Métodos Numéricos

Prof. Dr. Jonatha Costa

2024

MN aplicados à Engenharia

- Apresentar conteúdo de Ajuste de Curvas
  - Interpolação e extrapolação
  - Regressão Linear por Mínimos Quadrados
  - Linearização de Equações não lineares
  - Polinômio de Lagrange e de Newton
  - Spline linear, quadrática e cúbica

Organização

1. Ajustes de curvas
   - Conceitos
   - Regressão Linear - mínimos quadrados
   - Linearização
   - Regressão Polinomial
   - Polinômio interpolador de Lagrange
   - Polinômio interpolador de Newton
   - Splines: introdução
   - Splines Linear
   - Splines Quadráticas
   - Splines Cúbicas

MN aplicados à Engenharia

Introdução

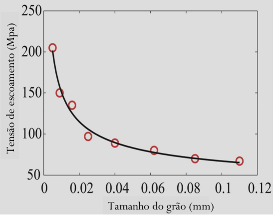
- Muitas observações de engenharia são feitas em experimentos nos quais grandezas físicas são medidas e gravadas;
- São chamados de dados ou pontos experimentais;
- Os dados armazenados são *discretos*, podendo perder informação do sinal original;
- As curvas obtidas são representadas por uma equação específica com parâmetros que representem da melhor forma possível o conjunto de dados;
- O procedimento de ajuste de curvas também é utilizado para determinar os valores dos parâmetros (coeficientes) nas equações. Esse procedimento pode ser feito com muitas funções diferentes e com polinômios de várias ordens.

MN aplicados à Engenharia

Definição

- Interpolação: procedimento utilizado para encontrar valores entre pontos medidos.
- Extrapolação: procedimento utilizado para *predizer* valores **além** do intervalo no qual foram medidos.
- Ajuste de curvas: procedimento no qual uma fórmula matemática é usada para produzir uma curva que melhor represente um conjunto de dados.

Figura: Tamanho de grão × tensão



Fonte: GILAT,(2008)

$$\sigma = Cd^m$$

Em que:

- σ - Tensão de escoamento
- C - é uma constante
- d - é o tamanho do grão
- m - é o grau da equação

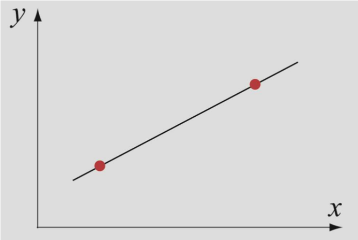
MN aplicados à Engenharia

Ajuste de curvas – Polinômio de 1º grau:

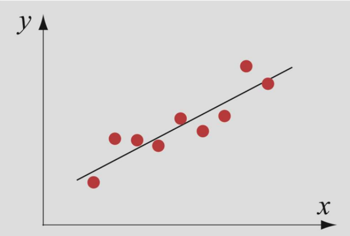
- Procedimento mais simples de se encontrar uma curva;
- É definido por $y = a_1 x + a_0$ e feito com a determinação das constantes a_1 e a_0, que fornecem o menor erro quando os pontos medidos são substituídos em y.

Figura: Curvas

(a) Dois pontos



(b) Vários pontos



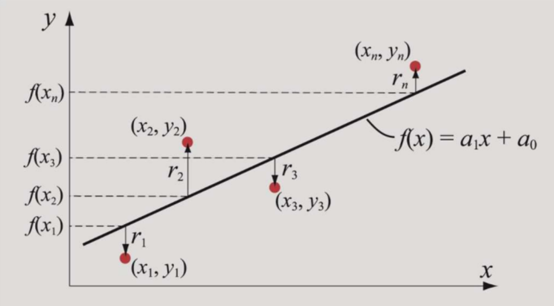
Fonte: GILAT,(2008)

MN aplicados à Engenharia

Melhor ajuste de curvas – Polinômio de 1º grau:

- O quanto uma curva está representando de forma aproximada um conjunto de dados;
- É preciso utilizar um número que quantifique esta aproximação. No caso, utiliza-se o erro ou resíduo;
- O erro é a diferença entre cada ponto pertencente ao conjunto de dados e o valor da função aproximada;
- O melhor ajuste é definido como o **menor** *erro total* calculado.

Figura: Polinômio de grau 1



Fonte: GILAT,(2008)

- Resíduo ou erro em cada ponto:

$$r_i = y_i - f(x)$$

- Erro global

$$E = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} \left[y_i - (a_1 x_i + a_0)\right]^2$$

* Com essa definição, o erro global é sempre um número positivo, visto que os resíduos positivos e negativos não promoverão interferência mútua.

Organização

MN aplicados à Engenharia

Regressão Linear pelo método dos Mínimos Quadrados

- Utilizado para determinar os coeficientes a_0 e a_1 de tal forma que a função represente o melhor ajuste do conjunto de dados.
- O melhor ajuste é obtido quando o erro global, definido pela equação a seguir, é mínimo:

$$E = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} \Big[y_i - (a_i x_i + a_0) \Big]^2$$

- Derivando o erro E em função dos coeficientes a_0 e a_1, tem-se:

$$\frac{\partial E}{\partial a_0} = -2 \sum_{i=1}^{n} \Big[y_i - a_i x_i - a_0 \Big] = 0$$

$$\frac{\partial E}{\partial a_1} = -2 \sum_{i=1}^{n} \Big[y_i - a_i x_i - a_0 \Big] x_i = 0$$

MN aplicados à Engenharia

Regressão Linear pelo método dos Mínimos Quadrados

- Resolvendo as derivadas, tem-se que:
  - * $a_1 = \dfrac{n\sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$
  - * $a_0 = \dfrac{\left(\sum_{i=1}^{n} x_i^2\right)\left(\sum_{i=1}^{n} y_i\right) - \left(\sum_{i=1}^{n} x_i y_i\right)\left(\sum_{i=1}^{n} x_i\right)}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$
- Por conveniência pode-se fazer:
  - * $S_x = \sum_{i=1}^{n} x_i, S_y = \sum_{i=1}^{n} y_i, S_{xy} = \sum_{i=1}^{n} x_i y_i, S_{xx} = \sum_{i=1}^{n} x_i^2$, portanto,
  - * $a_1 = \dfrac{nS_{xy} - S_x S_y}{nS_{xx} - (S_x)^2}$, $a_0 = \dfrac{S_{xy} S_y - S_{xy} S_x}{nS_{xx} - (S_x)^2}$

MN aplicados à Engenharia

Figura: *Script em Python* de Regressão Linear no Octave/Matlab

```
% AJUSTE DE CURVAS - REGRESSÃO LINEAR
clc; clear all;

x=0:10:100;
y=[ 0.94 0.96 1.0 1.05 1.07 1.09 ...
 1.14 1.17 1.21 1.24 1.28];
%*******************************************
if (length(x)!=length(y))
  disp('Falha! X e Y tem dimensões diferentes!')
else
  n=length(x);
  Sx=sum(x);      Sy=sum(y);
  Sxy=sum(x.*y);  Sxx=sum(x.^2);
  a1=(n*Sxy-Sx*Sy)/(n*Sxx-Sx^2);
  a0=(Sxx*Sy-Sxy*Sx)/(n*Sxx-Sx^2);
end
p=[a1 a0];y2=polyval(p,x);
%*******************************************
plot(x,y,'*r',x,y2)
legend('Medições','Regressão Linear')
```

Fonte: Autor,(2020)

MN aplicados à Engenharia

Figura: *Script* de Regressão Linear no Python

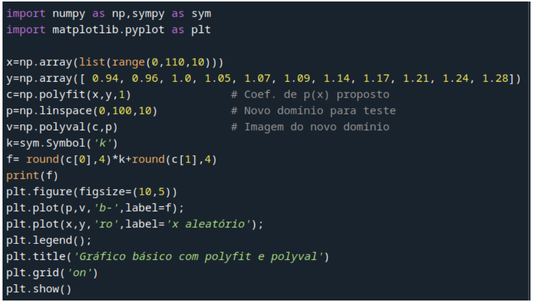
```
import numpy as np,sympy as sym
import matplotlib.pyplot as plt

x=np.array(list(range(0,110,10)))
y=np.array([ 0.94, 0.96, 1.0, 1.05, 1.07, 1.09, 1.14, 1.17, 1.21, 1.24, 1.28])
c=np.polyfit(x,y,1)                # Coef. de p(x) proposto
p=np.linspace(0,100,10)            # Novo domínio para teste
v=np.polyval(c,p)                  # Imagem do novo domínio
k=sym.Symbol('k')
f= round(c[0],4)*k+round(c[1],4)
print(f)
plt.plot(p,v,'b-',label=f);
plt.plot(x,y,'ro',label='x aleatório');
plt.legend();
plt.title('Gráfico básico com polyfit e polyval')
plt.show()
```

Fonte: Autor,(2024)

