Cálculo Numérico: Aplicação em Matlab

Professor: Eduardo Preto

Cálculo Numérico: Aplicação em Matlab

- Erros
- Raízes
- Sistemas lineares
- Interpolação
- Ajuste de curvas
- Derivadas
- Integrais
- Solução de EDOs

Cálculo numérico: Aplicação em Matlab

- Aplicações diretas
- Foco maior na implementação
- Aulas curtas e práticas
- Apresentação de vários métodos que lhe ajudará nos seus projetos e trabalhos

Cálculo Numérico: Aplicação em Matlab

TI e software › Mais opções em TI e software › MATLAB

Curso MATLAB: Do básico ao avançado

Conhecimentos básicos e avançados em MATLAB

Classificação mais alta 4,5 ★★★★☆ (195 classificações) 1.020 alunos

Criado por Eduardo Preto

Última atualização em 2/2021 · Português · Português [Automático]

Wishlist · Compartilhar · Presentear este curso

Cálculo Numérico: Aplicação em Matlab

- Tendo dúvidas não esite em perguntar

- Feedback

Aula 1 - Erros

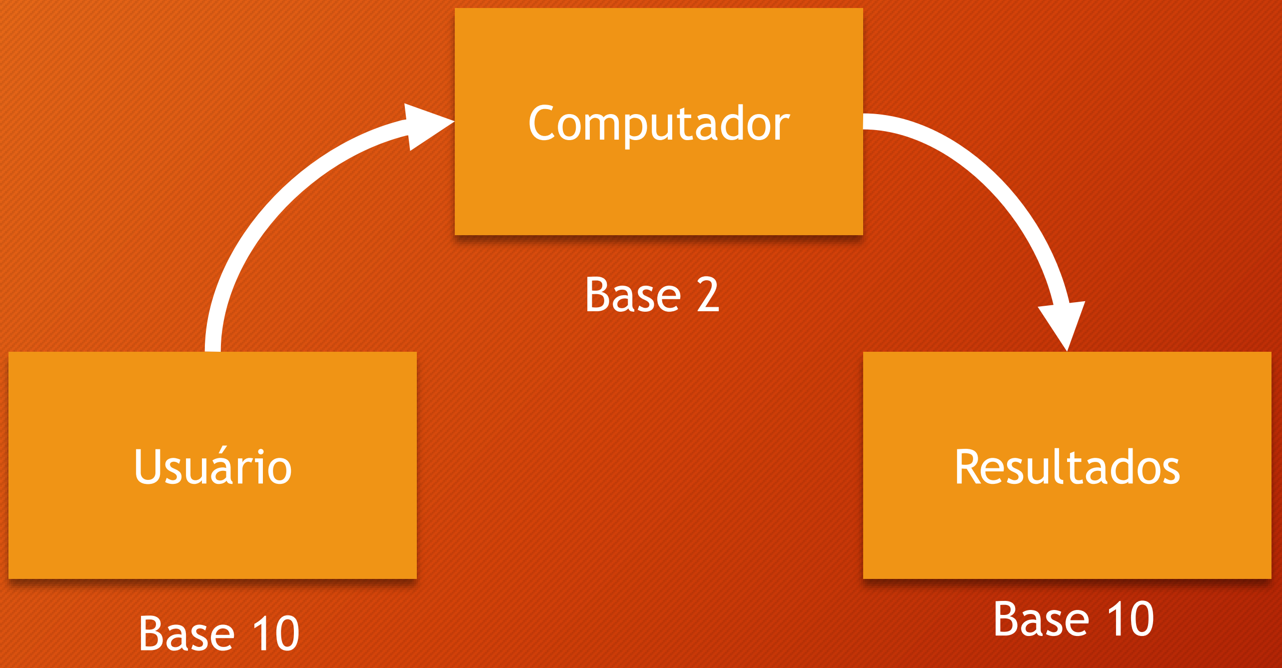
Modelagem

- 1- Interação com o assunto
  - Situação problema;
  - Pesquisa
- 2- Matemática
  - Hipóteses
  - Resolução do problema em termos de modelos
- 3- Modelo
  - Validação
  - Interpretação
- 4- Conclusão

Tipos de erros

- Conversão de base
- Arredondamento
- Truncamento

Conversão de base

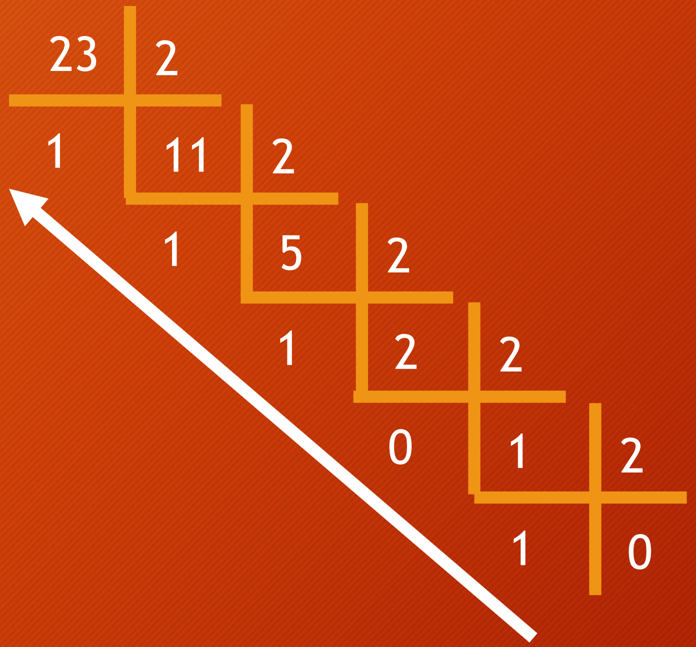


Conversão de base

- Conversão de base 2 para base 10

$$(10111)_2 = 1\ x\ 2^4 + 0\ x\ 2^3 + 1\ x\ 2^2 + 1\ x\ 2^1 + 1\ x\ 2^0 = 23$$

- Conversão de base 10 para base 2



Conversão de base

- Conversão de base 2 para base 10

$$(10{,}11)_2 = 1\ x\ 2^1 + 0\ x\ 2^0 + 1\ x\ 2^{-1} + 1\ x\ 2^{-2} = 2{,}75$$

- Conversão de base 10 para base 2

$$0{,}1875 \text{ x } 2 = \boxed{0}{,}3750$$

$$0{,}3750 \text{ x } 2 = \boxed{0}{,}750$$

$$0{,}75 \text{ x } 2 = \boxed{1}{,}50$$

$$0{,}5 \text{ x } 2 = \boxed{1}{,}0$$

$$(0{,}0011)_2 = 0{,}1875$$

Arredondamento

- Ocorre devido ao armazenamento dos dados, visto que o número precisa ser finito.
- Por exemplo: 2/3 = 0,6666 ou 0,6667. Ele sofre um corte ou arredondamento, e isto introduz erro.
- 0,3212867 – exato
- 0,3213 – arredondado
- Função “floor”, “ceil” no Matlab

Truncamento

- Ocorre devido a métodos numéricos que geram valores aproximados.
- Por exemplo:

$$Sen\ (x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} + \cdots$$

Elaboração dos códigos

- Conversão de bases
- Mostrando erros de arredondamento
- Mostrar erro de truncamento
- Converter 11000101.101

Aula 2 – Raízes de funções reais

Equações quadráticas - Resolução analítica

- As equações serão dadas por apenas uma variável

$$f(x) = 0$$

- Temos que as raízes serão os pontos em que a função toca abscissa.
- Equação de 2°grau:

$$ax^2 + bx + c = 0$$

- Aplicar Bhaskara

$$x = -\frac{b \pm \sqrt{b^2 - 4ac}}{2a}$$

Elaboração dos códigos

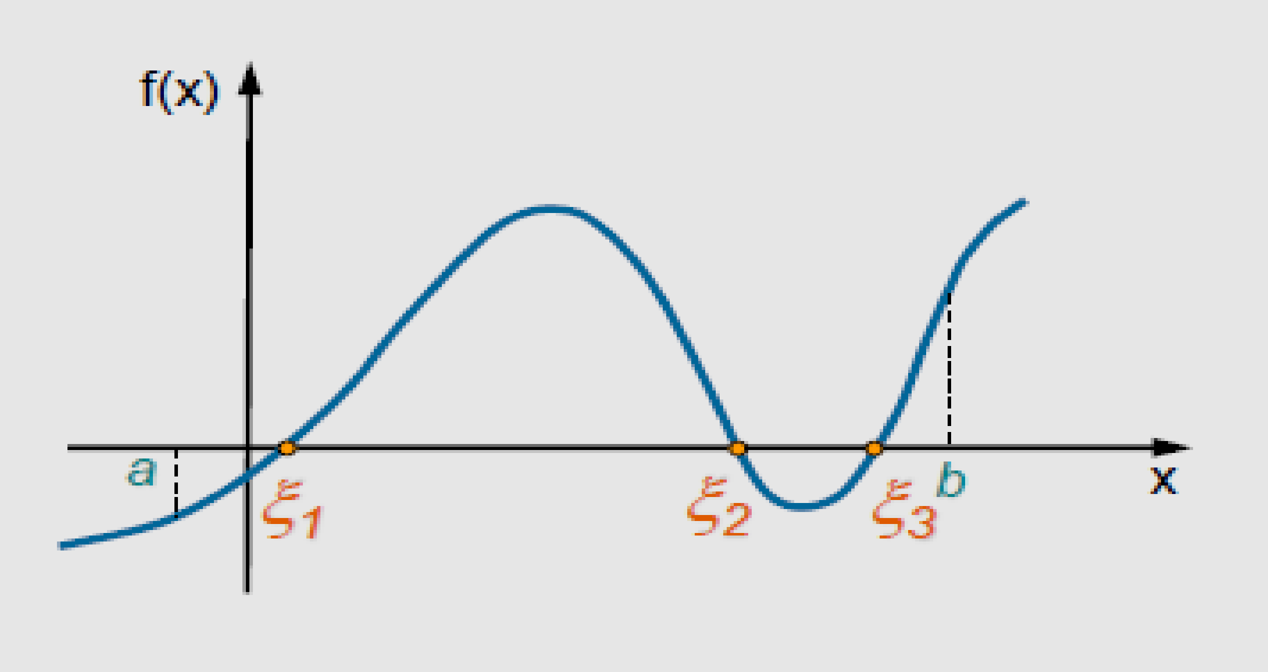
- Elaboração código para resolver por equação de segunda ordem

