equações
- $s_i(x)$ é um polinômio de grau 3 $4n$ coeficiente
- Condições de contorno
  1. Interpolação $\Rightarrow$ $s(x_i) = y_i$ n equações
  2. Continuidade de $s(x)$ $\Rightarrow$ $s_i(x_i) = s_{i-1}(x_i) = y_i$ n equações
  3. Continuidade de $s'(x)$ $\Rightarrow$ $s'_i(x_i) = s'_{i-1}(x_i)$ $(n-1)$ equações
  4. Continuidade de $s''(x)$ $\Rightarrow$ $s''_i(x_i) = s''_{i-1}(x_i)$ $(n-1)$ equações
  5. Condições de contorno livres nos extremos
     $s''(x_0) = s''(x_n) = 0$ 2 equações

Spline cúbico – objetivo

- Interpolar um conjunto de $n+1$ pontos $\{x_i, y_i\}$ com uma função
  $s(x) = s_i(x) \quad \text{para} \quad x_i < x \leq x_{i+1}$ n equações
- $s_i(x)$ é um polinômio de grau 3 $4n$ coeficiente
- Condições de contorno
  1. Interpolação $\Rightarrow \; s(x_i) = y_i$ n equações
  2. Continuidade de $s(x) \Rightarrow s_i(x_i) = s_{i-1}(x_i) = y_i$ n equações
  3. Continuidade de $s'(x) \Rightarrow s'_i(x_i) = s'_{i-1}(x_i)$ $(n-1)$ equações
  4. Continuidade de $s''(x) \Rightarrow s''_i(x_i) = s''_{i-1}(x_i)$ $(n-1)$ equações
  5. Condições de contorno livres nos extremos
     $s''(x_0) = s''(x_n) = 0$ 2 equações

Spline cúbico – objetivo

- Interpolar um conjunto de $n+1$ pontos $\{x_i, y_i\}$ com uma função
  $s(x) = s_i(x) \quad \text{para} \quad x_i < x \leq x_{i+1}$ n equações
- $s_i(x)$ é um polinômio de grau 3 $4n$ coeficiente
- Condições de contorno
  1. Interpolação $\Rightarrow \; s(x_i) = y_i$ n equações
  2. Continuidade de $s(x) \;\Rightarrow\; s_i(x_i) = s_{i-1}(x_i) = y_i$ n equações
  3. Continuidade de $s'(x) \;\Rightarrow\; s_i'(x_i) = s_{i-1}'(x_i)$ $(n-1)$ equações
  4. Continuidade de $s''(x) \;\Rightarrow\; s_i''(x_i) = s_{i-1}''(x_i)$ $(n-1)$ equações
  5. Condições de contorno livres nos extremos
     $s''(x_0) = s''(x_n) = 0$ 2 equações

Determinação de $s_i(x)$

$$s_i(x) = a_i(x-x_i)^3 + b_i(x-x_{i+1})^3 + c_i(x-x_i) + d_i(x-x_{i+1})$$

Precisamos, portanto, calcular $\{a_i, b_i, c_i, d_i\}$

$$\begin{aligned} s_i''(x_i) &= y_i'' = 6b_i(x_i - x_{i+1}) \\ &\Rightarrow b_i = -\frac{y_i''}{6(x_{i+1}-x_i)} \\ s_i''(x_{i+1}) &= s_{i+1}(x_{i+1}) = y_{i+1}'' = 6a_i(x_{i+1}-x_i) \\ &\Rightarrow a_i = \frac{y_{i+1}''}{6(x_{i+1}-x_i)} \end{aligned}$$

$$s_i(x) = \frac{y_{i+1}''}{6h_i}(x-x_i)^3 - \frac{y_i''}{6h_i}(x-x_{i+1})^3 + c_i(x-x_i) + d_i(x-x_{i+1})$$

Determinação de $s_i(x)$

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_{i+1})^3 + c_i(x - x_i) + d_i(x - x_{i+1})$$

Definindo y_i'' como a segunda derivada em x_i (a determinar), temos

$$\begin{aligned}
s_i''(x_i) &= y_i'' = 6b_i(x_i - x_{i+1}) \\
&\Rightarrow b_i = -\frac{y_i''}{6(x_{i+1} - x_i)} \\
s_i''(x_{i+1}) &= s_{i+1}(x_{i+1}) = y_{i+1}'' = 6a_i(x_{i+1} - x_i) \\
&\Rightarrow a_i = \frac{y_{i+1}''}{6(x_{i+1} - x_i)}
\end{aligned}$$

$$s_i(x) = \frac{y_{i+1}''}{6h_i}(x - x_i)^3 - \frac{y_i''}{6h_i}(x - x_{i+1})^3 + c_i(x - x_i) + d_i(x - x_{i+1})$$

Determinação de $s_i(x)$

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_{i+1})^3 + c_i(x - x_i) + d_i(x - x_{i+1})$$

Definindo y_i'' como a segunda derivada em x_i (a determinar), temos

$$\begin{aligned}
s_i''(x_i) &= y_i'' = 6b_i(x_i - x_{i+1}) \\
&\Rightarrow b_i = -\frac{y_i''}{6(x_{i+1} - x_i)} \\
s_i''(x_{i+1}) &= s_{i+1}(x_{i+1}) = y_{i+1}'' = 6a_i(x_{i+1} - x_i) \\
&\Rightarrow a_i = \frac{y_{i+1}''}{6(x_{i+1} - x_i)}
\end{aligned}$$

$$s_i(x) = \frac{y_{i+1}''}{6h_i}(x - x_i)^3 - \frac{y_i''}{6h_i}(x - x_{i+1})^3 + c_i(x - x_i) + d_i(x - x_{i+1})$$

Determinação de $s_i(x)$

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_{i+1})^3 + c_i(x - x_i) + d_i(x - x_{i+1})$$

Definindo y_i'' como a segunda derivada em x_i (a determinar), temos

$$\begin{aligned}
s_i''(x_i) &= y_i'' = 6b_i(x_i - x_{i+1}) \\
&\Rightarrow b_i = -\frac{y_i''}{6(x_{i+1} - x_i)} \\
s_i''(x_{i+1}) &= s_{i+1}(x_{i+1}) = y_{i+1}'' = 6a_i(x_{i+1} - x_i) \\
&\Rightarrow a_i = \frac{y_{i+1}''}{6(x_{i+1} - x_i)}
\end{aligned}$$

$$s_i(x) = \frac{y_{i+1}''}{6h_i}(x - x_i)^3 - \frac{y_i''}{6h_i}(x - x_{i+1})^3 + c_i(x - x_i) + d_i(x - x_{i+1})$$

Determinação de $s_i(x)$

$$s_i(x) = \frac{y''_{i+1}}{6h_i}(x - x_i)^3 - \frac{y''_i}{6h_i}(x - x_{i+1})^3 + c_i(x - x_i) + d_i(x - x_{i+1})$$

Mas, $s_i(x_i) = y_i$ e $s_i(x_{i+1}) = y_{i+1}$, logo

$$y_{i+1} = \frac{y''_{i+1}}{6}h_i^2 + c_i h_i \Rightarrow c_i = \frac{y_{i+1}}{h_i} - \frac{y''_{i+1}}{6}h_i$$

$$y_i = \frac{y''_i}{6}h_i^2 - d_i h_i \Rightarrow d_i = -\frac{y_i}{h_i} + \frac{y''_i}{6}h_i$$

$$s_i(x) = \frac{y''_{i+1}}{6h_i}(x - x_i)^3 - \frac{y''_i}{6h_i}(x - x_{i+1})^3$$
$$+ \left(\frac{y_{i+1}}{h_i} - \frac{y''_{i+1}}{6}h_i\right)(x - x_i) + \left(-\frac{y_i}{h_i} + \frac{y''_i}{6}h_i\right)(x - x_{i+1})$$

Determinação de $s_i(x)$

$$s_i(x) = \frac{y''_{i+1}}{6h_i}(x - x_i)^3 - \frac{y''_i}{6h_i}(x - x_{i+1})^3 + c_i(x - x_i) + d_i(x - x_{i+1})$$

Mas, $s_i(x_i) = y_i$ e $s_i(x_{i+1}) = y_{i+1}$, logo

$$\begin{aligned} y_{i+1} &= \frac{y''_{i+1}}{6}h_i^2 + c_i h_i \Rightarrow c_i = \frac{y_{i+1}}{h_i} - \frac{y''_{i+1}}{6}h_i \\ y_i &= \frac{y''_i}{6}h_i^2 - d_i h_i \Rightarrow d_i = -\frac{y_i}{h_i} + \frac{y''_i}{6}h_i \end{aligned}$$

$$\begin{aligned} s_i(x) = {} & \frac{y''_{i+1}}{6h_i}(x - x_i)^3 - \frac{y''_i}{6h_i}(x - x_{i+1})^3 \\ & + \left(\frac{y_{i+1}}{h_i} - \frac{y''_{i+1}}{6}h_i\right)(x - x_i) + \left(-\frac{y_i}{h_i} + \frac{y''_i}{6}h_i\right)(x - x_{i+1}) \end{aligned}$$

