UNIVERSIDADE FEDERAL DO PARANÁ

CURSO DE ENGENHARIA MECÂNICA

TMEC-001 Cálculo Numérico

Professor **Luciano Kiyoshi Araki**
(sala 7-30/Lena-2, lucianoaraki@gmail.com, fone: 3361-3126)

Internet: http://ftp.demec.ufpr.br/disciplinas/TMEC001/Prof.Luciano_Araki

ALGORITMOS PARA SOLUÇÃO DE SISTEMAS DE EQUAÇÕES LINEARES

ALGORITMO – ELIMINAÇÃO DE GAUSS COM SUBSTITUIÇÃO REGRESSIVA

Para resolver o sistema linear *n* x *n*

$$\begin{aligned} E_1: &\quad a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = a_{1,n+1} \\ E_2: &\quad a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = a_{2,n+1} \\ \\ E_n &\quad a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = a_{n,n+1} \end{aligned}$$

DADOS DE ENTRADA: número de incógnitas e número de equações *n*; matriz aumentada $A = [a_{ij}]$, em que $1 \le i \le n$ e $1 \le j \le n+1$

SAÍDA: solução $x_1, x_2, \ldots, x_n$ ou mensagem de erro de que o sistema não possui solução única

Passo 1: Iniciar um ciclo com *i* variando de 1 até *n* – 1 (Correspondente ao processo de eliminação)
Executar os passo 2 a 4.

Passo 2: Fazer *p* ser o menor número inteiro com $i \le p \le n$ e $a_{pi} \ne 0$ (encontrar o pivô)
Se nenhum número inteiro *p* puder ser encontrado, então:
Mensagem de saída: Não existe solução única

Passo 3: Se $p \ne i$, então trocar linhas *i* e *p*: $(E_p) \leftrightarrow (E_i)$

Passo 4: Iniciar um ciclo com *j* variando de *i* até *n*.
Executar os passos 5 e 6

Passo 5: Fazer $m_{ji} = a_{ji} / a_{ii}$

Passo 6: Executar a seguinte substituição para as linhas: $(E_j - m_{ji} E_i) \rightarrow (E_j)$
Finalizar o ciclo para *j* iniciado no passo 4.
Finalizar o ciclo para *i* iniciado no passo 1.

Passo 7: Se $a_{nn} = 0$ então: Não existe solução única e deve-se parar a resolução.

Passo 8: Fazer $x_n = a_{n,n+1} / a_{nn}$ (Início do processo de substituição regressiva)

Passo 9: Iniciar um ciclo com *i* variando de *n* – 1 até 1, com passo –1.
Fazer $x_i = \left[a_{i,n+1} - \sum_{j=i+1}^{n} a_{ij} x_j \right] \Big/ a_{ii}$

Passo 10: Apresentar os resultados para $x_1, x_2, \ldots, x_n$.

ALGORITMO – FATORAÇÃO/DECOMPOSIÇÃO LU

Para fatorar a matriz $A=[a_{ij}]$, n x n, no produto da matriz triangular inferior $L=[l_{ij}]$ pela matriz triangular superior $U=[u_{ij}]$; isto é, $A = LU$, em que a diagonal principal de L ou de U consiste apenas de elementos unitários.

DADOS DE ENTRADA: dimensão n; os elementos a_{ij}, $1 \le i \le n$ e $1 \le j \le n$ de A; a diagonal principal de L ($l_{11}=l_{22}=\ldots=l_{nn}=1$) ou a diagonal principal de U ($u_{11}=u_{22}=\ldots=u_{nn}=1$).

SAÍDA: os elementos l_{ij}, $1 \le j \le i$ e $1 \le i \le n$ de L e os elementos u_{ij}, $i \le j \le n$ e $1 \le i \le n$ de U.

Passo 1: Selecionar l_{11} e u_{11} satisfazendo $l_{11}u_{11}=a_{11}$.
Se $l_{11}u_{11}=0$ então:
Mensagem de saída: "Fatoração impossível"

Passo 2: Realizar um ciclo com j variando de 2 até n.
Fixar $u_{1j}=a_{1j}/l_{11}$ (Primeira linha de U)
Fixar $l_{j1}=a_{j1}/u_{11}$ (Primeira coluna de L)

Passo 3: Iniciar um ciclo com i variando de 2 até $n-1$.
Executar os passos 4 e 5.