MN aplicados à Engenharia

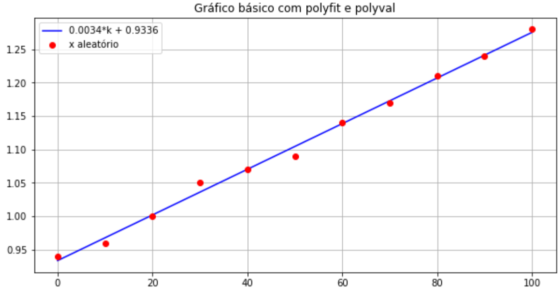
Figura: *Script* e Gráfico de Regressão Linear

(a) *Script*



```python
import numpy as np,sympy as sym
import matplotlib.pyplot as plt

x=np.array(list(range(0,110,10)))
y=np.array([ 0.94, 0.96, 1.0, 1.05, 1.07, 1.09, 1.14, 1.17, 1.21, 1.24, 1.28])
c=np.polyfit(x,y,1)              # Coef. de p(x) proposto
p=np.linspace(0,100,10)          # Novo domínio para teste
v=np.polyval(c,p)                # Imagem do novo domínio
k=sym.Symbol('k')
f= round(c[0],4)*k+round(c[1],4)
print(f)
plt.figure(figsize=(10,5))
plt.plot(p,v,'b-',label=f);
plt.plot(x,y,'ro',label='x aleatório');
plt.legend();
plt.title('Gráfico básico com polyfit e polyval')
plt.grid('on')
plt.show()
```

(b) Gráfico



Fonte: Autor,(2024)

Organização

MN aplicados à Engenharia

Linearização de Equações

- Muitas situações na ciência e na engenharia mostram que a relação entre as grandezas envolvidas **não é linear**;
- Entretanto, funções não-lineares que podem ser escritas em uma forma tal que possibilite a determinação dos coeficientes que levam ao melhor ajuste com o emprego do método da regressão linear por mínimos quadrados;
  - $y = b \cdot x^m$
- Para que a regressão linear possa ser utilizada, a equação não-linear de duas variáveis deve ser modificada de tal forma que a nova equação seja linear com termos contendo as variáveis originais. Dessa forma, para a equação acima tem-se:
  - $y = b \cdot x^m \rightarrow ln(y) = ln(b \cdot x^m) = m \cdot ln(x) + ln(b)$,
  - equação linear em termos de $ln(x)$ tal que a equação está na forma $Y = a_1 X + a_0$, em que $Y = ln(y)$, $a_1 = m$,$X = ln(x)$ e $a_0 = ln(b)$.
  - Isso significa que uma regressão linear por mínimos quadrados pode ser usada para fazer com que uma equação na forma $y = b \cdot x^m$ se ajuste a um conjunto de pontos x_i, y_i.

MN aplicados à Engenharia

Linearização de Equações Não-lineares

Figura: Equações não lineares e formas lineares

Equação não-linear	Forma linear	Relação com $Y = a_1X + a_0$	Valores para a regressão linear por mínimos quadrados	Gráficos onde os dados medidos parecem se ajustar a uma linha reta
$y = bx^m$	$\ln(y) = m\ln(x) + \ln(b)$	$Y = \ln(y)$, $X = \ln(x)$ $a_1 = m$, $a_0 = \ln(b)$	$\ln(x_i)$ e $\ln(y_i)$	Gráfico y vs. x em eixos x e y logarítmicos. Gráfico $\ln(y)$ vs. $\ln(x)$ em eixos x e y lineares.
$y = be^{mx}$	$\ln(y) = mx + \ln(b)$	$Y = \ln(y)$, $X = x$ $a_1 = m$, $a_0 = \ln(b)$	x_i e $\ln(y_i)$	Gráfico y vs. x em eixos x linear e y logarítmico. Gráfico $\ln(y)$ vs. x em eixos x e y lineares.
$y = b10^{mx}$	$\log(y) = mx + \log(b)$	$Y = \log(y)$, $X = x$ $a_1 = m$, $a_0 = \log(b)$	x_i e $\log(y_i)$	Gráfico y vs. x em eixos x linear e y logarítmico. Gráfico $\log(y)$ vs. x em eixos x e y lineares.
$y = \frac{1}{mx + b}$	$\frac{1}{y} = mx + b$	$Y = \frac{1}{y}$, $X = x$ $a_1 = m$, $a_0 = b$	x_i e $1/y_i$	Gráfico $1/y$ vs. x em eixos x e y lineares.
$y = \frac{mx}{b + x}$	$\frac{1}{y} = \frac{b}{m}\frac{1}{x} + \frac{1}{m}$	$Y = \frac{1}{y}$, $X = \frac{1}{x}$ $a_1 = \frac{b}{m}$, $a_0 = \frac{1}{m}$	$1/x_i$ e $1/y_i$	Gráfico $1/y$ vs. $1/x$ em eixos x e y lineares.

Fonte: GILAT,(2008)

MN aplicados à Engenharia

Linearização de Equações

Considerações importantes com relação à escolha da função não-linear adequada para o ajuste de uma curva são as seguintes:

- Traçado dos pontos medidos de uma maneira específica, verificando se esses pontos parecem formar uma linha reta;
- Funções exponenciais não podem passar pela origem;
- Funções exponenciais só são capazes de fazer o ajuste de dados nos quais todos os valores de y são positivos ou negativos;
- Funções logarítmicas não podem incluir $x = 0$ ou valores negativos de x;
- Para função de potência $y = 0$ quando $x = 0$;
- A equação inversa não pode incluir $y = 0$.

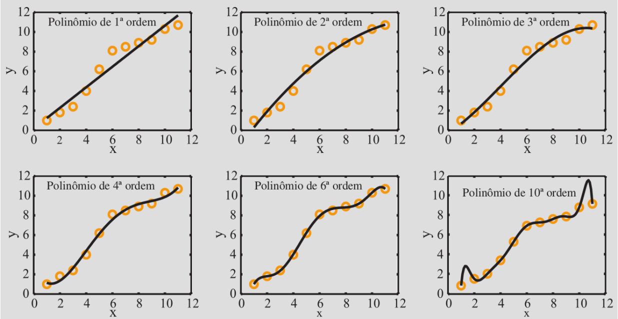
Organização

MN aplicados à Engenharia

Ajuste de curvas com polinômios de ordem superior:

- Polinômio de 1ª ordem é uma reta, 2ª ordem é uma parábola e 3ª ordem gera um ponto de inflexão em uma curva;
- Quanto maior a ordem do polinômio, maior a flexibilidade da curva;
- Concluir *qual dos polinômios fornece o melhor ajuste* depende do tipo e da origem dos dados, de sua aplicação, e do propósito do ajuste;
- Observe a discrepância na região central e marginal em cada gráfico da figura.

Figura: *Polinômio de grau 'n'*



Fonte: GILAT,(2008)

MN aplicados à Engenharia

Ajuste de curvas com polinômios quadráticos e de ordem superior:

Considerações

- É possível encontrar um polinômio que passe por n pontos, desde que a ordem do polinômio seja $(n-1)$;
- Se os dados não são precisos, não se faz necessário utilizar um polinômio de ordem elevada;
- Embora um polinômio de ordem elevada forneça os valores exatos em todos os pontos, muitas vezes ele apresenta um desvio significativo entre alguns pontos;
- Neste caso, o polinômio de ordem elevada não pode ser usado de forma confiável para a interpolação e extrapolação dos dados.

MN aplicados à Engenharia

Regressão polinomial

- Seja o polinômio de ordem m, definido por:

$$f(x) = a_m x^m + a_{m-1} x^{m-1} + \ldots + a_1 x + a_0$$

- Para um conjunto de n pontos (m é igual a $n-1$), tem-se que o erro E é dado por:

$$E = \sum_{i=1}^{n} \left[y_i - (a_m x_i^m + a_{m-1} x_i^{m-1} + \ldots + a_1 x_i + a_0) \right]^2$$

- Calculando-se as derivadas parciais em relação aos coeficientes $a_0, a_1, \ldots, a_m$, encontra-se um sistema com n equações e m incógnitas.
- A solução do sistema de equações fornece os valores dos coeficientes $a_0, a_1, \ldots, a_n$ do polinômio que melhor se ajusta aos n pontos (x_i, y_i).