Determinação de $s_i(x)$

$$s_i(x) = \frac{y''_{i+1}}{6h_i}(x - x_i)^3 - \frac{y''_i}{6h_i}(x - x_{i+1})^3 + c_i(x - x_i) + d_i(x - x_{i+1})$$

Mas, $s_i(x_i) = y_i$ e $s_i(x_{i+1}) = y_{i+1}$, logo

$$\begin{aligned} y_{i+1} &= \frac{y''_{i+1}}{6}h_i^2 + c_i h_i \Rightarrow c_i = \frac{y_{i+1}}{h_i} - \frac{y''_{i+1}}{6}h_i \\ y_i &= \frac{y''_i}{6}h_i^2 - d_i h_i \Rightarrow d_i = -\frac{y_i}{h_i} + \frac{y''_i}{6}h_i \end{aligned}$$

$$s_i(x) = \frac{y''_{i+1}}{6h_i}(x - x_i)^3 - \frac{y''_i}{6h_i}(x - x_{i+1})^3 + \left(\frac{y_{i+1}}{h_i} - \frac{y''_{i+1}}{6}h_i\right)(x - x_i) + \left(-\frac{y_i}{h_i} + \frac{y''_i}{6}h_i\right)(x - x_{i+1})$$

Determinação de y_i''

Usando a continuidade de $s'(x)$ $\left[s_i'(x_i) = s_{i-1}'(x_i)\right]$, temos

$$-\frac{y_i''}{2}h_i + \left[\frac{y_{i+1}}{h_i} - \frac{y_{i+1}''}{6}h_i\right] + \left[-\frac{y_i}{h_i} + \frac{y_i''}{6}h_i\right]$$
$$= \frac{y_i''}{2}h_{i-1} + \left[\frac{y_i}{h_{i-1}} - \frac{y_i''}{6}h_{i-1}\right] + \left[-\frac{y_{i-1}}{h_{i-1}} + \frac{y_{i-1}''}{6}h_{i-1}\right]$$

Donde

$$h_{i-1}y_{i-1}'' + 2(h_{i-1} + h_i)y_i'' + h_i y_{i+1}''$$
$$= 6\left[\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right]$$

Determinação de y_i''

Usando a continuidade de $s'(x)$ $\left[s_i'(x_i) = s_{i-1}'(x_i)\right]$, temos

$$
\begin{aligned}
-\frac{y_i''}{2}h_i + \left[\frac{y_{i+1}}{h_i} - \frac{y_{i+1}''}{6}h_i\right] + \left[-\frac{y_i}{h_i} + \frac{y_i''}{6}h_i\right] \\
= \frac{y_i''}{2}h_{i-1} + \left[\frac{y_i}{h_{i-1}} - \frac{y_i''}{6}h_{i-1}\right] + \left[-\frac{y_{i-1}}{h_{i-1}} + \frac{y_{i-1}''}{6}h_{i-1}\right]
\end{aligned}
$$

Donde

$$
\begin{aligned}
h_{i-1}y_{i-1}'' + 2(h_{i-1} + h_i)y_i'' + h_i y_{i+1}'' \\
= 6\left[\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right]
\end{aligned}
$$

Determinação de y_i''

Usando a continuidade de $s'(x)$ $\left[s_i'(x_i) = s_{i-1}'(x_i)\right]$, temos

$$
\begin{aligned}
-\frac{y_i''}{2}h_i + \left[\frac{y_{i+1}}{h_i} - \frac{y_{i+1}''}{6}h_i\right] + \left[-\frac{y_i}{h_i} + \frac{y_i''}{6}h_i\right] \\
= \frac{y_i''}{2}h_{i-1} + \left[\frac{y_i}{h_{i-1}} - \frac{y_i''}{6}h_{i-1}\right] + \left[-\frac{y_{i-1}}{h_{i-1}} + \frac{y_{i-1}''}{6}h_{i-1}\right]
\end{aligned}
$$

Donde

$$
\begin{aligned}
h_{i-1}y_{i-1}'' + 2(h_{i-1} + h_i)y_i'' + h_i y_{i+1}'' \\
= 6\left[\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right]
\end{aligned}
$$

Determinação de y_i''

Definindo $u_i = 2(h_{i-1} + h_i)$ e $v_i = \frac{y_{i+1}-y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}$ temos que resolver o seguinte sistema de equações: ▸ sistema tridiagonal

$$\begin{pmatrix} u_1 & h_1 & 0 & \cdots & & & & \\ h_1 & u_2 & h_2 & 0 & \cdots & & & \\ 0 & h_2 & u_3 & h_3 & 0 & \cdots & & \\ & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\ & & & \cdots & h_{n-4} & u_{n-3} & h_{n-3} & 0 \\ & & & & \cdots & h_{n-3} & u_{n-2} & h_{n-2} \\ & & & & & \cdots & h_{n-2} & u_{n-1} \end{pmatrix} \begin{pmatrix} y_1'' \\ y_2'' \\ y_3'' \\ \cdots \\ y_{n-3}'' \\ y_{n-2}'' \\ y_{n-1}'' \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \cdots \\ v_{n-3} \\ v_{n-2} \\ v_{n-1} \end{pmatrix}$$

Algoritmo

1. Dado o conjunto de pontos $\{x_i, y_i\}$, resolvemos o sistema de equações acima, obtendo y_i''.
2. Para um ponto qualquer x_0, encontramos i, tal que $x_i \leq x_0 < x_{i+1}$.
3. Calculamos a interpolação em x_0

$$s_i(x_0) = \frac{y_{i+1}''}{6h_i}(x_0 - x_i)^3 - \frac{y_i''}{6h_i}(x_0 - x_{i+1})^3$$
$$+ \left(\frac{y_{i+1}}{h_i} - \frac{y_{i+1}''}{6}h_i\right)(x_0 - x_i) + \left(-\frac{y_i}{h_i} + \frac{y_i''}{6}h_i\right)(x_0 - x_{i+1})$$

Algoritmo

1. Dado o conjunto de pontos $\{x_i, y_i\}$, resolvemos o sistema de equações acima, obtendo y_i''.
2. Para um ponto qualquer x_0, encontramos i, tal que $x_i \leq x_0 < x_{i+1}$.
3. Calculamos a interpolação em x_0

$$s_i(x_0) = \frac{y''_{i+1}}{6h_i}(x_0 - x_i)^3 - \frac{y''_i}{6h_i}(x_0 - x_{i+1})^3$$
$$+ \left(\frac{y_{i+1}}{h_i} - \frac{y''_{i+1}}{6}h_i\right)(x_0 - x_i) + \left(-\frac{y_i}{h_i} + \frac{y''_i}{6}h_i\right)(x_0 - x_{i+1})$$

Algoritmo

1. Dado o conjunto de pontos $\{x_i, y_i\}$, resolvemos o sistema de equações acima, obtendo y_i''.
2. Para um ponto qualquer x_0, encontramos i, tal que $x_i \leq x_0 < x_{i+1}$.
3. Calculamos a interpolação em x_0

$$s_i(x_0) = \frac{y_{i+1}''}{6h_i}(x_0 - x_i)^3 - \frac{y_i''}{6h_i}(x_0 - x_{i+1})^3$$
$$+ \left(\frac{y_{i+1}}{h_i} - \frac{y_{i+1}''}{6}h_i\right)(x_0 - x_i) + \left(-\frac{y_i}{h_i} + \frac{y_i''}{6}h_i\right)(x_0 - x_{i+1})$$

Observações finais

- O método Spline gera uma função que pode ser avaliada para qualquer ponto
- Chamadas sucessivas para diferentes pontos não requerem novas interpolações
- Suavidade da curva é garantida

Observações finais

- O método Spline gera uma função que pode ser avaliada para qualquer ponto
- Chamadas sucessivas para diferentes pontos não requerem novas interpolações
- Suavidade da curva é garantida

Observações finais

- O método Spline gera uma função que pode ser avaliada para qualquer ponto
- Chamadas sucessivas para diferentes pontos não requerem novas interpolações
- Suavidade da curva é garantida

Objetivo e Tipos

Objetivo

- Modelar a relação entre uma ou mais variáveis dependentes e as variáveis de controle
- Conhecendo a dependência funcional entre as variáveis dependentes e de controle, encontrar os parâmetros que melhor ajustam a função aos dados
- A função não passa necessariamente por todos os pontos, mas deve minimizar o erro (mínimos quadrados)

