

Métodos Numéricos

Prof. Dr. Jonatha Costa

2024

Organização

1 Métodos

Ajustes de curvas

Conceitos

Regressão Linear - mínimos quadrados

Linearização

Regressão Polinomial

Polinômio interpolador de Lagrange

Polinômio interpolador de Newton

Splines

Splines Quadráticas

Splines Cúbicas

Resolução de Sistemas Lineares

Introdução

Método de Eliminação de Gauss

Método de Eliminação de Gauss-Jordan

Método de Fatoração LU

LU - Método de Crout

Inversa de uma Matriz

Conceito de Métodos Iterativos

Método de Jacobi

Método de Gauss-Seidel

MN aplicados à Engenharia

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

MN aplicados à Engenharia

Introdução

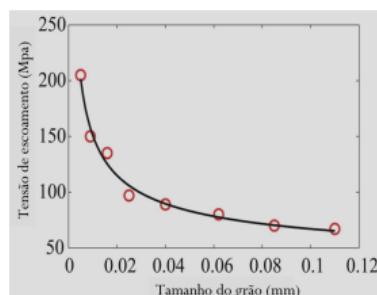
- Muitas observações de engenharia são feitas em experimentos nos quais grandezas físicas são medidas e gravadas.
- São chamados de dados ou pontos experimentais.
- Os dados armazenados são *discretos*, podendo perder informação do sinal original.
- As curvas obtidas são representadas por uma equação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



$$\sigma = Cd^m$$

Em que:

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

Fonte: GILAT,(2008)

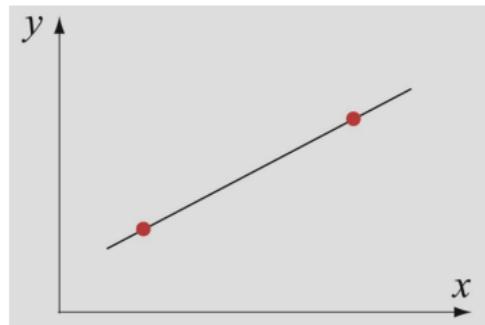
MN aplicados à Engenharia

Ajuste de curvas – Polinômio de 1 grau:

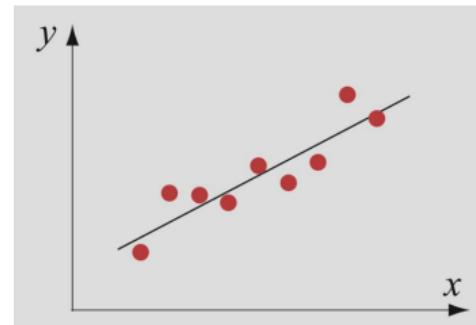
- É a forma mais simples de se encontrar uma curva.
- É definida por: $y = a_1x + a_0$ - feito com a determinação das constantes a_1 e a_0 que fornecem o menor erro quando os pontos medidos são substituídos em y .

Figura: Curvas

(a) Dois pontos



(b) Vários pontos



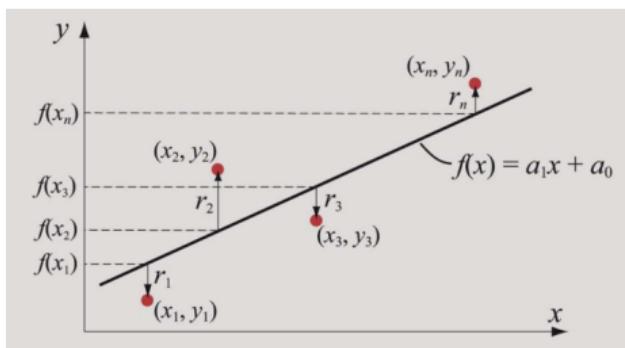
Fonte: GILAT,(2008)

MN aplicados à Engenharia

Melhor ajuste de curvas – Polinômio de 1 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, usa-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

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

- * Com essa definição, o erro global é sempre um número positivo (resíduos positivos e negativos não se cancelam)

MN aplicados à Engenharia

Regressão Linear pelo método dos Mínimos Quadrados

- Usado 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:

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

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

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

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

MN aplicados à Engenharia

Regressão Linear pelo método dos Mínimos Quadrados

- Resolvendo as derivadas, tem-se que:

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

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

- Por conveniência pode-se fazer:

$$* \quad S_x = \sum_{i=1}^n x_i, S_y = \sum_{i=1}^n y_i, S_{xy} = \sum_{i=1}^n x_i y_i, S_{xx} = \sum_{i=1}^n x_i^2, \text{ portanto}$$

$$* \quad a_1 = \frac{n S_{xy} - S_x S_y}{n S_{xx} - (S_x)^2}, \quad a_0 = \frac{S_{xy} S_y - S_x S_{xy}}{n S_{xx} - (S_x)^2}$$

MN aplicados à Engenharia

Figura: *Script* de Regressão Linear

```
1 % AJUSTE DE CURVAS - REGRESSÃO LINEAR
2 clc; clear all;
3
4 x=0:10:100;
5 y=[ 0.94 0.96 1.0 1.05 1.07 1.09 ...
6 1.14 1.17 1.21 1.24 1.28];
7 %*****
8 if (length(x)!=length(y))
9   disp('Falha! X e Y tem dimensões diferentes!')
10 else
11   n=length(x);
12   Sx=sum(x);      Sy=sum(y);
13   Sxy=sum(x.*y); Sxx=sum(x.^2);
14   a1=(n*Sxy-Sx*Sy)/(n*Sxx-Sx^2);
15   a0=(Sxx*Sy-Sxy*Sx)/(n*Sxx-Sx^2);
16 end
17 p=[a1 a0];y2=polyval(p,x);
18 %*****
19 plot(x,y,'*r',x,y2)
20 legend('Medições','Regressão Linear')
```

Fonte: Autor,(2020)

MN aplicados à Engenharia

Linearização de Equações Não-lineares

- 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;
 - $y = bx^m$
- 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:
 - $y = bx^m \rightarrow \ln(y) = \ln(bx^m) = m \ln(x) + \ln(b)$,
 - equação linear em termos de $\ln(x)$ tal que a equação está na forma $Y = a_1X + a_0$, 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 $y = bx^m$ 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.
$y = \frac{mx}{b+x}$	$\frac{1}{y} = \frac{b}{mx} + \frac{1}{m}$	$Y = \frac{1}{y}, X = \frac{1}{x}$ $a_1 = \frac{b}{m}, a_0 = \frac{1}{m}$	$1/x_i$ 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 Não-lineares

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

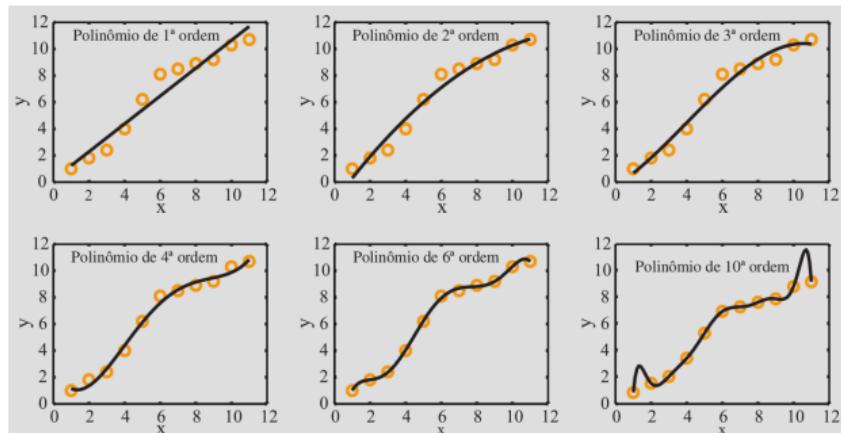
- Traçado dos pontos medidos de uma maneira específica, verificando se esses pontos parecem formar uma linha reta;
- Funções exponenciais não podem passar pela origem.
- Funções exponenciais só são capazes de fazer o ajuste de dados nos quais todos os valores de y são positivos ou negativos.
- Funções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

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:

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

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

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

- Calculando as derivadas parciais em relação aos coeficientes a_0, a_1, \dots, 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, \dots, a_n do polinômio que melhor se ajusta aos n pontos (x_i, y_i) .

MN aplicados à Engenharia

Regressão polinomial

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 as derivadas parciais em relação aos coeficientes a_0, a_1, \dots, a_m encontra-se um sistema com n equações e m incógnitas.
- Derivando 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

Regressão polinomial

Resolvendo 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 $y = a_2 x_i^2 + a_1 x_i + a_0$ que melhor se ajusta aos n pontos (x_i, y_i) .
- Os coeficientes de **polinômios de ordem superior** são deduzidos da mesma forma.

MN aplicados à Engenharia

Regressão polinomial

Estruturando na forma matricial tem-se:

$$\bullet \quad \begin{bmatrix} n & \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i^3 \\ \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i^3 & \sum_{i=1}^n x_i^4 \end{bmatrix} * \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n x_i^2 y_i \end{bmatrix}$$

- E, portanto, um sistema do tipo $Ax = B$, tal que $x = A^{-1}B$.
- 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:
 $p(x) = a_m x^m + a_{m-1} x^{m-1} + \dots + a_1 x + a_0$ ou $p(x) = a_0 + a_1 x + \dots + a_{m-1} x^{m-1} + a_m x^m$.

MN aplicados à Engenharia

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

```

1 % REGRESSÃO POLINOMIAL
2 clc; clear all; format short
3
4 x=1:0.25:5; y=(x.^2-2*x-3); % Função de referência
5 m=4; % Grau de p(x)
6
7 n=length(x);
8 for(i=1:2*m) v(i)=sum(x.^(i)); end % Termos de potências x num só vetor
9 v=[n v];A=[]; % Inclusão do termo n
10 for(i=1:m+1) A=[A; v(i:(m+i))]; end % Matriz A
11 for(i=1:m) u(i)=sum(x.^(i).*y); end % Termos de potências xy num só vetor
12 B=[sum(y) u]'; % Inclusão do termo yi na Matriz B
13 sol=(A\B)';
14 p=sol(end:-1:1); % Saída [a0 a1 a2 ... an]
15 y2=polyval(p,x); % Ordenando [an ... a0]
16 px=polyout(p,'x'); % Y do p(x)
17
18 plot(x,y,'or',x,y2)
19 legend(px);

```

Fonte: Autor,(2020)

MN aplicados à Engenharia

Figura: Exemplo de *script* de Regressão Polinomial no Python

```

1 % REGRESSÃO POLINOMIAL
2 clc; clear all; format short
3
4 x=1:0.25:5; y=(x.^2-2*x-3);           % Função de referência
5 m=4;                                     % Grau de p(x)
6
7 n=length(x);
8 for(i=1:2*m) v(i)=sum(x.^i)); end      % Termos de potências x num só vetor
9 v=[n v];A=[];                            % Inclusão do termo n
10 for(i=1:m+1) A=[A; v(i:(m+i))]; end   % Matriz A
11 for(i=1:m) u(i)=sum(x.^i).*y); end      % Termos de potências xy num só vetor
12 B=[sum(y) u]';
13 sol=(A\B)';
14 p=sol(end:-1:1);                        % Inclusão do termo yi na Matriz B
15 y2=polyval(p,x);                      % Saída [a0 a1 a2 ... an]
16 px=polyout(p,'x');                     % Ordenando [an ... a0]
17
18 plot(x,y,'or',x,y2);                  % Y do p(x)
19 legend(px);

```

Fonte: Autor,(2020)

MN aplicados à Engenharia

Polinômio de Lagrange

- A solução de um ajuste de curvas envolve com polinômio **padrão**:
 - ① Definir o polinômio, cujo formato padrão é: $f(x) = a_m x^m + a_{m-1} x^{m-1} + \dots + a_1 x + a_0$;
 - 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.
 - ④ Entretanto, não é eficiente quando polinômios de **ordem mais elevada** estão envolvidos. E, adicionalmente, a matriz dos coeficientes é frequentemente mal condicionada.
- Os *polinômios interpoladores de Lagrange*
 - ① formam uma classe específica de polinômios que podem ser usados 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 que para os pontos a e b resulta em:
 - * $y_1 = a_1(x_1 - x_2) + a_2(x_1 - x_1) \rightarrow a_1 = \frac{y_1}{(x_1 - x_2)}$
 - * $y_2 = a_1(x_2 - x_2) + a_2(x_2 - x_1) \rightarrow a_2 = \frac{y_2}{(x_2 - x_1)}$
 Recompondo a equação, temos:
 - $f(x) = \frac{(x-x_2)}{(x_1-x_2)}y_1 + \frac{(x-x_1)}{(x_2-x_1)}y_2$
- Para três pontos $(x_1, y_1), (x_2, y_2)$ e (x_3, y_3) o polinômio de Lagrange tem a seguinte forma:

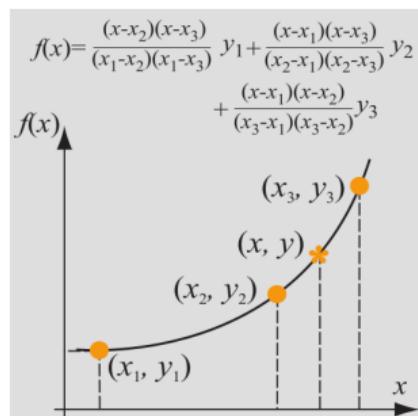
MN aplicados à Engenharia

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 2^a ordem



Fonte: GILAT,(2008)

MN aplicados à Engenharia

Polinômio de Lagrange

- Escrito de forma compacta usando 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 $L_i(x) = \prod_{j=1, j \neq i}^n \frac{(x - x_j)}{(x_i - x_j)}$ 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

```

1 % INTERPOLAÇÃO LAGRANGE : f(x) = soma( Y*L(x) )
2 clc; clear all; pkg load symbolic
3
4 x=[1 2 4 5 7]; y=[52 5 -5 -40 10];
5 m=3;
6 n=length(x);tic;
7 for(i=1:n) L(i)=1; % Função de Lagrange
8 for(j=1:n)
9 if (i~=j)
10 L(i)= L(i)*(m -x(j)) / ( x(i)-x(j));
11 end
12 end
13 end
14
15 clc;
16 Yint=(sum(y.*L)) % f(x) = soma(Y*L(x))
17 fprintf('Tempo de processamento: %.5fs\n',toc);

```

Fonte: Autor,(2020)

