ORDENAÇÃO DA ELIMINAÇÃO GAUSSIANA
EM SISTEMAS DE POTÊNCIA

Por

MARCOS JOSÉ DE ALMEIDA GAMA

TESE DE MESTRADO

Apresentada à Coordenação Setorial de Pós-Graduação e Pesquisa da Pró-Reitoria para Asssuntos do Interior da Universidade Federal da Paraíba, em cumprimento às exigências para obtenção do GRAU DE MESTRE EM CIÊNCIAS.

Campina Grande, setembro de 1982

G184o Gama, Marcos José de Almeida.
Ordenação da eliminação gaussiana em sistemas de potência / Marcos José de Almeida Gama. - Campina Grande, 1982.
ca 90 f.

Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal da Paraiba, 1982.
"Orientação: Prof. Dr. Drumond Xavier Cavalcanti".
Referências.

1. Sistemas de Potência. 2. Eliminação Gaussiana. 3. Sistemas de Equações Lineares. 4. Engenharia Elétrica - Dissertação. I. Cavalcanti, Drumond Xavier. II. Universidade Federal da Paraiba - Campina Grande (PB). III. Título

CDU 621.3.016.2(043)

AGRADECIMENTOS

Agradeço a todos que prestaram colaboração indispensável à elaboração deste trabalho: a Coordenação Setorial de Pós-Graduação em Engenharia Elétrica e ao Grupo de Sistemas de Potência do CCT da UFPb, pela atenção que me diepensaram; a minha esposa ANA CRISTINA RABELO LOUREIRO pela compreensão, incentivo, e correção dos manuscritos; a ALEXANDRE GAMA pelo companheirismo; aos meus pais e irmãos pelas condições básicas que proporcionaram a minha vida escolar. Agradecimento especial devo ao Professor Dr. DRUMOND XAVIER CAVALCANTI que tão dignamente exerceu sua função de Orientador, corrigindo as partes mal formuladas, incentivando nos momentos difíceis, apresentando idéias, imbuindo-me do gosto pela pesquisa.

O autor.

Í N D I C E

CAPÍTULO I

INTRODUÇÃO

Em estudo de fluxo de cargas, estabilidade,curto-circuito, e em outras aplicações de sistemas de potência, faz-se necessário solucionar problemas que envolvem soluções de sistemas de equações lineares, que são geralmente esparsas, utilizando métodos computacionais.

O método de triangularização/decomposição de Gauss, quando utilizado na solução de sistemas de equações lineares esparsas, gera elementos não nulos em posições antes ocupadas por zero.

Ordenar um sistema de equações lineares, é estabelecer uma ordem que seguida na decomposição/triangularização das variáveis do sistema, atenda a objetivos pré-estabelecidos.Para aplicações em sistemas de potência, esquemas de ordenações foram propostos por Tinney |4| e Sasson|5|.

No capítulo II deste trabalho, descreve-se o método de eliminação de Gauss e, mostra-se a necessidade de ordenar as barras de um sistema.

No capítulo III, apresenta-se os esquemas de ordenações propostos por Tinney e, o esquema proposto por Sasson.

No capítulo IV, propõe-se um esquema de ordenação que é uma extensão do esquema-III Tinney, denominado esquema de ordenação ITERATIVA. Alguns truncamentos são propostos para minimizar o tempo de processamento do esquema ITERATIVA.

No capítulo V, faz-se uma série de comparações entre os esquemas de ordenações apresentados.

No apêndice, apresenta-se o programa dos esquemas propostos por Tinney |4|,o do esquema de ordenação ITERATIVA e o do esquema ORDEM-3, escritas em Fortran IV.

CAPÍTULO II

MÉTODO DE ELIMINAÇÃO DE GAUSS E A NECESSIDADE DE UMA ORDENAÇÃO

Neste capítulo descreve-se o método de eliminação de Gauss, usado para a solução de sistemas de equações lineares |1|. Particulariza-se a aplicação do método para matrizes esparsas, que aparecem em vários problemas de simulação do comportamento de sistemas de potência.