Objetivo e Tipos

Objetivo

- Modelar a relação entre uma ou mais variáveis dependentes e as variáveis de controle
- Conhecendo a dependência funcional entre as variáveis dependentes e de controle, encontrar os parâmetros que melhor ajustam a função aos dados
- A função não passa necessariamente por todos os pontos, mas deve minimizar o erro (mínimos quadrados)

Objetivo e Tipos

Objetivo

- Modelar a relação entre uma ou mais variáveis dependentes e as variáveis de controle
- Conhecendo a dependência funcional entre as variáveis dependentes e de controle, encontrar os parâmetros que melhor ajustam a função aos dados
- A função não passa necessariamente por todos os pontos, mas deve minimizar o erro (mínimos quadrados)

Mínimos quadrados

Definições

- Assim como no caso da interpolação, conhecemos N pontos $\{x_i, y_i\}$
- Temos também uma função $f(x; a_1, a_2, \ldots, a_k)$, onde $\{a_j\}$ são os parâmetros ajustarão a função
- Por exemplo, $f(x) = a_1 x + a_2$
- Podemos "medir" a distância entre a função e os pontos (erro) pela soma dos quadrados das diferenças

$$\chi^2 = \sum_{i=1}^{N} [f(x_i) - y_i]^2$$

Mínimos quadrados $\Rightarrow$ minimizar χ^2

Mínimos quadrados

Definições

- Assim como no caso da interpolação, conhecemos N pontos $\{x_i, y_i\}$
- Temos também uma função $f(x; a_1, a_2, \ldots, a_k)$, onde $\{a_j\}$ são os parâmetros ajustarão a função
- Por exemplo, $f(x) = a_1 x + a_2$
- Podemos "medir" a distância entre a função e os pontos (erro) pela soma dos quadrados das diferenças

$$\chi^2 = \sum_{i=1}^{N} [f(x_i) - y_i]^2$$

Mínimos quadrados $\Rightarrow$ minimizar χ^2

Mínimos quadrados

Definições

- Assim como no caso da interpolação, conhecemos N pontos $\{x_i, y_i\}$
- Temos também uma função $f(x; a_1, a_2, \ldots, a_k)$, onde $\{a_j\}$ são os parâmetros ajustarão a função
- Por exemplo, $f(x) = a_1 x + a_2$
- Podemos ”medir” a distância entre a função e os pontos (erro) pela soma dos quadrados das diferenças

$$\chi^2 = \sum_{i=1}^{N} [f(x_i) - y_i]^2$$

Mínimos quadrados $\Rightarrow$ minimizar χ^2

Mínimos quadrados

Definições

- Assim como no caso da interpolação, conhecemos N pontos $\{x_i, y_i\}$
- Temos também uma função $f(x; a_1, a_2, \ldots, a_k)$, onde $\{a_j\}$ são os parâmetros ajustarão a função
- Por exemplo, $f(x) = a_1 x + a_2$
- Podemos ”medir” a distância entre a função e os pontos (erro) pela soma dos quadrados das diferenças

$$\chi^2 = \sum_{i=1}^{N} [f(x_i) - y_i]^2$$

Mínimos quadrados $\Rightarrow$ minimizar χ^2

Mínimos quadrados

Definições

- Assim como no caso da interpolação, conhecemos N pontos $\{x_i, y_i\}$
- Temos também uma função $f(x; a_1, a_2, \ldots, a_k)$, onde $\{a_j\}$ são os parâmetros ajustarão a função
- Por exemplo, $f(x) = a_1 x + a_2$
- Podemos ”medir” a distância entre a função e os pontos (erro) pela soma dos quadrados das diferenças

$$\chi^2 = \sum_{i=1}^{N} [f(x_i) - y_i]^2$$

Mínimos quadrados $\Rightarrow$ minimizar χ^2

Regressão linear: um exemplo

$f(x) = a_0 + a_1 x$

Minimizar $\chi^2 = \sum_{i=1}^{N} [a_1 x_i + a_0 - y_i]^2$

$$\frac{\partial \chi^2}{\partial a_1} = 2\left[a_1 \left(\sum_{i=1}^{N} x_i^2\right) + a_0 \left(\sum_{i=1}^{N} x_i\right) - \left(\sum_{i=1}^{N} x_i y_i\right)\right] = 0$$

$$\frac{\partial \chi^2}{\partial a_0} = 2\left[a_1 \left(\sum_{i=1}^{N} x_i\right) + N a_0 - \left(\sum_{i=1}^{N} y_i\right)\right] = 0$$

Regressão linear

Resolvendo o sistema para a_1 e a_0 temos

$$a_1 = \frac{N\left(\sum_{i=1}^{N} x_i y_i\right) - \left(\sum_{i=1}^{N} x_i\right)\left(\sum_{i=1}^{N} y_i\right)}{N\left(\sum_{i=1}^{N} x_i^2\right) - \left(\sum_{i=1}^{N} x_i\right)^2}$$

$$a_0 = \frac{\left(\sum_{i=1}^{N} x_i^2\right)\left(\sum_{i=1}^{N} y_i\right) - \left(\sum_{i=1}^{N} x_i\right)\left(\sum_{i=1}^{N} x_i y_i\right)}{N\left(\sum_{i=1}^{N} x_i^2\right) - \left(\sum_{i=1}^{N} x_i\right)^2}$$

Regressão linear

Interpretação física

$$a_1 = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\langle x^2 \rangle - \langle x \rangle^2}$$

$$a_0 = \frac{\langle x^2 \rangle \langle y \rangle - \langle x \rangle \langle xy \rangle}{\langle x^2 \rangle - \langle x \rangle^2}$$

Qualidade da regressão

Coeficiente de correlação

$$r = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\sqrt{\left(\langle x^2 \rangle - \langle x \rangle^2\right)\left(\langle y^2 \rangle - \langle y \rangle^2\right)}}$$

- Dados correlacionados:
  $y = f(x) = a_0 + a_1 x \;\Rightarrow\; r = \mathrm{sinal}(a_1)$
  - $r > 0 \;\rightarrow\;$ dados correlacionados
  - $r < 0 \;\rightarrow\;$ dados anti-correlacionados
- $r = 0 \;\Rightarrow\;$ y não depende de x

Quanto mais próximo de 1 for o valor de r^2, melhor a regressão

Qualidade da regressão

Coeficiente de correlação

$$r = \frac{\langle xy\rangle - \langle x\rangle\langle y\rangle}{\sqrt{\left(\langle x^2\rangle - \langle x\rangle^2\right)\left(\langle y^2\rangle - \langle y\rangle^2\right)}}$$

- Dados correlacionados:
  $y = f(x) = a_0 + a_1 x \;\Rightarrow\; r = \text{sinal}(a_1)$
  - $r > 0 \;\rightarrow$ dados correlacionados
  - $r < 0 \;\rightarrow$ dados anti-correlacionados
- $r = 0 \;\Rightarrow\;$ y não depende de x

Quanto mais próximo de 1 for o valor de r^2, melhor a regressão

Qualidade da regressão

Coeficiente de correlação

$$r = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\sqrt{\left(\langle x^2 \rangle - \langle x \rangle^2\right)\left(\langle y^2 \rangle - \langle y \rangle^2\right)}}$$

- Dados correlacionados:
  $y = f(x) = a_0 + a_1 x \;\Rightarrow\; r = \text{sinal}(a_1)$
  - $r > 0 \;\rightarrow$ dados correlacionados
  - $r < 0 \;\rightarrow$ dados anti-correlacionados
- $r = 0 \;\Rightarrow\; y$ não depende de x

Quanto mais próximo de 1 for o valor de r^2, melhor a regressão

Qualidade da regressão

Coeficiente de correlação

$$r = \frac{\langle xy\rangle - \langle x\rangle\langle y\rangle}{\sqrt{\left(\langle x^2\rangle - \langle x\rangle^2\right)\left(\langle y^2\rangle - \langle y\rangle^2\right)}}$$

- Dados correlacionados:
  $y = f(x) = a_0 + a_1 x \;\Rightarrow\; r = \mathrm{sinal}(a_1)$
  - $r > 0 \;\rightarrow$ dados correlacionados
  - $r < 0 \;\rightarrow$ dados anti-correlacionados
- $r = 0 \;\Rightarrow$ y não depende de x

Quanto mais próximo de 1 for o valor de r^2, melhor a regressão

Qualidade da regressão

Coeficiente de correlação

$$r = \frac{\langle xy\rangle - \langle x\rangle\langle y\rangle}{\sqrt{\left(\langle x^2\rangle - \langle x\rangle^2\right)\left(\langle y^2\rangle - \langle y\rangle^2\right)}}$$

