INSTITUTO FEDERAL DE SÃO PAULO – CAMPUS VOTUPORANGA
GRADUAÇÃO EM ENGENHARIA ELÉTRICA

JOÃO VITOR DA SILVA SOUZA

FATORAÇÃO LU NA PLATAFORMA SCILAB

VOTUPORANGA/SP
2019

JOÃO VITOR DA SILVA SOUZA

FATORAÇÃO LU NA LINGUAGEM SCILAB

Relatório apresentado como parte dos requisitos para aprovação na disciplina de Cálculo Numérico, do curso de graduação em Engenharia Elétrica, do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – IFSP.

Professora: Bruna Gonçalves de Lima

VOTUPORANGA/SP

2019

SUMÁRIO

1. INTRODUÇÃO

O presente relatório tem como finalidade mostrar o desenvolvimento de um algoritmo capaz de executar o método de Fatoração LU na resolução de sistemas lineares. Todo o algoritmo foi programado na linguagem SCILAB e executado na plataforma de mesmo nome. Esse método de fatoração consiste em decompor uma dada matriz A em outras duas matrizes triangulares, resultando em sistemas de fácil resolução. Além disso, mostraremos como esse método pode ser usado não apenas para a resolução de sistemas lineares, mas também para o cálculo de determinantes. Veremos ainda o porquê desse método ser recomendável em situações onde é necessário se resolver vários sistemas lineares com a mesma matriz dos coeficientes.

Palavras chave: Sistemas lineares, Fatoração LU, SCILAB.

2. DESENVOLVIMENTO TEÓRICO

2.1. Sistemas lineares

Um sistema linear é um conjunto de equações com incógnitas em comum, sendo que uma vez encontrados os valores das incógnitas todas as equações devem ser verdadeiras. Sendo linear, o sistema não pode ter produto de incógnitas. Um sistema linear pode ser escrito como:

$$S = \begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots \cdots + a_{1n}x_n = b_2 \\ \vdots \quad \cdots \cdots \quad \vdots \quad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots \cdots + a_{mn}x_n = b_m \end{cases} \tag{2.1}$$

Sendo os termos a_{ij}, $i = 1, 2, \ldots, m, j = 1, 2, \ldots, n$, os coeficientes das incógnitas x_j, e b_i os chamados termos independentes.

Basicamente existem dois tipos de métodos para a resolução de um sistema, uma delas é o método iterativo, aonde por sucessivas etapas os valores das incógnitas vão se atualizando e se aproximando de seu valor real, e a outra maneira é o método exato, que resulta em valores bem definidos para as incógnitas, sendo a decomposição LU um exemplo de método exato.

A seguir será apresentado o conceito de matriz e como elas podem auxiliar na resolução de sistemas lineares.

2.2. Matriz

Matriz, no seu conceito mais básico, é uma tabela de elementos a_{ij} dispostos em m linhas e n colunas sobre um corpo numérico K, onde $a_{ij} \in K$.

2.2.1. Dimensão

$$A = [a_{ij}]_{mxn} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}_{mxn} \tag{2.2}$$

A dimensão de uma matriz A é definida como $\dim(A) = mxn$, ou seja, a dimensão de uma matriz é representada pelo seu número de linhas (m) e o seu número de colunas (n). Os subíndices dos elementos a_{ij} dispostos na matriz A simbolizam a respectiva posição desse elemento $(i = \text{linha}, j = \text{coluna})$.

2.2.2. Adição de Matrizes

$$A + B = \begin{bmatrix} a_{11} + b_{11} & \cdots & a_{1n} + b_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & \cdots & a_{mn} + b_{mn} \end{bmatrix}_{mxn} \tag{2.3}$$

Na soma de matrizes realizamos uma adição usual entre os termos de mesmo índice. A única condição necessária para que a soma de matrizes seja possível é que a dimensão de A seja igual a dimensão de B.

2.2.3. Multiplicação por Escalar

$$\lambda A = \begin{bmatrix} \lambda a_{11} & \cdots & \lambda a_{1n} \\ \vdots & \ddots & \vdots \\ \lambda a_{m1} & \cdots & \lambda a_{mn} \end{bmatrix}_{mxn} \tag{2.4}$$

Na multiplicação de uma matriz A por um escalar $\lambda \in K$, multiplicamos cada elemento a_{ij} da matriz A pelo escalar λ, resultando em uma matriz com elementos λa_{ij}.