MN aplicados à Engenharia

Polinômio de Newton

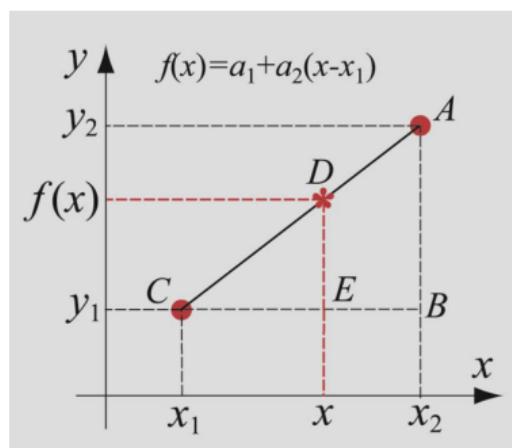
- Método alternativo ao polinômio interpolador de Lagrange
- Nos *polinômios interpoladores de Newton*
 - A determinação dos coeficientes não requer a solução de um sistema com n equações;
 - Os pontos do conjunto de dados não precisam estar ordenados de forma ascendente ou descendente;
 - Os coeficientes a_1 a a_n , uma vez determinados, podem ser usados 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
$$f(x) = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2) + \dots + a_n(x - x_1)(x - x_2) \dots (x - x_{n-1})$$

MN aplicados à Engenharia

Polinômio de Newton de primeira ordem

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

Figura: Polinômio de Newton



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 a semelhança de triângulos na figura. Logo,

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

$$\text{e, portanto, } a_1 = y_1 \text{ e } a_2 = \frac{y_2 - y_1}{x_2 - x_1}.$$

Fonte: GILAT,(2008)

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 $a_2 = \frac{y_2 - y_1}{x_2 - x_1}$
Através de manipulações matemáticas, tem-se:
- O termo adicional para um polinômio de ordem 2 seria:

$$a_3 = \frac{\frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_1}$$

- O termo adicional para um polinômio de ordem 3 seria:

$$a_4 = \frac{\frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}}{x_4 - x_1}$$

MN aplicados à Engenharia

Polinômio de Newton - Ordem superior

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

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

MN aplicados à Engenharia

Polinômio de Newton - Forma geral

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

- $f[x_k, x_{k-1}, x_2, x_1] = \frac{f[x_k, x_{k-1}, x_3, x_2] - f[x_{k-1}, x_{k-2}, \dots, x_2, x_1]}{x_k - x_1}$,

em que a ordem máxima é $(n - 1)$

- **Forma geral:**

$$f(x) = \left\{ \begin{array}{l} y_1 + f[x_2, x_1](x - x_1) + f[x_3, x_2, x_1](x - x_1)(x - x_2) + \dots \\ \dots + f[x_n, x_{n-1}, x_2, x_1](x - x_1)(x - x_2) \dots (x - x_{n-1}) \end{array} \right.$$

MN aplicados à Engenharia

Figura: Exemplo de *script* de Interpolação por Newton

```

1 % INTERPOLAÇÃO NEWTON :
2 clc; clear all; pkg load symbolic
3
4 x=[1 2 4 5 7]; y=[52 5 -5 -40 10];
5 xint=3;
6 n=length(x);
7 a(1)=y(1);
8 for (i=1:n-1)
9   divDIF(i,1)=(y(i+1)-y(i))/(x(i+1)-x(i));      %Diferenças na coluna 1
10 end
11 for(j=2:n-1)
12   for(i=1:n-j)
13     divDIF(i,j)=(divDIF(i+1,j-1)-divDIF(i,j-1))/(x(j+i)-x(i));
14   end
15 end
16 for(j=2:n) a(j)=divDIF(1,j-1);end
17 Yint=a(1); xn=1;
18 for(k=2:n) xn=xn*(xint -x(k-1)); Yint=Yint+a(k)*xn; end
19 Yint

```

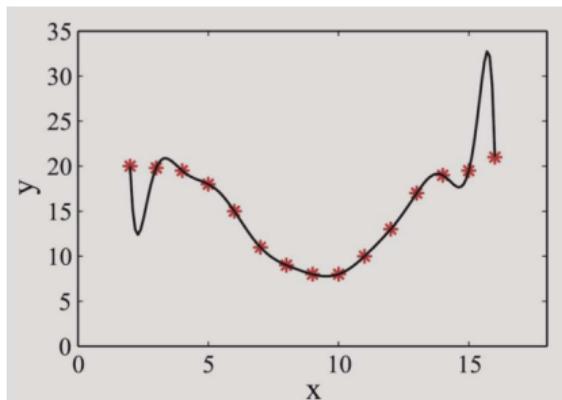
Fonte: Adaptado de GILAT,(2008)

MN aplicados à Engenharia

Splines - Introdução

- Quando um conjunto de dados contendo n pontos é dado e um *único polinômio* é usado 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



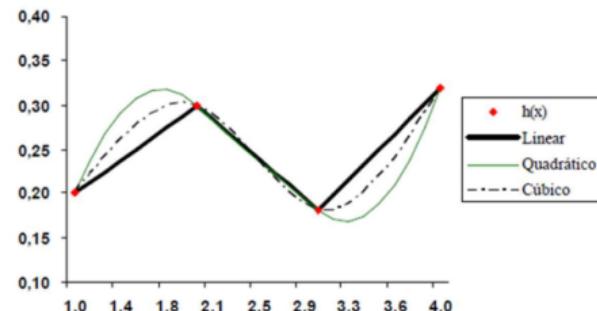
Fonte: GILAT,(2008)

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



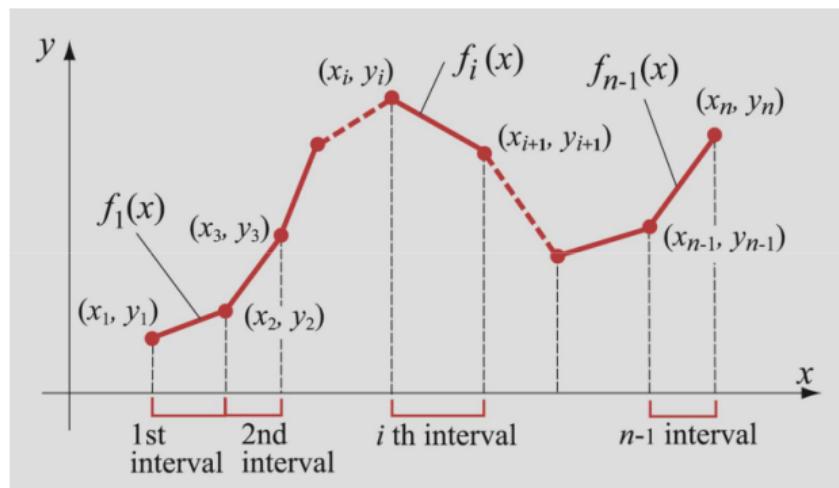
Fonte: GILAT,(2008)

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

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 usando um polinômio de primeira ordem, e os pontos são conectados por linhas retas.
- Usando 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 usando a equação da linha reta que conecta o ponto (x_i, y_i) ao ponto (x_{i+1}, y_{i+1}) :

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

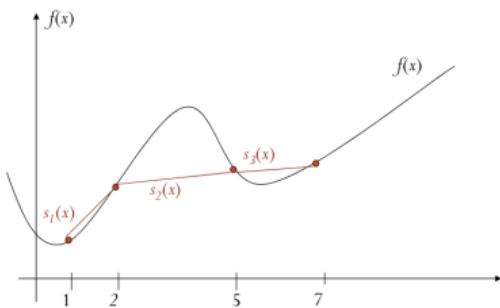
para $i = 1, 2, \dots, n - 1$

MN aplicados à Engenharia

Splines Linear

- Note que $s_i(x)$ é um polinômio de grau 1 no intervalo.
- $s_i(x)$ 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

Splines Linear - 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* e, 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

Splines Linear

Figura: Script Spline Linear

```

1 % Interpolação Parcial Linear - SPLINE LINEAR
2 clc; clear all; pkg load symbolic
3
4 %*****
5 x=[8 11 15 18 22]; y=[5 9 10 8 7]; x_int=12.7;
6
7 %*****
8 n=length(x); % Encontrando o intervalo de interpolacao
9 for(i=1:n)
10 if x_int < x(1) | x_int > x(n)
11 error('Interpolação fora do intervalo')
12 elseif x_int< x(i+1) break % Fim da busca pelo intervalo
13 end
14 end
15 %*****
16 syms m x1 x2 y1 y2
17 clc;
18 px=simplify((m-x2)*y1/(x1-x2) + (m-x1)*y2/(x2-x1)); % Polinômio
19 f=matlabFunction(px); % Função com 5 termos
20 y_int = f( x_int, x(i), x(i+1), y(i), y(i+1)); % Dados de entrada para i
21 pol=subs(px,'m',x_int)
22 fprintf('Intervalo de interpolação para x= %d é %d < x < %d.\n\n',
23 x_int,x(i),x(i+1));
24 fprintf('O ponto f(%2.2f) = %2.2f \n\n',x_int,y_int);
25
26 %*****
27 plot(x_int,y_int,'*'); hold on; plot(x,y,'0');
28 legend('Pontos x e y');

```

MN aplicados à Engenharia

Splines Linear

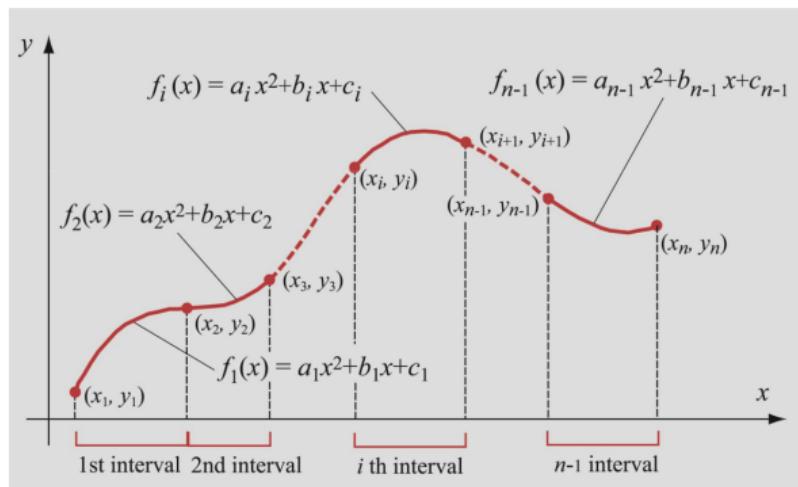
- Uma desvantagem da interpolação linear é que a função de interpolação é geralmente não diferenciável nos pontos extremos de cada intervalo.
- Dessa forma, a função não é suave nos pontos.
- Para uma melhor aproximação de sistemas reais, a característica de suavidade é deseável, logo, a função de interpolação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

MN aplicados à Engenharia

Splines Quadrática

- Nas *splines* quadráticas, a interpolação é feita com polinômios de **segunda ordem**;
- Em um conjunto de n pontos, há $n - 1$ intervalos e, 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 $i = 1, 2, \dots, n - 1$;

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:

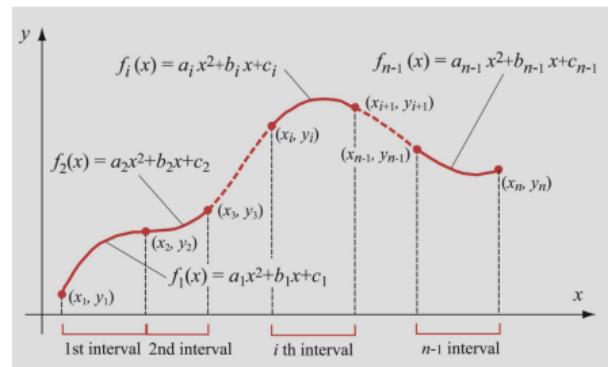
- * Cada polinômio $f_i(x)$ deve passar pelos pontos finais do intervalo, (x_i, y_i) e (x_{i+1}, y_{i+1}) , o que significa que $f_i(x_i) = y_i$ e $f_i(x_{i+1}) = y_{i+1}$ e, portanto:

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

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

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

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: $f'(x) = \frac{df}{dx} = 2a_i x + b_i$

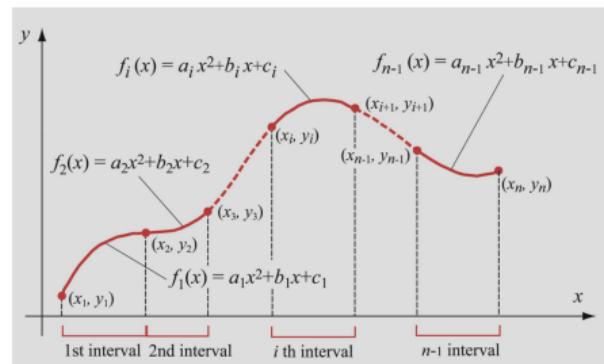
* Para n pontos, o primeiro ponto interno é $i = 2$ e o último é $i = n - 1$. Igualando as derivadas primeiras em todos os **pontos internos**, obtém-se:

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

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

* Como há $n - 2$ pontos internos, essa condição fornece $n - 2$ equações e, 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

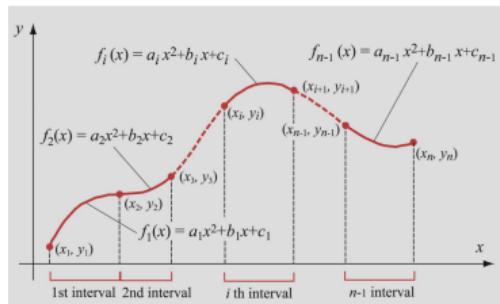
Splines Quadráticas

- (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.

Figura: Splines Quadráticas

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.