MN aplicados à Engenharia

Considere $m = 2$ como sendo a ordem do polinômio. Tem-se que o erro E é dado por:

$$E = \sum_{i=1}^{n} \left[y_i - (a_2 x_i^2 + a_1 x_i + a_0) \right]^2$$

- Calculando-se as derivadas parciais em relação aos coeficientes $a_0, a_1, \ldots, a_m$ encontra-se um sistema com n equações e m incógnitas.
- Derivando-se o erro E em função dos coeficientes a_0, a_1 e a_2, tem-se:

$$\frac{\partial E}{\partial a_0} = -2 \sum_{i=1}^{n} [y_i - (a_2 x_i^2 + a_1 x_i + a_0)] = 0$$

$$\frac{\partial E}{\partial a_1} = -2 \sum_{i=1}^{n} [y_i - (a_2 x_i^2 + a_1 x_i + a_0) x_i] = 0$$

$$\frac{\partial E}{\partial a_2} = -2 \sum_{i=1}^{n} [y_i - (a_2 x_i^2 + a_1 x_i + a_0) x_i^2] = 0$$

MN aplicados à Engenharia

Resolvendo-se as derivadas, tem-se que:

$$na_0 + \left(\sum_{i=1}^{n} x_i\right)a_1 + \left(\sum_{i=1}^{n} x_i^2\right)a_2 = \sum_{i=1}^{n} y_i$$

$$\left(\sum_{i=1}^{n} x_i\right)a_0 + \left(\sum_{i=1}^{n} x_i^2\right)a_1 + \left(\sum_{i=1}^{n} x_i^3\right)a_2 = \sum_{i=1}^{n} x_i y_i$$

$$\left(\sum_{i=1}^{n} x_i^2\right)a_0 + \left(\sum_{i=1}^{n} x_i^3\right)a_1 + \left(\sum_{i=1}^{n} x_i^4\right)a_2 = \sum_{i=1}^{n} x_i^2 y_i$$

- A solução do sistema de equações fornece os valores dos coeficientes a_0, a_1 e a_2 do polinômio $y = a_2x_i^2 + a_1x_i + a_0$ que melhor se ajusta aos n pontos (x_i, y_i).
- Note que coeficientes de **polinômios de ordem superior** são deduzidos utilizando-se o mesmo procedimento acima apresentado.

MN aplicados à Engenharia

Estruturando-se na forma matricial tem-se:

- $$\begin{bmatrix} n & \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^3 \\ \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^4 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} x_i^2 y_i \end{bmatrix}$$
- Portanto, um sistema do tipo $A \cdot x = B$, tal que $x = A^{-1} \cdot B$.
- *Para fins de programação* perceba:
  - Um padrão nos elementos da matrizes A e B;
  - A disposição dos coeficientes $[a_0, a_1, a_2]$, na ordem do termo independente para o termo de maior grau;
  - A necessidade de inverter a ordem do vetor de coeficientes a fim de compor: $p(x) = a_m x^m + a_{m-1} x^{m-1} + \ldots + a_1 x + a_0$ ou $p(x) = a_0 + a_1 x + \ldots + a_{m-1} x^{m-1} + a_m x^m$.

MN aplicados à Engenharia

Figura: Exemplo de *script* de Regressão Polinomial no Octave/Matlab

```
% REGRESSÃO POLINOMIAL
clc; clear all;   format short

x=1:0.25:5; y=(x.^2-2*x-3);                      % Função de referência
m=4;                                             % Grau de p(x)

n=length(x);
for(i=1:2*m) v(i)=sum(x.^(i));    end            % Termos de potências x num só vetor
v=[n v];A=[];                                    % Inclusão do termo n
for(i=1:m+1) A=[A; v(i:(m+i))];  end             % Matriz A
for(i=1:m) u(i)=sum(x.^(i).*y);  end             % Termos de potências xy num só vetor
B=[sum(y) u]';                                   % Inclusão do termo yi na Matriz B
sol=(A\B)';                                        % Saída [a0 a1 a2 ... an]
p=sol(end:-1:1);                                   % Ordenando [an ... a0]
y2=polyval(p,x);                                 % Y do p(x)
px=polyout(p,'x')

plot(x,y,'or',x,y2)
legend(px);
```

Fonte: Autor,(2020)

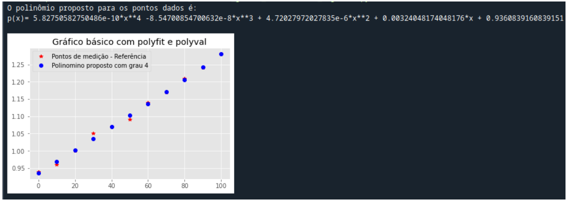
MN aplicados à Engenharia

Figura: *Script* e Gráfico de Regressão Polinomial

(a) *Script*

```python
import numpy as np
import matplotlib.pyplot as plt
from fun import pol
x=np.array(list(range(0,110,10)))
y=np.array([ 0.94, 0.96, 1.0, 1.05, 1.07, 1.09, 1.14, 1.17, 1.21, 1.24, 1.28])
m=4                                # Grau do polinômio

#%%==========================================================================
# # Regressão Polinomial - Soluções via comandos polyval e polyfit
# ==========================================================================
c=np.polyfit(x,y,m)                # Coef. de p(x) proposto
v=np.polyval(c,x)                  # Imagem do novo domínio
print(f'O polinômio proposto para os pontos dados é:\np(x)= {pol(c)}')
plt.plot(x,y,'r*',label='Pontos de medição - Referência')
plt.plot(x,v,'bo',label=f'Polinomino proposto com grau {m}')
plt.title('Gráfico básico com polyfit e polyval')
plt.style.use('ggplot');plt.grid(visible='on');
plt.legend();plt.show()
```

(b) Gráfico para polinômino de grau 4



Fonte: Autor,(2024)

Organização

MN aplicados à Engenharia

Polinômio de Lagrange

- A solução de um ajuste de curvas envolve com polinômio **padrão**. Procedimento:
  1. Definir o polinômio cujo formato padrão é: $f(x) = a_m x^m + a_{m-1} x^{m-1} + \ldots + a_1 x + a_0$;
  2. Estruturar o sistema na forma matricial $(Ax = B)$, para obter $x = A^{-1}B$;
  3. Verificar se o grau do polinômio encontrado converge nos pontos de análise e região de vizinhança sem *saltos* entre pontos;

  Nota: Ineficiente para polinômios de **ordem mais elevada** e quando a matriz dos coeficientes é mal condicionada.
- Os *polinômios interpoladores de Lagrange*
  1. Formam uma classe específica de polinômios que podem ser utilizados para fazer o ajuste de um determinado conjunto de dados simplesmente a partir dos valores dos pontos;
  2. Os polinômios podem ser escritos diretamente e os coeficientes são determinados sem a necessidade de nenhum cálculo preliminar.

MN aplicados à Engenharia

Polinômio de Lagrange

- Para dois pontos $a = (x_1, y_1)$ e $b = (x_2, y_2)$, o polinômio de Lagrange tem a seguinte forma:
  - $f(x) = y = a_1(x - x_2) + a_2(x - x_1)$, o qual para os pontos a e b resultado em:
  * $y_1 = a_1(x_1 - x_2) + a_2(x_1 - x_1) \rightarrow a_1 = \dfrac{y_1}{(x_1 - x_2)}$
  * $y_2 = a_1(x_2 - x_2) + a_2(x_2 - x_1) \rightarrow a_2 = \dfrac{y_2}{(x_2 - x_1)}$

  Recompondo a equação, temos:
  - $f(x) = \dfrac{(x - x_2)}{(x_1 - x_2)} y_1 + \dfrac{(x - x_1)}{(x_2 - x_1)} y_2$

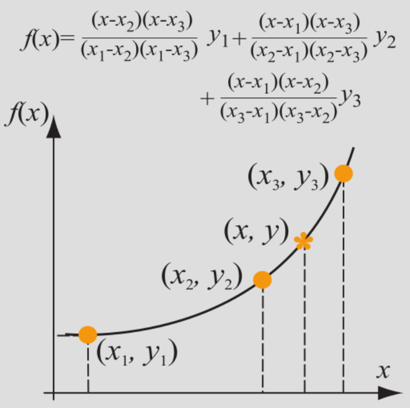
MN aplicados à Engenharia

Polinômio de Lagrange

- Para três pontos $(x_1, y_1), (x_2, y_2)$ e (x_3, y_3) o polinômio de Lagrange tem a seguinte forma:

$$f(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}y_1 + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}y_2 + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}y_3$$

Figura: Lagrange de 2ª ordem



Fonte: GILAT,(2008)

MN aplicados à Engenharia

Polinômio de Lagrange

- Escrito de forma compacta utilizando-se a notação de soma e produto o polinômio fica:

$$f(x) = \sum_{i=1}^{n} y_i L_i(x) = \sum_{i=1}^{n} y_i \prod_{j=1,\ j\neq i}^{n} \frac{(x - x_j)}{(x_i - x_j)},$$

  em que as funções $L_i(x) = \displaystyle\prod_{j=1,\ j\neq i}^{n} \frac{(x - x_j)}{(x_i - x_j)}$ são chamadas de funções de Lagrange.

- **Considerações**
  - Os pontos não precisam estar equidistantes em si;
  - Deve-se calcular a expressão completa do polinômio interpolador para cada valor de x a ser interpolado;
  - Uma vez ampliado o conjunto de dados, todos os termos do polinômio de Lagrange devem ser calculados novamente.