2.2.4. Produto de Matrizes

$$C = A \cdot B = \begin{bmatrix} a_{11}b_{11} + \cdots + a_{1n}b_{n1} & \cdots & a_{11}b_{1k} + \cdots + a_{1n}b_{nk} \\ \vdots & \ddots & \vdots \\ a_{m1}b_{11} + \cdots + a_{mn}b_{n1} & \cdots & a_{m1}b_{1k} + \cdots + a_{mn}b_{nk} \end{bmatrix}_{mxk} \tag{2.5}$$

A condição necessária para que seja possível o produto de matrizes é que o número de colunas de A seja igual ao número de linhas de B. No produto de matrizes somamos o produto dois a dois dos elementos da primeira linha de A pela primeira coluna de B, essa soma resulta no elemento c_{11} da matriz resultante, ao fazermos o mesmo processo só que com a segunda coluna de B teremos o elemento c_{12}, e assim por diante. Os subíndices dos elementos resultantes mostram qual linha de A (i) foi multiplicada pela coluna de B (j). A dimensão da matriz resultante será o número de linhas de A pelo número de colunas de B.

Propriedades do produto de matrizes:

1 – Não é comutativo: $A \cdot B \neq B \cdot A$; (2.6)

2 – É associativo: $(A \cdot B) \cdot C = A \cdot (B \cdot C)$; (2.7)

3 – É distributivo: $\begin{cases} A \cdot (B + C) = A \cdot B + A \cdot C \\ (B + C) \cdot A = B \cdot A + C \cdot A \end{cases}$. (2.8)

2.2.5. Determinante

O determinante é um número associado a uma matriz nxn (quadrada) que apresenta algumas propriedades significativas no estudo das matrizes, como mostrar se uma matriz admite inversa $(\det(\mathrm{A}) \neq 0)$, ou se um conjunto de equações dispostos em uma matriz são linearmente dependentes $(\det(\mathrm{A}) = 0)$, resultando em um possível sistema indeterminado, entre outras funções. O determinante deriva de uma sequência lógica de operações algébricas sobre os elementos de uma matriz e essa sequência varia de acordo com a ordem (n) da matriz.

Para $\mathrm{n} = 1$

$$\begin{gathered} \mathrm{A} = \left[\mathrm{a_{ij}}\right]_{\mathrm{1x1}} \\ \det(\mathrm{A}) = \mathrm{a_{11}} \end{gathered} \tag{2.9}$$

Para $\mathrm{n} = 2$

$$\begin{gathered} \mathrm{A} = \left[\mathrm{a_{ij}}\right]_{\mathrm{2x2}} \\ \det(\mathrm{A}) = \mathrm{a_{11}a_{22}} - \mathrm{a_{12}a_{21}} \end{gathered} \tag{2.10}$$

Para $\mathrm{n} = 3$

$$\begin{gathered} \mathrm{A} = \left[\mathrm{a_{ij}}\right]_{\mathrm{3x3}} \\ \det(\mathrm{A}) = \mathrm{a_{11}a_{22}a_{33}} + \mathrm{a_{21}a_{32}a_{13}} + \mathrm{a_{12}a_{23}a_{31}} \\ -\mathrm{a_{13}a_{22}a_{31}} - \mathrm{a_{12}a_{21}a_{33}} - \mathrm{a_{23}a_{32}a_{11}} \end{gathered} \tag{2.11}$$

Para $\mathrm{n} \geq 4$ (Teorema de Laplace)

Quando a ordem de uma matriz A for maior ou igual a quatro é recomendável que se use o Teorema de Laplace que diz que o determinante de qualquer matriz é igual à soma dos cofatores A_{ij} dessa matriz.

$$\det(A) = \sum A_{ij} \tag{2.12}$$

$$A_{ij} = (-1)^{i+j} D_{ij} \tag{2.13}$$

O elemento D_{ij} presente em (2.13) é chamado de menor complementar e ele é encontrado ao fazer o determinante da matriz resultante quando se elimina a linha e a coluna do elemento a_{ij} escolhido para o cálculo do cofator.

Propriedades dos determinantes:

1 - $\det(AB) = \det(A)\det(B)$; (2.14)

2 - $\det(\lambda A) = \lambda^n \det(A), \lambda \in K$; (2.15)

3 - $\det(A^\lambda) = [\det(A)]^\lambda, \lambda \in K$; (2.16)

4 - $\det(A^{-1}) = \frac{1}{\det(A)}, \det(A) \neq 0$; (2.17)

5 - Se A é uma matriz triangular: $\det(A) = \prod a_{ii}$; (2.18)

6 - Se λ multiplica alguma linha de A: $\det(A') = \lambda \det(A)$; (2.19)

7 - Se houver permutação entre duas linhas ou duas colunas de A: $\det(A') = (-1)^i \det(A)$, onde i é o número de permutações; (2.20)

8 - Se somarmos a uma linha um múltiplo de outra linha de A: $\det(A') = \det(A)$; (2.21)

9 - Se A tem pelo menos uma linha ou coluna composta apenas de zeros, então $\det(A) = 0$;

10 - Se A tem duas ou mais linhas múltiplas umas das outras: $\det(A) = 0$.