Passo 4: Selecionar l_{11} e u_{11} satisfazendo: $l_{ii}u_{ii}=a_{ii}-\sum_{k=1}^{i-1} l_{ik}u_{ki}$
Se $l_{ii}u_{ii} = 0$, então:
Mensagem de saída: "Fatoração impossível"

Passo 5: Iniciar um ciclo com j variando de $i+1$ até n:
Fazer $u_{ij}=\frac{1}{l_{ii}}\left[a_{ij}-\sum_{k=1}^{i-1} l_{ik}u_{kj}\right]$ (i-ésima linha de U)
Fazer $l_{ji}=\frac{1}{u_{ii}}\left[a_{ji}-\sum_{k=1}^{i-1} l_{jk}u_{ki}\right]$ (i-ésima coluna de L)

Passo 6: Selecionar l_{nn} e u_{nn} satisfazendo $l_{nn}u_{nn}=a_{nn}-\sum_{k=1}^{n-1} l_{nk}u_{kn}$
(Observação: Se $l_{nn}u_{nn}=0$, então $A = LU$, porém A é singular).

Passo 7: Apresentar as matrizes L e U.

ALGORITMO – FATORAÇÃO/DECOMPOSIÇÃO CHOLESKI

Para fatorar a matriz A n x n definida positiva em LL^t, em que L é triangular inferior:

DADOS DE ENTRADA: dimensão n; os elementos a_{ij}, $1 \le i \le n$ e $1 \le j \le n$ de A.

SAÍDA: os elementos l_{ij}, $1 \le j \le i$ e $1 \le i \le n$ de L (Os elementos de $U = L^t$ são $u_{ij} = l_{ji}$ para $1 \le i \le n$ e $1 \le j \le n$).

Passo 1: Fazer $l_{11}=\sqrt{a_{11}}$

Passo 2: Realizar um ciclo com j variando de 2 até n.
Fazer $l_{j1}=a_{j1}/l_{11}$

Passo 3: Iniciar um ciclo com i variando de 2 até $n-1$.

Executar os passos 4 e 5.

Passo 4: Fazer $l_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2 \right)^{1/2}$

Passo 5: Iniciar um ciclo com j variando de i +1 até n:

Fazer $l_{ji} = \dfrac{1}{l_{ii}} \left[a_{ji} - \sum_{k=1}^{i-1} l_{jk} l_{ik} \right]$

Passo 6: Fazer $l_{nn} = \left(a_{nn} - \sum_{k=1}^{n-1} l_{nk}^2 \right)^{1/2}$

Passo 7: Apresentar a matriz L.

ALGORITMO – FATORAÇÃO DE CROUT PARA SISTEMAS LINEARES TRIDIAGONAIS

Para resolver o sistema linear n x n

$$
\begin{array}{lrl}
E_1: & a_{11}x_1 + a_{12}x_2 & = a_{1,n+1} \\
E_2: & a_{21}x_1 + a_{22}x_2 + a_{23}x_3 & = a_{2,n+1} \\
 & & \\
E_{n-1}: & a_{n-1,n-2}x_{n-2} + a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n & = a_{n-1,n+1} \\
E_n: & a_{n,n-1}x_{n-1} + a_{n,n}x_n & = a_{n,n+1}
\end{array}
$$

DADOS DE ENTRADA: a dimensão n; matriz aumentada $A = [a_{ij}]$, em que $1 \le i \le n$ e $1 \le j \le n+1$

SAÍDA: solução $x_1, x_2, \ldots, x_n$ ou mensagem de erro de que o sistema não possui solução única

Passo 1: Fazer $l_{11} = a_{11}$;

$u_{12} = a_{12} / l_{11}$;

$z_1 = a_{1,n+1} / l_{11}$

Passo 2: Realizar um ciclo com i variando de 2 até n – 1

Fazer $l_{i,i-1} = a_{i,i-1}$

$l_{ii} = a_{ii} - l_{i,i-1} u_{i-1,i}$

$u_{i,i+1} = a_{i,i+1} / l_{ii}$

$z_i = (a_{i,n+1} - l_{i,i-1} z_{i-1}) / l_{ii}$

Passo 3: Fazer $l_{n,n-1} = a_{n,n-1}$

$l_{nn} = a_{nn} - l_{n,n-1} u_{n-1,n}$

$z_n = (a_{n,n+1} - l_{n,n-1} z_{n-1}) / l_{nn}$

Passo 4: Fazer $x_n = z_n$

Passo 5: Realizar um ciclo regressivo com i variando de n – 1 até 1 (com passo –1)

Fazer $x_i = z_i - u_{i,i+1} x_{i+1}$

Passo 6: Apresentar os resultados para $x_1, x_2, \ldots, x_n$.

ALGORITMO – MÉTODO DE JACOBI

Para resolver o sistema linear **A*x* = *b*** dada uma aproximação inicial $\boldsymbol{x}^{(0)}$.