MN aplicados à Engenharia

Figura: Exemplo de *script* de interpolação por Lagrange no Octave/Matlab

```
% INTERPOLAÇÃO LAGRANGE : f(x) = soma( Y*L(x) )
clc; clear all; pkg load symbolic

x=[1 2 4 5 7]; y=[52 5 -5 -40 10];
m=3;
n=length(x);tic;
for(i=1:n) L(i)=1;                    % Função de Lagrage
  for(j=1:n)
    if (i~=j)
      L(i)= L(i)*(m -x(j)) / ( x(i)-x(j));
    end
  end
end

clc;
Yint=(sum(y.*L))      % f(x) = soma(Y*L(x))
fprintf('Tempo de processamento: %.5fs\n',toc);
```

Fonte: Autor,(2020)

MN aplicados à Engenharia

Figura: Exemplo de *script* de interpolação por Lagrange no Python

```
import numpy as np
# ==========================================================================
x=np.array([1,2,4,5,7])
y=np.array([52,5,-5,-40,10])
#%% OPÇÃO 01===============================================================
p=3
k=np.ones(len(x))                           # vetor de uns para 1ª elemento
for i in range(len(x)):
    for j in range(len(x)):
        if (i!=j):
            k[i]=k[i]*(p-x[j])/(x[i]-x[j])
Yint=sum(y*k)
print(Yint)
```

Fonte: Autor,(2024)

Organização

MN aplicados à Engenharia

Polinômio de Newton

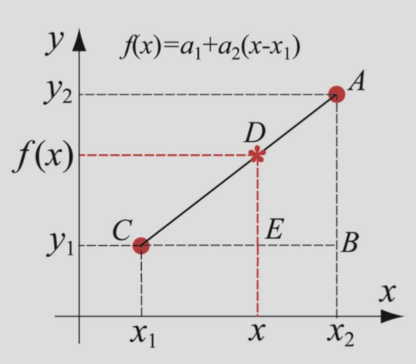
- Método alternativo ao polinômio interpolador de Lagrange
- Nos *polinômios interpoladores de Newton*
  - A determinação dos coeficientes não requer a solução de um sistema com n equações;
  - Os pontos do conjunto de dados não precisam estar ordenados de forma ascendente ou descendente;
  - Os coeficientes a_1 a a_n, uma vez determinados, podem ser utilizados para interpolar quaisquer dos pontos que compõem o conjunto de dados;
  - Após a determinação dos n coeficientes de um polinômio interpolador de Newton de ordem $n-1$, mais pontos podem ser adicionados ao conjunto de dados, sendo necessário apenas determinar os coeficientes adicionais;
  - A forma geral do polinômio de Newton de ordem $n-1$ que passa por n pontos é:
    $f(x) = a_1 + a_2(x-x_1) + a_3(x-x_1)(x-x_2) + \ldots + a_n(x-x_1)(x-x_2)\ldots(x-x_{n-1})$

MN aplicados à Engenharia

Polinômio de Newton de primeira ordem

$$f(x) = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2) + \ldots + a_n(x - x_1)(x - x_2)\ldots(x - x_{n-1})$$

Figura: Polinômio de Newton



Fonte: GILAT,(2008)

Para dois pontos (x_1, y_1) e (x_2, y_2) o polinômio de Newton tem a seguinte forma:

$$f(x) = a_1 + a_2(x - x_1)$$

Os coeficientes a_1 e a_2 podem ser calculados utilizando-se semelhança de triângulos, conforme figura ao lado.

Desse modo: $f(x) = y1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$, portanto, $a_1 = y_1$ e $a_2 = \dfrac{y_2 - y_1}{x_2 - x_1}$.

MN aplicados à Engenharia

Polinômio de Newton de ordem 2 e 3

- Para três pontos $(x_1, y_1), (x_2, y_2)$ e (x_3, y_3) o polinômio de Newton tem a seguinte forma:

$$f(x) = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2)$$

- Os valores de a_1 e a_2 são os mesmos obtidos para dois pontos. $a_1 = y_1$ e $a_2 = \dfrac{y_2 - y_1}{x_2 - x_1}$

  Através de manipulações matemáticas, tem-se:

- O termo adicional para um polinômio de ordem 2 seria: $a_3 = \dfrac{\frac{y_3-y_2}{x_3-x_2} - \frac{y_2-y_1}{x_2-x_1}}{x_3 - x_1}$
- O termo adicional para um polinômio de ordem 3 seria: $a_4 = \dfrac{\frac{y_3-y_2}{x_3-x_2} - \frac{y_2-y_1}{x_2-x_1}}{x_4 - x_1}$

MN aplicados à Engenharia

Polinômio de Newton - Ordem superior

Para encontrar os coeficientes maiores que a_3, utiliza-se o conceito das **diferenças divididas** $(\Delta x/\Delta y)$.

- $f[x_2, x_1] = \frac{y_2 - y_1}{x_2 - x_1} = a_2$
- $f[x_3, x_2, x_1] = \frac{f[x_3, x_2] - f[x_2, x_1]}{x_3 - x_1} = \frac{\left(\frac{y_3 - y_2}{x_3 - x_2}\right) - \left(\frac{y_2 - y_1}{x_2 - x_1}\right)}{x_3 - x_1} = a_3$
- $f[x_4, x_3, x_2, x_1] = \frac{f[x_4, x_3, x_2] - f[x_3, x_2, x_1]}{x_4 - x_1} = \frac{\frac{f[x_4, x_3] - f[x3, x_2]}{x_4 - x_2} - \frac{f[x_3, x_2] - f[x_2, x_1]}{x_3 - x_1}}{x_4 - x_1} = a_4 \ldots$
- $f[x_5, x_4, x_3, x_2, x_1] = \frac{f[x_5, x_4, x_3, x_2] - f[x_4, x_3, x_2, x_1]}{x_5 - x_1} = a_5 \ldots$
- $f[x_6, x_5, x_4, x_3, x_2, x_1] = \frac{f[x_6, x_5, x_4, x_3, x_2] - f[x_5, x_4, x_3, x_2, x_1]}{x_6 - x_1} = a_6 \ldots$

MN aplicados à Engenharia

Polinômio de Newton - Forma geral

Portanto, a k-ésima *diferença dividida*, ordem 2 ou superior, é dada por:

- $f[x_k, x_{k-1}, x_2, x_1] = \dfrac{f[x_k, x_{k-1}, x_3, x_2] - f[x_{k-1}, x_{k-2}, \ldots, x_2, x_1]}{x_k - x_1}$, em que a ordem máxima é $(n-1)$

- **Forma geral**:

$$f(x) = \begin{cases} y_1 + f[x_2, x_1](x - x_1) + f[x_3, x_2, x_1](x - x_1)(x - x_2) + \ldots \\ \cdots + f[x_n, x_{n-1}, x_2, x_1](x - x_1)(x - x_2)\ldots(x - x_{n-1}) \end{cases}$$

MN aplicados à Engenharia

Figura: Exemplo de *script* de interpolação por Newton no Octave/Matlab

```
% INTERPOLAÇÃO NEWTON :
clc; clear all;  pkg load symbolic

x=[1 2 4 5 7]; y=[52 5 -5 -40 10];
xint=3;
n=length(x);
a(1)=y(1);
for (i=1:n-1)
  divDIF(i,1)=(y(i+1)-y(i))/(x(i+1)-x(i));     %Diferenças na coluna 1
end
for(j=2:n-1)
  for(i=1:n-j)
  divDIF(i,j)=(divDIF(i+1,j-1)-divDIF(i,j-1))/(x(j+i)-x(i));
  end
end
for(j=2:n) a(j)=divDIF(1,j-1);end
Yint=a(1); xn=1;
for(k=2:n) xn=xn*(xint -x(k-1)); Yint=Yint+a(k)*xn; end
Yint
```

Fonte: Adaptado de GILAT,(2008)

MN aplicados à Engenharia

Figura: Exemplo de *script* de interpolação por Newton em Python

```
import numpy as np
x=np.array([1,2,4,5,7])
y=np.array([52,5,-5,-40,10])
print('\014')
xint=3
a,s= [y[0]],y                                   # s é vetor das divisoes
for i in range(len(x)-1):
    y=s                                         # Atualiza o vetor y
    s=(y[1:]-y[:-1])/(x[(1+i):]-x[:-(1+i)])     # Reduz o vetor x
    a.append(s[0])
xn,Yint=1,a[0]
for k in range(1,len(x)):
    xn=xn*(xint - x[k-1])
    Yint=Yint+a[k]*xn
print('\nA aproximação encontrada para f(%.1f) = %.2f'%(xint,Yint))
```

Fonte: Auttor,(2024)

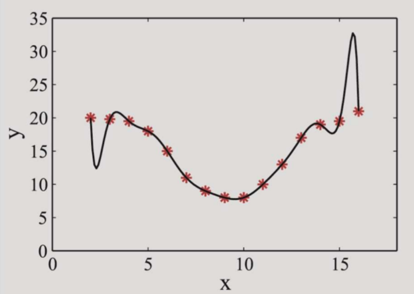
Organização

MN aplicados à Engenharia

Splines: introdução

- Quando um conjunto de dados contendo n pontos é dado e um *único polinômio* é utilizado para fazer a sua interpolação, esse polinômio fornece os valores exatos nos pontos e determina valores estimados (interpolados) entre eles;
- Entretanto, quanto maior o valor de n, a natureza oscilatória dos polinômios pode causar fenômenos indesejáveis, conforme figura a seguir.