Determinação de raízes por MN

Método → Isolamento de raízes - intervalo → Refinamento

Determinação de raízes por MN

- Intervalo da raiz
  - f(a)f(b) < 0 – existe pelo menos um ponto que é zero;



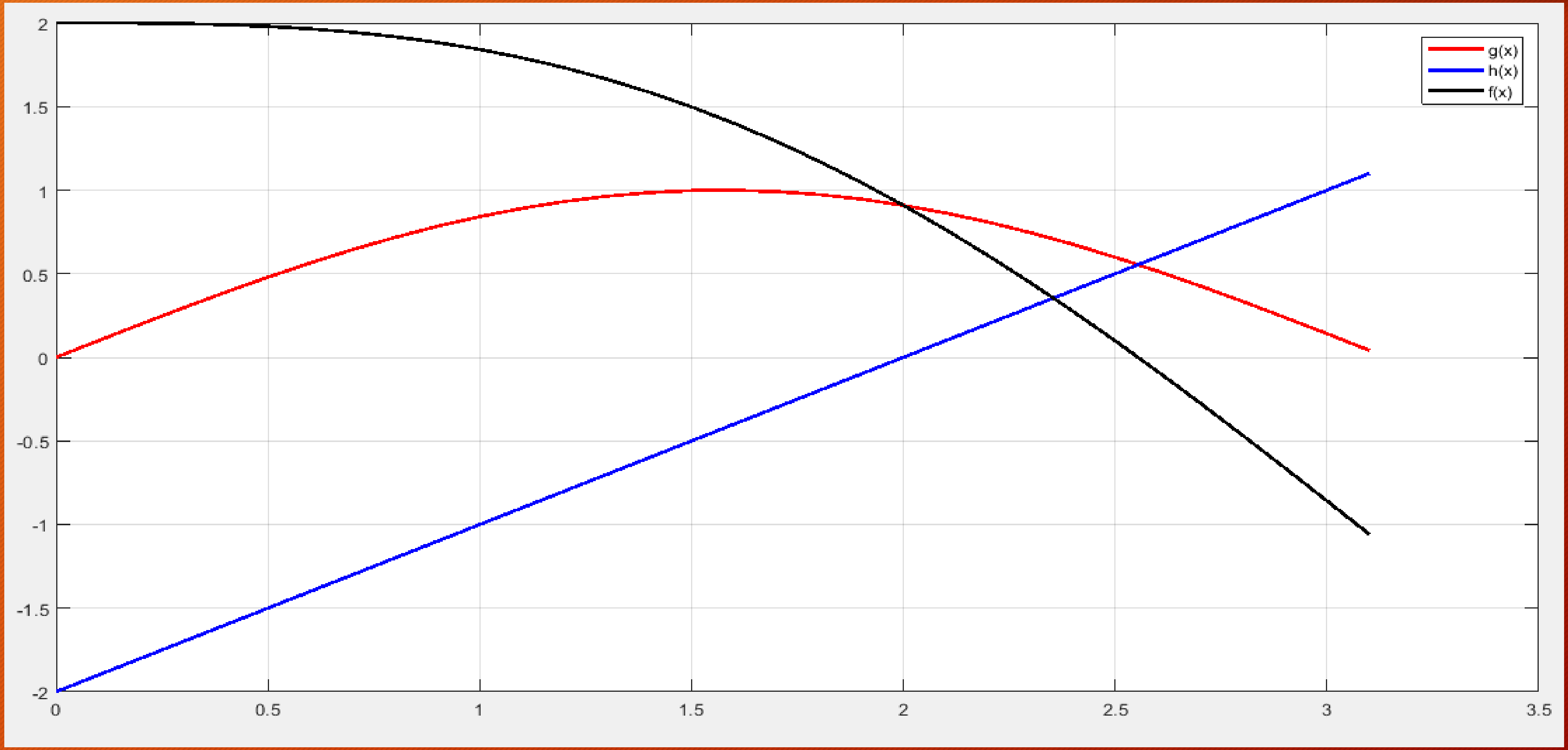
Determinação de raízes por MN

- Método gráfico
  - Separar $f(x)$ em $g(x)$ e $h(x)$
  - $f(x) = 0$, desse modo posso escrever $g(x) = h(x)$

  - Ex: $f(x) = \sin(x) - x + 2$

  $g(x) = \sin(x)$; $h(x) = x - 2 \rightarrow f(x) = g(x) - h(x) = 0$

Determinação de raízes por MN



Determinação de raízes por MN

- Refinamento
  - Usado métodos iterativos – série de intruções executáveis sequencialmente até atingir o critério de parada.

  - Critério de parada: Finalização do método ao atingir um valor especificado
  - Erro absoluto ou relativo são usados como critério de parada.

$$|x - x_0| < \varepsilon$$

$$\frac{|x - x_0|}{|x|} < \varepsilon$$

Determinação de raízes por MN - Bisseção

- Bisseção – subdivisão sucessivas do intervalo que contém o ponto médio entre a e b.

- $1^\circ: a_0 = a;\ \ b_0 = b;\ \ x_0 = \frac{a_0+b_0}{2}$

$$f(a_0) < 0$$
$$f(b_0) > 0$$
$$f(x_0) > 0$$

- $2^\circ: a_1 = a_0;\ \ b_1 = x_0;\ \ x_1 = \frac{a_1+b_1}{2}$

$$f(a_1) < 0$$
$$f(b_1) > 0$$
$$f(x_1) < 0$$

- $3^\circ: a_2 = x_1;\ \ b_2 = b_1;\ \ x_2 = \frac{a_2+b_2}{2}$

$$f(a_1) < 0$$
$$f(b_1) > 0$$
$$f(x_1) < 0$$

Determinação de raízes por MN - Bisseção

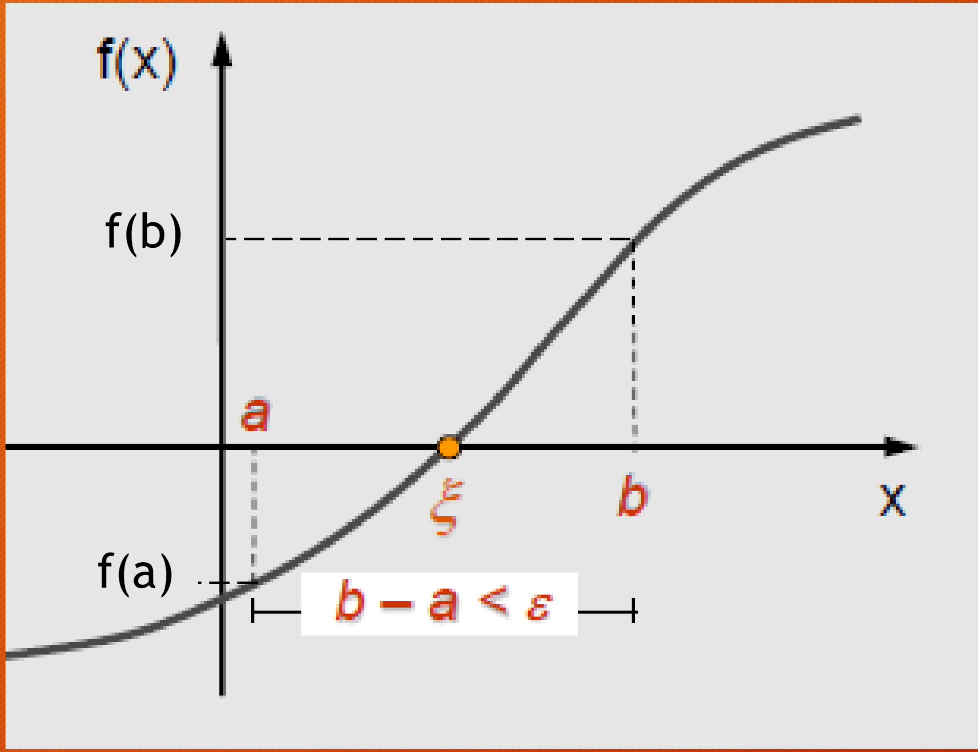
- Vantagens:
  - Fácil implementação;
  - Convergência e estabilidade;
  - Desempenho previsível.
- Desvantagens
  - Lentidão;
  - Conhecimento do intervalo da raiz.