- Dados correlacionados:
  $y = f(x) = a_0 + a_1 x \;\Rightarrow\; r = \text{sinal}(a_1)$
  - $r > 0 \;\rightarrow$ dados correlacionados
  - $r < 0 \;\rightarrow$ dados anti-correlacionados
- $r = 0 \;\Rightarrow\;$ y não depende de x

Quanto mais próximo de 1 for o valor de r^2, melhor a regressão

Qualidade da regressão

Coeficiente de correlação

$$r = \frac{\langle xy\rangle - \langle x\rangle\langle y\rangle}{\sqrt{\left(\langle x^2\rangle - \langle x\rangle^2\right)\left(\langle y^2\rangle - \langle y\rangle^2\right)}}$$

- Dados correlacionados:
  $y = f(x) = a_0 + a_1 x \;\Rightarrow\; r = \text{sinal}(a_1)$
  - $r > 0 \;\rightarrow$ dados correlacionados
  - $r < 0 \;\rightarrow$ dados anti-correlacionados
- $r = 0 \;\Rightarrow$ y não depende de x

Quanto mais próximo de 1 for o valor de r^2, melhor a regressão

Regressão linear: teoria geral

- No exemplo anterior, consideramos a função

$$f(x) = a_0 + a_1 x$$

- O termo linear **NÃO** se refere à variável x
- Para a regressão linear o importante é a linearidade de a_0 e a_1
- Em geral, podemos fazer uma regressão linear da função

$$f(x) = \sum_k a_k g_k(x)$$

onde $g_k(x)$ pode ser qualquer.

Regressão linear: teoria geral

- No exemplo anterior, consideramos a função

$$f(x) = a_0 + a_1 x$$

- O termo linear **NÃO** se refere à variável x
- Para a regressão linear o importante é a linearidade de a_0 e a_1
- Em geral, podemos fazer uma regressão linear da função

$$f(x) = \sum_k a_k g_k(x)$$

onde $g_k(x)$ pode ser qualquer.

Regressão linear: teoria geral

- No exemplo anterior, consideramos a função

$$f(x) = a_0 + a_1 x$$

- O termo linear **NÃO** se refere à variável x
- Para a regressão linear o importante é a linearidade de a_0 e a_1
- Em geral, podemos fazer uma regressão linear da função

$$f(x) = \sum_k a_k g_k(x)$$

onde $g_k(x)$ pode ser qualquer.

Regressão linear: teoria geral

$f(x) = \sum_k a_k g_k(x)$

Minimizar $\chi^2 = \sum_{i=1}^{N} [(\sum_k a_k g_k(x_i)) - y_i]^2$

$$\frac{\partial \chi^2}{\partial a_j} = 2\left[\sum_k a_k \left(\sum_{i=1}^{N} g_k(x_i) g_j(x_i)\right) - \left(\sum_{i=1}^{N} g_j(x_i) y_i\right)\right] = 0$$

Definindo $G_{kj} = \sum_{i=1}^{N} g_k(x_i) g_j(x_i)$ e $Y_j = \sum_{i=1}^{N} g_j(x_i) y_i$ temos:

$$\sum_k a_k G_{kj} = Y_j$$

Sistema linear $\Rightarrow$ solução analítica

Linearização

Algumas funções não-lineares podem ser linearizadas

1. Exponencial – $f(x) = A\exp(Bx)$
   Transformação: $g(x) = \ln f(x) = Bx + \ln A = a_1 x + a_0$
   Regressão linear de $\{x_i, \ln(y_i)\}$
2. Lei de potência – $f(x) = Ax^B$
   Transformação:
   $g(x) = \ln f(x) = B\ln(x) + \ln A = a_1 \ln(x) + a_0$
   Regressão linear de $\{\ln(x_i), \ln(y_i)\}$
3. Logaritmo – $f(x) = A\ln(Bx)$
   Notando que $f(x) = A\ln(x) + A\ln B = a_1 \ln(x) + a_0$
   Regressão linear de $\{\ln(x_i), y_i\}$

Linearização

Algumas funções não-lineares podem ser linearizadas

1. Exponencial – $f(x) = A\exp(Bx)$
   Transformação: $g(x) = \ln f(x) = Bx + \ln A = a_1 x + a_0$
   Regressão linear de $\{x_i, \ln(y_i)\}$
2. Lei de potência – $f(x) = Ax^B$
   Transformação:
   $g(x) = \ln f(x) = B\ln(x) + \ln A = a_1 \ln(x) + a_0$
   Regressão linear de $\{\ln(x_i), \ln(y_i)\}$
3. Logaritmo – $f(x) = A\ln(Bx)$
   Notando que $f(x) = A\ln(x) + A\ln B = a_1 \ln(x) + a_0$
   Regressão linear de $\{\ln(x_i), y_i\}$

Linearização

Algumas funções não-lineares podem ser linearizadas

1. Exponencial – $f(x) = A\exp(Bx)$
   Transformação: $g(x) = \ln f(x) = Bx + \ln A = a_1 x + a_0$
   Regressão linear de $\{x_i, \ln(y_i)\}$
2. Lei de potência – $f(x) = Ax^B$
   Transformação:
   $g(x) = \ln f(x) = B\ln(x) + \ln A = a_1 \ln(x) + a_0$
   Regressão linear de $\{\ln(x_i), \ln(y_i)\}$
3. Logaritmo – $f(x) = A\ln(Bx)$
   Notando que $f(x) = A\ln(x) + A\ln B = a_1 \ln(x) + a_0$
   Regressão linear de $\{\ln(x_i), y_i\}$

Linearização

Algumas funções não-lineares podem ser linearizadas

1. Exponencial – $f(x) = A\exp(Bx)$
   Transformação: $g(x) = \ln f(x) = Bx + \ln A = a_1 x + a_0$
   Regressão linear de $\{x_i, \ln(y_i)\}$
2. Lei de potência – $f(x) = Ax^B$
   Transformação:
   $g(x) = \ln f(x) = B\ln(x) + \ln A = a_1 \ln(x) + a_0$
   Regressão linear de $\{\ln(x_i), \ln(y_i)\}$
3. Logaritmo – $f(x) = A\ln(Bx)$
   Notando que $f(x) = A\ln(x) + A\ln B = a_1 \ln(x) + a_0$
   Regressão linear de $\{\ln(x_i), y_i\}$

Linearização

Algumas funções não-lineares podem ser linearizadas

1. Exponencial – $f(x) = A\exp(Bx)$
   Transformação: $g(x) = \ln f(x) = Bx + \ln A = a_1 x + a_0$
   Regressão linear de $\{x_i, \ln(y_i)\}$
2. Lei de potência – $f(x) = Ax^B$
   Transformação:
   $g(x) = \ln f(x) = B\ln(x) + \ln A = a_1 \ln(x) + a_0$
   Regressão linear de $\{\ln(x_i), \ln(y_i)\}$
3. Logaritmo – $f(x) = A\ln(Bx)$
   Notando que $f(x) = A\ln(x) + A\ln B = a_1 \ln(x) + a_0$
   Regressão linear de $\{\ln(x_i), y_i\}$

Linearização

Algumas funções não-lineares podem ser linearizadas

1. Exponencial – $f(x) = A\exp(Bx)$
   Transformação: $g(x) = \ln f(x) = Bx + \ln A = a_1 x + a_0$
   Regressão linear de $\{x_i, \ln(y_i)\}$
2. Lei de potência – $f(x) = Ax^B$
   Transformação:
   $g(x) = \ln f(x) = B\ln(x) + \ln A = a_1 \ln(x) + a_0$
   Regressão linear de $\{\ln(x_i), \ln(y_i)\}$
3. Logaritmo – $f(x) = A\ln(Bx)$
   Notando que $f(x) = A\ln(x) + A\ln B = a_1 \ln(x) + a_0$
   Regressão linear de $\{\ln(x_i), y_i\}$

Linearização

Algumas funções não-lineares podem ser linearizadas

1. Exponencial – $f(x) = A\exp(Bx)$
   Transformação: $g(x) = \ln f(x) = Bx + \ln A = a_1 x + a_0$
   Regressão linear de $\{x_i, \ln(y_i)\}$
2. Lei de potência – $f(x) = Ax^B$
   Transformação:
   $g(x) = \ln f(x) = B\ln(x) + \ln A = a_1 \ln(x) + a_0$
   Regressão linear de $\{\ln(x_i), \ln(y_i)\}$
3. Logaritmo – $f(x) = A\ln(Bx)$
   Notando que $f(x) = A\ln(x) + A\ln B = a_1 \ln(x) + a_0$
   Regressão linear de $\{\ln(x_i), y_i\}$