2.2.6. Matriz Quadrada

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}_{nxn} \tag{2.22}$$

A matriz quadrada nada mais é do que uma matriz com o número de linhas igual ao número de colunas $(m = n)$. Apenas matrizes quadradas apresentam determinantes.

2.2.7. Matriz Identidade

$$I_n = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}_{nxn} \tag{2.23}$$

A matriz identidade é uma matriz definida pela expressão:

$$a_{ij} := \begin{cases} 1, se\ i = j \\ 0, se\ i \neq j \end{cases} \tag{2.24}$$

ou seja, os elementos da diagonal principal da matriz identidade são iguais a 1 e todos os outros são nulos. A matriz identidade é o elemento neutro na multiplicação de matrizes.

2.2.8. Matriz Ampliada

$$C = \begin{bmatrix} a_{11} & \cdots & a_{1n} & \vdots & b_{11} & \dots & b_{1k} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} & \vdots & b_{m1} & \dots & b_{mk} \end{bmatrix}_{mx(n+k)} \tag{2.25}$$

Seja a matriz $A = [a_{ij}]_{mxn}$ e a matriz $B = [b_{ij}]_{mxk}$, a matriz ampliada de A e B será a matriz $C = [c_{ij}]_{mx(n+k)}$.

2.2.9. Matriz Triangular

Existem dois tipos de matrizes triangulares, as superiores (*upper*) e as inferiores (*lower*). O formato geral de uma matriz triangular superior é:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & a_{nn} \end{bmatrix}_{nxn} \tag{2.26}$$

e o de uma triangular inferior é:

$$A = \begin{bmatrix} a_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}_{nxn} \tag{2.27}$$

2.2.10. Característica

A característica $\rho(A)$ de uma matriz A é o valor que representa o número de linhas não nulas de A quando esta está em sua forma escalonada.

2.3. Teorema de Rouché-Capelli

Sejam as matrizes $A = [a_{ij}]_{nxn}$, $B = [b_i]_{nx1}$ e a matriz ampliada $C = [A \vdots B]_{nx(n+1)}$. O Teorema de Rouché-Capelli nos diz que:

Se $\rho(A) = \rho(C) = n$, então o sistema tem o mesmo número de equações e incógnitas, portanto é possível e determinado (SPD).

Se $n > \rho(A) = \rho(C)$, então o sistema tem mais incógnitas do que equações, logo o sistema é possível e indeterminado (SPI). Para esse mesmo caso de sistema temos que $\omega = n - \rho(C)$, onde ω é o número de variáveis livres do sistema.

E se $\rho(C) > \rho(A)$ o sistema é impossível (SI), pois em algum momento do escalonamento iremos inevitavelmente se deparar com um absurdo do tipo:

$$0 = b_i \neq 0 \rightarrow 0 \neq 0 \downarrow$$

2.4. Eliminação de Gauss

A Eliminação de Gauss é um método que nos permite simplificar (escalonar) matrizes de maneira a facilitar a resolução de sistemas lineares. O método consiste em eliminar termos das equações de um sistema linear disposto em uma matriz usando operações elementares. Essas operações não alteram a solução do sistema, de modo que se resolvermos um sistema escalonado também conseguiremos a solução para o sistema original. Existem três operações elementares, e são elas:

1 – Permutar duas linhas do sistema;

2 – Multiplicar uma linha por uma constante diferente de zero;

3 – Somar um múltiplo de uma linha a outra linha.

Por meio da Eliminação de Gauss podemos conseguir um tipo de matriz que é essencial para a resolução de sistemas lineares, essas matrizes são chamadas de matrizes escalonadas e a sua forma geral é:

$$C = \begin{bmatrix} a_{11} & \cdots & a_{1n} & \vdots & b_1 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & a_{mn} & \vdots & b_m \end{bmatrix} \tag{2.28}$$

A Eliminação de Gauss pode ser expressa por meio de um processo onde as linhas são atualizadas de modo a conseguirmos matrizes escalonadas. Algebricamente temos que, para uma matriz $A = [a_{ij}]_{nxn}$, vale a seguinte expressão:

$$R_j \leftarrow R_j - \alpha_{i,j} R_i \quad , i < j \tag{2.29}$$

onde R_j é a j-ésima linha de A, R_i é a i-ésima linha de A, e $\alpha_{i,j}$ é dado por:

$$\alpha_{i,j} = \frac{a_{ji}}{a_{ii}} \tag{2.30}$$

A expressão (2.29) pode ser entendida da seguinte maneira:

Dada uma linha R_j de A, cujo primeiro elemento a_{ji} é não-nulo, podemos atualizá-la de tal forma que ao ser subtraída de um múltiplo de R_i, no caso $\alpha_{i,j} R_i$, a_{ji} passará a ser igual a zero.