Fonte: GILAT,(2008)

MN aplicados à Engenharia

Splines Quadráticas - 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:

$$\text{i=1} \quad f_1(x) = a_1 x_1^2 + b_1 x_1 + c_1 = (0).8^2 + b_1 8 + c_1 = 5$$

$$f_1(x) = a_1 x_2^2 + b_1 x_2 + c_1 = (0).11^2 + b_1 11 + c_1 = 9$$

$$\text{i=2} \quad f_2(x) = a_2 x_2^2 + b_2 x_2 + c_2 = a_2 11^2 + b_2 11 + c_2 = 9$$

$$f_2(x) = a_2 x_3^2 + b_2 x_3 + c_2 = a_2 15^2 + b_2 15 + c_2 = 10$$

$$\text{i=3} \quad f_3(x) = a_3 x_3^2 + b_3 x_3 + c_3 = a_3 15^2 + b_3 15 + c_3 = 10$$

$$f_3(x) = a_3 x_4^2 + b_3 x_4 + c_3 = a_3 18^2 + b_3 18 + c_3 = 8$$

$$\text{i=4} \quad f_4(x) = a_4 x_4^2 + b_4 x_4 + c_4 = a_4 18^2 + b_4 18 + c_4 = 8$$

$$f_4(x) = a_4 x_5^2 + b_4 x_5 + c_4 = a_4 22^2 + b_4 22 + c_4 = 7$$

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:

$$\begin{aligned} i=2 \quad & 2a_1x_2 + b_1 = 2a_2x_2 + b_2 \rightarrow \\ & b_1 = 2a_211 + b_2 \end{aligned}$$

$$\begin{aligned} i=3 \quad & 2a_2x_3 + b_2 = 2a_3x_3 + b_3 \rightarrow \\ & 2a_215 + b_2 = 2a_315 + b_3 \end{aligned}$$

$$\begin{aligned} i=4 \quad & 2a_3x_4 + b_3 = 2a_4x_4 + b_4 \rightarrow \\ & 2a_318 + b_3 = 2a_418 + b_4 \end{aligned}$$

- 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, $coeff_i$, B

8	1	0	0	0	0	0	0	0	0	0	\bar{b}_1	5
11	1	0	0	0	0	0	0	0	0	0	\bar{c}_1	9
0	0	121	11	1	0	0	0	0	0	0	\bar{a}_2	9
0	0	225	15	1	0	0	0	0	0	0	\bar{b}_2	10
0	0	0	0	0	225	15	1	0	0	0	\bar{c}_2	10
0	0	0	0	0	324	18	1	0	0	0	\bar{a}_3	8
0	0	0	0	0	0	0	0	324	18	1	\bar{b}_3	8
0	0	0	0	0	0	0	0	484	22	1	\bar{c}_3	7
1	0	-22	-1	0	0	0	0	0	0	0	\bar{a}_4	0
0	0	30	1	0	-30	-1	0	0	0	0	\bar{b}_4	0
0	0	0	0	0	36	1	0	-36	-1	0	\bar{c}_4	0

MN aplicados à Engenharia

Splines Quadráticas - Exemplo

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

- O vetor de coeficientes das splines quadráticas é, portanto:

$$\text{coef}' = [a_1 \ b_1 \ c_1 \ a_2 \ b_2 \ c_2 \ a_3 \ b_3 \ c_3 \ a_4 \ b_4 \ c_4]$$

$$\text{coef}' = [0.00 \ 1.33 \ -5.67 \ \color{red}{-0.2708} \ \color{blue}{7.2917} \ \color{red}{-38.4375} \ 0.06 \ -2.50 \ 35.00 \ 0.06 \ -2.75 \ 37.25]$$

- Portanto, $f(12, 7)$ 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

Splines Quadrática

Figura: Script Spline Linear - parte 01

```

1 % Interpolação Parcial - SPLINE QUADRATICA
2 clear all;clc; pkg load symbolic; format bank
3 %*****
4 x=[8 11 15 18 22]; y=[5 9 10 8 7]; x_int=12.7;
5
6 %PASSO 1*****
7 n=length(x); % Comprimento do vetor x
8 for(i=1:n) % Encontrando o intervalo de interesse;
9 if (x_int < x(1)) or (x_int > x(n))
10 error('n Interpolação fora do intervalo')
11 elseif x_int <= x(i + 1); break end
12 end
13 intervalo=i;
14 %PASSO 2*****
15 eq0=2*(n-1); eq1=n-2; % Quant. spline por intervalo e de nós
16 eq=eq0+eq1; % Total de equações
17 A=zeros(eq+1); %Matriz A - com coluna adicional
18 for(i=1:eq+1)
19 for(j=1:eq+1)
20 if i==1 for(j=1) A(i,j)=1; end end
21 if i==2 for(j=1:3) A(i,j)=x(i-1)^abs(3-j);end; end;
22 if i==3 for(j=1:3) A(i,j)=x(i-1)^abs(3-j);end; end;
23 if i==4 for(j=4:6) A(i,j)=x(i-2)^abs(6-j);end; end;
24 if i==5 for(j=4:6) A(i,j)=x(i-2)^abs(6-j);end; end;
25 if i==6 for(j=7:9) A(i,j)=x(i-3)^abs(9-j);end; end;
26 if i==7 for(j=7:9) A(i,j)=x(i-3)^abs(9-j);end; end;
27 if i==8 for(j=10:12) A(i,j)=x(i-4)^abs(12-j);end; end;
28 if i==9 for(j=10:12) A(i,j)=x(i-4)^abs(12-j);end; end;
29 %*****

```

Fonte: AUTOR,(2020)

MN aplicados à Engenharia

Splines Quadrática

Figura: Script Spline Linear - parte 02

```

29      %*****
30      if i==10
31          for(j=1:2) A(i,j)=(2*x(2))^abs(2-j);end;
32          for(j=4:5) A(i,j)=-A(i,j-3);end;
33      end;
34      if i==11
35          for(j=4:5) A(i,j)=(2*x(3))^abs(5-j);end;
36          for(j=7:8) A(i,j)=-A(i,j-3);end;
37      end;
38      if i==12
39          for(j=7:8) A(i,j)=(2*x(4))^abs(8-j);end;
40          for(j=10:11) A(i,j)=-A(i,j-3);end;
41      end;
42  end
43 end
44 B=zeros(1,eq+1);                               %PASSO 3
45 for(i=2:eq+1)
46     for(i=2) B(i)=y(1);end
47     for(i=3:4) B(i)=y(2);end
48     for(i=5:6) B(i)=y(3);end
49     for(i=7:8) B(i)=y(4);end
50     for(i=9) B(i)=y(5); end
51 end
52 coef=(A\B'); j=3*intervalo - 2;             % %PASSO 4
53 px=polyout([coef(j) coef(j+1) coef(j+2)],'(x_int)')
54 y_int=coef(j)*x_int^2 + coef(j+1)*x_int + coef(j+2);
55 fprintf('\nLogo: f(%d) vale: %f\n',x_int,y_int);

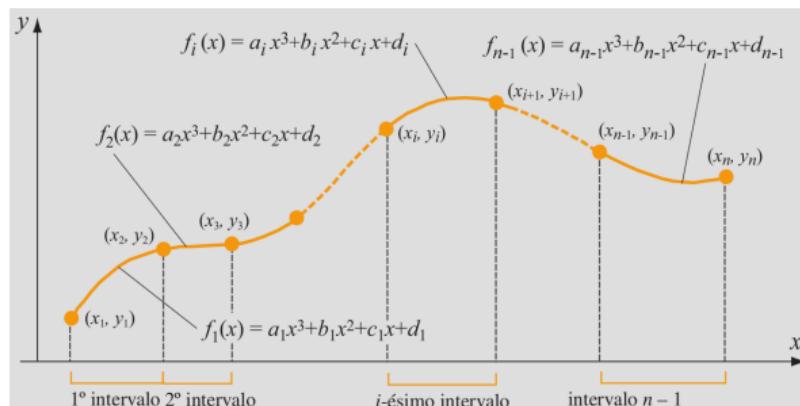
```

MN aplicados à Engenharia

Splines Cúbica

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

Figura: Splines Cúbicas



Fonte: GILAT,(2008)

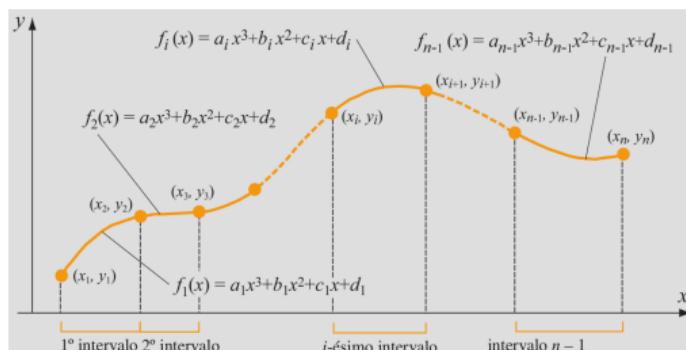
MN aplicados à Engenharia

Splines Cúbicas

- (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 $f_i(x_{i+1}) = y_{i+1}$ e, portanto:

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

Figura: Splines Quadráticas



Fonte: GILAT,(2008)

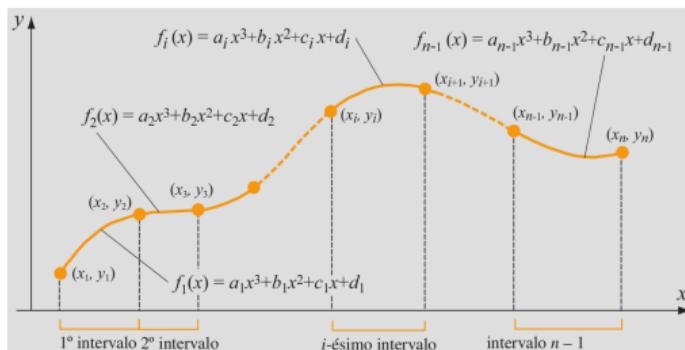
MN aplicados à Engenharia

Splines Cúbicas

(2) Nos nós internos, as inclinações (derivadas primeiras) dos polinômios de intervalos adjacentes são iguais e, 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: $f'_i(x) = \frac{df_i}{dx} = 3a_i x^2 + 2b_i x + c_i$
- 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

Splines Cúbicas

- (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, e portanto:

- $f_i''(x) = \frac{d^2 f_i}{dx^2} = 6a_i x + 2b_i$
- 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** e, 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 usando a forma de Lagrange resulta no termo geral:

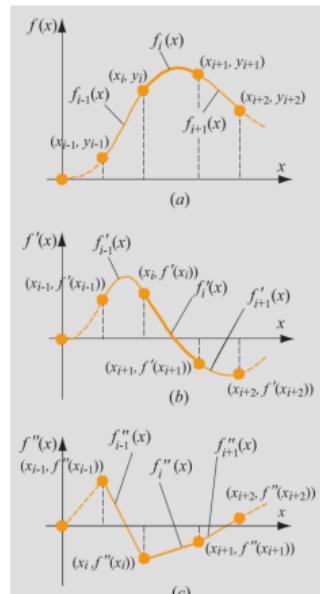
$$f_i(x) = \begin{cases} \frac{a_i}{6h_i}(x_{i+1} - x)^3 + \frac{a_{i+1}}{6h_i}(x - x_i)^3 \\ + \left[\frac{y_i}{h_i} - \frac{a_i h_i}{6} \right] (x_{i+1} - x) \\ + \left[\frac{y_{i+1}}{h_i} - \frac{a_{i+1} h_i}{6} \right] (x - x_i), \end{cases},$$

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

- E o termo a_i vale:

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

Figura: Splines Cúbica



Fonte: GILAT,(2008)

MN aplicados à Engenharia

Splines cúbicas - 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:

$$f_i(x) = \frac{a_i}{6h_i}(x_{i+1}-x)^3 + \frac{a_{i+1}}{6h_i}(x-x_i)^3 + \left[\frac{y_i}{h_i} - \frac{a_i h_i}{6} \right](x_{i+1}-x) + \left[\frac{y_{i+1}}{h_i} - \frac{a_{i+1} h_i}{6} \right](x-x_i), \text{ para } i = 1, \dots, 4,$$
 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:

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

$$3.0 + 2(3+4)a_2 + 4a_3 = 6 \left[\frac{10-9}{4} - \frac{9-5}{3} \right] \rightarrow 14a_2 + 4a_3 = -6,5$$

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

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

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

$$3.a_3 + 2(3+4)a_4 + 4.0 = 6 \left[\frac{7-8}{4} - \frac{8-10}{3} \right] \rightarrow 3a_3 + 14a_4 = 2,5$$

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

Portanto:

$$a_1 = 0; a_2 = -0,3665; a_3 = -0,3421; a_4 = 0,2519 \text{ e } a_5 = 0$$

MN aplicados à Engenharia

Splines cúbicas - Exemplo

De modo que tem-se :

- 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 ...

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

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

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

i=3 ...

i=4 ...

MN aplicados à Engenharia

Splines cúbicas - Exemplo

De modo que:

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

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 os intervalos estajam equidistantes para mitigar erros.

Organização

1 Métodos

Ajustes de curvas

Conceitos

Regressão Linear - mínimos quadrados

Linearização

Regressão Polinomial

Polinômio interpolador de Lagrange

Polinômio interpolador de Newton

Splines

Splines Quadráticas

Splines Cúbicas

Resolução de Sistemas Lineares

Introdução

Método de Eliminação de Gauss

Método de Eliminação de Gauss-Jordan

Método de Fatoração LU

LU - Método de Crout

Inversa de uma Matriz

Conceito de Métodos Iterativos

Método de Jacobi

Método de Gauss-Seidel

MN aplicados à Engenharia

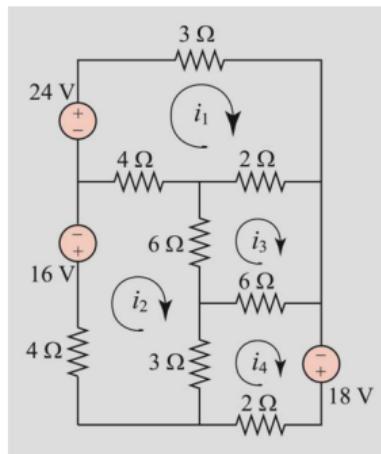
- Apresentar conteúdo de Resolução de Equações Lineares
 - Métodos diretos
 - * Eliminação de Gauss
 - * Eliminação de Gauss-Jordan
 - * Fatoração LU
 - * Método de Crout
 - * Inversa de uma matriz
 - Métodos Iterativos
 - * Gauss-Jacobi
 - * Gauss-Seidel

MN aplicados à Engenharia

Introdução

Exemplo: Seja um problema de engenharia que requer a solução de um sistema de equações. Usando a lei de kirchhoff, as correntes i_1, i_2, i_3 e i_4 podem ser determinadas com a solução do seguinte sistema de quatro equações:

Figura: Circuito Elétrico



$$\left\{ \begin{array}{l} +9i_1 - 4i_2 - 2i_3 = 24 \\ -4i_1 + 17i_2 - 6i_3 - 3i_4 = -16 \\ -2i_1 - 6i_2 + 14i_3 - 6i_4 = 0 \\ -3i_2 - 6i_3 + 11i_4 = 18 \end{array} \right.$$

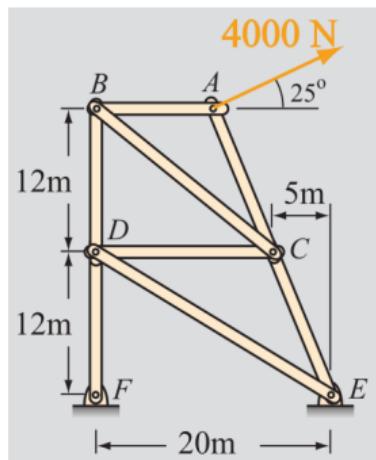
Fonte: GILAT,(2008)

MN aplicados à Engenharia

Introdução

Exemplo: Seja o cálculo de força nos membros de uma treliça. As forças nos oito membros da treliça são determinadas a partir da solução do seguinte sistema de oito equações:

Figura: Dinâmica de forças



$$0,9231F_{AC} = 1690$$

$$F_{AB} - 0,7809F_{BC} = 0$$

$$F_{CD} + 0,8575F_{DE} = 0$$

$$0,3846F_{CE} - 0,3846F_{AC} - 0,7809F_{BC} - F_{CD} = 0$$

$$0,9231F_{AC} + 0,6247F_{BC} - 0,9231F_{CE} = 0$$

$$-F_{AB} - 0,3846F_{AC} = 3625$$

$$0,6247F_{BC} - F_{BD} = 0$$

$$F_{BD} - 0,5145F_{DE} - F_{DF} = 0$$

Fonte: GILAT,(2008)

MN aplicados à Engenharia

Introdução

Um sistema de m equações e n variáveis é chamado de sistema de equações lineares e tem a seguinte forma genérica:

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{array} \right.$$

onde: a_{ij} são os coeficientes, para $1 \leq i \leq m$, $1 \leq j \leq n$, x_j são as variáveis e b_i são as constantes

A resolução de um sistema linear consiste em calcular os valores de x_j , para $j = 1, \dots, n$, caso eles existam, que satisfaçam as m equações simultaneamente.