Figura: Spline



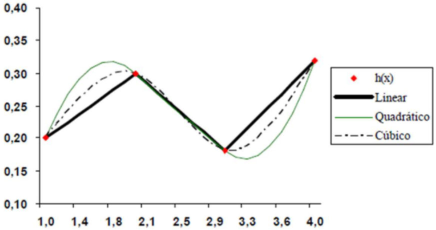
Fonte: GILAT,(2008)

MN aplicados à Engenharia

Splines - Introdução

- Quando se trabalha com um grande número de pontos, uma melhor interpolação pode ser feita com o uso de *muitos polinômios de baixa ordem* ao invés de um único polinômio de ordem elevada;
- Cada polinômio de baixa ordem é válido em um intervalo entre dois ou vários pontos;
- Tipicamente, todos os polinômios utilizados têm a mesma ordem, mas os coeficientes são diferentes em cada intervalo;
- A interpolação feita dessa forma é chamada de interpolação por partes, ou *spline*.
- Os pontos do conjunto de dados em que se encontram os polinômios de intervalos adjacentes são chamados de nós.

Figura: Spline em interpolação



Fonte: GILAT,(2008)

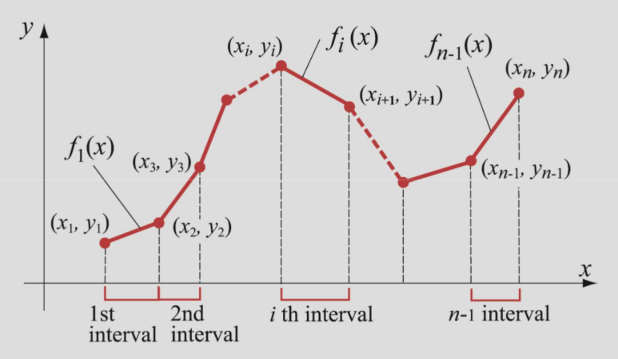
- Os três tipos de interpolação *spline*:
  - Linear
  - Quadrática
  - Cúbica

Organização

MN aplicados à Engenharia

Splines Linear

Figura: Spline Linear



Fonte: GILAT,(2008)

MN aplicados à Engenharia

Splines Linear

- Numa *spline* linear, a interpolação é feita utilizando-se um polinômio de primeira ordem, e os pontos são conectados por linhas retas;
- Utilizando-se o polinômio interpolador de Lagrange, a equação da linha reta que conecta os dois primeiros pontos é dada por:

$$p(x) = s_1(x) = \frac{(x - x_2)}{(x_1 - x_2)} y_1 + \frac{(x - x_1)}{(x_2 - x_1)} y_2$$

- A interpolação no intervalo i, que está entre os pontos x_i e x_{i+1} $(x_i \leq x \leq x_{i+1})$, é feita pela equação da linha reta que conecta o ponto (x_i, y_i) ao ponto (x_{i+1}, y_{i+1}):

$$p(x) = s_i(x) = \frac{(x - x_{i+1})}{(x_i - x_{i+1})} y_i + \frac{(x - x_i)}{(x_{i+1} - x_i)} y_{i+1},$$

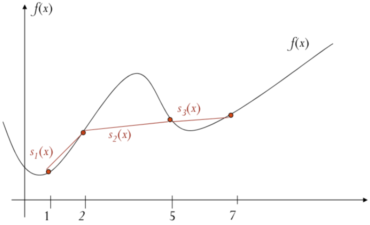
para $i = 1, 2, \ldots, n - 1$

MN aplicados à Engenharia

Splines Linear

- Note que $s_i(x)$ é um polinômio de grau 1 no intervalo;
- $s_1(x)$ é **contínua** em todo intervalo $[x_{i-1}, x_i]$, já que os dois polinômios adjacentes têm o mesmo valor em um nó comum;
- Nos nós $s_i(x_i) = f(x_i)$;
- Entretanto, uma descontinuidade na inclinação das *splines* lineares nos nós está presente.

Figura: Spline Linear



Fonte: DIAS,(2019)

MN aplicados à Engenharia

Exemplo: Determine as *splines* lineares que fazem o ajuste dos dados: $x = [8, 11, 15, 18, 22]; y = [5, 9, 10, 8, 7]$.

- **Os cinco pontos geram quatro *splines*, portanto:**
  - Para $8 \leq x \leq 11$
    $$f_1(x) = \frac{(x - x_2)}{(x_1 - x_2)}y_1 + \frac{(x - x_1)}{(x_2 - x_1)}y_2 = \frac{(x - 11)}{(8 - 11)}5 + \frac{(x - 8)}{(11 - 8)}9 = \frac{5}{-3}(x - 11) + \frac{9}{2}(x - 8)$$
  - Para $11 \leq x \leq 15$
    $$f_2(x) = \frac{(x - x_3)}{(x_2 - x_3)}y_2 + \frac{(x - x_2)}{(x_3 - x_2)}y_3 = \frac{(x - 15)}{(11 - 15)}9 + \frac{(x - 11)}{(15 - 11)}10 = \frac{9}{-4}(x - 15) + \frac{10}{4}(x - 11)$$
  - Para $15 \leq x \leq 18$
    $$f_3(x) = \frac{(x - x_4)}{(x_3 - x_4)}y_3 + \frac{(x - x_3)}{(x_4 - x_3)}y_4 = \frac{(x - 18)}{(15 - 18)}10 + \frac{(x - 15)}{(18 - 15)}8 = \frac{10}{-3}(x - 18) + \frac{8}{3}(x - 15)$$
  - Para $18 \leq x \leq 22$
    $$f_4(x) = \frac{(x - x_5)}{(x_4 - x_5)}y_4 + \frac{(x - x_4)}{(x_5 - x_4)}y_5 = \frac{(x - 22)}{(18 - 22)}8 + \frac{(x - 18)}{(22 - 18)}7 = -2(x - 18) + \frac{7}{4}(x - 22)$$

- **O valor interpolado de y em $x = 12,7$ é obtido com a substituição do valor x na equação de $f_2(x)$ acima resultando em:**
  $$f_2(x) = \frac{9}{-4}(x - 15) + \frac{10}{4}(x - 11) = \frac{9}{-4}(12,7 - 15) + \frac{10}{4}(12,75 - 11) = \mathbf{9{,}425}$$

MN aplicados à Engenharia

Figura: *Script* Spline Linear no Octave/Matlab

```
% Interpolaçao Parcial Linear - SPLINE LINEAR
clc; clear all;  pkg load symbolic

%**************************************************************
x=[8 11 15 18 22]; y=[5 9 10 8 7]; x_int=12.7;

%**************************************************************
n=length(x);                        % Encontrando o intervalo de interpolacao
for(i=1:n)
      if x_int < x(1) | x_int > x(n)
      error('Interpolação fora do intervalo')
      elseif x_int<x(i+1) break  % Fim da busca pelo intervalo
      end
 end
%**************************************************************
syms m x1 x2 y1 y2
clc;
px=simplify((m-x2)*y1/(x1-x2) + (m-x1)*y2/(x2-x1));     % Polinômio
f=matlabFunction(px);                                  % Função com 5 termos
y_int = f( x_int, x(i), x(i+1), y(i), y(i+1));  % Dados de intrada para i
pol=subs(px,'m',x_int)
fprintf('Intervalo de interpolação para x= %d é %d < x < %d.\n\n',
x_int,x(i),x(i+1));
fprintf('O ponto f(%.2f) = %.2f \n\n',x_int,y_int);

%***********************************************************
plot(x_int,y_int,'*'); hold on; plot(x,y,'0');
legend('Pontos x e y');
```

Fonte: AUTOR,(2020)

MN aplicados à Engenharia

Figura: *Script* Spline Linear em Pyhton

```
import matplotlib.pyplot as plt
# ==========================================================================
x=[8,11,15,18]
y=[5,9,10,8]
xint=12.7
# ==========================================================================
print('\014')
# n=len(x)
for i in range(len(x)):
    if (xint < x[0] and xint >x[-1]):
        print('Erro!\nInterpolação fora do intervalo.')
    elif xint<x[i+1]:
        i+=1
        break
yint=(xint-x[i])* y[i-1]/ (x[i-1]-x[i]) + (xint-x[i-1])*y[i]/(x[i]-x[i-1])
print('\nA aproximação encontrada para f(%.1f) = %.2f'%(xint,yint))

plt.plot(x,y,'b',label='Pontos de medição')
plt.plot(xint,yint,'r*',label='y interpolado')
plt.legend()
plt.title('Interpolação via spline linear')
plt.style.use('ggplot')
```