Vale ressaltar que pela condição $i < j$, o pivô a_{ii} (primeiro elemento não nulo da linha) sempre será o ponto de partida para que possamos zerar os elementos logo abaixo de sua coluna, chegando assim em uma matriz escalonada.

2.5. Fatoração LU

A Fatoração LU é um método para a resolução de sistemas lineares, onde dado um sistema linear S:

$$S = \begin{cases} a_{11}x_1 + a_{12}x_2 + & \cdots & \cdots & +a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + & \cdots & \cdots & +a_{1n}x_n & = & b_2 \\ \vdots & \cdots & \cdots & \vdots & & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + & \cdots & \cdots & +a_{nn}x_n & = & b_n \end{cases} \tag{2.31}$$

sendo a matriz dos coeficientes A:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \tag{2.32}$$

a matriz das incógnitas X:

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \tag{2.33}$$

e a matriz dos termos independentes B:

$$B = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \tag{2.34}$$

fatoramos A em duas matrizes triangulares, uma triangular inferior $L = [l_{ij}]_{nxn}$ e outra triangular superior $U = [u_{ij}]_{nxn}$, tal que:

$$A = L \cdot U \tag{2.35}$$

Para começar a desenvolver o método de fatoração vamos escrever o sistema (2.31) da seguinte forma:

$$A \cdot X = B \tag{2.36}$$

Levando (2.35) em consideração podemos escrever (2.36) como:

$$(L \cdot U) \cdot X = B \tag{2.37}$$

$$L \cdot (U \cdot X) = B \tag{2.38}$$

$$L \cdot Y = B \tag{2.39}$$

$$Y = U \cdot X \tag{2.40}$$

Dessa forma temos que o sistema original se resume em resolver dois sistemas triangulares, primeiro resolvemos $L \cdot Y = B$ para Y e depois resolvemos $U \cdot X = Y$ para X. Em comparação a um sistema não triangular, resolver sistemas triangulares é muito mais fácil já que uma das incógnitas sempre já está definida.

Para encontrar L devemos escalonar A. Ao final do escalonamento podemos escrever L como:

$$L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \alpha_{1,2} & 1 & \cdots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \alpha_{1,n} & \cdots & \alpha_{n-1,n} & 1 \end{bmatrix} \tag{2.41}$$

sendo os termos $\alpha_{i,j}$ os já definidos em (2.30).

Em relação à matriz U temos que:

$$U = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}' & \cdots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{nn}' \end{bmatrix} \tag{2.42}$$

onde os elementos a_{ij} e a_{ij}' são os termos que restaram do escalonamento de A.

Uma vez encontradas as matrizes L e U, podemos resolver o sistema $L \cdot Y = B$ usando substituição progressiva, e assim encontrar os termos y_i. Essa substituição é definida formalmente pela expressão (2.43):

$$y_i = \frac{1}{l_{ii}} \left(b_i - \sum_{j=1}^{i-1} l_{ij} y_j \right) \tag{2.43}$$

Encontrada a matriz Y, podemos resolver o sistema $U \cdot X = Y$ usando substituição regressiva determinando assim os termos de X. A substituição regressiva é expressa da seguinte forma:

$$x_i = \frac{1}{u_{ii}} \left(y_i - \sum_{j=1+1}^{n} u_{ij} x_j \right) \tag{2.44}$$

Embora seja esse o método empregado para se resolver sistemas lineares usando o algoritmo de Fatoração LU, percebe-se que podemos encontrar um problema durante o escalonamento. Como o cálculo de $\alpha_{i,j}$ envolve uma divisão de termos da matriz A, caso $\alpha_{i,i}$ seja nulo então teremos uma indefinição ou indeterminação matemática, pois teríamos uma divisão por zero. Para evitar esse problema multiplicamos a matriz A por uma matriz de permutação P. A matriz de

permutação é uma matriz que troca as linhas de A, evitando assim uma possível divisão por zero.