O sistema linear pode ter:

- * Mais equações do que incógnitas ($m > n$);
- * Mais incógnitas do que equações ($m < n$)
- * O mesmo número de incógnitas e equações ($m = n$).

MN aplicados à Engenharia

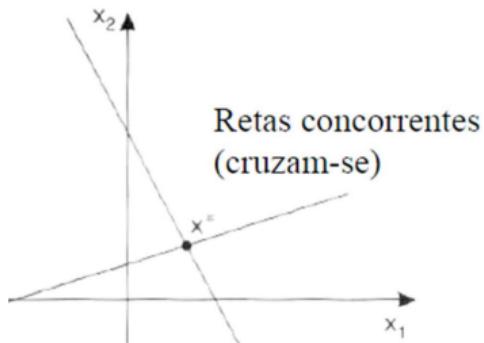
Introdução

Solução única

$$\begin{cases} 2x_1 + x_2 = 3 \\ x_1 - 3x_2 = -2 \end{cases}$$

$$x^* = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Figura: Retas concorrentes



Fonte: DIAS,(2019)

MN aplicados à Engenharia

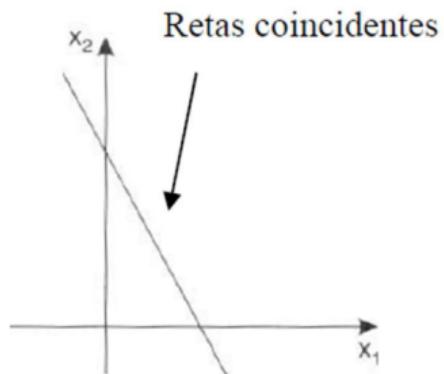
Introdução

Infinitas Soluções

Figura: Retas coincidentes

$$\begin{cases} 2x_1 + x_2 = 3 \\ 4x_1 - 2x_2 = 6 \end{cases}$$

para o qual $\forall x^* = (\alpha, 3 - 2\alpha)^t$ com $\alpha \in \mathbb{R}$, é solução.



Fonte: DIAS,(2019)

MN aplicados à Engenharia

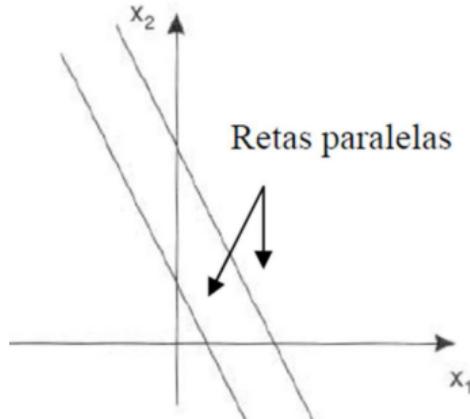
Introdução

Nenhuma solução

Figura: Retas Parelelas

$$\begin{cases} 2x_1 + x_2 = 3 \\ 4x_1 + 2x_2 = 2 \end{cases}$$

Determinante de A é nulo.
 $\det(A) = 0 \rightarrow \nexists x^* \text{ in } \mathbb{R}$.



Fonte: DIAS,(2019)

MN aplicados à Engenharia

Introdução

As operações elementares entre equações de um sistema linear são:

- ① Trocar as equações de posição
- ② Multiplicar uma ou mais equações por constantes (chamamos múltiplos de equações):
- ③ Somar o múltiplo de uma equação por outra e aplicarmos qualquer operação elementar entre equações, em um sistema linear o resultado (x_1, x_2, \dots, x_n) sempre será o mesmo.

MN aplicados à Engenharia

Introdução

- O sistema pode ser escrito na forma de um produto matricial $Ax = b$, onde as matrizes são definidas por:

$$[\mathbf{A}] = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{vmatrix}, [\mathbf{x}] = \begin{vmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{vmatrix}, [\mathbf{b}] = \begin{vmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{vmatrix}$$

Em que:

- A é a matriz (m, n) dos coeficientes
- x é o vetor das variáveis (n linhas)
- b é o vetor das constantes (m linhas), termos independentes
- Obter a solução de $Ax = b$ implica em se obter os escalares x_1, x_2, \dots, x_n que permitam escrever b como combinação linear das n colunas de A .

$$[\mathbf{b}] = x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} + \dots + x_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix}$$

MN aplicados à Engenharia

Métodos de soluções para Sistemas Lineares

Métodos Diretos - São aqueles que fornecem uma solução exata, a menos que existam erros de arredondamento.

- $x = A^{-1}b$
- Eliminação de Gauss;
- Pivotamento
- Fatoração LU

Métodos Indiretos - São aqueles que geram uma sequência de vetores $x(k)$ a partir de uma aproximação inicial $x(0)$.

- Método Iterativo de Gauss
- Método Iterativo de Gauss-Jacobi;

MN aplicados à Engenharia

Método da Eliminação de Gauss

Este método consiste em transformar o sistema linear original $Ax = b$ em um sistema linear equivalente $A^*x = b^*$ com matriz dos coeficientes **triangular superior**.

Figura: Eliminação de Gauss

Sistema original

$$\underbrace{\begin{bmatrix} * & * & \dots & * \\ * & * & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \dots & * \end{bmatrix}}_{Matrix...A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{vetor...X} = \underbrace{\begin{bmatrix} * \\ * \\ \vdots \\ * \end{bmatrix}}_{vetor...b}$$

Sistema transformado

$$\underbrace{\begin{bmatrix} * & * & \dots & * \\ 0 & * & \dots & * \\ \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & * \end{bmatrix}}_{Matrix...A^*} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{vetor...X} = \underbrace{\begin{bmatrix} * \\ * \\ \vdots \\ * \end{bmatrix}}_{vetor...b^*}$$

Operações elementares

Fonte: DIAS,(2019)

MN aplicados à Engenharia

Método da Eliminação de Gauss

Considerações

- Supor que a matriz A seja quadrada $m = n$ e não singular.
- Adotado as seguintes notações
 - * $i = 1, 2, \dots, m$ (i-ésima linha);
 - * $j = 1, 2, \dots, n$ (j-ésima coluna);
 - * $k = 1, \dots$ (k-ésima etapa da eliminação)
 - * $a_{ij}^{(k)}$ e $b_i^{(k)}$

Uma **matriz é singular** se e somente se seu determinante é nulo.

MN aplicados à Engenharia

Método da Eliminação de Gauss

Procedimento:

Para cada fase $k = 1, 2, \dots, n$ da eliminação (ou pivoteamento):

- Determinar o pivô $a_{kk}^{(k)} \neq 0$ (ou não muito pequeno).
- Eliminar (zerar) os elementos da coluna $a_{ik}^{(k)}$ abaixo da k -ésima linha do pivô, para $i = k + 1, \dots, n$.
- Determinar uma constante m_{ik} , de modo que, ao multiplicá-la pela k -ésima linha do pivô e subtrair com a i -ésima linha, esse elemento deverá ser zerado.

$$a_{ik}^{(k)} - m_{ik} a_{kk}^{(k)} = 0 \quad m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$$

MN aplicados à Engenharia

Método da Eliminação de Gauss

Procedimento:

Figura: Eliminação de Gauss - Procedimento

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

Conjunto inicial de equações

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ \cancel{a_{21}} & a'_{22} & a'_{23} & a'_{24} \\ \cancel{a_{31}} & a'_{32} & a'_{33} & a'_{34} \\ \cancel{a_{41}} & a'_{42} & a'_{43} & a'_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x'_2 \\ x'_3 \\ x'_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b'_3 \\ b'_4 \end{bmatrix}$$

Passo 1

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & a'_{32} & a''_{33} & a''_{34} \\ 0 & a'_{42} & a''_{43} & a''_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x'_2 \\ x''_3 \\ x'''_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \\ b'''_4 \end{bmatrix}$$

Passo 2

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & a''_{43} & a''_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x'_2 \\ x''_3 \\ x'''_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \\ b'''_4 \end{bmatrix}$$

Passo 3

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & 0 & a''''_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x'_2 \\ x''_3 \\ x''''_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \\ b''''_4 \end{bmatrix}$$

Equações na forma triangular superior



Fonte: DIAS,(2019)

MN aplicados à Engenharia

Figura: Pivotação

Após o Passo 1, a segunda equação tem um elemento pivô zero

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & 0 & a'_{23} & a'_{24} \\ 0 & a'_{32} & a'_{33} & a'_{34} \\ 0 & a'_{42} & a'_{43} & a'_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b'_3 \\ b'_4 \end{bmatrix}$$

Usando pivotação, troca-se a segunda equação pela terceira.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{32} & a'_{33} & a'_{34} \\ 0 & 0 & a'_{23} & a'_{24} \\ 0 & a'_{42} & a'_{43} & a'_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_3 \\ b'_2 \\ b'_4 \end{bmatrix}$$

Método da Eliminação de Gauss com Pivotação

- Se durante o procedimento de Eliminação de Gauss uma equação pivô tiver um elemento pivô nulo e o sistema de equações tiver solução, uma equação com um elemento pivô diferente de zero sempre poderá ser encontrada.

MN aplicados à Engenharia

Resolva o sistema linear: $\begin{cases} 10x_1 + 2x_2 + x_3 = 7 \\ x_1 + 5x_2 + x_3 = -8 \\ 2x_1 + 3x_2 + 10x_3 = 6 \end{cases}$

- Forma matricial $Ax=b$

$$\left| \begin{array}{ccc|c} 10 & 2 & 1 & x_1 \\ 1 & 5 & 1 & x_2 \\ 2 & 3 & 10 & x_3 \end{array} \right| \cdot \left| \begin{array}{c} 7 \\ -8 \\ 6 \end{array} \right|$$

(1) Método de Gauss

$$\left| \begin{array}{cccc|c} 10 & 2 & 1 & 7 & m_{21} = \frac{A(2,1)}{A(1,1)} = \frac{1}{10} = 0,1 \\ 1 & 5 & 1 & -8 & L2 = L2 - m_{21} * L1 \\ 2 & 3 & 10 & 6 & \Rightarrow \end{array} \right| \left| \begin{array}{cccc|c} 10 & 2 & 1 & 7 & \\ 0 & 4,8 & 0,9 & -8,7 & \Rightarrow \\ 2 & 3 & 10 & 6 & \end{array} \right|$$

$$\left| \begin{array}{cccc|c} 10 & 2 & 1 & 7 & m_{31} = \frac{A(3,1)}{A(1,1)} = \frac{2}{10} = 0,2 \\ 0 & 4,8 & 0,9 & -8,7 & L3 = L3 - m_{31} * L1 \\ 2 & 3 & 10 & 6 & \Rightarrow \end{array} \right| \left| \begin{array}{cccc|c} 10 & 2 & 1 & 7 & \\ 0 & 4,8 & 0,9 & -8,7 & \Rightarrow \\ 0 & 2,6 & 9,8 & 4,6 & \end{array} \right|$$

$$\left| \begin{array}{cccc|c} 10 & 2 & 1 & 7 & m_{32} = \frac{A(3,2)}{A(2,2)} = \frac{2,6}{4,8} = 0,5416 \\ 0 & 4,8 & 0,9 & -8,7 & L3 = L3 - m_{32} * L2 \\ 0 & 2,6 & 9,8 & 4,6 & \Rightarrow \end{array} \right| \left| \begin{array}{cccc|c} 10 & 2 & 1 & 7 & \\ 0 & 4,8 & 0,9 & -8,7 & \text{Logo:} \\ 0 & 0 & 9,31 & 9,31 & \end{array} \right|$$

$$\begin{cases} x_3 = 9,31/9,31 & x_1 = 1 \\ x_2 = (-8,7 - 0,9 * x_3)/4,8 & x_2 = -2 \\ x_1 = (7 - 1 * x_3 - 2 * x_2)/10 & x_3 = 1 \end{cases}$$

MN aplicados à Engenharia

Método de Eliminação de Gauss-Jordan

- A equação pivô é normalizada com a **divisão de todos os seus termos pelo coeficiente pivô**. Isso faz com que o coeficiente pivô seja igual a 1.
 - A equação pivô é utilizada na eliminação dos elementos fora da diagonal principal em TODAS as demais equações, ou seja, o processo de eliminação aplica-se às equações acima e abaixo da equação pivô.
 - A manipulação da equação pivô segue a mesma estrutura de Gauss para a obtenção dos elementos elementos fora da **diagonal**.