Fonte: AUTOR,(2024)

MN aplicados à Engenharia

Splines Linear

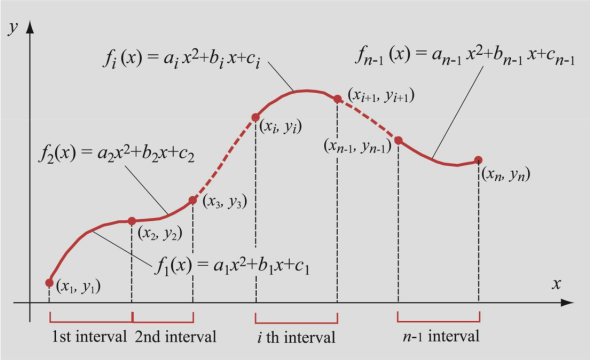
- Uma desvantagem da interpolação linear é que a função de interpolação é geralmente não diferenciável nos pontos extremos de cada intervalo.
- Dessa forma, a função não é suave nos pontos.
- Para uma melhor aproximação de sistemas reais, a característica de suavidade é desejável, logo, a função de interpolação deve ser contínua e diferenciável.
- Proposições de melhor ajuste utilizam função de ordem superior.

Organização

MN aplicados à Engenharia

- Nas *splines* quadráticas, a interpolação é feita com polinômios de **segunda ordem**;
- Em um conjunto de n pontos, há $n-1$ intervalos, portanto, a equação no i-ésimo intervalo, localizado entre os pontos x_i e x_{i+1} é dada por $f_i(x) = a_i x^2 + b_i x + c_i$ para $i = 1, 2, \ldots, n-1$;

Figura: Splines Quadráticas



Fonte: GILAT,(2008)

MN aplicados à Engenharia

Splines Quadráticas

(1) De forma geral, há $n-1$ equações e, como cada equação tem três coeficientes, um total de $3(n-1) = 3n-3$ coeficientes precisam determinados de tal forma que:

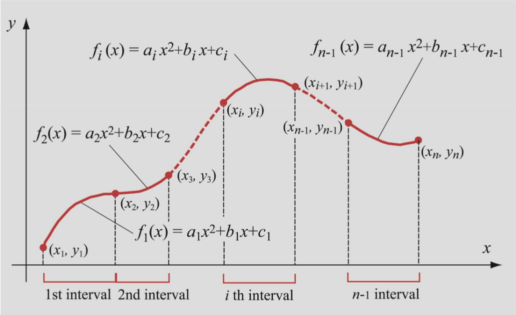
* Cada polinômio $f_i(x)$ deve passar pelos pontos finais do intervalo, (x_i, y_i) e (x_{i+1}, y_{i+1}), o que significa que $f_i(x_i) = y_i$ e $f_i(x_{i+1}) = y_{i+1}$, portanto:

$$a_i x_i^2 + b_i x_i + c_i = y_i$$

$$a_i x_{i+1}^2 + b_i x_{i+1} + c_i = y_{i+1}$$

para $i = 1, 2, \ldots, n-1$.

Figura: Splines Quadráticas



Fonte: GILAT,(2008)

MN aplicados à Engenharia

Splines Quadráticas

(2) Nos nós internos, as inclinações (derivadas primeiras) dos polinômios de intervalos adjacentes são iguais. Isso significa que, com a transição da curva que passa por um nó interno de um polinômio para outro, a inclinação deve ser contínua.
A derivada primeira do i-ésimo polinômio é: $f'(x) = \frac{df}{dx} = 2a_i x + b_i$

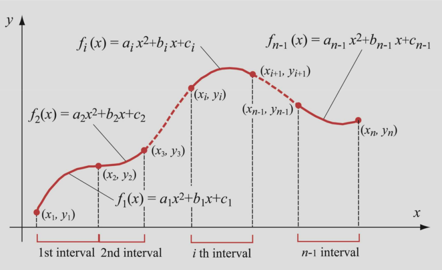
* Para n pontos, o primeiro ponto interno é $i = 2$ e o último é $i = n - 1$. Igualando-se as derivadas primeiras em todos os **pontos internos**, obtém-se:

$$2a_{i-1}x_i + b_{i-1} = 2a_i x_i + b_i$$

para $i = 2, 3, \ldots, n-1$.

* Como há $n - 2$ pontos internos, essa condição fornece $n - 2$ equações, portanto, juntas às equações do passo (1) tem-se um total de $3n - 4$ equações.

Figura: Splines Quadráticas



Fonte: GILAT,(2008)

MN aplicados à Engenharia

Splines Quadráticas

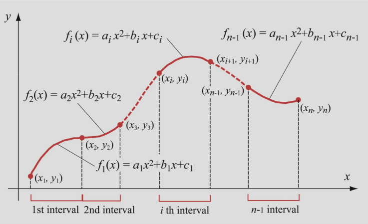
(3) Entretanto, os $n-1$ polinômios têm $3n-3$ coeficientes, de forma que uma equação adicional (condição) é necessária para que os coeficientes sejam obtidos. A condição comumente aplicada assume que a derivada segunda seja nula no **primeiro ou no último ponto**.

- A derivada **segunda** no primeiro ponto, (x_1, y_1), é nula. O polinômio no primeiro intervalo (entre o primeiro e o segundo ponto) é:
  - $f_1(x) = a_1x^2 + b_1x + c_1$
  - A derivada segunda do polinômio é $f_1''(x) = 2a_1$, que, quando igualada a zero, resulta em $a_1 = 0$
  - A interpretação dessa condição é que uma linha reta conecta os dois primeiros pontos.

Considerações

- *Splines* quadráticas têm derivada primeira contínua em pontos internos (nós);
- Em um conjunto de n pontos, elas requerem a solução de um sistema linear com $3n-4$ equações para que os coeficientes dos polinômios sejam determinados.

Figura: Splines Quadráticas



Fonte: GILAT (2008)

MN aplicados à Engenharia

Exemplo: Determine as *splines* quadráticas que fazem o ajuste dos dados: $x = [8, 11, 15, 18, 22]; y = [5, 9, 10, 8, 7]$ e encontre $f(12, 7)$.

- **Os cinco pontos geram quatro *splines***. A equação quadrática para a i-ésima spline é: $f_i(x) = a_i x^2 + b_i x + c_i$
  - Há quatro polinômios, e, como cada polinômio tem três coeficientes, 12 coeficientes têm que ser determinados no total.
  - Os coeficientes são $a_1, b_1, c_1, a_2, b_2, c_2, a_3, b_3, c_3, a_4, b_4 e c_4$.
  - O coeficiente a_1 **é igual a zero** e os outros 11 coeficientes são determinados a partir de um sistema linear de 11 equações.
  - Oito equações são obtidas a partir da condição que diz que, em cada intervalo, o polinômio deve passar pelos pontos finais e, portanto:

$$
\begin{array}{ll}
\text{i=1} & f_1(x) = a_1 x_1^2 + b_1 x_1 + c_1 = (0).8^2 + b_1 8 + c_1 = 5 \\
 & f_1(x) = a_1 x_2^2 + b_1 x_2 + c_1 = (0).11^2 + b_1 11 + c_1 = 9 \\
\text{i=2} & f_2(x) = a_2 x_2^2 + b_2 x_2 + c_2 = a_2 11^2 + b_2 11 + c_2 = 9 \\
 & f_2(x) = a_2 x_3^2 + b_2 x_3 + c_2 = a_2 15^2 + b_2 15 + c_2 = 10 \\
\text{i=3} & f_3(x) = a_3 x_3^2 + b_3 x_3 + c_3 = a_3 15^2 + b_3 15 + c_3 = 10 \\
 & f_3(x) = a_3 x_4^2 + b_3 x_4 + c_3 = a_3 18^2 + b_3 18 + c_3 = 8 \\
\text{i=4} & f_4(x) = a_4 x_4^2 + b_4 x_4 + c_4 = a_4 18^2 + b_4 18 + c_4 = 8 \\
 & f_4(x) = a_4 x_5^2 + b_4 x_5 + c_4 = a_4 22^2 + b_4 22 + c_4 = 7
\end{array}
$$

MN aplicados à Engenharia

Splines Quadráticas - Exemplo

- Três equações são obtidas a partir da condição que diz que, nos nós interiores, as inclinações (derivadas primeiras) dos polinômios de intervalos adjacentes são iguais, portanto:

  i=2 $2a_1x_2 + b_1 = 2a_2x_2 + b_2 \rightarrow$
  $b_1 = 2a_2 11 + b_2$

  i=3 $2a_2x_3 + b_2 = 2a_3x_3 + b_3 \rightarrow$
  $2a_2 15 + b_2 = 2a_3 15 + b_3$

  i=4 $2a_3x_4 + b_3 = 2a_4x_4 + b_4 \rightarrow$
  $2a_3 18 + b_3 = 2a_4 18 + b_4$