Como dito no início, podemos usar o método de Fatoração LU para encontrar o determinante de A, pois como a matriz A pode ser escrita sendo $P \cdot L \cdot U$, aplicamos a função determinante em ambos os lados da equação, resultando em:

$$\det(A) = \det(P \cdot L \cdot U) \tag{2.45}$$

$$\det(A) = \det(P)\det(L)\det(U) \tag{2.46}$$

$$\det(A) = (-1)^{i} \prod u_{ii} \tag{2.47}$$

O determinante de uma matriz triangular é o produtório dos elementos da diagonal principal, então $\det(L) = 1$, e pela 7ª propriedade dos determinantes podemos escrever $\det(P) = (-1)^{i}$, sendo i o número de permutações da matriz P.

Até esse ponto já se deve ter percebido que a fatoração da matriz A independe do valor de B, de modo que uma vez fatorada podemos resolver qualquer sistema da forma $A \cdot X = B$ para qualquer valor de B sem que precisemos escalonar todo o sistema novamente, bastando apenas resolver dois sistemas triangulares e já teremos o vetor X. Por comparação ao escalonamento completo, fica clara a vantagem computacional de se resolver sistemas com a mesma matriz dos coeficientes (A) pelo método de Fatoração LU. Resolver sistemas triangulares é muito mais simples do que sistemas completos.

3. ALGORITMO

Como já mencionado, vamos programar na linguagem e plataforma SCILAB uma função que terá como dados de entrada as matrizes A e B, e a saída será a matriz solução X. O algoritmo está anexado no Apêndice A e todas as etapas estão comentadas para um melhor entendimento do leitor. Quando necessário citaremos em qual seção do Desenvolvimento Teórico foi explicado como funciona tal processo ou como é uma determinada matriz.

3.1. Aplicação

Existem três principais leis que regem o funcionamento de um circuito elétrico de corrente contínua, uma delas é a chamada Primeira Lei de Ohm que é enunciada como:

$$U = R \cdot i \tag{3.1}$$

onde U é a queda de tensão sobre um determinado componente do circuito, R é a resistência elétrica desse componente e i é a corrente elétrica que o percorre. As outras duas leis fazem parte do conjunto das Leis de Kirchhoff. A primeira delas é chamada de Lei dos Nós, que diz que a corrente que sai de um encontro entre dois ou mais condutores interligados é igual a soma de todas as correntes que entram no nó por estes condutores. Essa lei se baseia no princípio da conservação de carga. A segunda Lei de Kirchhoff é chamada de Lei das Malhas. Essa lei diz que a soma de todos os ganhos de tensões e quedas de tensão em uma malha fechada é igual a zero. Enunciada as três principais leis vamos examinar como a corrente se comporta no circuito elétrico da Figura 1.

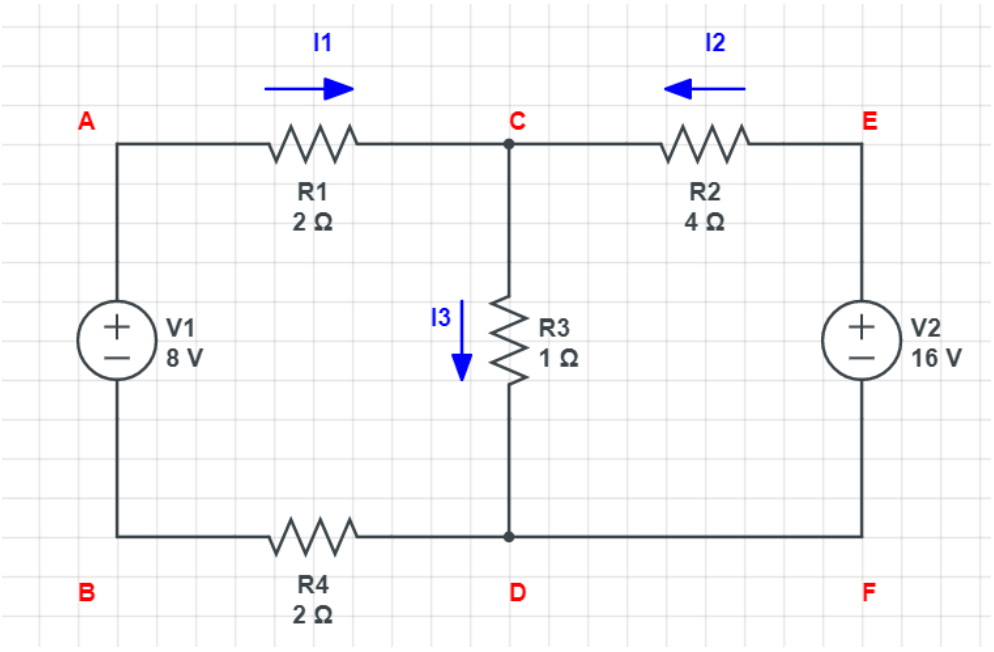

Figura 1: Circuito elétrico

Fonte: Próprio autor

Na malha ABCD tem-se:

$$4i_1 + i_3 = 8 \tag{3.2}$$

Pela malha CDEF:

$$4i_2 + i_3 = 16 \tag{3.3}$$

Aplicando a Lei dos Nós no nó C obtemos a última expressão necessária para construirmos um sistema linear possível e determinado. Portanto:

$$i_3 = i_1 + i_2 \tag{3.4}$$

Agrupando as equações (3.2), (3.3) e (3.4) em um sistema linear temos:

$$\begin{cases} 4i_1 + i_3 = 8 \\ 4i_2 + i_3 = 16 \\ i_1 + i_2 - i_3 = 0 \end{cases} \tag{3.5}$$

Criando a matriz dos coeficientes A, a matriz dos termos independentes B e a matriz das incógnitas X para o sistema (3.5):

$$A \cdot X = B \rightarrow \begin{bmatrix} 4 & 0 & 1 \\ 0 & 4 & 1 \\ 1 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 16 \\ 0 \end{bmatrix} \tag{3.6}$$

Aplicando as matrizes A e B no algoritmo teremos os dados apresentados na Tabela 1.

Tabela 1: Dados do algoritmo.

Matriz dos coeficientes	$A = \begin{bmatrix} 4 & 0 & 1 \\ 0 & 4 & 1 \\ 1 & 1 & -1 \end{bmatrix}$
Matriz dos termos independentes	$B = \begin{bmatrix} 8 \\ 16 \\ 0 \end{bmatrix}$
Determinante de A	$\det(A) = -24$
Determinante de U	$\det(U) = -24$
Característica de A	$\rho(A) = 3$
Característica de $[A : B]$	$\rho([A : B]) = 3$

Matriz de permutação	$\mathrm{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Matriz triangular inferior	$\mathrm{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.25 & 0.25 & 1 \end{bmatrix}$
Matriz triangular superior	$\mathrm{U} = \begin{bmatrix} 4 & 0 & 1 \\ 0 & 4 & 1 \\ 0 & 0 & -1.5 \end{bmatrix}$
Matriz Y	$\mathrm{Y} = \begin{bmatrix} 8 \\ 16 \\ -6 \end{bmatrix}$
Matriz solução	$\mathrm{X} = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix}$

Realizando a prova real:

$$\mathrm{A} \cdot \mathrm{X} = \begin{bmatrix} 4 & 0 & 1 \\ 0 & 4 & 1 \\ 1 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 4 \cdot 1 + 0 \cdot 3 + 1 \cdot 4 \\ 0 \cdot 3 + 4 \cdot 3 + 1 \cdot 4 \\ 1 \cdot 1 + 1 \cdot 3 + (-1) \cdot 4 \end{bmatrix} = \begin{bmatrix} 8 \\ 16 \\ 0 \end{bmatrix} = \mathrm{B}$$

Portanto, temos como solução do sistema:

$$\begin{cases} \mathrm{i}_1 = 1\mathrm{A} \\ \mathrm{i}_2 = 3\mathrm{A} \\ \mathrm{i}_3 = 4\mathrm{A} \end{cases}$$

Pela Tabela 1 percebe-se o que já havia sido discutido antes sobre os determinantes. O determinante de A de fato é igual ao determinante de U. Ainda vemos que para esse exemplo a matriz de permutação é igual a matriz identidade de ordem três pois não se precisou permutar as linhas de A durante o escalonamento.

4. CONCLUSÃO

O exemplo estudado apresentava apenas três incógnitas, sendo não muito complexo resolve-lo sem recursos computacionais, no entanto, quando o número de incógnitas aumenta os cálculos se tornam muito extensos, sendo inviável faze-los manualmente, nesses casos o algoritmo se torna uma ferramenta indispensável. Na aplicação real dos sistemas lineares, principalmente na engenharia civil e engenharia elétrica, o número de incógnitas é absurdamente grande, e na maioria das vezes o custo computacional deve ser reduzido, e nesse aspecto a Fatoração LU se mostra muito eficaz. Se percebermos, por exemplo, que se alterarmos o valor das fontes de tensão do circuito estudado só estaríamos alterando os valores da matriz B, então não precisamos escalonar a matriz A novamente, pois esta já se encontra fatorada e não teve seus termos alterados, reduzindo assim o custo computacional. Conclui-se, portanto, que para aplicações onde a matriz dos coeficientes se mantém inalterada, a Fatoração LU se torna uma ferramenta absurdamente econômica em se tratando de tempo e custo para se resolver um sistema linear.