Figura: Gauss-Jordan

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ a_{21} & a_{22} & a_{23} & a_{24} & b_2 \\ a_{31} & a_{32} & a_{33} & a_{34} & b_3 \\ a_{41} & a_{42} & a_{43} & a_{44} & b_4 \end{bmatrix}$$

Gauss-Jordan

$$\begin{bmatrix} 1 & 0 & 0 & 0 & b'_1 \\ 0 & 1 & 0 & 0 & b'_2 \\ 0 & 0 & 1 & 0 & b'_3 \\ 0 & 0 & 0 & 1 & b'_4 \end{bmatrix}$$

Fonte: GILAT,(2008)

MN aplicados à Engenharia

Resolva o sistema linear:

$$\begin{cases} 10x_1 + 2x_2 + x_3 = 7 \\ x_1 + 5x_2 + x_3 = -8 \\ 2x_1 + 3x_2 + 10x_3 = 6 \end{cases}$$

(2) Método de Gauss-Jordan

$$\left| \begin{array}{cccc} 10 & 2 & 1 & 7 \\ 1 & 5 & 1 & -8 \\ 2 & 3 & 10 & 6 \end{array} \right| \rightarrow \left| \begin{array}{cccc} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 1 \end{array} \right| \rightarrow \begin{cases} x_1 = 1 \\ x_2 = -2 \\ x_3 = 1 \end{cases}$$

MN aplicados à Engenharia

Fatoração LU

A base deste método, assim como o método da eliminação de Gauss, é o uso de uma propriedade elementar de sistemas de equações lineares que estabelece o seguinte:

- A solução de um sistema linear $Ax = b$ não se altera se o submetivermos a uma sequência de operações tais como:
 - * Multiplicação de uma equação (linha) por uma constante não nula;
 - * Soma do múltiplo de uma equação a outra;
 - * Troca de posição de duas ou mais equações.
- Seja o sistema linear $Ax = b$, este processo de fatoração consiste em decompor a matriz A em um produto de dois ou mais fatores, na forma:

$$[A] = [L][U]$$

Onde:

- * L = Matriz triangular inferior (Decomposição LU)
- * U = Matriz triangular superior (Eliminação de Gauss)

MN aplicados à Engenharia

Fatoração LU

- Seja o sistema linear: $\begin{cases} Ax = b \\ LUx = b \end{cases}$
- Considerando, $Ux = y$ temos então dois sistemas: $Ly = b$
- Decomposição de Matriz $[a]$:

$$\left[\begin{array}{cccc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{array} \right] = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ m_{21} & 1 & 0 & 0 \\ m_{31} & m_{32} & 1 & 0 \\ m_{41} & m_{42} & m_{43} & 1 \end{array} \right] \left[\begin{array}{cccc} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{array} \right]$$

- * Os elementos diagonais de $[L]$ são todos iguais a 1 e os elementos abaixo desta são os multiplicadores m_{ij} que multiplicam a equação pivô quando ela é usada para eliminar os elementos abaixo do coeficiente pivô no método de Gauss;
- * A matriz triangular superior $[U]$ é a matriz de coeficientes $[a]$ obtida ao final do procedimento de Eliminação de Gauss.

MN aplicados à Engenharia

Resolva o sistema linear:

$$\begin{cases} 10x_1 + 2x_2 + x_3 = 7 \\ x_1 + 5x_2 + x_3 = -8 \\ 2x_1 + 3x_2 + 10x_3 = 6 \end{cases}$$

(3) Fatoração LU

- Aplicar o método de Eliminação de Gauss na Matriz A para obtenção de L e U;
- Então, para a matriz A dada, tem-se:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0,1 & 1 & 0 \\ 0,2 & 0,54 & 1 \end{pmatrix} \text{ e } U = \begin{pmatrix} 10 & 2 & 1 \\ 0 & 4,8 & 0,9 \\ 0 & 0 & 9,31 \end{pmatrix}$$

- Resolver os dois sistemas lineares equivalentes obtidos pelas substituição $A = LU$

- $Ly = b; L = \begin{vmatrix} 1 & 0 & 0 \\ 0,1 & 1 & 0 \\ 0,2 & 0,54 & 1 \end{vmatrix} . \begin{vmatrix} y_1 \\ y_2 \\ y_3 \end{vmatrix} = \begin{vmatrix} 7 \\ -8 \\ 6 \end{vmatrix} \rightarrow y = \begin{vmatrix} 7 \\ -8,7 \\ 9,31 \end{vmatrix}$

- $Ux = y; U = \begin{vmatrix} 10 & 2 & 1 \\ 0 & 4,8 & 0,9 \\ 0 & 0 & 9,31 \end{vmatrix} . \begin{vmatrix} x_1 \\ x_2 \\ x_3 \end{vmatrix} = \begin{vmatrix} 7 \\ -8,7 \\ 9,31 \end{vmatrix} \rightarrow x = \begin{vmatrix} 1 \\ -2 \\ 1 \end{vmatrix}$

MN aplicados à Engenharia

Fatoração LU - Método de Crout

As matrizes L e U são da forma

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 & 0 \\ L_{21} & L_{22} & 0 & 0 \\ L_{31} & L_{32} & L_{33} & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} \end{bmatrix} \begin{bmatrix} 1 & U_{12} & U_{13} & U_{14} \\ 0 & 1 & U_{23} & U_{24} \\ 0 & 0 & 1 & U_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Resolvendo o produto L e U, tem-se:

$$= \begin{bmatrix} L_{11} & L_{11}U_{12} & L_{11}U_{13} & L_{11}U_{14} \\ L_{21} & L_{21}U_{12} + L_{22} & L_{21}U_{13} + L_{22}U_{23} & L_{21}U_{14} + L_{22}U_{24} \\ L_{31} & L_{31}U_{12} + L_{32} & L_{31}U_{13} + L_{32}U_{23} + L_{33} & L_{31}U_{14} + L_{32}U_{24} + L_{33}U_{34} \\ L_{41} & L_{41}U_{12} + L_{42} & L_{41}U_{13} + L_{42}U_{23} + L_{43} & L_{41}U_{14} + L_{42}U_{24} + L_{43}U_{34} + L_{44} \end{bmatrix}$$

MN aplicados à Engenharia

Fatoração LU - Método de Crout

Igualando os elementos correspondentes em ambos os lados da equação, pode-se encontrar os elementos das matrizes $[L]$ e $[U]$, como sendo:

- Na primeira linha:

$$L_{11} = a_{11}; U_{12} = \frac{a_{12}}{L_{11}}; U_{13} = \frac{a_{13}}{L_{11}}; U_{14} = \frac{a_{14}}{L_{11}}$$

- Na segunda linha:

$$\begin{aligned} L_{21} &= a_{21}; L_{22} = a_{22} - L_{21}U_{12}; \\ U_{23} &= \frac{a_{23} - L_{21}U_{13}}{L_{22}}; U_{24} = \frac{a_{24} - L_{21}U_{14}}{L_{22}} \end{aligned}$$

E assim sucessivamente!

MN aplicados à Engenharia

Inversa de uma Matriz

- A inversa de uma matriz quadrada $[a]$ é a matriz $[a]^{-1}$ tal que o produto das duas matrizes fornece a identidade.

$$[A][A]^{-1} = [I]$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \cdot \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- A solução da identidade é obtida através da solução das quatro equações:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \cdot \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \\ x_{41} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \cdot \begin{bmatrix} x_{12} \\ x_{22} \\ x_{32} \\ x_{42} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \cdot \begin{bmatrix} x_{13} \\ x_{23} \\ x_{33} \\ x_{43} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \cdot \begin{bmatrix} x_{14} \\ x_{24} \\ x_{34} \\ x_{44} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

MN aplicados à Engenharia

Método Iterativos

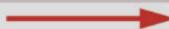
Consiste em colocar cada incógnita x_i em função das outras variáveis, conforme segue:

Figura: Iterativos

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 &= b_3 \\ a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 &= b_4 \end{aligned}$$

(a)

Escrevendo as equações
em uma forma explícita



$$\begin{aligned} x_1 &= [b_1 - (a_{12}x_2 + a_{13}x_3 + a_{14}x_4)]/a_{11} \\ x_2 &= [b_2 - (a_{21}x_1 + a_{23}x_3 + a_{24}x_4)]/a_{22} \\ x_3 &= [b_3 - (a_{31}x_1 + a_{32}x_2 + a_{34}x_4)]/a_{33} \\ x_4 &= [b_4 - (a_{41}x_1 + a_{42}x_2 + a_{43}x_3)]/a_{44} \end{aligned}$$

(b)

Fonte: GILAT,(2008)

Destacam-se os métodos:

- **Jacobi** - os valores das incógnitas no lado direito da equação são atualizados todos de uma vez no final de cada iteração.
- **Gauss-Seidel** - onde o valor de cada incógnita é atualizado (e usado no cálculo da nova estimativa das demais incógnitas dentro da mesma iteração) assim que se calcula uma nova estimativa para essa incógnita.

MN aplicados à Engenharia

Método Iterativos

- O processo de solução começa com a escolha de valores iniciais para as incógnitas (primeira solução estimada).
- Na primeira iteração, a primeira solução assumida é substituída no lado direito das equações, e os novos valores calculados para as incógnitas formam a segunda solução estimada.
- Na segunda iteração, a segunda solução é substituída de volta nas equações para que novos valores sejam obtidos para as incógnitas, e isso constitui a terceira solução estimada.
- As iterações continuam da mesma forma e, quando o método "dá certo", as soluções obtidas durante as iterações sucessivas convergem para a solução real.

Destacam-se os métodos:

- **Jacobi** - os valores das incógnitas no lado direito da equação são atualizados todos de uma vez no final de cada iteração.
- **Gauss-Seidel** - onde o valor de cada incógnita é atualizado (e usado no cálculo da nova estimativa das demais incógnitas dentro da mesma iteração) assim que se calcula uma nova estimativa para essa incógnita.

MN aplicados à Engenharia

Método Iterativos

Em um sistema com n equações:

- As equações explícitas para as incógnitas $[x_i]$ são:

$$x_i = \frac{1}{a_{ii}} \left[b_i - \left(\sum_{j=1, j \neq i}^n a_{ij} x_j \right) \right] \quad (1)$$

$$i = 1, 2, \dots, n$$

- Uma condição suficiente para a convergência ocorre se, em cada uma das linhas da matriz de coeficientes $[a]$, o valor absoluto do elemento diagonal for maior que a soma dos valores absolutos dos elementos fora da diagonal.

$$|a_{ii}| \leq \sum_{j=1, j \neq i}^n |a_{ij}|$$

- * Essa condição é suficiente, mas não necessária para a convergência do método iterativo e, quando ocorre a matriz $[a]$ é classificada como diagonalmente dominante.

MN aplicados à Engenharia

Método Iterativos de Jacobi

- Um valor inicial é escolhido para cada uma das incógnitas, $x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}$ assumindo-se que o valor inicial de todas seja zero, caso não haja informações iniciais a respeito.
- A segunda estimativa da solução, $x_1^{(2)}, x_2^{(2)}, \dots, x_n^{(2)}$, é calculada com a substituição do lado direito da equação 1 de modo que se tem:

$$x_i^{(2)} = \frac{1}{a_{ii}} \left[b_i - \left(\sum_{j=1, j \neq i}^n a_{ij} x_j^{(1)} \right) \right], \quad i = 1, 2, \dots, n$$

- Em geral, a $(k+1)$ -ésima estimativa da solução é calculada a partir da k -ésima estimativa usando:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left[b_i - \left(\sum_{j=1, j \neq i}^n a_{ij} x_j^{(k)} \right) \right], \quad i = 1, 2, \dots, n$$

- As iterações continuam até que as diferenças entre os valores obtidos nas iterações sucessivas sejam pequenas ou que o valor absoluto do erro relativo estimado de todas as incógnitas for menor que um valor ε predeterminado:

$$\left| \frac{x_i^{(k+1)} - x_i^{(k)}}{x_i^{(k)}} \right| \leq \varepsilon, \quad i = 1, 2, \dots, n$$

MN aplicados à Engenharia

Resolva o sistema linear:

$$\begin{cases} 10x_1 + 2x_2 + x_3 = 7 \\ x_1 + 5x_2 + x_3 = -8 \\ 2x_1 + 3x_2 + 10x_3 = 6 \end{cases}$$

(4) Método de Jacobi

- Descrição de equações x_i :

- $x_1 = (7 - (2x_2 + x_3))/10$
- $x_2 = (-8 - (x_1 + x_3))/5$
- $x_3 = (6 - (2x_1 + 3x_2))/10$
- Ponto de partida: $x_1 = 0, x_2 = 0, x_3 = 0$

- Segunda iteração $x_i^{(k=2)}$:

- $x_1^{(2)} = (7 - (2.0 + 0))/10 = 0,7$
- $x_2^{(2)} = (-8 - (0+0))/5 = -1,6$
- $x_3^{(2)} = (6 - (2.0 + 3.0))/10 = 0,6$
- Após a segunda iteração: $x_1 = 0,7, x_2 = -1,6, x_3 = 0,6$

- Terceira iteração $x_i^{(k=3)}$:

- $x_1^{(3)} = (7 - (2.(-1,6) + 0,6))/10 = 0,96$
- $x_2^{(3)} = (-8 - (0,7 + 0,6))/5 = -1,86$
- $x_3^{(3)} = (6 - (2.(0,7) + 3.(-1,6))/10 = 0,94$
- Após a terceira iteração: $x_1 = 0,96, x_2 = -1,86, x_3 = 0,94$

- Quarta iteração $x_i^{(k=4)}$:

- $x_1^{(4)} = (7 - (2.(-1,86) + 0,94))/10 = 0,978$
- $x_2^{(4)} = (-8 - (0,96 + 0,94))/5 = -1,98$
- $x_3^{(4)} = (6 - (2.(0,96) + 3.(-1,86))/10 = 0,966$
- Após a quarta iteração: $x_1 = 0,978, x_2 = -1,98, x_3 = 0,966$
- ...

MN aplicados à Engenharia

Método Iterativos de Gauss-Seidel

- No método de Gauss-Seidel, valores iniciais são assumidos para as incógnitas x_2, x_3, \dots, x_n , (**exceto x_1**), assumindo-se que o valor inicial de todas seja zero, caso não haja informações iniciais a respeito.
- x_1 é obtido pela substituição dos valores assumidos na equação 1.
- Em seguida, o novo valor de x_2 é obtido pela substituição dos valores assumidos na equação 1 para $i = 2$, e assim por diante até que $i = n$, para concluir a **primeira iteração**.
- Em seguida, a segunda iteração começa com $i = 1$, onde um novo valor é calculado para x_1 , e assim por diante.
- No método de Gauss-Seidel, **os valores atuais das incógnitas são utilizados no cálculo do novo valor da próxima incógnita**.