- O sistema de 11 equações lineares pode ser escrito na forma matricial;
- Note, na figura, que os coeficientes (a_i, b_i, c_i) na coluna 12 estão dispostos do menor para o maior, sendo que $a_1 = 0$;

Figura: Matriz expandida A,$coef_i$, B

$$
\begin{bmatrix}
8 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \overline{b_1} & 5 \\
11 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \overline{c_1} & 9 \\
0 & 0 & 121 & 11 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \overline{a_2} & 9 \\
0 & 0 & 225 & 15 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \overline{b_2} & 10 \\
0 & 0 & 0 & 0 & 0 & 225 & 15 & 1 & 0 & 0 & 0 & \overline{c_2} & 10 \\
0 & 0 & 0 & 0 & 0 & 324 & 18 & 1 & 0 & 0 & 0 & \overline{a_3} & 8 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 324 & 18 & 1 & \overline{b_3} & 8 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 484 & 22 & 1 & \overline{c_3} & 7 \\
1 & 0 & -22 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \overline{a_4} & 0 \\
0 & 0 & 30 & 1 & 0 & -30 & -1 & 0 & 0 & 0 & 0 & \overline{b_4} & 0 \\
0 & 0 & 0 & 0 & 0 & 36 & 1 & 0 & -36 & -1 & 0 & \overline{c_4} & 0
\end{bmatrix}
$$

Fonte: AUTOR,(2020)

MN aplicados à Engenharia

Portanto, o polinômio $p(x)$ gerado é:

- O vetor de coeficientes das splines quadráticas é, portanto:
  coef'=[a_1 b_1 c_1 a_2 b_2 c_2 a_3 b_3 c_3 a_4 b_4 c_4]
  coef'=[0.00 1.33 -5.67 **-0.2708 7.2917 -38.4375** 0.06 -2.50 35.00 0.06 -2.75 37.25]

- Desse modo, $f(12,7)$ pertence ao polinômio $f_2(x)$ tal que:

$$f_2(x) = (-0,27)x^2 + 7,29x - 38,43 =$$

$$f_2(12,7) = (-0,2708)(12,7)^2 + 7,2917.(12,7) - 38,4375 = 10,4898$$

MN aplicados à Engenharia

Figura: *Script* Spline Linear - parte 01

```
% Interpolaçao Parcial - SPLINE QUADRATICA
clear all;clc; pkg load symbolic; format bank
%**************************************************************
x=[8 11 15 18 22]; y=[5 9 10 8 7]; x_int=12.7;

%PASSO 1*******************************************************
n=length(x);                % Comprimento do vetor x
for(i=1:n)                  % Encontrando o intervalo de interesse;
      if (x_int < x(1)) or (x_int > x(n))
      error('\n Interpolação fora do intervalo')
      elseif x_int <= x(i + 1); break end
end
intervalo=i;
%PASSO 2********************************************************
eq0=2*(n-1); eq1=n-2;    % Quant. spline por intervalo e de nós
eq=eq0+eq1;              % Total de equações
A=zeros(eq+1);            %Matriz A - com coluna adicional
for(i=1:eq+1)
  for(j=1:eq+1)
        if i==1 for(j=1) A(i,j)=1; end end
        if i==2 for(j=1:3) A(i,j)=x(i-1)^abs(3-j);end;  end;
        if i==3 for(j=1:3) A(i,j)=x(i-1)^abs(3-j);end;  end;
        if i==4 for(j=4:6) A(i,j)=x(i-2)^abs(6-j);end;  end;
        if i==5 for(j=4:6) A(i,j)=x(i-2)^abs(6-j);end;  end;
        if i==6 for(j=7:9) A(i,j)=x(i-3)^abs(9-j);end;  end;
        if i==7 for(j=7:9) A(i,j)=x(i-3)^abs(9-j);end;  end;
        if i==8 for(j=10:12) A(i,j)=x(i-4)^abs(12-j);end;  end;
        if i==9 for(j=10:12) A(i,j)=x(i-4)^abs(12-j);end;  end;
          %*************************
```

Fonte: AUTOR,(2020)

MN aplicados à Engenharia

Figura: *Script* Spline Linear - parte 02

```
          %************************
        if i==10
           for(j=1:2) A(i,j)=(2*x(2))^abs(2-j);end;
           for(j=4:5) A(i,j)=-A(i,j-3);end;
        end;
        if i==11
           for(j=4:5) A(i,j)=(2*x(3))^abs(5-j);end;
           for(j=7:8) A(i,j)=-A(i,j-3);end;
        end;
        if i==12
           for(j=7:8) A(i,j)=(2*x(4))^abs(8-j);end;
           for(j=10:11) A(i,j)=-A(i,j-3);end;
    end;
  end
end
B=zeros(1,eq+1);                                  %PASSO 3
for(i=2:eq+1)
    for(i=2) B(i)=y(1);end
    for(i=3:4) B(i)=y(2);end
    for(i=5:6) B(i)=y(3);end
    for(i=7:8) B(i)=y(4);end
    for(i=9) B(i)=y(5); end
end
coef=(A\B'); j=3*intervalo - 2;                          % %PASSO 4
px=polyout([coef(j) coef(j+1) coef(j+2)],'(x_int)')
y_int=coef(j)*x_int^2 + coef(j+1)*x_int + coef(j+2);
fprintf('\nLogo: f(%d) vale: %f\n',x_int,y_int);
```

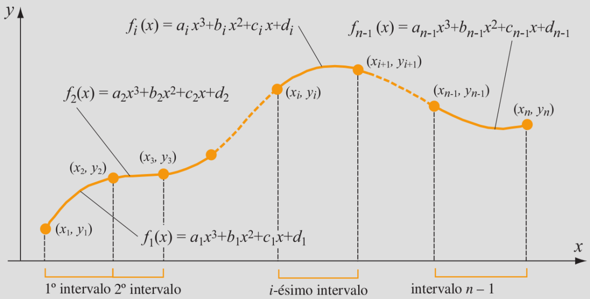
Fonte: AUTOR,(2020)

Organização

MN aplicados à Engenharia

- Em *splines* cúbicas, a interpolação é feita com polinômios de terceira ordem.
- Para um conjunto de dados com n pontos, há $n-1$ intervalos, portanto, um número expressivo de equações, a depender da forma de representação polinomial utilizada (padrão, Lagrange, Newton);

Figura: Splines Cúbicas



Fonte: GILAT,(2008)

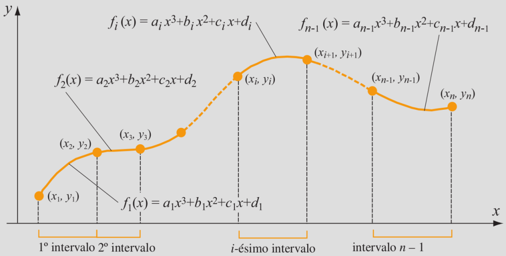
MN aplicados à Engenharia

(1) De forma geral, há $n-1$ equações e, como cada equação tem, *neste caso*, **quatro** coeficientes, um total de $4(n-1) = 4n-4$ coeficientes a serem determinados, tais que:

* Cada polinômio $f_i(x)$ deve passar pelos pontos finais do intervalo, (x_i, y_i) e (x_{i+1}, y_{i+1}), o que significa que $f_i(x_i) = y_i$ e $f_i(x_{i+1}) = y_{i+1}$, portanto:

$$\begin{cases} a_i x_i^3 + b_i x_i^2 + c_i x_i + d_i = y_i \\ a_i x_{i+1}^3 + b_i x_{i+1}^2 + c_i x_{i+1} + d_{i+1} = y_{i+1} \end{cases} \quad \text{para } i = 1, 2, \ldots, n-1$$

Figura: Splines Quadráticas



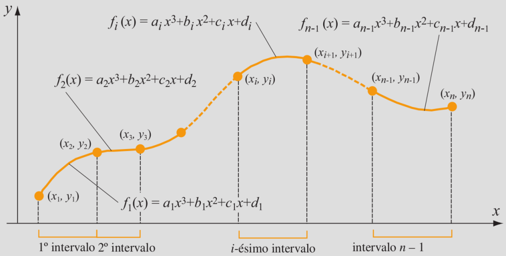
Fonte: GILAT,(2008)

MN aplicados à Engenharia

(2) Nos nós internos, as inclinações (derivadas primeiras) dos polinômios de intervalos adjacentes são iguais, portanto, com a transição da curva que passa por um nó interno de um polinômio para outro:

- A derivada primeira do i-ésimo polinômio é: $f_i'(x) = \dfrac{df_i}{dx} = 3a_i x^2 + 2b_i x + c_i$
- Para n pontos, o primeiro ponto interno é $i = 2$ e o último é $i = n - 1$, o que resulta em: $3a_{i-1}x_i^2 + 2b_{i-1}x_i + c_i = 3a_i x_i^2 + 2b_i x_i + c_i$
- Como há $n - 2$ pontos internos, essa condição fornece $n - 2$ equações.