5. BIBLIOGRAFIA

[1] BOLDRINI, J. L. **Álgebra Linear**. 3ª. ed. [S.l.]: Harbra, 1984.

[2] KAISARE, D. N. MATLAB Tutorial: Learn MATLAB Programming for Numerical Computation. **Youtube**, 2017. Disponivel em: <https://www.youtube.com/playlist?list=PLEJxKK7AcSEFtiPAlDTJywUIAIF8FGYXA>. Acesso em: Maio 2019.

[3] PESCADOR, A.; POFFO, J.; ROBERTO, C. **Aplicação de Álgebra Linear na Engenharia**. Congresso Brasileiro de Educação em Engenharia. [S.l.]: [s.n.]. 2011. p. 2-3.

[4] VALLE, M. E. Aula 6 Eliminação de Gauss e Fatoração LU. **Instituto de Matemática, Estatística e Computação Científica**. Disponivel em: <https://www.ime.unicamp.br/~valle/Teaching/2015/MS211/Aula6.pdf>. Acesso em: 28 Maio 2019.

APÊNDICE A

A.1 Implementação do algoritmo no SCILAB

```
// Função para o método de Fatoração LU na resolução de sistemas lineares.

// Nossa função será chamada de FatoracaoLU, e os inputs serão justamente
// as matrizes A e B do sistema linear AX = B.

function FatoracaoLU(A, B)

// Criamos uma variável 'Ae' que irá assumir o valor da matriz A.

Ae = A;

// Calculamos a dimensão (2.2.1) das matrizes iniciais usando a função size
// e armazenamos esse valor em diferentes variáveis. As variáveis m e c
// armazenarão quantas linhas a matriz A e a matriz B tem, respectivamente.
// Enquanto n e d armazenam o número de colunas.

[m,n] = size(A);
[c,d] = size(B);

// Criamos uma condição para que A seja quadrada (2.2.6), ou seja, m precisa
// ser igual a n, caso contrário o código se encerra.

if m~=n
   printf('\nA matriz A não é quadrada, logo não pode ser fatorada.\n\n');
   return
end

// Calculamos a característica de A (2.2.10) usando a função rank e salvamos
// esse valor na variável CA. O determinante de A (2.2.5) também é calculado
// usando a função det.

CA = rank(A);
detA = det(A);

// Para criarmos a matriz ampliada [A,B] (2.2.8) precisamos de duas condições.
// A matriz B precisa ser uma matriz coluna (só ter uma coluna) e o número de
// linhas de A deve ser igual ao de B.

if d~=1
   printf('\nA matriz B deve ser uma matriz coluna.\n\n');
   return
end
```

```
if m~=c
   printf('\nO número de linhas de A é diferente do número de linhas de B.\n\n');
   return
end

// Estabelecida as duas condições podemos criar [A,B] e salvar seu valor na
// varíavel Ab. Aqui a característica de Ab também é calculada.

Ab = [A,B];
CAb = rank(Ab);

// Como visto em (2.3), pelo Teorema Rouché-Capelli se p(Ab)>p(A) o sistema é
// impossível (SI). E se det(A)=0 (o que implica em um número maior de equações
// do que incógnitas) e p(Ab)=p(A), então o sistema é possível e indeterminado (SPI).

if CAb>CA
   printf('\nNão há soluções\n\n');
   return
end

if (detA==0)&&(CAb==CA)
   printf('\nHá infinitas soluções\n\n');
   return
end

// Se o algoritmo chegou até aqui então quer dizer que A tem os requisitos para
// poder ser fatorada. Vamos começar definindo os valores inicias para alpha (2.30)
// e I que será usado para o cálculo do determinante de U caso a matriz A precise
// ser permutada (2.20). Nesse caso o valor inicial será zero para as duas variáveis.

alpha = 0;
I = 0;

// Definimos também o valor inicial para a matriz L e para a matriz de permutação
// P. A função eye(n,n) cria uma matriz identidade (2.2.7) de ordem n e armazena
// essa matriz em uma determinada variável, no caso em L e P.

L = eye(n,n);
P = eye(n,n);

// Criamos um loop que irá percorrer todas as colunas da matriz ampliada Ab da
// esquerda para a direita.

for j = 1:n

   // As operações elementares (2.4) podem ser expressas na forma de matrizes,
   // estas são chamadas de matrizes elementares. Vamos definir uma matriz
   // elementar que irá armazenar quais linhas de A foram permutadas (caso seja
   // necessário) em cada loop, sendo que de um loop para o outro a matriz é
   // resetada ao seu valor inicial, que é a matriz identidade (eye).
```