MN aplicados à Engenharia

Método Iterativos de Gauss-Seidel

- A aplicação da equações (1) no método de Gauss-Seidel leva à fórmula iterativa:

$$x_1^{(k+1)} = \frac{1}{a_{11}} \left[b_1 - \sum_{j=2}^{j=n} a_{1j} x_j^{(k)} \right]$$

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left[b_i - \left(\sum_{j=1}^{j=i-1} a_{ij} x_j^{(k+1)} + \sum_{j=i+1}^{j=n} a_{ij} x_j^{(k)} \right) \right], \quad i = 2, 3, \dots, n-1$$

$$x_n^{(k+1)} = \frac{1}{a_{nn}} \left[b_n - \sum_{j=1}^{j=n-1} a_{nj} x_j^{(k+1)} \right]$$

- Note que os valores das incógnitas na iteração $k+1$, são calculados obtidos na iteração $k+1$ para $j \leq i$ e usando os valores usando os valores para $j \leq i$.
- O critério de interrupção das iterações é o mesmo utilizado no método de Jacobi.
- O método de Gauss-Seidel converge mais rápido do que o método de Jacobi e requer menos memória computacional quando programado

MN aplicados à Engenharia

Resolva o sistema linear:

$$\begin{cases} 10x_1 + 2x_2 + x_3 = 7 \\ x_1 + 5x_2 + x_3 = -8 \\ 2x_1 + 3x_2 + 10x_3 = 6 \end{cases}$$

(5) Método de Gauss-Seidal

- Descrição de equações x_i :

- $x_1 = (7 - (2x_2 + x_3))/10$
- $x_2 = (-8 - (x_1 + x_3))/5$
- $x_3 = (6 - (2x_1 + 3x_2))/10$
- Ponto de partida: $x_1 = 0, x_2 = 0, x_3 = 0$

- Segunda iteração $x_i^{(k=2)}$:

- $x_1^{(2)} = (7 - (2.0 + 0))/10 = 0,7$
- $x_2^{(2)} = (-8 - (0,7 + 0))/5 = -1,74$
- $x_3^{(2)} = (6 - (2.0,7 + 3. -1,74))/10 = -0,982$

- Após a segunda itera-

- Terceira iteração $x_i^{(k=3)}$:

- $x_1^{(3)} = (7 - (2.(-1,74) + 0,982))/10 = 0,9498$
- $x_2^{(3)} = (-8 - (0,9498 + 0,982))/5 = -1,9863$
- $x_3^{(3)} = (6 - (2.0,7 + 3. -1,9863))/10 = -0,062$
- Após a terceira iteração: $x_1 = 0,9498, x_2 = -1,9863, x_3 = 1,005$

- Quarta iteração $x_i^{(k=4)}$:

- $x_1^{(4)} = (7 - (2. - 1,9863 + 1,005))/10 = 0,9966$
- $x_2^{(4)} = (-8 - (0,9966 + 1,005))/5 = -2,000$
- $x_3^{(4)} = (6 - (2.0,9966 + 3. -2,000))/10 = 1,000$
- Após a quarta iteração: $x_1 = 0,9966, x_2 = -2,000, x_3 = 1,000$

MN aplicados à Engenharia

Resultados Jacobi e Gauss-Seidal

Figura: Comparativo de iterações para erro de 0,001

***** Método Iterativo de Jacobi *****			
k	x1	x2	x3
2	0.700000	-1.600000	0.600000
3	0.960000	-1.860000	0.940000
4	0.978000	-1.980000	0.966000
5	0.999400	-1.988800	0.998400
6	0.997920	-1.999560	0.996760
7	1.000236	-1.998936	1.000284
8	0.999759	-2.000104	0.999634
9	1.000057	-1.999878	1.000079
10	0.999968	-2.000027	0.999952
***** Método Iterativo de Gauss-Seidal *****			
k	x1	x2	x3
2	0.700000	-1.740000	0.982000
3	0.949800	-1.986360	1.005948
4	0.996677	-2.000525	1.000822
5	1.000023	-2.000169	1.000046
6	1.000029	-2.000015	0.999999

Fonte: AUTOR,(2020)

Caso erro seja superado antes de 20 iterações o *script* interrompe as iterações!

MN aplicados à Engenharia

Script Jacobi e Gauss-Seidal

Figura: Script comparativo de iterações com Erro de 0,001

```

1 % Sistema Linear Iteração de Jacobi e Gauss-Seidel.
2 clc; clear all; format bank
3 %*****
4 fprintf('\n***** Método Iterativo de Jacobi *****\n\n');
5 x1=0;x2=0;x3=0;x4=0; n=20; err=0.0001;
6 a=0;b=0;c=0;
7 disp('k          x1          x2          x3 ');
8
9 for(k=2:n)
10    a=(7-(2*x2+x3))/10;           % Atualiza a e usa Xi anterior
11    b=(-8-(x1+x3))/5;           % Atualiza b e usa Xi anterior
12    c=(6-(2*x1+3*x2))/10;       % Atualiza c e usa Xi anterior
13    if((abs(x1-a) < err) && (abs(x2-b)<err) && (abs(x3-c)<err) break end;
14    x1=a;x2=b;x3=c;
15    fprintf('%2.0f \t%-8.6f \t%-8.6f \t%-8.6f \t\n',k,x1,x2,x3);
16 end
17
18 %*****
19 %break
20 %*****
21 fprintf('\n***** Método Iterativo de Gauss-Seidal *****\n\n')
22 x1=0;x2=0;x3=0;x4=0;
23 disp('k          x1          x2          x3 ');
24 for(k=2:n)
25    a=x1; x1=(7-(2*x2+x3))/10;      % Guarda x1(k-1) em a;
26    b=x2; x2=(-8-(x1+x3))/5;        % Guarda x2(k-1) em b;
27    c=x3; x3=(6-(2*x1+3*x2))/10;    % Guarda x3(k-1) em c;
28    if((abs(x1-a) < err) && (abs(x2-b)<err) && (abs(x3-c)<err) break end;
29    fprintf('%2.0f \t%-8.6f \t%-8.6f \t%-8.6f \t\n',k,x1,x2,x3);

```

Fonte: AUTOR,(2020)

MN aplicados à Engenharia

Recursividade Octave / MatLab

Sejam as matrizes: $A_{(n \times n)}$ e $b_{(n \times 1)}$, tal que $Ax = b$;

- $x = A \backslash b$ - A divisão à esquerda - equivalente a $x = \text{linsolve}(A, b)$
- $x = b^t / A^t$ - A divisão à direita
- $x = \text{inv}(A) * B$ ou $x = A^{-1} * b$
- $\text{rref}([A \ b])$ - Solução pelo método de Gauss-Jordan
- $[L, U, P] = \text{lu}(A)$ - Solução pela fatoração LU
 L - é uma matriz triangular inferior;
 U - é uma matriz triangular superior;
 P - é a matriz de permutação.

Organização

1 Métodos

Ajustes de curvas

Conceitos

Regressão Linear - mínimos quadrados

Linearização

Regressão Polinomial

Polinômio interpolador de Lagrange

Polinômio interpolador de Newton

Splines

Splines Quadráticas

Splines Cúbicas

Resolução de Sistemas Lineares

Introdução

Método de Eliminação de Gauss

Método de Eliminação de Gauss-Jordan

Método de Fatoração LU

LU - Método de Crout

Inversa de uma Matriz

Conceito de Métodos Iterativos

Método de Jacobi

Método de Gauss-Seidel

MN aplicados à Engenharia

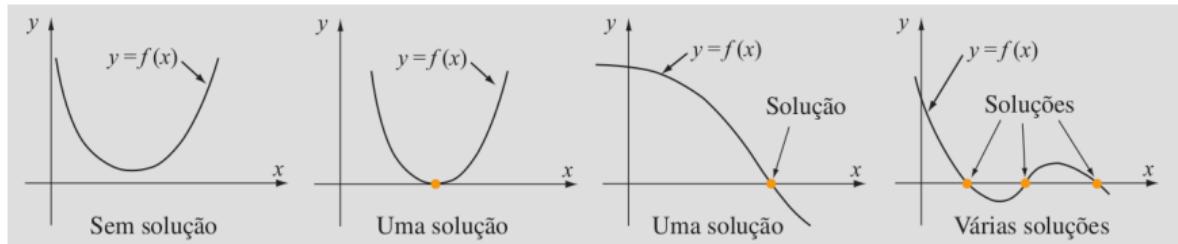
- Apresentar conteúdo de Resolução de Equações Não-Lineares
 - Método da Bisseção
 - Método Regula-Falsi
 - Método Newton-Raphson
 - Método da Secante

MN aplicados à Engenharia

Fundamentos

- Equações precisam ser resolvidas em todas áreas da engenharia.
- Consiste em determinar o(s) valores de x tal que $f(x) = 0$.

Figura: Existência de solução em curvas



Fonte: GILAT,(2008)

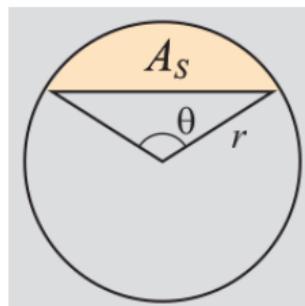
- Um número real ξ é um zero da função $f(x)$ ou uma raiz da equação $f(x) = 0$ se $f(\xi) = 0$.

MN aplicados à Engenharia

Fundamentos

- Em alguns casos as soluções podem ser reais ou imaginárias.
- Seja o exemplo: A área do segmento sombreado é dado por:

Figura: Área de setor



- $A_s = \frac{1}{2}r^2(\theta - \sin(\theta))$
- A determinação de θ em função de A_s e r conhecidos não é possível através de solução analítica.
- Usam-se os métodos numéricos para encontrar uma solução aproximada para tanto.

Fonte: GILAT, (2008)

MN aplicados à Engenharia

Zeros de funções reais

- Quais os zeros das funções:

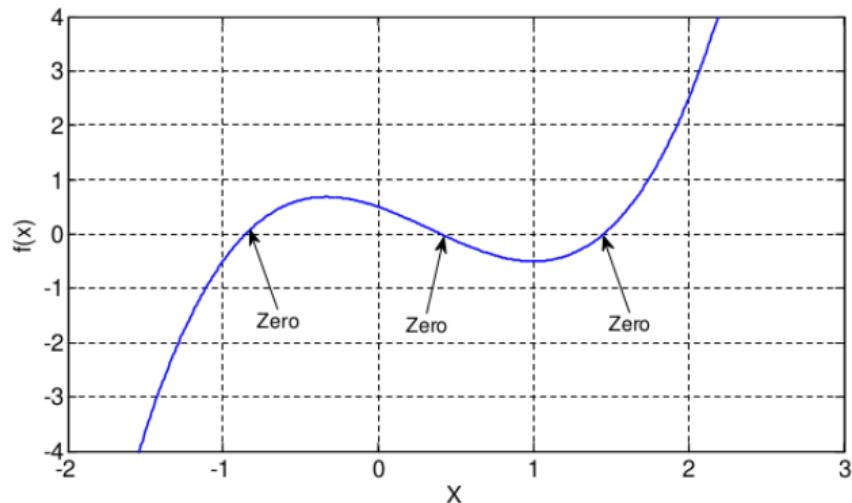
- $f(x) = x - 3;$
- $f(x) = x^2 - 4x - 5;$
- $f(x) = x^3 - x^2 - x + 1/2$

MN aplicados à Engenharia

Zeros de funções reais

- Seja uma função $f(x)$ contínua em um intervalo I . Denomina-se zero desta função todo $x \in I$, tal que $f(x) = 0$.

Figura: $f(x) = x^3 - x^2 - x + 1/2$



Fonte: DIAS,(2019)

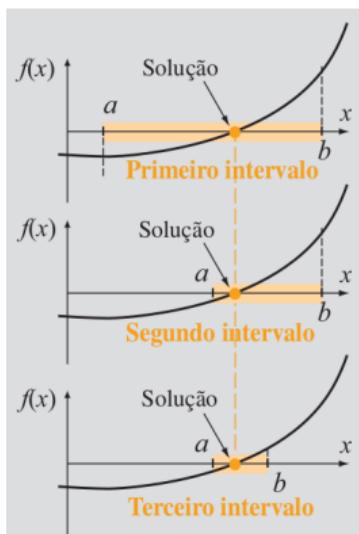
MN aplicados à Engenharia

Como obter as raízes de uma equação qualquer?

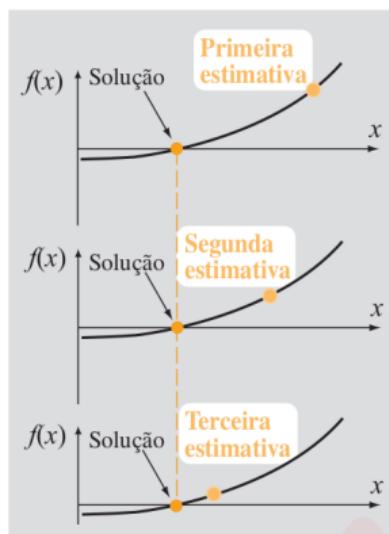
- Os métodos numéricos de resolução são divididos nos grupos:

Figura: Método de obtenção das raízes

(a) Método de confinamento



(b) Método aberto



MN aplicados à Engenharia

Boas práticas

- Definir a faixa de busca ou isolamento das raízes;
- Estabelecer uma faixa de erro aceitável e tolerância para a aproximação;
- Refinar a busca por meio de um processo iterativo até que a solução tenha uma precisão prefixada;

MN aplicados à Engenharia

- **Estimação de Erros em Soluções Numéricas - Erro real**

Seja x_{TS} a solução exata tal que $f(x_{TS}) = 0$, e seja x_{NS} uma solução numérica aproximada tal que $f(x_{NS}) = \varepsilon$. Onde ε é um número muito pequeno.

- Deve ser criado algum critério para verificar se uma solução é precisa.
- *Erro real*

$$\text{erro real} = x_{TS} - x_{NS}$$

Este critério não é tão útil, pois, em princípio, x_{TS} não é conhecido!

- **Estimação de Erros em Soluções Numéricas - Tolerância em $f(x)$:**

- Ao invés de considerar o erro na solução, verifica-se o desvio de $f(x_{NS})$ em relação a zero.
- Tolerância em f : $\text{tolerancia} = |f(x_{TS}) - f(x_{NS})| = |0 - \varepsilon| = |\varepsilon|$

MN aplicados à Engenharia

- **Estimação de Erros em Soluções Numéricas - Tolerância na solução:**

- É útil quando métodos de confinamento são usados na determinação numérica.
- Assume-se que a solução numérica é o ponto central de um intervalo com uma tolerância tal que:

$$x_{NS} = \left(\frac{a+b}{2} \right)^{\pm \left(\frac{a-b}{2} \right)}$$

- **Estimação de Erros em Soluções Numéricas - Erro relativo estimado**

- Usado quando as soluções numéricas são calculadas iterativamente.
- É dado por:

$$\text{erro relativo estimado} = \left| \frac{x_{NS}^n - x_{NS}^{(n-1)}}{x_{NS}^{(n-1)}} \right|$$

MN aplicados à Engenharia

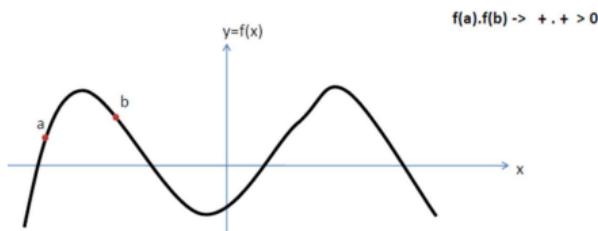
Fase 01 -Confinamento das raízes

- **Teorema 1.** Seja uma função contínua no intervalo $[a, b]$.
Se $f(a)f(b) \leq 0$, então \exists pelo menos um ponto $x = x_0$ entre a e b que é solução de $f(x) = 0$.