DADOS DE ENTRADA: o número de equações e incógnitas *n*; os elementos a_{ij}, $1 \le i \le n$ e $1 \le j \le n$ da matriz *A*; as componentes b_i, $1 \le i \le n$ de ***b***; as componentes XO_i, $1 \le i \le n$ de $\mathbf{XO} = \boldsymbol{x}^{(0)}$; tolerância TOL; número máximo de iterações *itmax*.

SAÍDA: a solução aproximada $x_1, x_2, ..., x_n$ ou uma mensagem de que o número máximo de iterações foi excedido.

Passo 1: Realizar um ciclo com *k* variando de 1 a *itmax*.
Executar os passos 2 a 4.
Passo 2: Realizar um ciclo com *i* variando de 1 a *n*.
Fazer $x_i = \dfrac{-\sum\limits_{\substack{j=1 \\ j \ne i}}^{n} \left(a_{ij} XO_j\right) + b_i}{a_{ii}}$
Passo 3: Se $\|\mathbf{x} - \mathbf{XO}\| < tol$ então:
Apresentar os resultados: $x_1, x_2, ..., x_n$.
Interromper o processo (O procedimento foi bem sucedido).
Passo 4: Realizar um ciclo com *i* variando de 1 a *n*.
Fazer $XO_i = x_i$.
Passo 5: Mensagem de advertência: O número máximo de iterações foi excedido.

ALGORITMO – MÉTODO DE GAUSS-SEIDEL

Para resolver o sistema linear **A*x* = *b*** dada uma aproximação inicial $\boldsymbol{x}^{(0)}$.

DADOS DE ENTRADA: o número de equações e incógnitas *n*; os elementos a_{ij}, $1 \le i \le n$ e $1 \le j \le n$ da matriz *A*; as componentes b_i, $1 \le i \le n$ de ***b***; as componentes XO_i, $1 \le i \le n$ de $\mathbf{XO} = \boldsymbol{x}^{(0)}$; tolerância TOL; número máximo de iterações *itmax*.

SAÍDA: a solução aproximada $x_1, x_2, ..., x_n$ ou uma mensagem de que o número máximo de iterações foi excedido.

Passo 1: Realizar um ciclo com *k* variando de 1 a *itmax*.
Executar os passos 2 a 4.
Passo 2: Realizar um ciclo com *i* variando de 1 a *n*.
Fazer $x_i = \dfrac{-\sum\limits_{j=1}^{i-1} \left(a_{ij} x_j\right) - \sum\limits_{j=i+1}^{n} \left(a_{ij} XO_j\right) + b_i}{a_{ii}}$
Passo 3: Se $\|\mathbf{x} - \mathbf{XO}\| < tol$ então:
Apresentar os resultados: $x_1, x_2, ..., x_n$.
Interromper o processo (O procedimento foi bem sucedido).
Passo 4: Realizar um ciclo com *i* variando de 1 a *n*.
Fazer $XO_i = x_i$.
Passo 5: Mensagem de advertência: O número máximo de iterações foi excedido.

ALGORITMO – MÉTODO DE SUB OU SOBRERRELAXAÇÕES SUCESSIVAS

Para resolver o sistema linear $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ dada uma aproximação inicial $\boldsymbol{x}^{(0)}$.

DADOS DE ENTRADA: o número de equações e incógnitas n; os elementos a_{ij}, $1 \le i \le n$ e $1 \le j \le n$ da matriz A; as componentes b_i, $1 \le i \le n$ de $\boldsymbol{b}$; as componentes XO_i, $1 \le i \le n$ de $\mathbf{XO} = \boldsymbol{x}^{(0)}$; o parâmetro ω (que serve para subrelaxações, se $0 < \omega < 1$, ou sobrerrelaxações se $\omega > 1$); tolerância TOL; número máximo de iterações *itmax*.

SAÍDA: a solução aproximada $x_1, x_2, ..., x_n$ ou uma mensagem de que o número máximo de iterações foi excedido.

Passo 1: Realizar um ciclo com k variando de 1 a *itmax*.
Executar os passos 2 a 4.

Passo 2: Realizar um ciclo com i variando de 1 a n.

Fazer $x_i = (1-\omega)XO_i + \dfrac{\omega\left[-\sum_{j=1}^{i-1}\left(a_{ij}x_j\right) - \sum_{j=i+1}^{n}\left(a_{ij}XO_j\right) + b_i\right]}{a_{ii}}$

Passo 3: Se $\|\mathbf{x} - \mathbf{XO}\| < tol$ então:

Apresentar os resultados: $x_1, x_2, ..., x_n$.
Interromper o processo (O procedimento foi bem sucedido).

Passo 4: Realizar um ciclo com i variando de 1 a n.
Fazer $XO_i = x_i$.

Passo 5: Mensagem de advertência: O número máximo de iterações foi excedido.