2.1. Descrição do método de eliminação de Gauss

Um dos métodos mais usados para resolver um sistema de equações lineares é atribuido a Gauss. Este consiste na eliminação sucessiva das variáveis de um sistema de equações.

Para uma descrição detalhada do método de eliminação de Gauss, será usado o sistema de quatro equações lineares apresentado abaixo:

$$
\begin{aligned}
a_{1,1}\,X_1 + a_{1,2}\,X_2 + a_{1,3}\,X_3 + a_{1,4}\,X_4 &= Y_1 \\
a_{2,1}\,X_1 + a_{2,2}\,X_2 + a_{2,3}\,X_3 + a_{2,4}\,X_4 &= Y_2 \\
a_{3,1}\,X_1 + a_{3,2}\,X_2 + a_{3,3}\,X_3 + a_{3,4}\,X_4 &= Y_3 \\
a_{4,1}\,X_1 + a_{4,2}\,X_2 + a_{4,3}\,X_3 + a_{4,4}\,X_4 &= Y_4
\end{aligned}
\qquad (2.1.)
$$

A etapa inicial, denominada NORMALIZAÇÃO da primeira equação, consiste em dividir a primeira equação pelo coeficiente ($a_{1,1} \neq 0$), obtendo-se:

$$X_1 + \frac{a_{1,2}}{a_{1,1}} X_2 + \frac{a_{1,3}}{a_{1,1}} X_3 + \frac{a_{1,4}}{a_{1,1}} X_4 = \frac{y_1}{a_{1,1}}$$

Definindo:

$$b_{1,j} = \frac{a_{1,j}}{a_{1,1}}$$

$$g_1 = \frac{y_1}{a_{1,1}}$$

O sistema de equações (2.1.) pode ser escrito da seguinte forma equivalente:

$$
\begin{aligned}
X_1 + b_{1,2} X_2 + b_{1,3} X_3 + b_{1,4} X_4 &= g_1 \\
a_{2,1} X_1 + a_{2,2} X_2 + a_{2,3} X_3 + a_{2,4} X_4 &= y_2 \qquad (2.2.) \\
a_{3,1} X_1 + a_{3,2} X_2 + a_{3,3} X_3 + a_{3,4} X_4 &= y_3 \\
a_{4,1} X_1 + a_{4,2} X_2 + a_{4,3} X_3 + a_{4,4} X_4 &= y_4
\end{aligned}
$$

A solução do sistema de equações (2.2.) será a mesma do sistema de equações (2.1.), a menos de erros de arrendondamento.

A etapa seguinte consiste em eliminar a variável X_1 nas segunda, terceira e quarta equações, do sistema de equações(2.2.), tornando os coeficientes de X_1 iguais a zero. Inicialmente, multiplica-se a primeira equação pelo coeficiente de X_1 da segunda equação($a_{2,1} \neq 0$, caso seja zero, X_1 não existe na segunda equação), em seguida subtrai-se o resultado da segunda equação, obtendo-se:

$$
(a_{2,1} - a_{2,1}) X_1 + (a_{2,2} - a_{2,1} b_{1,2}) X_2 + (a_{2,3} - a_{2,1} b_{1,3})
$$

$$
X_3 + (a_{2,4} - a_{2,1} b_{1,4}) X_4 = Y_2 - a_{2,1} g_1
$$

Designando os coeficientes e o termo independente por $a^{(1)}_{2,j}$, $y^{(1)}_2$ respectivamente, tem-se:

$$
a^{(1)}_{2,2} X_2 + a^{(1)}_{2,3} X_3 + a^{(1)}_{2,4} X_4 = Y^{(1)}_2
$$
(vê-se que X_1 foi eliminado)

O procedimento exposto acima deve ser utilizado na terceira e quarta equações, a fim de eliminar X_1 nestas equações.