Elaboração dos códigos

- Elaboração código para resolver por bisseção
- $f(x) = 8 - 4{,}5(x - \sin(x))$

Determinação de raízes por MN - Falsa posição

- Falsa Posição



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

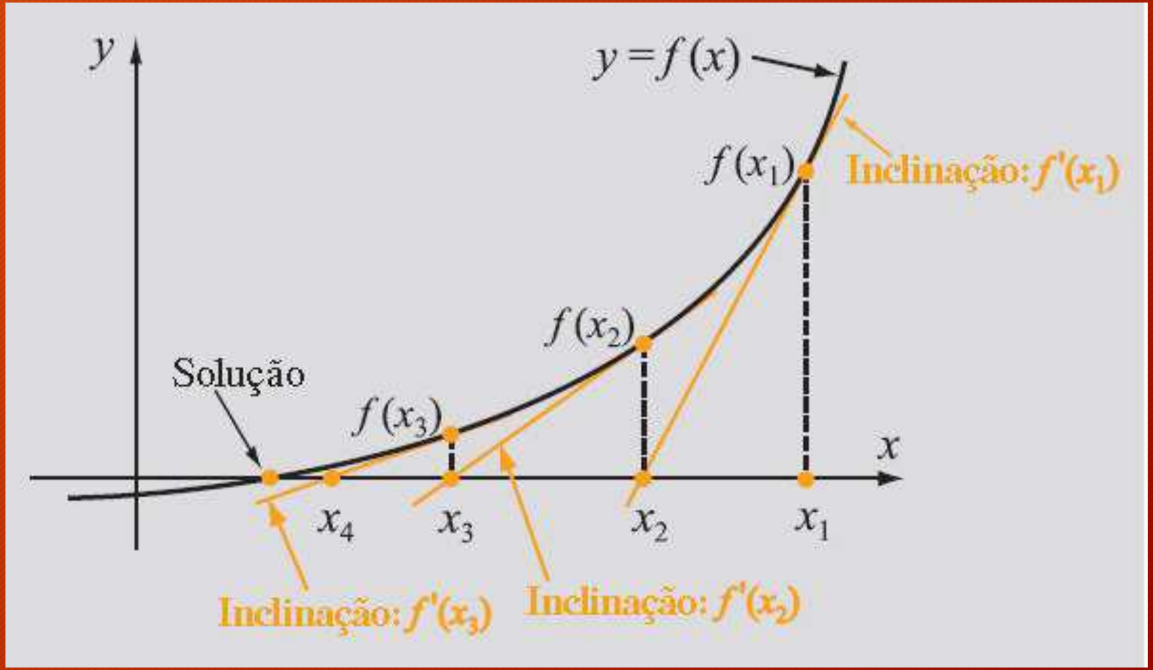
$$x = \frac{a\, |f(b)| - b\, |f(a)|}{|f(b)| - |f(a)|}$$

Elaboração dos códigos

- Elaboração código para resolver por falsa posição

Determinação de raízes por MN - Newton-Raphson

- Newton – Raphson
  - f(x) = 0;
  - Contínua;
  - Diferenciável;
  - Adotada uma primeira estimativa (x_1)



$$f'(x_1) = \frac{f(x_1) - 0}{x_1 - x_2}$$

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

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

Determinação de raízes por MN - Newton-Raphson

- Vantagens:
  - Rapidez na convergência
  - Desempenho elevado.
- Desvantagens:
  - Precisa de calcular a derivada
  - Difícil implementação.

Elaboração dos códigos

- Elaboração código para resolver por Newton-Raphson

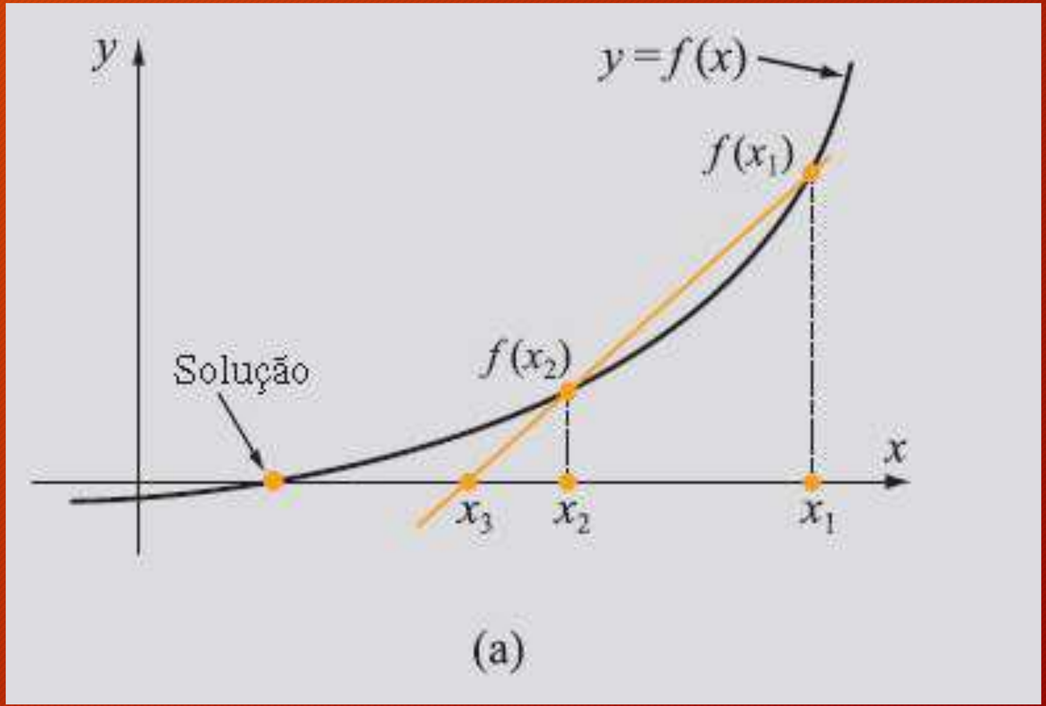
Determinação de raízes por MN - Secante

- Secante
  - f(x) = 0;
  - Contínua;
  - Usa dois pontos da vizinhança;

$$\frac{f(x_1) - f(x_2)}{x_1 - x_2} = \frac{f(x_2) - 0}{x_2 - x_3}$$

$$x_3 = x_2 - \frac{f(x_2)(x_1 - x_2)}{f(x_1) - f(x_2)}$$

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



Elaboração dos códigos

- Elaboração código para resolver por Secante

Aula 3 - Sistemas de equações lineares

Sistemas de equações lineares

- Métodos diretos
  - Método de Gauss
  - Método da Eliminação de Gauss
  - Método LU
- Métodos iterativos
  - Método de Jacobi
  - Método de Gauss - Seidel

Sistemas de equações lineares

- Método de Gauss
  - Sistema linear deve ser transformado em um sistema triangular;
  - Somar, subtrair linhas;
  - Dividir ou multiplicar a linha por um fator;
  - Obter a solução direta pelo formato triangular.

Sistemas de equações lineares

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \{x\} = \begin{bmatrix} z \\ w \\ v \end{bmatrix}$$

$$ax_1 + bx_2 + cx_3 = z$$
$$dx_1 + ex_2 + fx_3 = w$$
$$gx_1 + hx_2 + ix_3 = v$$

$$\begin{bmatrix} a & b & c \\ 0 & e & f \\ 0 & 0 & i \end{bmatrix} \{x\} = \begin{bmatrix} z \\ w \\ v \end{bmatrix}$$

Sistemas de equações lineares

- Método de Eliminação de Gauss
  - Sistema linear deve ser transformado em um sistema triangular;
  - Primeira linha é usada como pivô;
  - Ela é usada para eliminar as demais linhas;
  - O que fazer quando a primeira linha tiver zeros?
  - Aplicação no cálculo de treliças.

$$\begin{bmatrix} 0 & b & c \\ d & e & f \\ g & 0 & i \end{bmatrix} \Rightarrow \begin{bmatrix} d & e & f \\ 0 & b & c \\ g & 0 & i \end{bmatrix}$$

Elaboração dos códigos

- Elaboração código para resolver por Eliminação de Gauss
- Elaboração código para resolver por Eliminação de Gauss com pivotação

Sistemas de equações lineares

- Decomposição LU

$$[x] = [a]^{-1}[b]$$

$$[a] = [L][U]$$

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

Sistemas de equações lineares

- Decomposição LU

$$[L][U]\,[x] = [b]$$

Chamando:

$$[U][x] = [y]$$

Assim:

$$[L][y] = [b]$$

Sistemas de equações lineares

- Decomposição LU

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

Sistemas de equações lineares

- Decomposição LU
- Passo 1

$$U_{ii} = 1 \quad U_{1j} = \frac{a_{1j}}{L_{11}}$$

- Passo 2

$$L_{i1} = a_{i1}$$

- Passo 3

$$L_{ij} = a_{ij} - \sum_{k=1}^{k=j-1} L_{ik}U_{kj}$$

$$U_{ij} = \frac{a_{ij} - \sum_{k=1}^{k=j-1} L_{ik}U_{kj}}{L_{ii}}$$

Sistemas de equações lineares

- Decomposição LU - exemplo

$$\begin{bmatrix} 4 & -2 & -3 & 9 \\ -6 & 7 & 6{,}5 & -6 \\ 1 & 7{,}5 & 6{,}25 & 5{,}5 \\ -12 & 22 & 15{,}5 & -1 \end{bmatrix} \{x\} = \begin{bmatrix} 12 \\ -6{,}5 \\ 16 \\ 17 \end{bmatrix}$$