Figura: Splines Quadráticas



Fonte: GILAT,(2008)

MN aplicados à Engenharia

(3) Nos nós internos, as derivadas segundas dos polinômios de intervalos adjacentes devem ser iguais, isso significa que, com a transição da curva que passa por um nó interno de um polinômio para outro, a taxa de inclinação (curvatura) deve ser contínua, portanto,:

- $f_i''(x) = \dfrac{d^2 f_i}{dx^2} = 6a_i x + 2b_i$
- Para n pontos, o primeiro ponto interno é $i = 2$ e o último é $i = n - 1$, o que resulta em: $6a_{i-1}x_i + 2b_{i-1} = 6a_i x_i + 2b_i$
- Como há $n - 2$ pontos internos, essa condição fornece $n - 2$ equações.

- Juntas, as três condições fornecem $4n - 6$ equações, entretanto, os $n - 1$ *polinômios* têm $4n - 4$ *coeficientes*, e com isso duas equações (condições) adicionais são necessárias para que os coeficientes sejam obtidos. As condições geralmente escolhidas assumem que **a derivada segunda seja nula no primeiro e no último ponto**, portanto:
  $6a_1x_1 + 2b_1 = 0$ e $6a_{n-1}x_n + 2b_{n-1} = 0$

* Splines cúbicas com derivadas segundas igualadas a zero nos pontos finais do intervalo são chamadas de **splines cúbicas naturais**.

- A aplicação de todas as condições leva a um sistema de $4n - 4$ **equações com** $4n - 4$ **coeficientes**.

MN aplicados à Engenharia

Splines Cúbicas
Baseadas em polinômios na forma de Lagrange

- A dedução de splines cúbicas utilizando a forma de Lagrange resulta no termo geral:

$$f_i(x) = \begin{cases} \frac{a_i}{6h_i}(x_{i+1}-x)^3 + \frac{a_{i+1}}{6h_i}(x-x_i)^3 \\ \quad + \left[\frac{y_i}{h_i} - \frac{a_i h_i}{6}\right](x_{i+1}-x) \\ \quad + \left[\frac{y_{i+1}}{h_i} - \frac{a_{i+1} h_i}{6}\right](x-x_i), \end{cases},$$

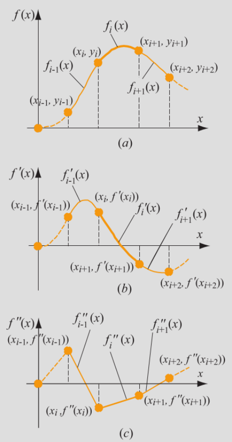
em que $x_i \leq x \leq x_{i+1}$ e $i = 1, 2, \ldots, n-1$

- E o termo a_i vale:

$$h_i a_i + 2(h_i + h_{i+1})a_{i+1} + h_{i+1}a_{i+2} =$$

$$6\left[\frac{y_{i+2}-y_{i+1}}{h_{i+1}} - \frac{y_{i+1}-y_i}{h_i}\right], \text{ para } i = 1, 2, \ldots, n-1.$$

Figura: Splines Cúbica



Fonte: GILAT,(2008)

MN aplicados à Engenharia

Exemplo: Determine as *splines* cúbicas que fazem o ajuste dos dados: $x = [8, 11, 15, 18, 22]; y = [5, 9, 10, 8, 7]$, e encontre $f(12, 7)$.

- **Os cinco pontos geram quatro *splines*.** A equação cúbica da i-ésima spline é:
  $f_i(x) = \frac{a_i}{6h_i}(x_{i+1}-x)^3 + \frac{a_{i+1}}{6h_i}(x-x_i)^3 + \left[\frac{y_i}{h_i} - \frac{a_i h_i}{6}\right](x_{i+1}-x) + \left[\frac{y_{i+1}}{h_i} - \frac{a_{i+1} h_i}{6}\right](x-x_i)$, para $i = 1, \ldots 4$, em que $h_i = x_{x+1} - x_i$
- As quatro equações contêm cinco coeficientes desconhecidos a_1, a_2, a_3, a_4 e a_5. Nas splines cúbicas naturais, os coeficientes a_1 e a_5 são iguais a zero.
- Os outros três coeficientes são determinados a partir de um sistema linear de três equações, a saber:
  - $h_1 = x_2 - x_1 = 11 - 8 = 3$ e $h_2 = x_3 - x_2 = 15 - 11 = 4$;
  - $h_3 = x_4 - x_3 = 18 - 15 = 3$ e $h_4 = x_5 - x_4 = 22 - 18 = 4$.

MN aplicados à Engenharia

Splines cúbicas - Exemplo

De modo que tem-se para:

i=1 $h_1 a_1 + 2(h_1 + h_2)a_2 + h_2 a_3 = 6\left[\frac{y_3 - y_2}{h_2} - \frac{y_2 - y_1}{h_1}\right]$

$3.\boxed{0} + 2(3+4)a_2 + 4a_3 = 6\left[\frac{10-9}{4} - \frac{9-5}{3}\right] \rightarrow 14a_2 + 4a_3 = -6,5$

i=2 $h_2 a_2 + 2(h_2 + h_3)a_3 + h_3 a_4 = 6\left[\frac{y_4 - y_3}{h_3} - \frac{y_3 - y_2}{h_2}\right]$

$4.a_2 + 2(3+4)a_3 + 3a_4 = 6\left[\frac{8-10}{3} - \frac{10-9}{4}\right] \rightarrow 4a_2 + 14a_3 + 3a_4 = -5,5$

i=3 $h_3 a_3 + 2(h_3 + h_4)a_4 + h_4 a_5 = 6\left[\frac{y_5 - y_4}{h_4} - \frac{y_4 - y_3}{h_3}\right]$

$3.a_3 + 2(3+4)a_4 + 4.\boxed{0} = 6\left[\frac{7-8}{4} - \frac{8-10}{3}\right] \rightarrow 3a_3 + 14a_4 = 2,5$

- Na forma matricial o sistema resulta em: $\begin{bmatrix} 14 & 4 & 0 \\ 4 & 14 & 3 \\ 0 & 3 & 14 \end{bmatrix} \begin{bmatrix} a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} \text{-6,5} \\ \text{-5,5} \\ \text{2,5} \end{bmatrix}$

Portanto:

$a_1 = 0; a_2 = -0,3665; a_3 = -0,3421; a_4 = 0,2519$ e $a_5 = 0$

MN aplicados à Engenharia

De modo que :

- O polinômio interpolado $p(x) = f(x)$ calculado com a substituição dos coeficientes para o intervalo que contém $x = 12,7$ é $f_2(x)$ com $11 \leq x \leq 15$

i=1 ...

i=2 $f_2(x) = \frac{a_2}{6h_2}(x_3 - x)^3 + \frac{a_3}{6h_2}(x - x_2)^3 + \left[\frac{y_2}{h_2} - \frac{a_2 h_2}{6}\right](x_3 - x) + \left[\frac{y_3}{h_2} - \frac{a_3 h_2}{6}\right](x - x_2)$

$f_2(x) = \frac{-0{,}3665}{6.4}(15 - 3)^3 + \frac{-0{,}3421}{6.4}(x - 11)^3 + \left[\frac{9}{4} - \frac{-0{,}3665.4}{6}\right](15 - x) + \left[\frac{10}{4} - \frac{-0{,}3421.4}{6}\right](x - 11)$

$f_2(x) = (-0,01527)(15 - x)^3 + (-0,01427)(x - 11)^3 + 2,494(15 - x) + 2,728(x - 11)$

i=3 ...

i=4 ...

MN aplicados à Engenharia

Portanto,

- $f_2(x) = (-0,01527)(15 - x)^3 + (-0,01427)(x - 11)^3 + 2,494(15 - x) + 2,728(x - 11)$
- $f_2(12,7) = (-0,01527)(15 - 12,7)^3 + (-0,01427)(12,7 - 11)^3 + 2,494(15 - 12,7) + 2,728(12,7 - 11) =$ **10,11**

Comparativo de resultados das splines para $x_{int} = 12,7$

$x = 12,7$	Linear	Quadrática	Cúbica
$f(12,7)$	$9,4250$	$10,4898$	$10,11$

MN aplicados à Engenharia

Funções auxiliares nativas

- polyfit(*vetor x, vetor y , grau*)
- interp1(*vetor x, vetor y, x interpolado (vetor ou escalar), método*)
    - Interpolação : 'nearest' e 'linear'
    - Interpolação e extrapolação: 'spline' e 'pchip'
    - 'spline' exige que o intervalos estajam equidistantes para mitigar erros.

Exercícios

- **Veja a lista de exercícios na web**
- **Veja a lista de códigos em:** `https://github.com/jonathacosta/NM`