Linearização

Algumas funções não-lineares podem ser linearizadas

1. Exponencial – $f(x) = A\exp(Bx)$
   Transformação: $g(x) = \ln f(x) = Bx + \ln A = a_1 x + a_0$
   Regressão linear de $\{x_i, \ln(y_i)\}$
2. Lei de potência – $f(x) = Ax^B$
   Transformação:
   $g(x) = \ln f(x) = B\ln(x) + \ln A = a_1\ln(x) + a_0$
   Regressão linear de $\{\ln(x_i), \ln(y_i)\}$
3. Logaritmo – $f(x) = A\ln(Bx)$
   Notando que $f(x) = A\ln(x) + A\ln B = a_1\ln(x) + a_0$
   Regressão linear de $\{\ln(x_i), y_i\}$

Linearização

Algumas funções não-lineares podem ser linearizadas

1. Exponencial – $f(x) = A\exp(Bx)$
   Transformação: $g(x) = \ln f(x) = Bx + \ln A = a_1 x + a_0$
   Regressão linear de $\{x_i, \ln(y_i)\}$
2. Lei de potência – $f(x) = Ax^B$
   Transformação:
   $g(x) = \ln f(x) = B\ln(x) + \ln A = a_1 \ln(x) + a_0$
   Regressão linear de $\{\ln(x_i), \ln(y_i)\}$
3. Logaritmo – $f(x) = A\ln(Bx)$
   Notando que $f(x) = A\ln(x) + A\ln B = a_1 \ln(x) + a_0$
   Regressão linear de $\{\ln(x_i), y_i\}$

Regressão não-linear

- Para regressões lineares, a minimização de χ^2 levou a um sistema de equações lineares
- Portanto, uma solução analítica foi possível
- No caso da regressão não-linear, isto não é possível
- Contudo, se olharmos para χ^2 no espaço de parâmetros $\{a_k\}$, o problema se resume a encontrar o mínimo $\longrightarrow$ solução iterativa

Regressão não-linear

- Para regressões lineares, a minimização de χ^2 levou a um sistema de equações lineares
- Portanto, uma solução analítica foi possível
- No caso da regressão não-linear, isto não é possível
- Contudo, se olharmos para χ^2 no espaço de parâmetros $\{a_k\}$, o problema se resume a encontrar o mínimo $\longrightarrow$ solução iterativa

Regressão não-linear

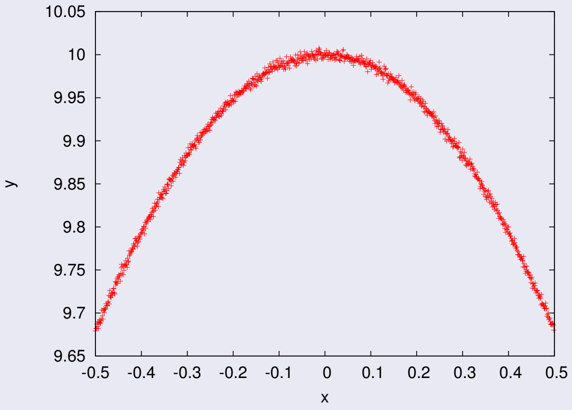
- Para regressões lineares, a minimização de χ^2 levou a um sistema de equações lineares
- Portanto, uma solução analítica foi possível
- No caso da regressão não-linear, isto não é possível
- Contudo, se olharmos para χ^2 no espaço de parâmetros $\{a_k\}$, o problema se resume a encontrar o mínimo $\longrightarrow$ solução iterativa

Regressão não-linear

- Para regressões lineares, a minimização de χ^2 levou a um sistema de equações lineares
- Portanto, uma solução analítica foi possível
- No caso da regressão não-linear, isto não é possível
- Contudo, se olharmos para χ^2 no espaço de parâmetros $\{a_k\}$, o problema se resume a encontrar o mínimo $\longrightarrow$ solução iterativa

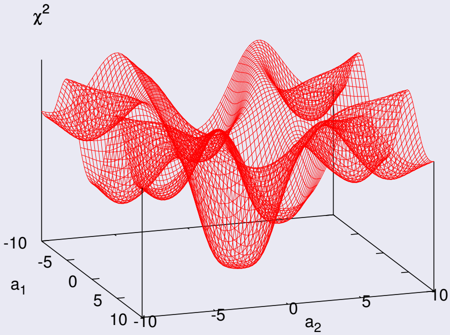
Múltiplos mínimos

Exemplo: $f(x) = 10\cos^2(a_1 x)\cos^2(a_2 x)$



Múltiplos mínimos

Exemplo: $f(x) = 10\cos^2(a_1 x)\cos^2(a_2 x)$



Como encontrar o mínimo?

O problema

Encontrar o mínimo de

$$\chi^2(a_1, a_2, \ldots, a_s) = \sum_{k=1}^{N} [F_k(a_1, a_2, \ldots, a_s)]$$

Métodos

Todos os métodos são **iterativos**

1. Steepest descent
2. Algoritmo de Gauss-Newton
3. Algoritmo de Levenberg-Marquardt

Como encontrar o mínimo?

O problema

Encontrar o mínimo de

$$\chi^2(a_1, a_2, \ldots, a_s) = \sum_{k=1}^{N} [F_k(a_1, a_2, \ldots, a_s)]$$

Métodos

Todos os métodos são **iterativos**

1. Steepest descent
2. Algoritmo de Gauss-Newton
3. Algoritmo de Levenberg-Marquardt

Como encontrar o mínimo?

O problema

Encontrar o mínimo de

$$\chi^2(a_1, a_2, \ldots, a_s) = \sum_{k=1}^{N} [F_k(a_1, a_2, \ldots, a_s)]$$

Métodos

Todos os métodos são **iterativos**

1. Steepest descent
2. Algoritmo de Gauss-Newton
3. Algoritmo de Levenberg-Marquardt

Como encontrar o mínimo?

O problema

Encontrar o mínimo de

$$\chi^2(a_1, a_2, \ldots, a_s) = \sum_{k=1}^{N} [F_k(a_1, a_2, \ldots, a_s)]$$

Métodos

Todos os métodos são **iterativos**

1. Steepest descent
2. Algoritmo de Gauss-Newton
3. Algoritmo de Levenberg-Marquardt

Como encontrar o mínimo?

O problema

Encontrar o mínimo de

$$\chi^2(a_1, a_2, \ldots, a_s) = \sum_{k=1}^{N} [F_k(a_1, a_2, \ldots, a_s)]$$

Métodos

Todos os métodos são **iterativos**

1. Steepest descent
2. Algoritmo de Gauss-Newton
3. Algoritmo de Levenberg-Marquardt

Steepest descent

- Intuitivamente é o método mais simples
- A partir de um ponto inicial (no espaço de parâmetros), se aproxima do mínimo local tomando passos proporcionais ao negativo do gradiente
-
- Portanto, $\vec{a}_{n+1} = \vec{a}_n - \lambda \nabla F$, com $\lambda > 0$ suficentemente pequeno, se aproxima sucessivamente do mínimo local

Steepest descent

- Intuitivamente é o método mais simples
- A partir de um ponto inicial (no espaço de parâmetros), se aproxima do mínimo local tomando passos proporcionais ao negativo do gradiente
-
- Portanto, $\vec{a}_{n+1} = \vec{a}_n - \lambda \nabla F$, com $\lambda > 0$ suficentemente pequeno, se aproxima sucessivamente do mínimo local

Steepest descent

- Intuitivamente é o método mais simples
- A partir de um ponto inicial (no espaço de parâmetros), se aproxima do mínimo local tomando passos proporcionais ao negativo do gradiente
- O gradiente diz a direção em que a função cresce mais rápido
- Portanto, $\vec{a}_{n+1} = \vec{a}_n - \lambda \nabla F$, com $\lambda > 0$ suficentemente pequeno, se aproxima sucessivamente do mínimo local