Elaboração dos códigos

- Elaboração código para resolver por Decomposição LU

Sistemas de equações lineares

- Métodos iterativos
  - Para sistemas grandes e esparsos
  - Escrever eles de forma explicita

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

(a)

Escrevendo as equações em uma forma explícita

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

(b)

Sistemas de equações lineares

- Métodos iterativos
  - Escolher os valores iniciais para as incógnitas;

$$x_i = \frac{1}{a_{ii}}\left[b_i - \sum_{\substack{j=1 \\ j\neq i}}^{j=n} a_{ij}x_j\right] \quad i = 1,2,3 \ldots, n$$

Sistemas de equações lineares

- Métodos iterativos
  - Condições de convergência

$$|a_{ii}| > \left[\sum_{\substack{j=1 \\ j \neq i}}^{j=n} |a_{ij}| \right] \quad i = 1,2,3 \ldots, n$$

  - O valor absoluto dos elementos da diagonal for maior que a soma dos valores absolutos dos elementos fora da diagonal
  - Essa condição é suficiente, mas não necessária para a convergência

Sistemas de equações lineares

- Métodos iterativos
  - Método de Jacobi
    - Os valores das incógnitas do lado direito são atualizados a cada iteração
  - Método de Gauss-Seidel
    - Os valores das incógnitas são atualizados assim que calculado a estimativa

Sistemas de equações lineares

- Métodos iterativos
  - Método de Jacobi
    - Valor inicial
    - Se não tiver informação, use zero

$$x_{ij}^{(1)} = \mathrm{C.I.}$$

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

Sistemas de equações lineares

- Métodos iterativos
  - Método de Gauss-Seidel
    - Os valores das incógnitas são atualizados assim que calculado a estimativa
    - Os valores iniciais se não conhecido, usar zero

$$x_{ij}^{(1)} = \mathrm{C.I.}$$

Sistemas de equações lineares

- Métodos iterativos
  - Método de Gauss-Seidel

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

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

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

Sistemas de equações lineares

- Métodos iterativos
  - Método de Gauss-Seidel

$$\begin{aligned}
9x_1 - 2x_2 + 3x_3 + 2x_4 &= 54{,}5\\
2x_1 + 8x_2 - 2x_3 + 3x_4 &= -14\\
-3x_1 + 2x_2 + 11x_3 - 4x_4 &= 12{,}5\\
-2x_1 + 3x_2 + 2x_3 + 10x_4 &= -21
\end{aligned}$$

$$\begin{aligned}
x_1 &= [54{,}5 - (-2x_2 + 3x_3 + 2x_4)]/9\\
x_2 &= [-14 - (2x_1 - 2x_3 + 3x_4)]/8\\
x_3 &= [12{,}5 - (-3x_1 + 2x_2 - 4x_4)]/11\\
x_4 &= [-21 - (-2x_1 + 3x_2 + 2x_3)]/10
\end{aligned}$$

Sistemas de equações lineares

- Comandos do Matlab
  - Resolver sistemas linear [A] [x] = [b]:
    - X = A\b
    - X = b/A → matriz A é a transposta
    - X = A^-1 *b
    - X = inv(A)*b
  - Resolver sistemas linear LU
    - Usar a função “lu” do Matlab
    - $[L, U, P] = lu(A)$
    - L – Matriz triangular inferior; U – matriz triangular superior; P – Permutação
    - A – Matriz a ser decomposta – [A] = [L][U]

Elaboração dos códigos

- Elaboração código para resolver por Jacobi
- Elaboração código para resolver por Gauss - Seidel

Aula 4 - Interpolação e ajuste de curva

Interpolação e ajuste de curva

- Tópico
  - Ajuste de curva por Regressão linear por mínimos quadrados
  - Ajuste de equações não lineares
  - Interpolação polinomial
  - Método de Lagrange
  - Polinômio Interpolador de Newton

Interpolação e ajuste de curva

- Ajuste de curva
  - Encontrar uma equação que melhor representa um conjunto de pontos
  - Obter uma representação geral
- Interpolação
  - Encontrar uma estimativa para um valor entre pontos conhecidos
  - Usado quando possui uma quantidade pontos pequena.

Interpolação e ajuste de curva

- Ajuste de curva por equações lineares
  - Podemos obter uma equação linear(reta) que represente um conjunto de dados
  - $y = a_1 x + a_0$
  - Ela pode não passar por todos ele, mas ela apresenta o melhor ajuste
  - Resíduo: $r_i = y_i - f(x_i)$
  - Erro total:

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

  - Regressão linear por mínimos quadrados:

Interpolação e ajuste de curva

- Ajuste de curva por equações lineares
  - Regressão linear por mínimos quadrados:

$$a_1 = \frac{n \sum_{i=1}^{i=n} x_i y_i \; - (\sum_{i=1}^{i=n} x_i)(\sum_{i=1}^{i=n} y_i)}{n \sum_{i=1}^{i=n} x_i^2 \; - (\sum_{i=1}^{i=n} x_i)^2}$$

$$a_0 = \frac{(\sum_{i=1}^{i=n} x_i^2\,) \sum_{i=1}^{i=n} y_i \; - (\sum_{i=1}^{i=n} x_i y_i)(\sum_{i=1}^{i=n} x_i)}{n \sum_{i=1}^{i=n} x_i^2 \; - (\sum_{i=1}^{i=n} x_i)^2}$$

$$y = a_1 x + a_0$$

Interpolação e ajuste de curva

- Ajuste de curva de equações não lineares

Equação não-linear	Forma linear	Relação com $Y = a_1X + a_0$
$y = bx^m$	$\ln(y) = m\ln(x) + \ln(b)$	$Y = \ln(y), X = \ln(x)$ $a_1 = m, a_0 = \ln(b)$
$y = be^{mx}$	$\ln(y) = mx + \ln(b)$ $\boxed{v_R = ve^{(-t/(RC))}}$	$Y = \ln(y), X = x$ $a_1 = m, a_0 = \ln(b)$
$y = b10^{mx}$	$\log(y) = mx + \log(b)$	$Y = \log(y), X = x$ $a_1 = m, a_0 = \log(b)$
$y = \frac{1}{mx+b}$	$\frac{1}{y} = mx + b$	$Y = \frac{1}{y}, \quad X = x$ $a_1 = m, \quad a_0 = b$
$y = \frac{mx}{b+x}$	$\frac{1}{y} = \frac{b}{m}\frac{1}{x} + \frac{1}{m}$	$Y = \frac{1}{y}, \quad X = \frac{1}{x}$ $a_1 = \frac{b}{m}, \quad a_0 = \frac{1}{m}$

Interpolação e ajuste de curva

- Ajuste de curva com polinômios quadráticos e de ordem superior

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

$$E = \sum_{i=1}^{i=n} [y_i - (a_m x^m + a_{m-1}\, x^{m-1} + \ ... + a_1 x + a_0)]^2$$

Interpolação e ajuste de curva

- Ajuste de curva com polinômios quadráticos e de ordem superior

$$\frac{\partial E}{\partial a_0} = -2\sum_{i=1}^{n}(y_i - a_2x_i^2 - a_1x_i - a_0) = 0$$

$$\frac{\partial E}{\partial a_1} = -2\sum_{i=1}^{n}(y_i - a_2x_i^2 - a_1x_i - a_0)x_i = 0$$

$$\frac{\partial E}{\partial a_2} = -2\sum_{i=1}^{n}(y_i - a_2x_i^2 - a_1x_i - a_0)x_i^2 = 0$$

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

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

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

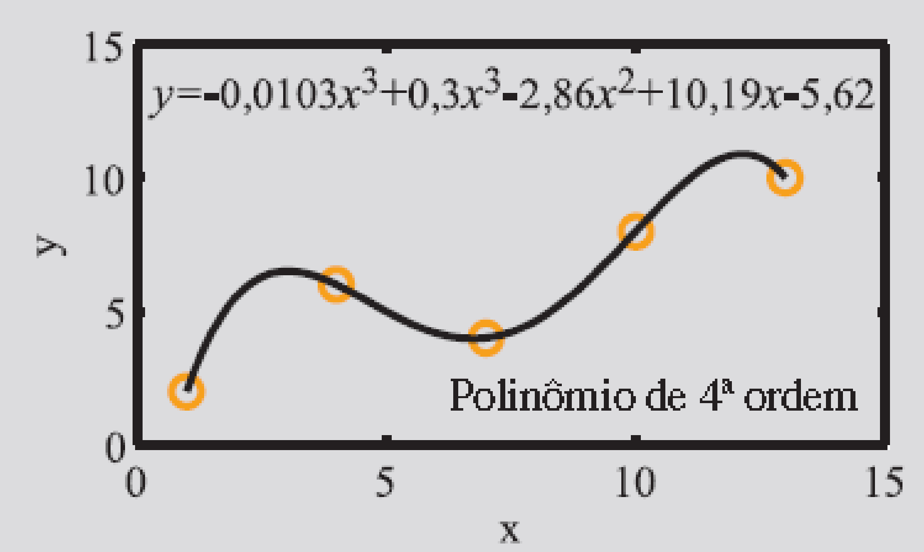
Interpolação e ajuste de curva

- Interpolação polinomial
  - Obtenção do polinômio que representa um conjunto de pontos
  - Formas: Padrão, Lagrange e Newton