```
E = eye(n,n);

 // Vamos verificar se o pivô da j-ésima linha precisa ter sua linha permu-
 // tada, para isso seu valor precisa ser zero.

 if Ae(j,j) == 0

   // Como a linha precisa ser permutada, vamos comparar os elementos da
   // j-ésima coluna e verificar qual é o de maior valor usando a função
   // max. Essa função salva o valor máximo na primeira variável (v) e o
   // número da sua linha na segunda (p). Quando usamos o comando Ae(:,j)
   // estaremos interessado em todas linhas, simbolizadas por :, que cruzam
   // a linha j, na matriz Ae. Em outras palavras estamos avaliando a
   // j-ésima coluna de Ae.

   colj = Ae(:,j);
   [v,p] = max(colj);

   // Salvamos o valor da j-ésima linha de E na variável memoria.

   memoria = E(j,:);

   // Permutamos as linhas de acordo com o valor definido para p.

   E(j,:) = E(p,:);
   E(p,:) = memoria;

   // Como dito antes, as operações elementares podem ser representadas por
   // matrizes, no nosso caso a matriz elementar é E. Quando encontrada
   // podemos usa-la para modificar certa matriz usando um produto de
   // matrizes(2.2.4). Como queremos permutar as linhas de Ae, multiplicamos ela
   // pela matriz elementar E.

   Ae = E*Ae;

   // Para salvarmos as alterações já feitas precisamos multiplicar P por E,
   // já que ao final do loop a matriz elementar será resetada.

   P = E*P;

   // Como as linhas foram permutadas atualizamos o valor de I em uma unidade.

   I = I + 1;

 end

     // Verificada a n-ésima coluna, podemos agora escalona-la (2.28).
```

```
        for i = j:(m-1)

            alpha = Ae(i+1,j)/Ae(j,j);
            Ae(i+1,:) = Ae(i+1,:) - alpha*Ae(j,:);

        end

end

// Definimos o termo final que será multiplicado pelo determinante de U.

Id = (-1)^I;

// Sendo U o que restou do escalonamento, igualamos U à matriz escalonada Ae.

U = Ae;

// Resetamos o valor de Ae para P*A. Dessa forma poderemos encontrar os valores
// alpha que constituem a matriz L. Note que não podíamos ter usado os valores
// de alpha encontrados no primeiro escalonamento pois a matriz Ae ainda estava
// sendo permutada, com isso a matriz L também seria permutada a cada loop, o
// que obviamente alteraria toda a sua estrutura, gerando valores incompatíveis.
// Ainda vale ressaltar que se a matriz Ae não precisasse ser permutada, o valor
// de P seria o seu valor inicial, P = I, o que não alteraria o valor de Ae, que
// inicialmente é igual a A.

Ae = P*A;

// Definindo L' para a matriz já permutada (L'=PL). Por praticidade ainda repre-
// sentaremos L' por L.

for j = 1:n

        for i = j:(m-1)

            alpha = Ae(i+1,j)/Ae(j,j);
            Ae(i+1,:) = Ae(i+1,:) - alpha*Ae(j,:);
            L(i+1,j) = alpha;

        end
end

// Para manter a igualdade dos termos em AX = B precisamos multiplicar P nos dois
// lados da equação, PAX = PB --- (PL)(UX) = PB --- L'Y = PB. Portanto, Y será
// calculado usando substituição progressiva (2.43) para L' e PB.

Bd = P*B;

// O valor inicial de Y e X será definido como um vetor coluna de n linhas compos-
// to apenas de zeros. Para fazer isso usamos a função zeros.
```

```
Y = zeros(n,1);

// O primeiro termo de Y será sempre igual ao primeiro termo de Bd.

Y(1,1) = Bd(1,1);

// Calculando Y usando substituição progressiva.

for i = 2:n
   Y(i,1) = Bd(i,1);
      for j = i:-1:2
         N(j) = Y(j-1,1)*L(i,j-1);
         Y(i,1) = Y(i,1) - N(j);
      end
end

X = zeros(n,1);

// O último termo de X será o n-ésimo termo de Y sobre o elemento de posição
// nn da matriz U.

X(n,1) = Y(n,1)/U(n,n);

// Calculando X usando substituição regressiva (2.44).

for i = (n-1):-1:1
   X(i,1) = Y(i,1)/U(i,i);
   for j = n:-1:(i+1)
      N(j) = X(j,1)*U(i,j);
      X(i,1) = X(i,1) - N(j)/U(i,i);
   end
end

// A função disp mostra na tela o valor do vetor X.

disp(X,'X = ');

// Encerra a função.

endfunction
```