i) na terceira equação,

$$(a_{3,1} - a_{3,1})\ X_1 + (a_{3,2} - a_{3,1}\ b_{1,2})\ X_2 + (a_{3,3} - a_{3,1}\ b_{1,3})\ X_3 +$$
$$+ (a_{3,4} - a_{3,1}\ b_{1,4})\ X_4 = Y_4 - a_{4,1}\ g_1$$

ou, escrevendo com a nova notação: $a^{(1)}_{4,2}\ X_2 + a^{(1)}_{4,3}\ X_3 + a^{(1)}_{4,4}\ X_4 = y^{(1)}_4$

ii) na quarta equação,

$$(a_{4,1} - a_{4,1})\ X_1 + (a_{4,2} - a_{4,1}\ b_{1,2})\ X_2 + (a_{4,3} - a_{4,1}\ b_{1,3})\ X_3 +$$
$$(a_{4,4} - a_{4,1}\ b_{1,4})\ X_4 = y_4 - a_{4,1}\ g_1$$

ou, escrevendo com a nova notação: $a^{(1)}_{4,2}\ X_2 + a^{(1)}_{4,3}\ X_3 + a^{(1)}_{4,4}\ X_4 = y^{(1)}_4$

Após a eliminação de X_1 na segunda, terceira e quarta equações, do sistema de equações (2.2.), tem-se:

$$\begin{aligned}
X_1 + b_{1,2}\ X_2 + b_{1,3}\ X_3 + b_{1,4}\ X_4 &= g_1 \\
a^{(1)}_{2,2}\ X_2 + a^{(1)}_{2,3}\ X_3 + a^{(1)}_{2,4}\ X_4 &= y^{(1)}_2 \\
a^{(1)}_{3,2}\ X_2 + a^{(1)}_{3,3}\ X_3 + a^{(1)}_{3,4}\ X_4 &= y^{(1)}_3 \\
a^{(1)}_{4,2}\ X_2 + a^{(1)}_{4,3}\ X_3 + a^{(1)}_{4,4}\ X_4 &= y^{(1)}
\end{aligned} \qquad (2.3.)$$

O procedimento seguido para eliminação de X_1 na segunda, terceira e quarta equações do sistema de equações (2.2.), deve ser repetido para eliminar X_2 na terceira e quarta equações, do sistema de equações (2.3.).

A primeira etapa é dividir a segunda equação, do sistema de equações (2.3.), por $a^{(1)}_{2,2} \neq 0$, obtendo-se:

$$X_2 + \frac{a^{(1)}_{2,3}}{a^{(1)}_{2,2}} X_3 + \frac{a^{(1)}_{2,4}}{a^{(1)}_{2,2}} X_4 = \frac{y^{(1)}_2}{a^{(1)}_{2,2}} .$$

Designando-se os coeficientes e o termo independente por $b_{2,j}$ e g_2 respectivamente, o sistema de equações (2.3.), pode ser escrito na seguinte forma:

$$\begin{aligned} X_1 + b_{1,2} X_2 + b_{1,3} X_3 + b_{1,4} X_4 &= g_1 \\ X_2 + b_{2,3} X_3 + b_{2,4} X_4 &= g_2 \\ a^{(1)}_{3,2} X_2 + a^{(1)}_{3,3} X_3 + a^{(1)}_{3,4} X_4 &= y^{(1)} \\ a^{(1)}_{4,2} X_2 + a^{(1)}_{4,3} X_3 + a^{(1)}_{4,4} X_4 &= y^{(1)} \end{aligned} \qquad (2.4.)$$