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

Interpolação e ajuste de curva

- Interpolação polinomial
  - Padrão
  - Resolução de sistema de equações
  - Equação para cada ponto dos dados



$$
\begin{aligned}
a_4 1^4 + a_3 1^3 + a_2 1^2 + a_1 1 + a_0 &= 2 \\
a_4 4^4 + a_3 4^3 + a_2 4^2 + a_1 4 + a_0 &= 6 \\
a_4 7^4 + a_3 7^3 + a_2 7^2 + a_1 7 + a_0 &= 4 \\
a_4 10^4 + a_3 10^3 + a_2 10^2 + a_1 10 + a_0 &= 8 \\
a_4 13^4 + a_3 13^3 + a_2 13^2 + a_1 13 + a_0 &= 10
\end{aligned}
$$

Interpolação e ajuste de curva

- Interpolação polinomial
  - Polinômio de Lagrange
  - Fazer de um conjunto de pontos
  - Tendo o conjunto de pontos (x_1, y_1) e (x_2, y_2)

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

$$a_1 = \frac{y_1}{(x_1 - x_2)} \qquad a_2 = \frac{y_2}{(x_2 - x_1)}$$

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

Interpolação e ajuste de curva

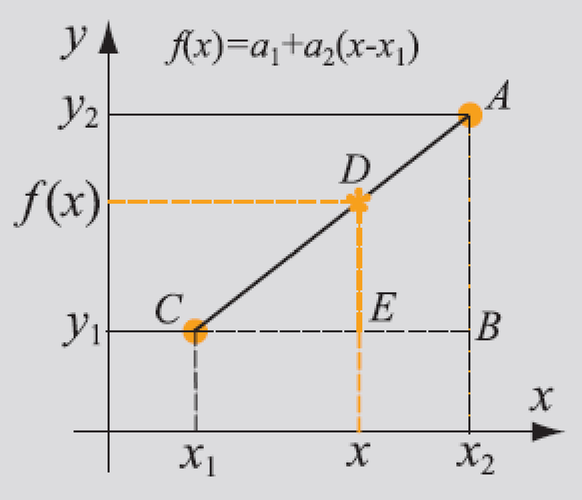
- Interpolação polinomial
  - Polinômio de Lagrange
  - De forma compacta, temos:

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

Interpolação e ajuste de curva

- Interpolação polinomial
  - Polinômio de Newton
  - Obter o ajuste exato de um conjunto de pontos

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



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

Interpolação e ajuste de curva

- Interpolação polinomial
  - Polinômio de Newton

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

$$f(x_1) = y_1 = a_1$$

$$f(x_2, x_1) = y_2 = y_1 + a_2(x_2 - x_1)$$

$$a_2 = \frac{y_2 - y_1}{x_2 - x_1}$$

Interpolação e ajuste de curva

- Interpolação polinomial
  - Polinômio de Newton

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

$$a_j = f(x_j, x_i) = \frac{y_j - y_i}{x_j - x_i}$$

Interpolação e ajuste de curva

- Interpolação polinomial
  - Polinômio de Newton

x	1	2	4	5	7
y	52	5	-5	-40	10

$a_1 = 52$ $\quad a_2 = -47$ $\quad a_3 = 14$ $\quad a_4 = -6$ $\quad a_5 = 2$

x_i	y_i
1	52
2	5
4	-5
5	-40
7	10

$$\frac{5-52}{2-1} = -47 \qquad \frac{-5-5}{4-2} = -5 \qquad \frac{-40-(-5)}{5-4} = -35 \qquad \frac{10-(-40)}{7-5} = 25$$

$$\frac{-5-(-47)}{4-1} = 14 \qquad \frac{-35-(-5)}{5-2} = -10 \qquad \frac{25-(-35)}{7-4} = 20$$

$$\frac{-10-14}{5-1} = -6 \qquad \frac{20-(-10)}{7-2} = 6$$

$$\frac{6-(-6)}{7-1} = 2$$

Interpolação e ajuste de curva

- Funções do matlab
  - Polyfit
    - P = polyfit (x,y,m)
    - P → vetor de coeficientes
    - m → grau do polinômio
    - x e y são os dados que possui
  - Interp1
    - yj = interp1 (x,y,xj,'método')
    - yj → valor interpolado
    - xj → valor de x que deseja encontrar yj
    - x e y são os dados que possui
    - 'método' → método de interpolação

Elaboração dos códigos

- Elaboração código para resolver por ajuste de curva
- Elaboração código para resolver por interpolação

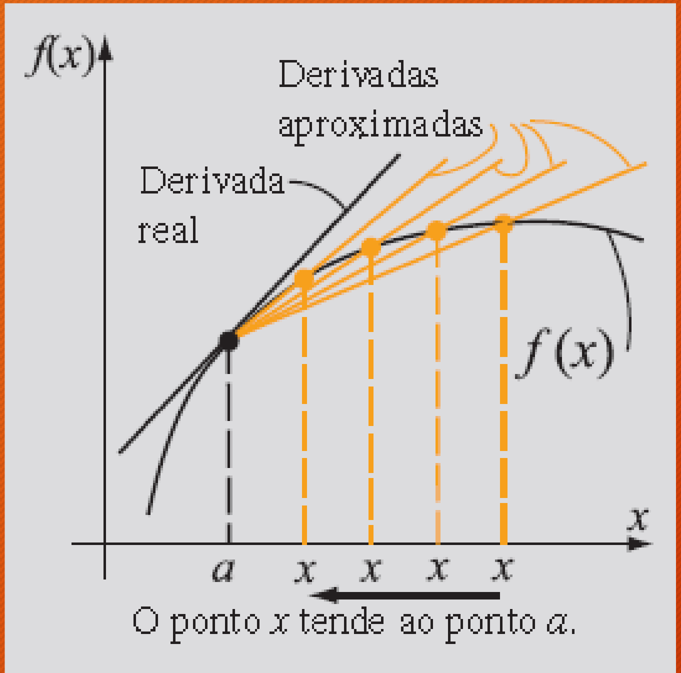
Aula 5 - Diferenciação numérica

Diferenciação numérica

- Tópicos
  - Aproximação da derivada por diferenças finitas
  - Série de Taylor
  - Funções do matlab

Diferenciação numérica

- Aproximação da derivada por diferenças finitas
  - Derivada é a inclinação de uma reta tangente a função no ponto de análise
  - Por exemplo, adotando um x = a, podemos fazer:



$$f'(a) = \lim_{x \to a} \frac{f(x) - f(a)}{x - a}$$

Diferenciação numérica

- Aproximação da derivada por diferenças finitas
  - Podemos ter 3 tipos de diferenças:
    - Progressiva

$$\left.\frac{df}{dx}\right|_{x=x_i} = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

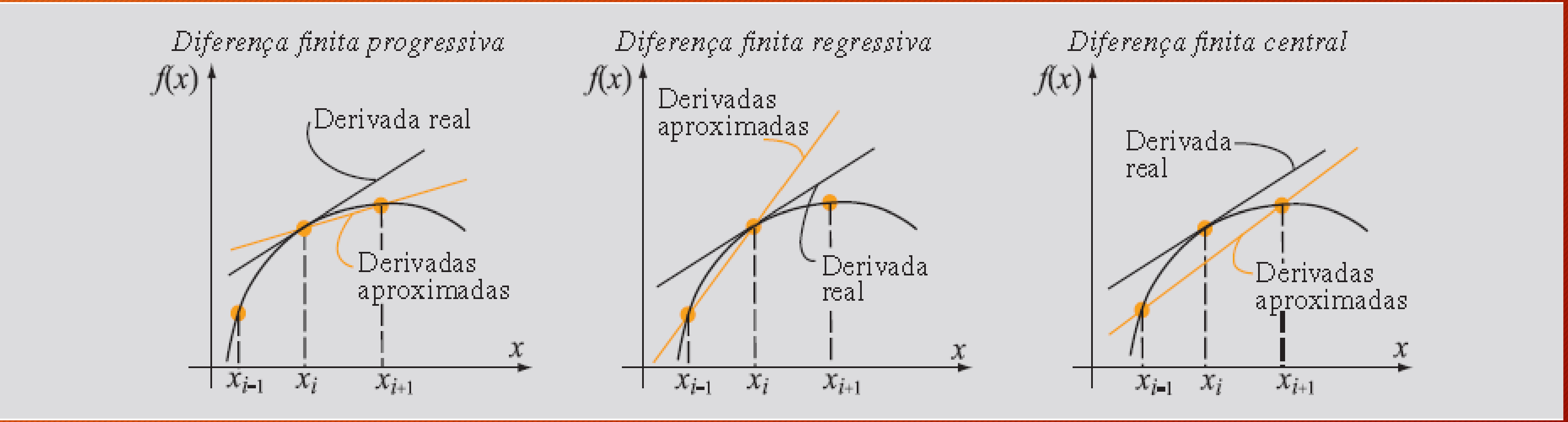
    - Regressiva

$$\left.\frac{df}{dx}\right|_{x=x_i} = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$

Diferenciação numérica

- Aproximação da derivada por diferenças finitas
  - Podemos ter 3 tipos de diferenças:
    - Central

$$\left.\frac{df}{dx}\right|_{x=x_i} = \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}}$$



Diferenciação numérica

- Série de Taylor
  - O método anterior pode ser obtido por meio da série de Taylor
  - Podemos obter o erro de truncamento
  - $h = (x_{i+1} - x_i)$