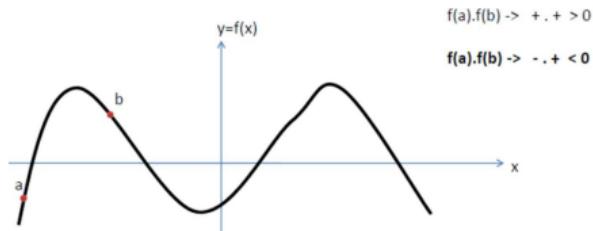
MN aplicados à Engenharia

Figura: Existência de zeros na função

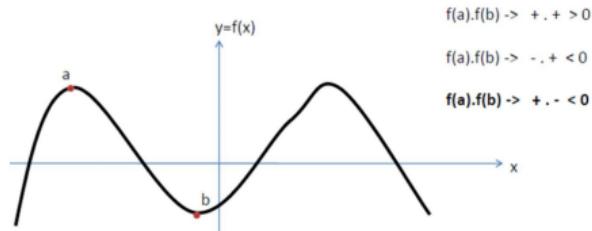
(a) caso 1



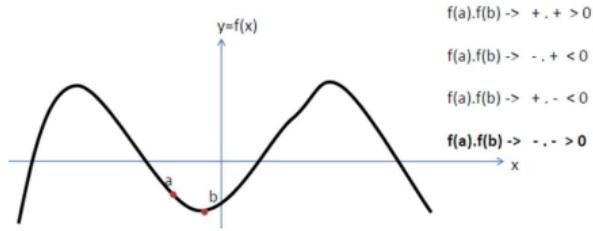
(b) caso 2



(c) caso 3



(d) caso 4



Fonte: DIAS,(2019)

MN aplicados à Engenharia

Fase 01 -Confinamento das raízes

Ex.: Para encontrar os zeros da função $f(x) = x^3 - 9x + 3$ pode-se construir uma tabela de valores para $f(x)$ e se analisar os sinais, conforme a tabela abaixo.

Figura: Confinamento de raízes

$f(-4)f(-3) = (-25).(3) = -75 < 0$	$f(2)f(3) = (-7).(3) = -21 < 0$																																				
<table border="1"> <thead> <tr> <th>x</th><th>-4</th><th>-3</th><th>-2</th><th>-1</th><th>0</th><th>1</th><th>2</th><th>3</th></tr> </thead> <tbody> <tr> <td>$f(x)$</td><td>-25</td><td>3</td><td>13</td><td>11</td><td>3</td><td>-5</td><td>-7</td><td>3</td></tr> </tbody> </table>	x	-4	-3	-2	-1	0	1	2	3	$f(x)$	-25	3	13	11	3	-5	-7	3	<table border="1"> <thead> <tr> <th>x</th><th>-4</th><th>-3</th><th>-2</th><th>-1</th><th>0</th><th>1</th><th>2</th><th>3</th></tr> </thead> <tbody> <tr> <td>$f(x)$</td><td>-25</td><td>3</td><td>13</td><td>11</td><td>3</td><td>-5</td><td>-7</td><td>3</td></tr> </tbody> </table>	x	-4	-3	-2	-1	0	1	2	3	$f(x)$	-25	3	13	11	3	-5	-7	3
x	-4	-3	-2	-1	0	1	2	3																													
$f(x)$	-25	3	13	11	3	-5	-7	3																													
x	-4	-3	-2	-1	0	1	2	3																													
$f(x)$	-25	3	13	11	3	-5	-7	3																													

$f(0)f(1) = (3).(-5) = -15 < 0$

Fonte: DIAS,(2019)

Portanto, nos intervalos $[-4, -3]$, $[0, 1]$ e $[2, 3]$, \exists pelo menos uma raiz real da função. Como trata-se de um polinômio do 3 grau, pode-se dizer que existe apenas uma raiz em cada intervalo.

MN aplicados à Engenharia

Fase 02 -Confinamento das raízes

Critério de Parada:

Considerando \bar{x} a raiz aproximada com precisão ε , o qual normalmente é da ordem de 10^{-6} :

- (i) $|\bar{x} - \xi| < \epsilon$
- (ii) $|f(\bar{x})| < \epsilon$

Como realizar o teste (i), uma vez desconhecido o ξ ?

Uma forma é reduzir o intervalo que contém a raiz a cada iteração.

Seja $[a, b]$ tal que $\xi \in [a, b]$ $(b - a) < \epsilon$.

Então $\forall x \in [a, b], |x - \varepsilon| < \xi$ e, portanto, $\forall x \in [a, b]$ pode ser tomado como \bar{x} .

MN aplicados à Engenharia

Fase 02 -Confinamento das raízes (Método da Bisseção)

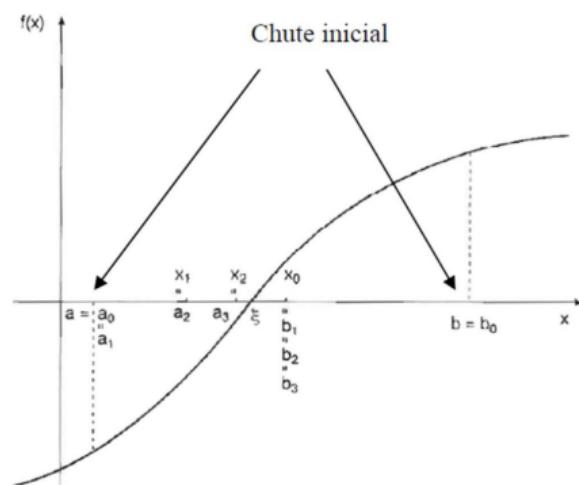
Para se aproximar de uma raiz, o princípio da bisseção consiste em reduzir o intervalo inicial testando o sinal de $f(x)$ para o ponto médio do intervalo.

- Considerando o intervalo $[a, b]$, $x = \frac{a+b}{2}$
- Se $f(a).f(x) < 0$, o novo intervalo é $[a, \frac{a+b}{2}]$
- Se $f(b).f(x) < 0$, o novo intervalo é $[\frac{a+b}{2}, b]$

MN aplicados à Engenharia

Fase 02 - Refinamento (Método da Bisseção)

Figura: Método da Bissecção



$$x_0 = \frac{a_0 + b_0}{2} \quad \left\{ \begin{array}{l} f(a_0) < 0 \\ f(b_0) > 0 \\ f(x_0) > 0 \end{array} \right\} \quad \Rightarrow \quad \left\{ \begin{array}{l} \xi \in (a_0, x_0) \\ a_1 = a_0 \\ b_1 = x_0 \end{array} \right.$$

$$x_1 = \frac{a_1 + b_1}{2} \quad \left\{ \begin{array}{l} f(a_1) < 0 \\ f(b_1) > 0 \\ f(x_1) < 0 \end{array} \right. \quad \Rightarrow \quad \left\{ \begin{array}{l} \xi \in (x_1, b_1) \\ a_2 = x_1 \\ b_2 = b_1 \end{array} \right.$$

$$x_2 = \frac{a_2 + b_2}{2} \quad \left\{ \begin{array}{l} f(a_2) < 0 \\ f(b_2) > 0 \\ f(x_2) < 0 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \xi \in (x_2, b_2) \\ a_3 = x_2 \\ b_3 = b_2 \end{array} \right.$$

$$[a_i \longleftrightarrow x_i \longleftrightarrow b_i]$$

MN aplicados à Engenharia

Método da Bisseção

Figura: *Script* Método da Bisseção

```

1 % Solução de equações não lineares
2 clear all;clc; pkg load symbolic; format bank
3 %*****
4 F=inline('8-4.5*(x-sin(x))');
5 %*****
6 a=2; b=3; % Kick-off
7 imax=50; tol=0.001;
8 %*****
9 disp("Método da Bisseção!")
10 fprintf("\niteração \ta\tb\tc\tFa\tFb\tFx\n")
11 tic
12 for(i=1:imax)
13 %*****
14 x=(a+b)/2; toli=(b-a)/2;
15 fprintf("%5d\t%1.4f %8.4f%8.4f%8.4f%8.4f%8.4f\n",i,a,b,x,F(a),F(b),F(x))
16 %*****
17 if (F(a)*F(x)<0) a=a; b=x; end %Raiz entre a e xmed => novo 'b'
18 if (F(a)*F(x)>0) a=x; b=b; end %Raiz entre b e xmed => novo 'a'
19 %*****
20 if (tol<tol)
21 fprintf("\nSolução %.4f alcançada após %d iterações! \n",x,i);
22 break
23 end
24 end
25
26 fprintf("Tempo de processamento t=%f(s)\n\n",toc)
27 %*****

```

MN aplicados à Engenharia

Método da Bisseção

Figura: Resultados Método da Bisseção

Método da Bisseção!

iteração	a	b	c	Fa	Fb	Fx
1	2.0000	3.0000	2.5000	3.0918	-4.8650	-0.5569
2	2.0000	2.5000	2.2500	3.0918	-0.5569	1.3763
3	2.2500	2.5000	2.3750	1.3763	-0.5569	0.4341
4	2.3750	2.5000	2.4375	0.4341	-0.5569	-0.0557
5	2.3750	2.4375	2.4062	0.4341	-0.0557	0.1907
6	2.4062	2.4375	2.4219	0.1907	-0.0557	0.0678
7	2.4219	2.4375	2.4297	0.0678	-0.0557	0.0062
8	2.4297	2.4375	2.4336	0.0062	-0.0557	-0.0248
9	2.4297	2.4336	2.4316	0.0062	-0.0248	-0.0093
10	2.4297	2.4316	2.4307	0.0062	-0.0093	-0.0016

Solução 2.4307 alcançada após 10 iterações!

Tempo de processamento t=0.002587(s)

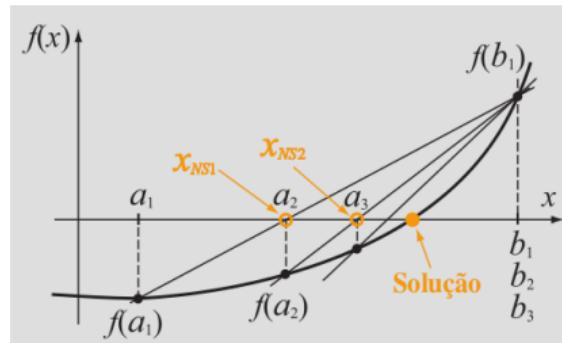
Fonte: AUTOR,(2020)

MN aplicados à Engenharia

Fase 02 - Refinamento (Método regula falsi)

- Também chamado de **falsa posição** ou de interpolação linear.
- É um método de confinamento usado para se obter a solução de uma equação $f(x) = 0$ quando se tem conhecimento que a solução está dentro de um intervalo $[a, b]$ e $f(x)$ é contínua.

Figura: Método da Bisseção



Fonte: GILAT,(2008)

MN aplicados à Engenharia

Fase 02 - Refinamento (Método regula falsi)

- Para um intervalo $[a, b]$, a equação da linha reta que conecta os dois pontos $(b, f(b))$ e $(a, f(a))$ é dada por:

$$y = \frac{f(b) - f(a)}{b - a}(x - b) + f(b)$$

- O ponto x_{NS} onde a reta cruza o eixo x é determinado pela equação a seguir, considerando $f(x) = 0$

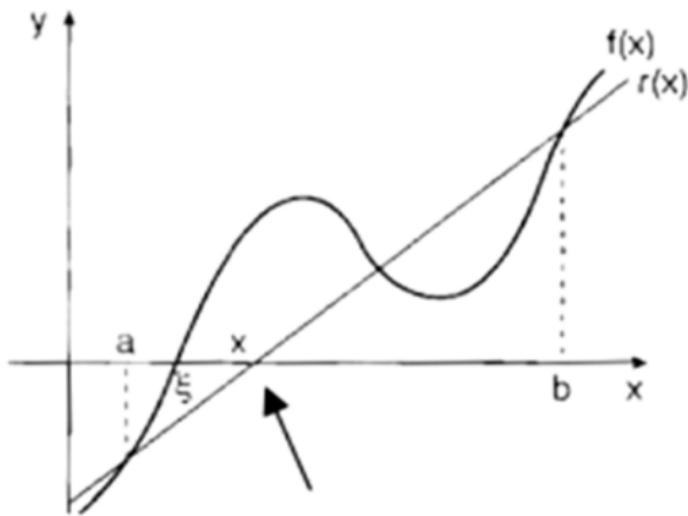
$$x_{NS} = \frac{a.f(b) - b.f(a)}{f(b) - f(a)}$$

MN aplicados à Engenharia

Fase 02 - Refinamento (Método regula falsi)

- Graficamente este ponto x é a interseção entre o eixo das abcissas (ox) e a reta $r(x)$ que passa pelos pontos $(a, f(a))$ e $(b, f(b))$.

Figura: Refinamento



MN aplicados à Engenharia

Método da Falsa Posição

Figura: Script Método da Falsa Posição

```
1 % Solução de equações não lineares
2 clear all;clc; pkg load symbolic; format bank
3 F=inline('8-4.5*(x-sin(x))');
4 %*****
5 a=2; b=3; imax=30; tol=0.001;
6 %*****
7 disp("Método da Falsa Posição!")
8 fprintf("\niteração \ta\tb\tc\tFa\tFb\tFx\n")
9 %*****
10 tic
11 for(i=1:imax)
12 x=(a*F(b)-b*F(a))/(F(b)-F(a));
13 toli=(b-a)/2;
14 fprintf("%5d\t%11.4f %8.4f%8.4f%8.4f%8.4f%8.4f\n",i,a,b,x,F(a),F(b),F(x))
15 if (F(a)*F(x)>0) a=x; else b=x; end
16 %*****
17 if (tol<tol)
18 fprintf("\nSolução %.4f alcançada após %d iterações! \n",x,i);
19 break
20 end
21 end
22 fprintf("Tempo de processamento t=%f(s)\n\n",toc)
```

MN aplicados à Engenharia

Método da Falsa Posição

Figura: Resultados da Falsa Posição

Método da Falsa Posição!						
iteração	a	b	c	Fa	Fb	Fx
1	2.0000	3.0000	2.3886	3.0918	-4.8650	0.3287
2	2.3886	3.0000	2.4273	0.3287	-4.8650	0.0252
3	2.4273	3.0000	2.4302	0.0252	-4.8650	0.0019
4	2.4302	3.0000	2.4304	0.0019	-4.8650	0.0001
5	2.4304	3.0000	2.4305	0.0001	-4.8650	0.0000
6	2.4305	3.0000	2.4305	0.0000	-4.8650	0.0000
7	2.4305	3.0000	2.4305	0.0000	-4.8650	0.0000
8	2.4305	3.0000	2.4305	0.0000	-4.8650	0.0000
9	2.4305	3.0000	2.4305	0.0000	-4.8650	0.0000
10	2.4305	3.0000	2.4305	0.0000	-4.8650	0.0000
11	2.4305	3.0000	2.4305	0.0000	-4.8650	0.0000
12	2.4305	3.0000	2.4305	0.0000	-4.8650	0.0000
13	2.4305	3.0000	2.4305	0.0000	-4.8650	0.0000
14	2.4305	3.0000	2.4305	0.0000	-4.8650	0.0000
15	2.4305	2.4305	2.4305	0.0000	0.0000	0.0000