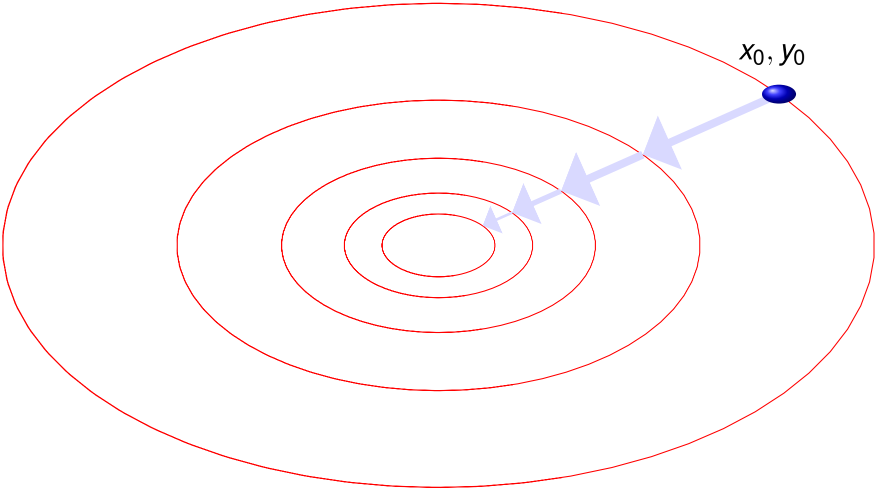
Steepest descent

- Intuitivamente é o método mais simples
- A partir de um ponto inicial (no espaço de parâmetros), se aproxima do mínimo local tomando passos proporcionais ao negativo do gradiente
- O negativo do gradiente diz a direção em que a função decresce mais rápido
- Portanto, $\vec{a}_{n+1} = \vec{a}_n - \lambda \nabla F$, com $\lambda > 0$ suficentemente pequeno, se aproxima sucessivamente do mínimo local

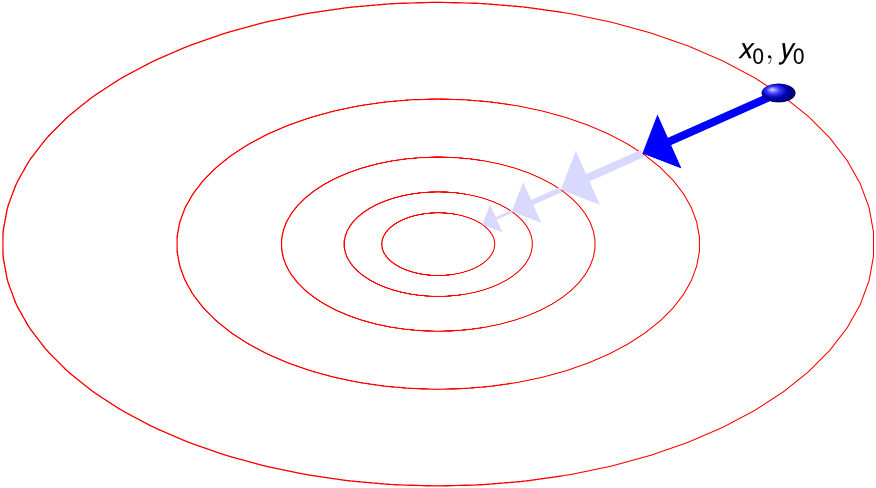
Steepest descent

- Intuitivamente é o método mais simples
- A partir de um ponto inicial (no espaço de parâmetros), se aproxima do mínimo local tomando passos proporcionais ao negativo do gradiente
- O negativo do gradiente diz a direção em que a função decresce mais rápido
- Portanto, $\vec{a}_{n+1} = \vec{a}_n - \lambda \nabla F$, com $\lambda > 0$ suficentemente pequeno, se aproxima sucessivamente do mínimo local

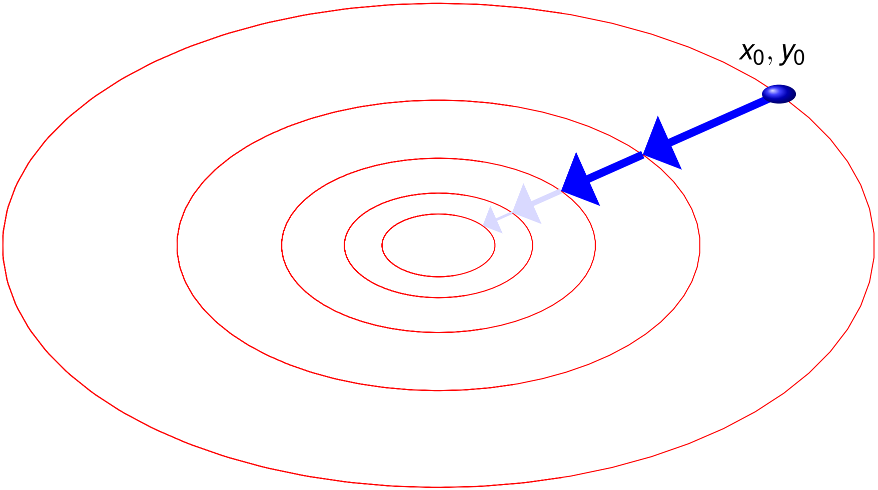
Steepest descent



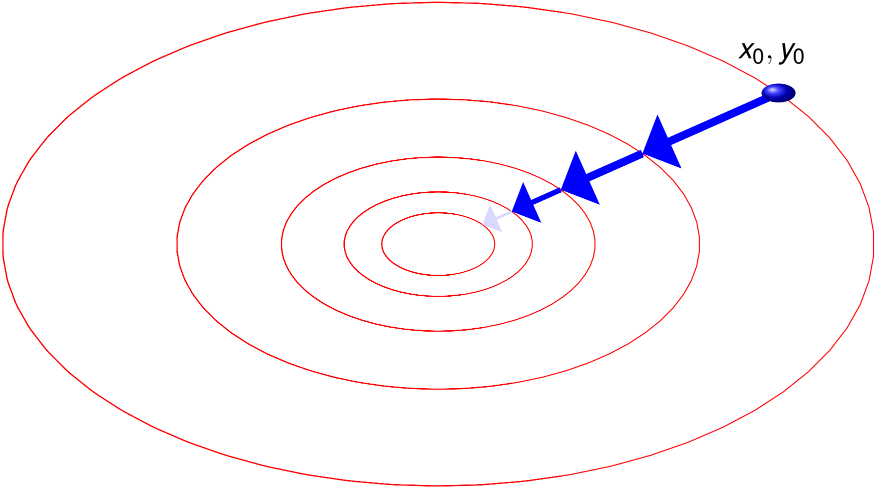
Steepest descent



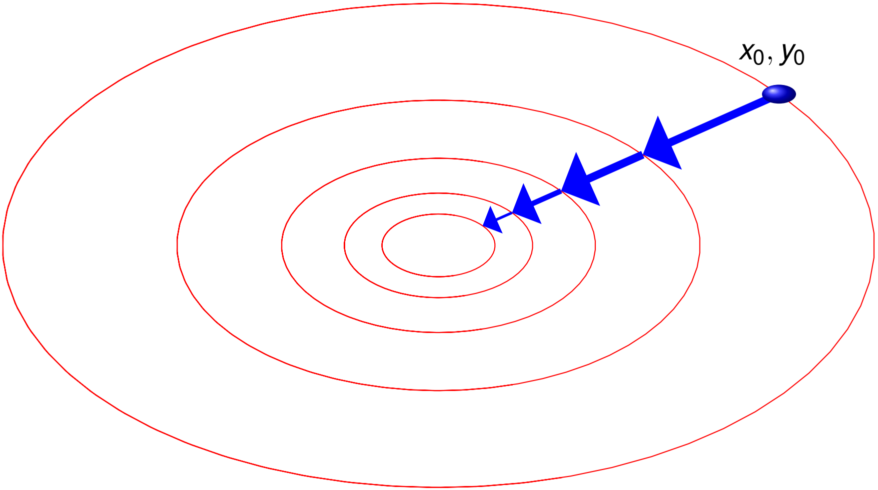
Steepest descent



Steepest descent



Steepest descent



Problemas do steepest descent

- O algoritmo pode levar muitas iterações para convergir para o mínimo local se as curvaturas forem diferentes nas diferentes direções
- Só funciona se começarmos próximo ao mínimo

Problemas do steepest descent

- O algoritmo pode levar muitas iterações para convergir para o mínimo local se as curvaturas forem diferentes nas diferentes direções
- Só funciona se começarmos próximo ao mínimo

Algoritmo de Gauss-Newton

- Para achar o mínimo, seria conveniente dar passos grandes quando o gradiente é pequeno e passos pequenos quando o gradiente é grande
- O método do steepest descent faz o oposto
- Uma possível solução é usar também a curvatura (segunda derivada) para determinar o passo
- Para tanto, notamos que no ponto de mínimo $\nabla f(\vec{a}) = 0$. Logo, podemos escrever

$$\nabla f(\vec{a}) = \nabla f(\vec{a}_0) + (\vec{a} - \vec{a}_0)\nabla^2 f(\vec{a}_0) \Rightarrow \vec{a} = \vec{a}_0 - \left[\nabla^2 f(\vec{a}_0)\right]^{-1} \nabla f(\vec{a}_0)$$