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)h^2}{2!} + \frac{f'''(x_i)h^3}{3!} + \frac{f^{(4)}(x_i)h^4}{4!} + \cdots$$

  - Resolvendo com dois termos:

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)h^2}{2!}$$

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f''(x_i)h}{2!}$$

Diferenciação numérica

- Série de Taylor
  - Desse modo, podemos chamar o termo negativo como sendo o erro de truncamento

$$E(h) = -\frac{f''(x_i)h}{2!}$$

Assim, temos a diferença finita progressiva

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

Diferenciação numérica

- Série de Taylor
  - Diferença finita regressiva

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

  - Diferença finita central

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} + E(h^2)$$

  - Devido o $E(h^2)$, a DF central apresenta maior precisão para a derivada

Diferenciação numérica

- Série de Taylor
  - Diferença finita progressiva para três pontos

$$f'(x_i) = \frac{-3f(x_i) + 4f(x_{i+1}) - f(x_{i+2})}{2h} + E(h^2)$$

  - Diferença finita regressiva para três pontos

$$f'(x_i) = \frac{f(x_{i-2}) - 4f(x_{i-1}) + 3f(x_i)}{2h} + E(h^2)$$

Diferenciação numérica

- Funções do Matlab
  - Diff – diferença entre os pontos adjacentes
    - $d = diff(x, 2)$
    - d → vetor com as diferenças entre os elementos
    - x → vetor de dados
    - 2 → quantidades de vezes que é executado ex: diff(dif(x)), quando tiver 2
  - Polyder – derivada de um polinômio
    - $d = polyder(x)$
    - d → vetor com os coeficientes dos polinômios
    - x → vetor de dados

Elaboração dos códigos

- Elaboração código para resolver por diferenças finitas
- Elaboração código para resolver por série de Taylor

Aula 6 - Integração

Integração numérica

- Tópicos
  - Fórmulas de Newton – Cotes
  - Método do Retângulo e do ponto central
  - Regra do trapézio
  - 1/3 de Simpson

Integração numérica

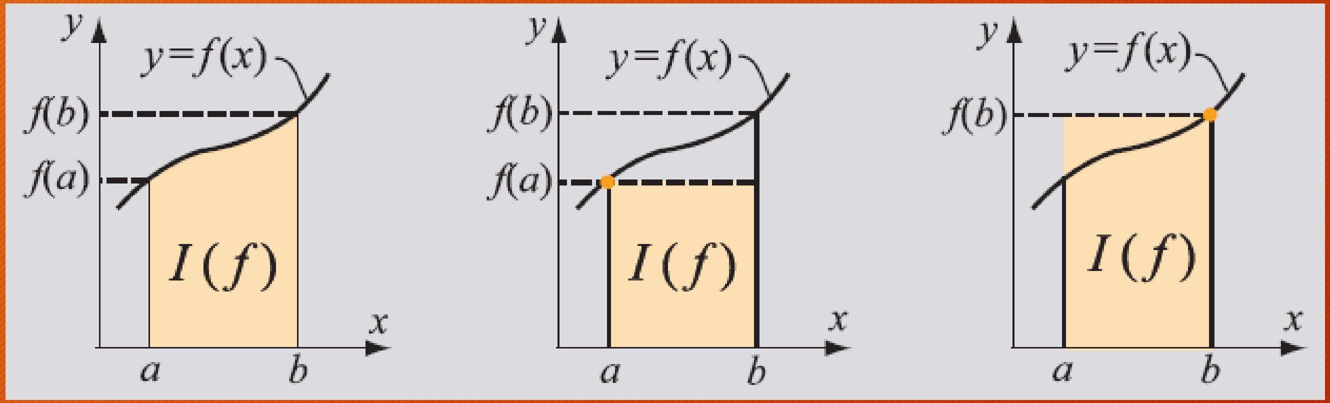
- Introdução
  - Os métodos utilizados podem ser de duas maneiras:
    - Newton – Cotes: Os valores de f(x) estão espaçados igualmente
    - Quadratura Gaussiana: Os pontos estão espaçados diferentemente

  - Newton – Cotes
    - $h = (x_{i+1} - x_i)$
    - Para um intervalo [a,b]
    - $h = \frac{(b-a)}{m}$ $m = n + 1$, sendo n o número de subdivisões

$$\int_a^b f(x)dx = A_0 f(x_0) + A_1 f(x_1) + \ldots + A_n f(x_n) = \sum_{i=0}^{n} A_i f(x_i)$$

Integração numérica

- Método do Retângulo
  - Consiste em obter a área de um retângulo formado entre o intervalo analisado

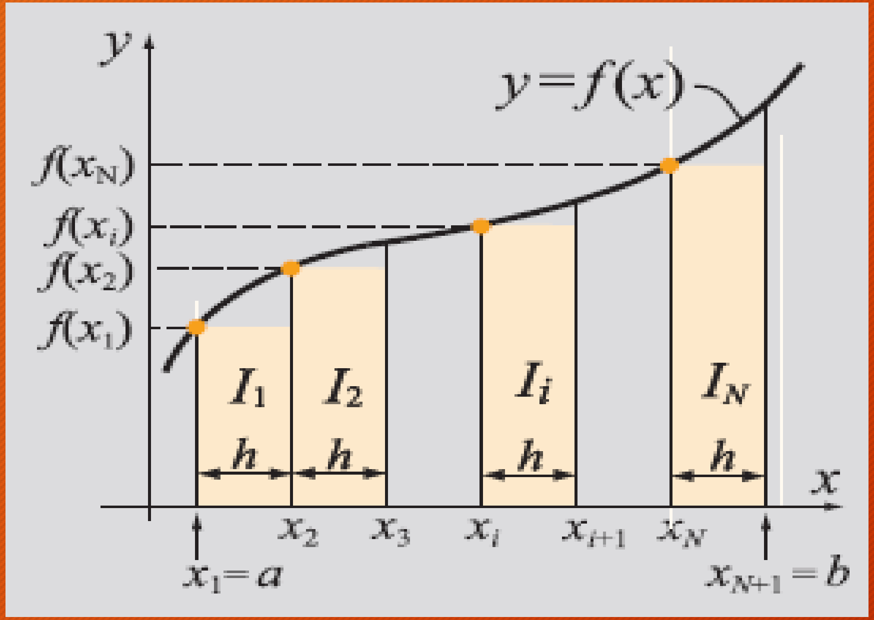


$$I = f(a)(b-a)$$

$$I = f(b)(b-a)$$

Integração numérica

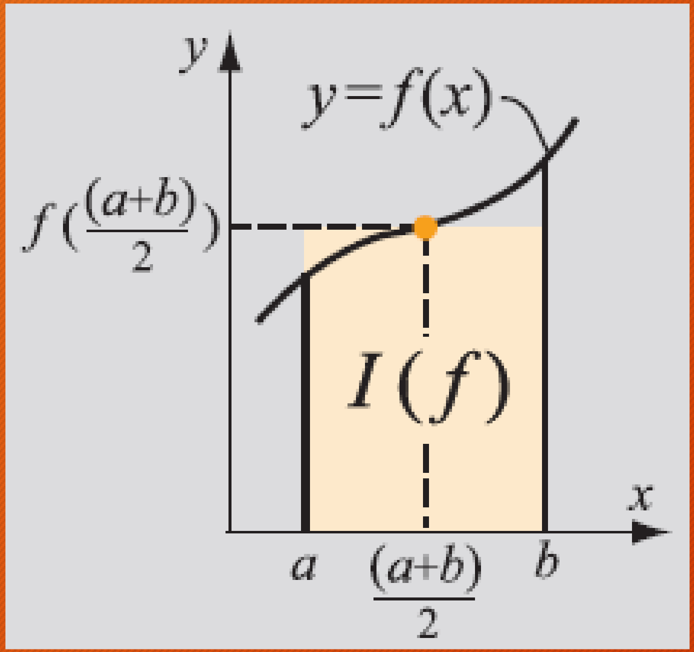
- Método do Retângulo composto
  - Divide o intervalo em n retângulos
  - Soma-se a área de cada retângulo



$$I = \int_a^b f(x)dx \sim h \sum_{i=1}^{n} f(x_i)$$

Integração numérica

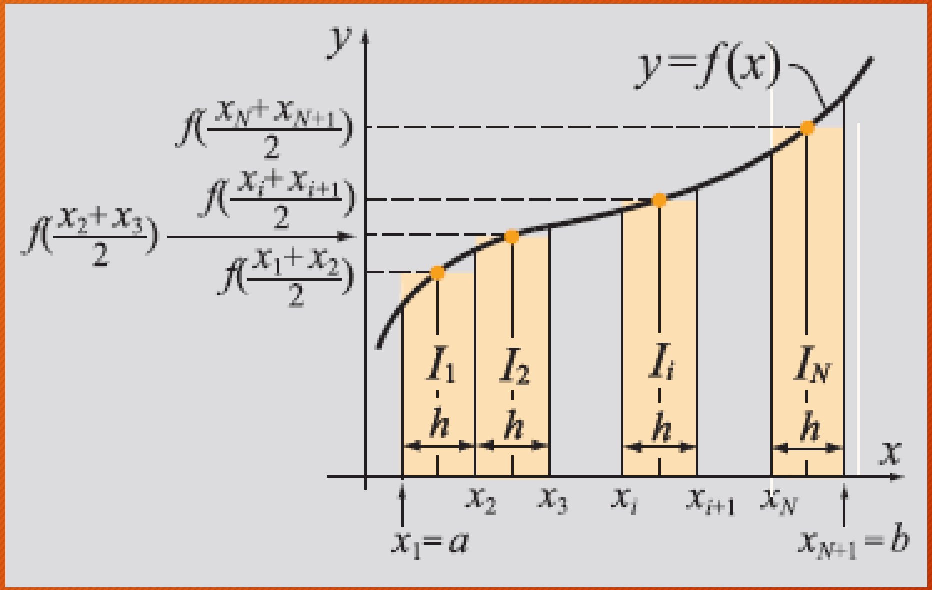
- Método do ponto central
  - Divide o intervalo em um retângulo e utiliza o ponto médio dele



$$I = \int_a^b f(x)dx \sim \int_a^b f\left(\frac{a+b}{2}\right)dx = f\left(\frac{a+b}{2}\right)(b-a)$$