Solução 2.4305 alcançada após 15 iterações!
Tempo de processamento t=0.003625(s)

MN aplicados à Engenharia

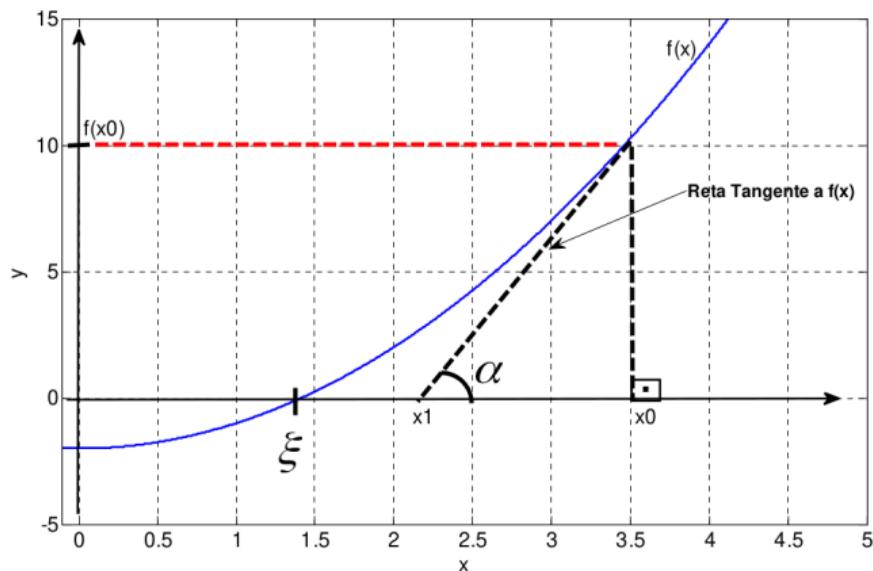
Método de Newton-Raphson

- Dada uma função $f(x)$ contínua no intervalo $[a, b]$ onde existe uma raiz única, é possível determinar uma aproximação de tal raiz a partir de interseção da tangente à curva em um ponto x_0 com o eixo das abscissas.
- x_0 - atribuído em função da geometria do método e do comportamento da curva da equação nas proximidades da raiz.

MN aplicados à Engenharia

Método de Newton-Raphson

Figura: Interpretação Gráfica

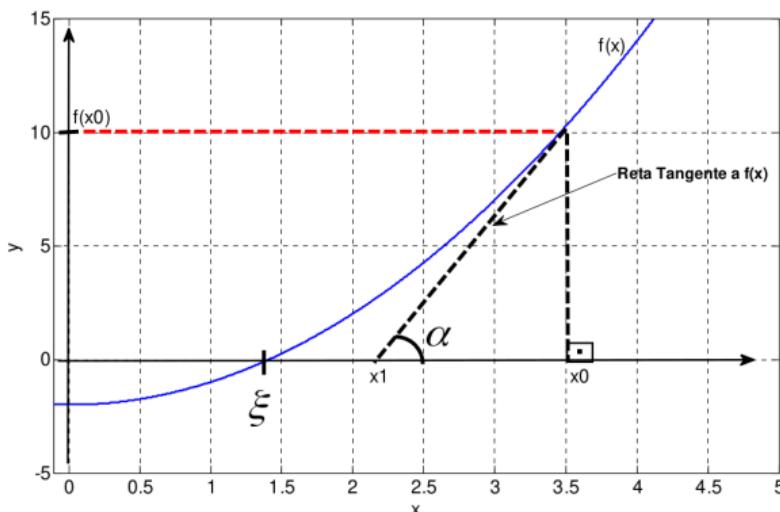


Fonte: DIAS,(2019)

MN aplicados à Engenharia

Método de Newton-Raphson

Figura: Interpretação Gráfica



$$\operatorname{tg}(\alpha) = \frac{f(x_0) - 0}{x_0 - x_1}$$

$$\operatorname{tg}(\alpha) = \frac{f(x_0)}{x_0 - x_1}$$

$$\text{Como: } \operatorname{tg}(\alpha) = f'(x_0)$$

Portanto:

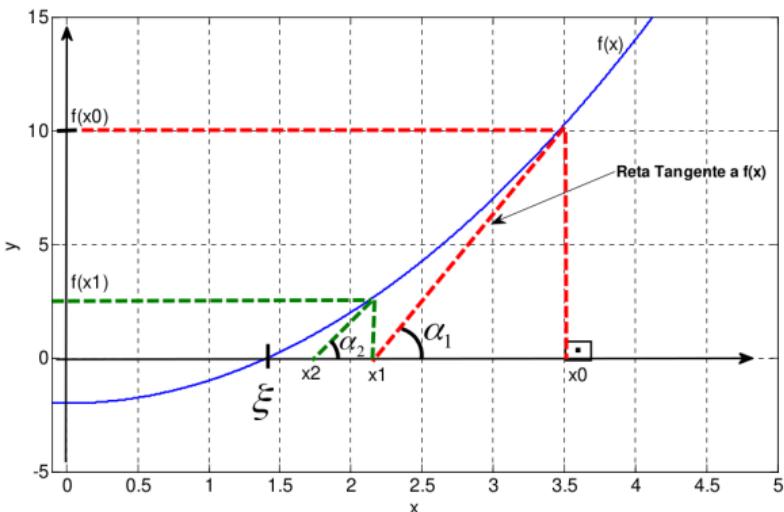
$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Fonte: DIAS,(2019)

MN aplicados à Engenharia

Método de Newton-Raphson

Figura: Interpretação Gráfica



$$\operatorname{tg}(\alpha) = \frac{f(x_1)-0}{x_1-x_2}$$

$$\operatorname{tg}(\alpha) = \frac{f(x_1)}{x_1-x_2}$$

Como: $\operatorname{tg}(\alpha) = f'(x_1)$

Portanto:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

Fonte: DIAS,(2019)

MN aplicados à Engenharia

Método de Newton-Raphson

Se forem realizadas diversas aproximações

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

$$\vdots \quad \vdots \quad \vdots$$

Conclui-se que:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)},$$

onde $k = 0, 1, 2, 3, \dots$

MN aplicados à Engenharia

Método de Newton-Raphson

Testes de parada:

- Erro estimado

$$|\varepsilon| = |x_{k+1} - x_k| < \varepsilon_d$$

- Erro relativo estimado

$$|\varepsilon_R| = \left| \frac{x_{k+1} - x_k}{x_{k+1}} \right| < \varepsilon$$

- Tolerância em $f(x)$

$$\left| f(x_{k+1}) \right| < \varepsilon_d$$

MN aplicados à Engenharia

Método de Newton-Raphson

Algoritmo:

- Avaliar o $f'(x)$
- Usar o x_i para estimar o próximo valor estimado da raiz (x_{i+1})

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

- Encontrar o erro relativo aproximado $| \in_a |$

$$| \in_a | = \left| \frac{x_{i+1} - x_i}{x_{i+1}} \right| x 100$$

- Se $| \in_a | > \in_s$, então volta-se ao passo 2, se não interrompe-se o algoritmo.

MN aplicados à Engenharia

Método de Newton-Raphson

Vantagens:

- Rapidez no processo de convergência
- Desempenho elevado

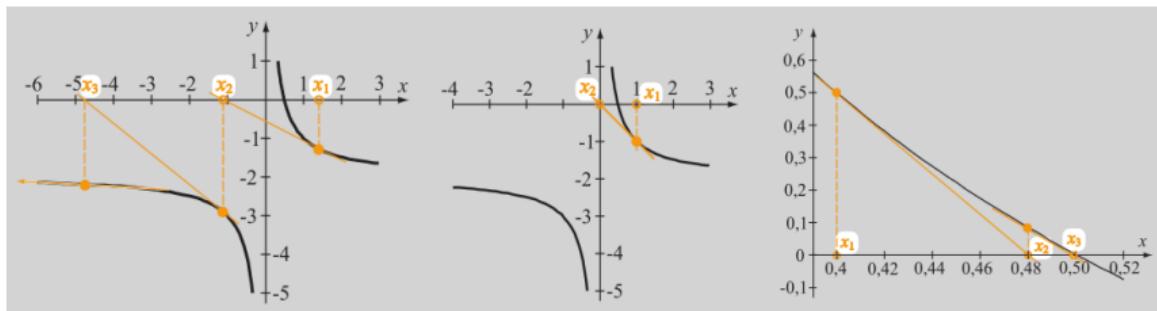
Desvantagens:

- Necessidade da obtenção de $f'(x)$ - o que pode ser impossível em determinados casos;
- O cálculo do valor numérico de $f'(x)$ a cada iteração

MN aplicados à Engenharia

Método de Newton-Raphson Erros de Convergência

Figura: Método de Newton usando diferentes pontos de partida



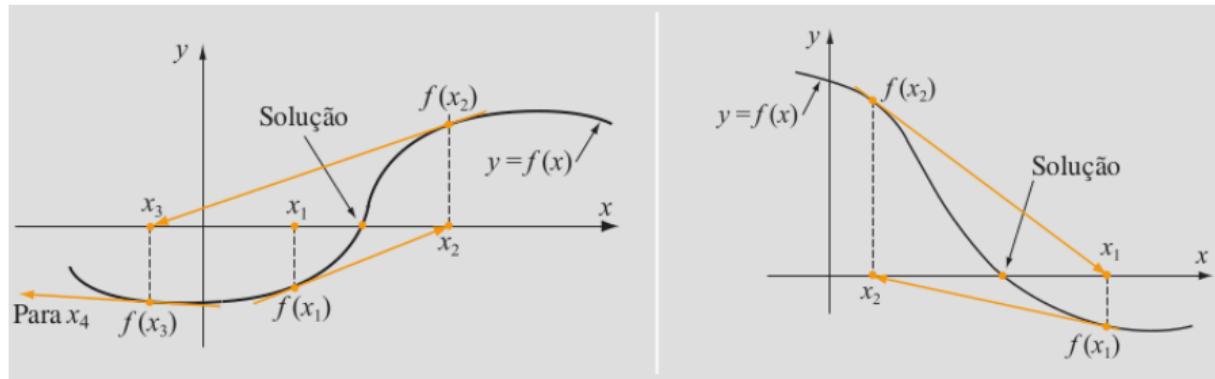
Fonte: GILAT,(2008)

Perceba a relação causa-efeito a partir de ponto de início (x_0) nos itens *a,b e c.*

MN aplicados à Engenharia

Método de Newton-Raphson

Figura: Erros de Convergência



Fonte: GILAT,(2008)

MN aplicados à Engenharia

Método de Newton-Raphson

Figura: Script Método da Newton-Raphson

```

1 clear all;clc; pkg load symbolic; format bank
2 %*****
3 f=@(x)(8-4.5*(x-sin(x))); g=(diff(sym(f))); df=matlabFunction(g);clc;
4 disp("Método de Newton-Raphson")
5 %*****
6 Err=0.001; tol=Err*0.1; imax=30;
7 Xest=2; tic % valor inicial
8 %*****
9 for(i=1:imax)
10    Xsn = Xest-f(Xest)/df(Xest); % Xsn = x(i+1) ; Xest=x(i);
11    %*****
12    if (abs((Xsn-Xest)/Xest) < Err) % Erro < x(i+1)-xi)/xi|
13        Xsn = Xest;
14        fprintf("\nSolução %.4f alcançada após %d iterações! \n",Xsn,i)
15        break
16    end
17    %*****
18    if (abs(f(Xsn))< Err) % Tol < f(xi)
19        fprintf('\nSolução %.4f alcançada com %d iterações e tolerância %.f!\n',
20        Xsn,i,f(Xsn))
21        break
22    end
23    %*****
24    if i==imax % Maximo de iterações
25        fprintf("\nSolução não alcançada com %d iterações\n", i)
26        Xsn='Sem resposta!'
27        break
28    end
29    %*****
30    Xest=Xsn;
31 end
32 %*****
33 fprintf("Tempo de processamento t=%f(s)\n",toc)

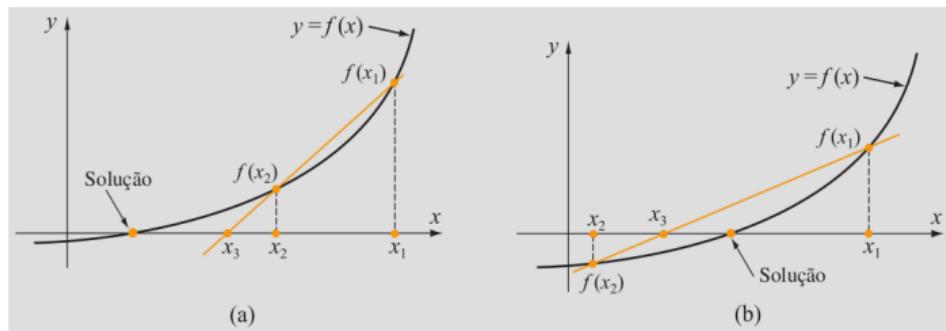
```

MN aplicados à Engenharia

Método da Secante

- Usa dois pontos na vizinhança da solução para determinar a nova solução estimada.

Figura: Método da Secante-Pontos de vizinhança



Fonte: GILAT,(2008)

- Em que $x_3 = x_2 - \frac{f(x_2)(x_1-x_2)}{f(x_1)-f(x_2)}$
- Termo geral: $x_{k+1} = x_k - \frac{f(x_k)(x_{k-1}-x_k)}{f(x_{k-1})-f(x_k)}$

MN aplicados à Engenharia

Método da Secante

Figura: *Script* Método da Secante

```

1 clear all;clc;
2 %*****
3 f=@(x)(8-4.5*(x-sin(x)));
4 Err=0.001; tol=Err; imax=30;
5 x1=2 ; x2=3; tic % valores inicial
6 %*****
7 for(i=1:imax)
8     Xsn=x2-f(x2)*(x1-x2)/(f(x1)-f(x2));
9     if (abs((Xsn-x2)/x2) < Err)           % Erro < x(i+1)-xi)/xi|
10        Xsn = x2;
11        fprintf("\nSolução %.4f alcançada após %d iterações! \n",Xsn,i)
12        break
13    end
14    %*****
15    if (abs(f(Xsn))< Err)                 % Tol < f(xi)
16        fprintf("\nSolução %.4f alcançada com %d iterações e tolerancia %.f! \n",
17        Xsn,i,f(Xsn))
18        break
19    end
20    %*****
21    if i==imax                         % Maximo de iterações
22        fprintf("\nSolução não alcançada com %d iterações\n", i)
23        Xsn='Sem resposta!';
24    break
25    end
26 %*****
27 x1=x2; x2=Xsn;
28 end
29 %*****
30 fprintf("Tempo de processamento t=%f(s)\n\n",toc)

```