- Que, iterativamente, se torna

$$\vec{a}_{n+1} = \vec{a}_n - \left[\nabla^2 f(\vec{a}_n)\right]^{-1} \nabla f(\vec{a}_n)$$

Algoritmo de Gauss-Newton

- Para achar o mínimo, seria conveniente dar passos grandes quando o gradiente é pequeno e passos pequenos quando o gradiente é grande
- O método do steepest descent faz o oposto
- Uma possível solução é usar também a curvatura (segunda derivada) para determinar o passo
- Para tanto, notamos que no ponto de mínimo $\nabla f(\vec{a}) = 0$. Logo, podemos escrever

$$\nabla f(\vec{a}) = \nabla f(\vec{a}_0) + (\vec{a} - \vec{a}_0)\nabla^2 f(\vec{a}_0) \;\Rightarrow\; \vec{a} = \vec{a}_0 - \left[\nabla^2 f(\vec{a}_0)\right]^{-1} \nabla f(\vec{a}_0)$$

- Que, iterativamente, se torna

$$\vec{a}_{n+1} = \vec{a}_n - \left[\nabla^2 f(\vec{a}_n)\right]^{-1} \nabla f(\vec{a}_n)$$

Algoritmo de Gauss-Newton

- Para achar o mínimo, seria conveniente dar passos grandes quando o gradiente é pequeno e passos pequenos quando o gradiente é grande
- O método do steepest descent faz o oposto
- Uma possível solução é usar também a curvatura (segunda derivada) para determinar o passo
- Para tanto, notamos que no ponto de mínimo $\nabla f(\vec{a}) = 0$. Logo, podemos escrever

$$\nabla f(\vec{a}) = \nabla f(\vec{a}_0) + (\vec{a} - \vec{a}_0)\nabla^2 f(\vec{a}_0) \Rightarrow \vec{a} = \vec{a}_0 - \left[\nabla^2 f(\vec{a}_0)\right]^{-1} \nabla f(\vec{a}_0)$$

- Que, iterativamente, se torna

$$\vec{a}_{n+1} = \vec{a}_n - \left[\nabla^2 f(\vec{a}_n)\right]^{-1} \nabla f(\vec{a}_n)$$

Algoritmo de Gauss-Newton

- Para achar o mínimo, seria conveniente dar passos grandes quando o gradiente é pequeno e passos pequenos quando o gradiente é grande
- O método do steepest descent faz o oposto
- Uma possível solução é usar também a curvatura (segunda derivada) para determinar o passo
- Para tanto, notamos que no ponto de mínimo $\nabla f(\vec{a}) = 0$. Logo, podemos escrever

$$\nabla f(\vec{a}) = \nabla f(\vec{a}_0) + (\vec{a} - \vec{a}_0)\nabla^2 f(\vec{a}_0) \;\Rightarrow\; \vec{a} = \vec{a}_0 - \left[\nabla^2 f(\vec{a}_0)\right]^{-1} \nabla f(\vec{a}_0)$$

- Que, iterativamente, se torna

$$\vec{a}_{n+1} = \vec{a}_n - \left[\nabla^2 f(\vec{a}_n)\right]^{-1} \nabla f(\vec{a}_n)$$

Algoritmo de Gauss-Newton

- Para achar o mínimo, seria conveniente dar passos grandes quando o gradiente é pequeno e passos pequenos quando o gradiente é grande
- O método do steepest descent faz o oposto
- Uma possível solução é usar também a curvatura (segunda derivada) para determinar o passo
- Para tanto, notamos que no ponto de mínimo $\nabla f(\vec{a}) = 0$. Logo, podemos escrever

$$\nabla f(\vec{a}) = \nabla f(\vec{a}_0) + (\vec{a} - \vec{a}_0) \nabla^2 f(\vec{a}_0) \;\Rightarrow\; \vec{a} = \vec{a}_0 - \left[\nabla^2 f(\vec{a}_0)\right]^{-1} \nabla f(\vec{a}_0)$$

- Que, iterativamente, se torna

$$\vec{a}_{n+1} = \vec{a}_n - \left[\nabla^2 f(\vec{a}_n)\right]^{-1} \nabla f(\vec{a}_n)$$

Algoritmo de Gauss-Newton

- Para achar o mínimo, seria conveniente dar passos grandes quando o gradiente é pequeno e passos pequenos quando o gradiente é grande
- O método do steepest descent faz o oposto
- Uma possível solução é usar também a curvatura (segunda derivada) para determinar o passo
- Para tanto, notamos que no ponto de mínimo $\nabla f(\vec{a}) = 0$. Logo, podemos escrever

$$\nabla f(\vec{a}) = \nabla f(\vec{a}_0) + (\vec{a} - \vec{a}_0)\nabla^2 f(\vec{a}_0) \;\Rightarrow\; \vec{a} = \vec{a}_0 - \left[\nabla^2 f(\vec{a}_0)\right]^{-1} \nabla f(\vec{a}_0)$$

- Que, iterativamente, se torna

$$\vec{a}_{n+1} = \vec{a}_n - \left[\nabla^2 f(\vec{a}_n)\right]^{-1} \nabla f(\vec{a}_n)$$

Algoritmo de Gauss-Newton

- O método possui uma convergência rápida
- Contudo, a taxa de convergência depende sensivelmente do ponto de partida
- Torna-se particularmente lento se $\nabla^2 f$ for grande

Algoritmo de Gauss-Newton

- O método possui uma convergência rápida
- Contudo, a taxa de convergência depende sensivelmente do ponto de partida
- Torna-se particularmente lento se $\nabla^2 f$ for grande

Algoritmo de Gauss-Newton

- O método possui uma convergência rápida
- Contudo, a taxa de convergência depende sensivelmente do ponto de partida
- Torna-se particularmente lento se $\nabla^2 f$ for grande

Algoritmo de Levenberg-Marquardt

- Os algoritmos do steepest descent e de Gauss-Newton são, de certa forma, complementares em suas vantagens
- O algoritmo de **Levenberg-Marquadt** é uma **mistura** dos dois

Algoritmo de Levenberg-Marquardt

- Os algoritmos do steepest descent e de Gauss-Newton são, de certa forma, complementares em suas vantagens
- O algoritmo de **Levenberg-Marquadt** é uma **mistura** dos dois

Algoritmo de Levenberg

1. Faça uma iteração usando a seguinte regra

$$\vec{a}_{n+1} = \vec{a}_n - \left[\nabla^2 f(\vec{a}_n) + \lambda \mathrm{I}\right]^{-1} \nabla f(\vec{a}_n)$$

2. Calcule o erro com o novo vetor de parâmetros
3. Se o erro aumentar, despreze o novo vetor de parâmetros e aumente λ por um certo fator (normalmente 10)
4. Se o erro diminuir, aceite o novo vetor e divida λ pelo mesmo fator
5. Repita o processo até obter convergência

Algoritmo de Levenberg

1. Faça uma iteração usando a seguinte regra

$$\vec{a}_{n+1} = \vec{a}_n - \left[\nabla^2 f(\vec{a}_n) + \lambda \mathrm{I}\right]^{-1} \nabla f(\vec{a}_n)$$

2. Calcule o erro com o novo vetor de parâmetros
3. Se o erro aumentar, despreze o novo vetor de parâmetros e aumente λ por um certo fator (normalmente 10)
4. Se o erro diminuir, aceite o novo vetor e divida λ pelo mesmo fator
5. Repita o processo até obter convergência

Algoritmo de Levenberg

1. Faça uma iteração usando a seguinte regra

$$\vec{a}_{n+1} = \vec{a}_n - \left[\nabla^2 f(\vec{a}_n) + \lambda \mathrm{I}\right]^{-1} \nabla f(\vec{a}_n)$$

2. Calcule o erro com o novo vetor de parâmetros
3. Se o erro aumentar, despreze o novo vetor de parâmetros e aumente λ por um certo fator (normalmente 10)
4. Se o erro diminuir, aceite o novo vetor e divida λ pelo mesmo fator
5. Repita o processo até obter convergência

Algoritmo de Levenberg

1. Faça uma iteração usando a seguinte regra

$$\vec{a}_{n+1} = \vec{a}_n - \left[\nabla^2 f(\vec{a}_n) + \lambda \mathrm{I}\right]^{-1} \nabla f(\vec{a}_n)$$

2. Calcule o erro com o novo vetor de parâmetros
3. Se o erro aumentar, despreze o novo vetor de parâmetros e aumente λ por um certo fator (normalmente 10)
4. Se o erro diminuir, aceite o novo vetor e divida λ pelo mesmo fator
5. Repita o processo até obter convergência