Integração numérica

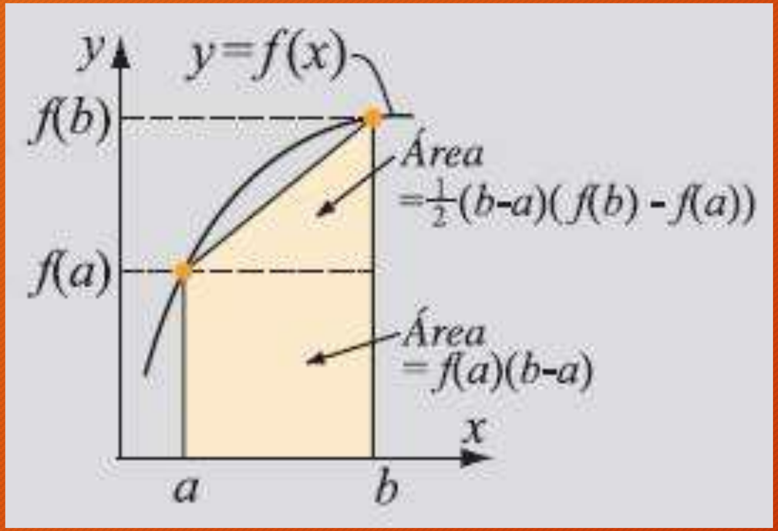
- Método do ponto central composto
  - Divide o intervalo em n retângulos



$$I = \int_a^b f(x)dx \sim h \sum_{i=1}^{n} f(\frac{x_i + x_{i+1}}{2})$$

Integração numérica

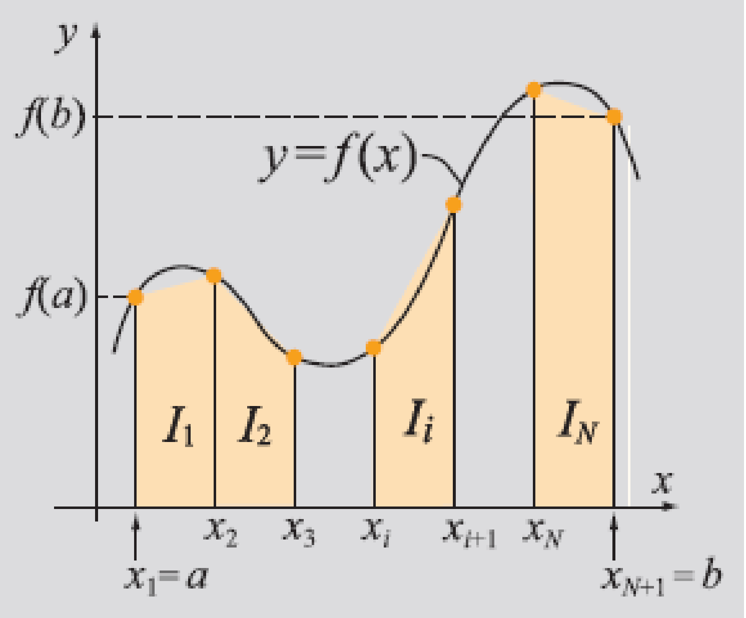
- Método trapezoidal
  - Seja um intervalo [a,b]
  - Retângulo + triângulo



$$I \sim \frac{[f(a) - f(b)]}{2}(b-a)$$

Integração numérica

- Método trapezoidal composto
  - Seja um intervalo [a,b]
  - Retângulo + triângulo



$$I \sim \frac{1}{2}\sum_{i=1}^{n}[f(x_i + f(x_{i+1})](x_{i+1} - x_i) \sim \frac{h}{2}\sum_{i=1}^{n}[f(x_i) + f(x_{i+1})]$$

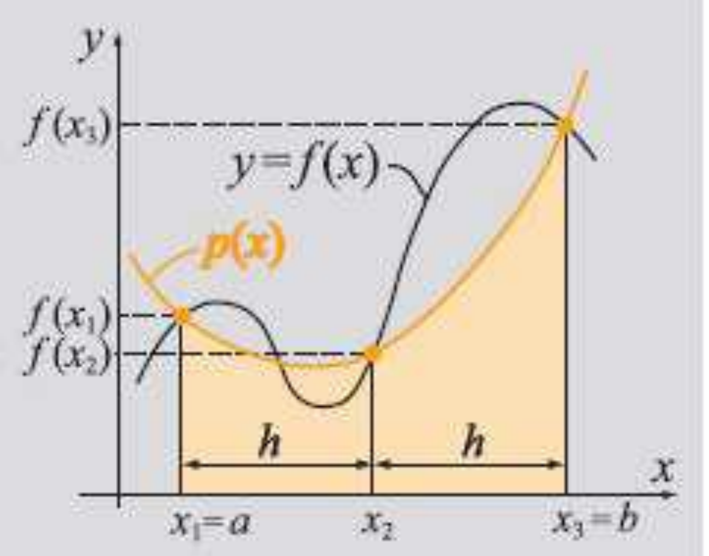
$$I \sim \frac{h}{2}[f(a) - f(b)] + h\sum_{i=1}^{n} f(x_i)$$

Integração numérica

- Método de Simpson
  - Utilização de um polinômio
  - Esse polinômio pode ser quadrático (1/3 de Simpson) ou cúbico (3/8 de Simpson)
  - Para a determinação dos coeficientes são necessários 3 pontos

$$p(x) = \alpha + \beta(x - x_1) + \gamma(x - x_1)(x - x_2)$$

$$I \sim \frac{h}{3}\left[f(a) + 4\, f\left(\frac{a+b}{2}\right) + f(b)\right]$$

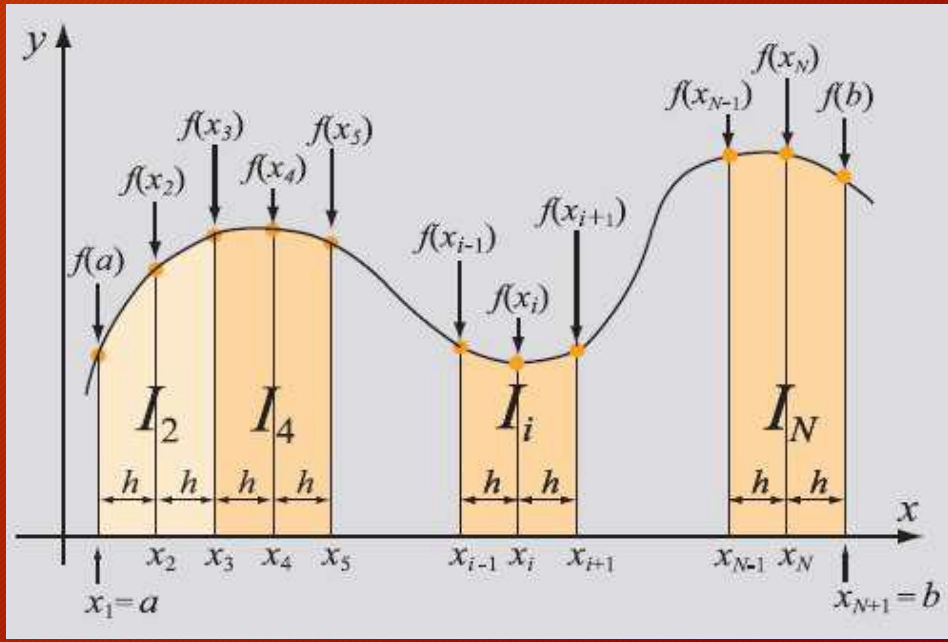


Integração numérica

- Método 1/3 de Simpson
  - Utilização de um polinômio
  - Divisão em n subintervalos
  - Os intervalos devem ter a mesma dimensão (largura)
  - Será um número impar a quantidade de h

$$I \sim \frac{h}{3}[f(x_{i-1}) + 4f(x_i) + f(x_{i+1})]$$

$$I \sim \frac{h}{3}\left[f(a) + 4\sum_{i=2,4,6}^{n} f(x_i) + 2\sum_{j=3,5,7}^{n-1} f(x_i) + f(b)\right]$$



Integração numérica

- Método 1/3 de Simpson
  - Utilização de um polinômio
  - Divisão em n subintervalos
  - Os intervalos devem ter a mesma dimensão (largura)

$$I \sim \frac{3h}{8}\left[f(a) + 3 \sum_{i=2,5,8}^{n-1} [f(x_i) + f(x_{i+1}) + 2 \sum_{j=4,7,10}^{n-2} f\left(x_j\right) + f(b) \right]$$

Integração numérica

- Funções do Matlab
  - Quad
    - $I = quad\ ('função', a, b)$
    - Função deve ser fornecida como um string
    - A e b são os extremos dos intervalos
    - Calcula a integral com erro absoluto menor que 10^{-6}
  - integral
    - $I = integral(função, a, b)$

Integração numérica

- Funções do Matlab
  - Trapz
    - $I = trapz\ (x, y)$
    - Necessário fornecer os pontos da função
  - Dblquad
    - $I = dblquad\ (função', xmin, xmax, ymin, ymax)$
    - Resolver integral dupla
    - Xmin, xmax,ymin e ymax são limites de integração

Elaboração dos códigos

- Elaboração código para resolver por 1/3 Simpson
- Elaboração código para resolver por Trapezoidal

Aula 7 - EDOs

EDOs

- Tópicos
  - Método de Euller
  - Método de Euller modificado
  - Método de Runge Kutta

EDOs

- Equações Diferenciais Ordinárias (EDOs)

$$\frac{dy}{dx} + ax^2 + by = 0 \qquad \text{(Linear)}$$

$$\frac{dy}{dx} + ayx^2 + b\sqrt{y} = 0 \qquad \text{(Não-Linear)}$$

  - Aplicações em diversas áreas

$$\frac{dx}{dt} = at^2 + bt$$

EDOs

- Métodos de Euller
  - Método explícito de Euller
    - Usado para EDOs de primeira ordem
    - Para h pequeno, a inclinação é constante e igual a do ponto (x_i, y_i)

$$x_{i+1} = x_i + h$$

$$y_{i+1} = y_i + f(x_i, y_i)h$$

$$y_{i+1} = y_i + inclinação\ h$$

EDOs

- Métodos de Euller
  - Método implícito de Euller
    - Usado para EDOs de primeira ordem
    - Para h pequeno, a inclinação é constante e igual a do ponto (x_{i+1}, y_{i+1})

$$x_{i+1} = x_i + h$$

$$y_{i+1} = y_i + f(x_{i+1}, y_{i+1})h$$

EDOs

- Métodos de Euller
  - Método implícito de Euller
    - Ex:

$$\frac{dn(t)}{dt} = -0{,}8n^{\frac{3}{2}} + 10n_1(1 - e^{-3t}) = f(t, n)$$

$$t_{i+1} = t_i + h$$

$$n_{i+1} = n_i + [-0{,}8n^{\frac{3}{2}} + 10n_1(1 - e^{-3t})]h$$

EDOs

- Métodos de Euller
  - Método implícito de Euller
    - Ex:

Chamando $x = n_{i+1}$

$$g(x) = x + 0{,}8x^{\frac{3}{2}}h - 10n_1(1 - e^{-3t})]h - n_i = 0$$

Aplicar o método de Newton. Para isso, calcula-se a derivada de g(x)

$$g'(x) = 1 + 0{,}8\frac{3}{2}\,x^{\frac{1}{2}}h$$

EDOs

- Métodos de Euller
  - Método implícito de Euller
    - Ex:

Assim:

$$x_{j+1} = x_j - \frac{\left[x_j + 0{,}8x_j^{\frac{3}{2}}h - 10n_1(1 - e^{-3t_{i+1}})h - n_i\right]}{1 + 0{,}8\frac{3}{2}\,x_j^{\frac{1}{2}}h}$$

EDOs

- Métodos de Euller
  - Método de Euller modificado
    - Versão modificado do Euller explicito
    - Assume que a derivada entre dois pontos (x_i, y_i) e (x_{i+1}, y_{i+1}) seja constante e igual a derivada de y(x) no ponto (x_i, y_i)
    - A inclinação é uma média de inclinações

$$y_{i+1}^E = y_i + f(x_i, y_i)h$$

$$x_{i+1} = x_i + h$$

$$y_{i+1} = y_i + \frac{f(x_i, y_i) + f\left(x_{i+1}, y_{i+1}^E\right)}{2}h$$

EDOs

- Métodos de Runge Kutta
  - Métodos explícitos
  - É classificado de acordo com a ordem
  - A ordem é a quantidade pontos usados para determinar a inclinação

$$y_{i+1} = y_i + inclinação\ h$$

  - São mais precisos que o método de Euller explícito
  - Precisão aumenta conforme aumenta a ordem

EDOs

- Métodos de Runge Kutta
  - Runge Kutta de Segunda ordem

$$y_{i+1} = y_i + (c_1 K_1 + c_2 K_2)h$$

  - Sendo:

$$K_1 = f(x_i, y_i)$$
$$K_2 = f(x_i + a_2 h, y_i + b_{21} K_1 h)$$

  - c_1, c_2, a_2 e b_{21} são constantes que depende do método de segunda ordem
  - Erro de truncamento local é de terceira ordem $E(h^3)$. O erro global é de segunda ordem $E(h^2)$

EDOs

- Métodos de Runge Kutta
  - Runge Kutta de Segunda ordem – Euller Modificado

$$c_1 = \frac{1}{2} \quad c_2 = \frac{1}{2} \quad a_2 = 1 \quad b_{21} = 1$$

  - Runge Kutta de Segunda ordem – Ponto central

$$c_1 = 0 \quad c_2 = 1 \quad a_2 = \frac{1}{2} \quad b_{21} = \frac{1}{2}$$

  - Runge Kutta de Segunda ordem – Método de Heun

$$c_1 = \frac{1}{4} \quad c_2 = \frac{3}{4} \quad a_2 = \frac{2}{3} \quad b_{21} = \frac{2}{3}$$

EDOs

- Métodos de Runge Kutta
  - Runge Kutta de terceira ordem

$$y_{i+1} = y_i + (c_1K_1 + c_2K_2 + c_3K_3)h$$

  - Sendo:

$$K_1 = f(x_i, y_i)$$
$$K_2 = f(x_i + a_2h, y_i + b_{21}K_1h)$$
$$K_3 = f(x_i + a_3h, y_i + b_{31}K_1h + b_{32}K_2h)$$

$$c_1 = \frac{1}{6} \quad c_2 = \frac{4}{6} \quad c_3 = \frac{1}{6} \quad a_2 = \frac{1}{2} \quad a_3 = 1 \quad b_{21} = \frac{1}{2} \quad b_{31} = -1 \quad b_{32} = 2$$

EDOs

- Métodos de Runge Kutta
  - Runge Kutta de terceira ordem

$$y_{i+1} = y_i + \frac{1}{6}(K_1 + 4K_2 + K_3)h$$

$$K_1 = f(x_i, y_i)$$

$$K_2 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_1 h\right)$$

$$K_3 = f(x_i + h, y_i - K_1 h + 2K_2 h)$$

Erro de truncamento local é de quarta ordem

Erro de truncamento global é de terceira ordem

EDOs

- Métodos de Runge Kutta
  - Runge Kutta de quarta ordem

$$y_{i+1} = y_i + (c_1K_1 + c_2K_2 + c_3K_3 + c_4K_4)h$$

  - Sendo:

$$K_1 = f(x_i, y_i)$$
$$K_2 = f(x_i + a_2h, y_i + b_{21}K_1h)$$
$$K_3 = f(x_i + a_3h, y_i + b_{31}K_1h + b_{32}K_2h)$$
$$K_4 = f(x_i + a_4h, y_i + b_{41}K_1h + b_{42}K_2h + b_{43}K_3h)$$

$$c_1 = c_4 = \frac{1}{6} \quad c_2 = c_3 = \frac{2}{6} \quad a_2 = a_3 = b_{21} = b_{32} = \frac{1}{2} \quad b_{43} = a_4 = 1 \quad b_{31} = b_{42} = b_{41} = 0$$

EDOs

- Métodos de Runge Kutta
  - Runge Kutta de quarta ordem

$$y_{i+1} = y_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)h$$

  - Sendo:

$$K_1 = f(x_i, y_i)$$

$$K_2 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_1 h\right)$$

$$K_3 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_2 h\right)$$

$$K_4 = f(x_i + h, y_i + K_3 h)$$

EDOs

- Funções do Matlab
  - Resolução numérica de EDOs com valores iniciais
  - ode45, ode23, ode113, ode15s, ode23s....
  - As mais usadas são o ode45 e ode23
  - ode45 é baseada no Runge Kutta explicito de quarta e quinta ordem
  - Ode 23 é baseada no Runge Kuta de segunda e terceira ordem
  - As funções que terminam em “s” são destinada a problemas em que a a solução decai rapidamente para zero
  - Sem o termo “s” indica que são sistemas rígidos e, por isso, não decaem rapidamente

EDOs

- Funções do Matlab
  - Ode45
  - $[t, x] = ode45\ ('EDO', tspan, y0)$
  - ode45 → nome da função do Matlab
  - 'EDO' → função para ser resolvida. Pode ser usado @EDO caso ela for uma função auxiliar
  - tspan → representa o vetor do domínio da variável independente que pode conter de dois a muitos elementos
  - y0 → condições iniciais
  - t → vetor coluna para o tempo
  - x → vetor coluna solução

EDOs

- Elaboração código para resolver por Método de Euller
- Elaboração código para resolver por Runge Kutta
- Elaboração código para resolver por função do Matlab
- Ex: $y_1'' - 2(1 - y_1^2)y_1' + y_1 = 0$

$$y_1' = y_2$$
$$y_2' = 2(1 - y_1^2)y_2 - y_1$$