Algoritmo de Levenberg

1. Faça uma iteração usando a seguinte regra

$$\vec{a}_{n+1} = \vec{a}_n - \left[\nabla^2 f(\vec{a}_n) + \lambda \mathbf{I}\right]^{-1} \nabla f(\vec{a}_n)$$

2. Calcule o erro com o novo vetor de parâmetros
3. Se o erro aumentar, despreze o novo vetor de parâmetros e aumente λ por um certo fator (normalmente 10)
4. Se o erro diminuir, aceite o novo vetor e divida λ pelo mesmo fator
5. Repita o processo até obter convergência

Algoritmo de Levenberg-Marquardt

- Para λ grande, o algoritmo anterior, despreza a curvatura
- É interessante, no entanto, dar passos maiores na direção em que a curvatura é menor
- Marquardt propôs, então, alterar a regra de iteração para

$$\vec{a}_{n+1} = \vec{a}_n - \left[\nabla^2 f(\vec{a}_n) + \lambda \mathrm{diag}(\nabla^2 f(\vec{a}_n))\right]^{-1} \nabla f(\vec{a}_n)$$

Algoritmo de Levenberg-Marquardt

- Para λ grande, o algoritmo anterior, despreza a curvatura
- É interessante, no entanto, dar passos maiores na direção em que a curvatura é menor
- Marquardt propôs, então, alterar a regra de iteração para

$$\vec{a}_{n+1} = \vec{a}_n - \left[\nabla^2 f(\vec{a}_n) + \lambda \mathrm{diag}(\nabla^2 f(\vec{a}_n))\right]^{-1} \nabla f(\vec{a}_n)$$

Algoritmo de Levenberg-Marquardt

- Para λ grande, o algoritmo anterior, despreza a curvatura
- É interessante, no entanto, dar passos maiores na direção em que a curvatura é menor
- Marquardt propôs, então, alterar a regra de iteração para

$$\vec{a}_{n+1} = \vec{a}_n - \left[\nabla^2 f(\vec{a}_n) + \lambda \mathrm{diag}(\nabla^2 f(\vec{a}_n))\right]^{-1} \nabla f(\vec{a}_n)$$

Algoritmo de Levenberg-Marquardt

- O algoritmo de Levenberg-Marquardt não é um algoritmo ótimo
- É heurístico
- Contudo, funciona muito bem na prática
- É o algoritmo mais usado
- Várias implementações estão disponíveis na internet

Algoritmo de Levenberg-Marquardt

- O algoritmo de Levenberg-Marquardt não é um algoritmo ótimo
- É heurístico
- Contudo, funciona muito bem na prática
- É o algoritmo mais usado
- Várias implementações estão disponíveis na internet

Algoritmo de Levenberg-Marquardt

- O algoritmo de Levenberg-Marquardt não é um algoritmo ótimo
- É heurístico
- Contudo, funciona muito bem na prática
- É o algoritmo mais usado
- Várias implementações estão disponíveis na internet

Algoritmo de Levenberg-Marquardt

- O algoritmo de Levenberg-Marquardt não é um algoritmo ótimo
- É heurístico
- Contudo, funciona muito bem na prática
- É o algoritmo mais usado
- Várias implementações estão disponíveis na internet

Algoritmo de Levenberg-Marquardt

- O algoritmo de Levenberg-Marquardt não é um algoritmo ótimo
- É heurístico
- Contudo, funciona muito bem na prática
- É o algoritmo mais usado
- Várias implementações estão disponíveis na internet

Resolução de sistema tridiagonal

$$\begin{pmatrix} b_1 & c_1 & 0 & \cdots & & & & \\ a_2 & b_2 & c_2 & 0 & \cdots & & & \\ 0 & a_3 & b_3 & c_3 & 0 & \cdots & & \\ & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\ & & & \cdots & a_{n-2} & b_{n-2} & c_{n-2} & 0 \\ & & & & \cdots & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & & \cdots & a_n & b_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_{n-2} \\ r_{n-1} \\ r_n \end{pmatrix}$$

Resolução de sistema tridiagonal

$$
\begin{pmatrix}
b_1 & c_1 & 0 & \cdots & & & & \\
a_2 & b_2 & c_2 & 0 & \cdots & & & \\
0 & a_3 & b_3 & c_3 & 0 & \cdots & & \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\
 & & & \cdots & a_{n-2} & b_{n-2} & c_{n-2} & 0 \\
 & & & & \cdots & a_{n-1} & b_{n-1} & c_{n-1} \\
 & & & & & \cdots & a_n & b_n
\end{pmatrix}
\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix}
=
\begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_{n-2} \\ r_{n-1} \\ r_n \end{pmatrix}
$$

multiplicando primeira linha por $-\frac{a_2}{b_1}$ e somando à segunda

Resolução de sistema tridiagonal

$$\begin{pmatrix} \beta_1 & c_1 & 0 & \cdots & & & & \\ 0 & \beta_2 & c_2 & 0 & \cdots & & & \\ 0 & a_3 & b_3 & c_3 & 0 & \cdots & & \\ & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\ & & & \cdots & a_{n-2} & b_{n-2} & c_{n-2} & 0 \\ & & & & \cdots & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & & \cdots & a_n & b_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} \rho_1 \\ \rho_2 \\ r_3 \\ \vdots \\ r_{n-2} \\ r_{n-1} \\ r_n \end{pmatrix}$$

definindo $\beta_1 = b_1,\ \beta_2 = b_2 - \frac{a_2}{\beta_1} c_1,\ \rho_1 = r_1$ e $\rho_2 = r_2 - \frac{a_2}{\beta_1}\rho_1$

Resolução de sistema tridiagonal

$$\begin{pmatrix} \beta_1 & c_1 & 0 & \cdots & & & & \\ 0 & \beta_2 & c_2 & 0 & \cdots & & & \\ 0 & a_3 & b_3 & c_3 & 0 & \cdots & & \\ & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\ & & & \cdots & a_{n-2} & b_{n-2} & c_{n-2} & 0 \\ & & & & \cdots & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & & \cdots & a_n & b_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} \rho_1 \\ \rho_2 \\ r_3 \\ \vdots \\ r_{n-2} \\ r_{n-1} \\ r_n \end{pmatrix}$$

multiplicando segunda linha por $-\frac{a_3}{\beta_2}$ e somando à terceira

Resolução de sistema tridiagonal

$$\begin{pmatrix} \beta_1 & c_1 & 0 & \cdots & & & & \\ 0 & \beta_2 & c_2 & 0 & \cdots & & & \\ 0 & 0 & \beta_3 & c_3 & 0 & \cdots & & \\ & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\ & & & \cdots & a_{n-2} & b_{n-2} & c_{n-2} & 0 \\ & & & & \cdots & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & & \cdots & a_n & b_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \vdots \\ r_{n-2} \\ r_{n-1} \\ r_n \end{pmatrix}$$

definindo $\beta_3 = b_3 - \frac{a_3}{\beta_2} c_2$ e $\rho_3 = r_3 - \frac{a_3}{\beta_2} \rho_2$

Resolução de sistema tridiagonal

$$\begin{pmatrix} \beta_1 & c_1 & 0 & \cdots & & & & \\ 0 & \beta_2 & c_2 & 0 & \cdots & & & \\ 0 & 0 & \beta_3 & c_3 & 0 & \cdots & & \\ & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\ & & & \cdots & 0 & \beta_{n-2} & c_{n-2} & 0 \\ & & & & \cdots & 0 & \beta_{n-1} & c_{n-1} \\ & & & & & \cdots & 0 & \beta_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \vdots \\ \rho_{n-2} \\ \rho_{n-1} \\ \rho_n \end{pmatrix}$$

procedendo recursivamente e definindo

$\beta_n = b_n - \frac{a_n}{\beta_{n-1}} c_{n-1}$ e $\rho_n = r_n - \frac{a_n}{\beta_{n-1}} \rho_{n-1}$

Resolução de sistema tridiagonal

$$\begin{pmatrix} \beta_1 & c_1 & 0 & \cdots & & & & \\ 0 & \beta_2 & c_2 & 0 & \cdots & & & \\ 0 & 0 & \beta_3 & c_3 & 0 & \cdots & & \\ & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\ & & & \cdots & 0 & \beta_{n-2} & c_{n-2} & 0 \\ & & & & \cdots & 0 & \beta_{n-1} & c_{n-1} \\ & & & & & \cdots & 0 & \beta_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \vdots \\ \rho_{n-2} \\ \rho_{n-1} \\ \rho_n \end{pmatrix}$$

agora, da última linha, temos que De volta ao spline cúbico

$$\boxed{x_n = \frac{\rho_n}{\beta_n} \text{ e } x_{n-j} = \frac{\rho_{n-j} - c_{n-j} x_{n-j+1}}{\beta_{n-j}}}$$