Multiplicando-se a segunda equação, do sistema de equações (2.4.), pelo coeficiente de X_2 da terceira equação, obtém-se:

$$(a^{(1)}_{3,2} - a^{(1)}_{3,2}) X_2 + (a^{(1)}_{3,3} - a^{(1)}_{3,2} b_{2,3}) X_3 + (a^{(1)}_{3,4} - a^{(1)}_{3,2} b_{2,4}) X_4$$

$$= y^{(1)}_3 - a^{(1)}_{3,2} g_2$$

Denominando-se os novos coeficientes e o termo independente por $a_{3,j}^{(2)}$ e $y_3^{(2)}$ respectivamente, tem-se:

$$a_{3,3}^{(2)} X_3 + a_{3,4}^{(2)} X_4 = y_3^{(2)}, \text{ eliminado } X_2.$$

Analogamente para a quarta equação, tem-se:

$$a_{4,3}^{(2)} X_3 + a_{4,4}^{(2)} X_4 = y_4^{(2)}$$

A eliminação de X_2 na terceira e quarta equações, do sistema de equações (2.4.), transformou-o na seguinte forma:

$$\begin{aligned} X_1 + b_{1,2} X_2 + b_{1,3} X_3 + b_{1,4} X_4 &= g_1 \\ X_2 + b_{2,3} X_3 + b_{2,4} X_4 &= g_2 \\ a_{3,3}^{(2)} X_3 + a_{3,4}^{(2)} X_4 &= y_3^{(2)} \\ a_{4,3}^{(2)} X_3 + a_{4,4}^{(2)} X_4 &= y^{(2)} \end{aligned} \qquad (2.5.)$$

Dividindo a terceira equação, do sistema de equações (2.5.), por $a_{3,3}^{(2)} \neq 0$ e usando-se a notação apresentada anteriormente, tem-se:

$$X_3 + b_{3,4} X_4 = g_3$$

Multiplicando-se a terceira equação, do sistema de equações (2.5.), por $a_{4,3} \neq 0$ e subtraindo-se o resultado da quarta equação, tem-se:

$$a_{4,4}^{(3)} X_4 = y_4^{(3)} \tag{2.6.}$$

Dividindo-se a equação (2.6.) por $a_{4,4}^{(3)} \neq 0$, o sistema de equação (2.5.) assume a forma final,

$$\begin{aligned} X_1 + b_{1,2} X_2 + b_{1,3} X_3 + b_{1,4} X_4 &= g_1 \\ X_2 + b_{2,3} X_3 + b_{2,4} X_4 &= g_2 \\ X_3 + b_{3,4} X_4 &= g_3 \\ X_4 &= g_4 \end{aligned} \tag{2.7.}$$

O sistema de equações (2.7.) é a forma triangular do sistema de equações (2.1.). A sequência de etapas para obtenção da forma triangular é denominada TRIANGULARIZAÇÃO.

A sequência de etapas apresentadas a seguir, denominada SUBSTITUIÇÃO, tem por objetivo o cálculo das variáveis a partir do sistema de equações (2.7.).

O valor de X_4 é obtido diretamente da quarta equação, da sua substituição na terceira equação obtém-se o valor de X_3. Os valores de X_3 e X_4 são substituídos na segunda equação, obtendo-se X_2. Os três valores obtidos são substituídos na primeira equação, obtendo-se X_1.

Na TRIANGULARIZAÇÃO, obtém-se a forma triangular do sistema de equações. Na substituição, obtém-se os valores das va riáveis a partir da forma triangular.

As equações gerais no método de eliminação de Gauss, para a K-ésima etapa são:

i) na triangularização:

$$a_{i,j}^{(k)} = a_{i,j}^{(k-1)} - \frac{a_{i,k}^{(k-1)} \cdot a_{k,j}^{(k-1)}}{a_{k,k}^{(k-1)}} \qquad i,\ j \geqslant k + 1$$

$$y_i^{(k)} = y_i^{(k-1)} - \frac{a_{i,k}^{(k-1)} \cdot y_k^{(k-1)}}{a_{k,k}^{(k-1)}} \qquad i \geqslant k$$

$$b_{k,j} = \frac{a_{k,j}^{(k-1)}}{a_{k,k}^{(k-1)}} \qquad j \geqslant k + 1$$

$$g_k = \frac{y_k^{(k-1)}}{a_{k,k}^{(k-1)}}$$

ii) na substituição:

$$K_i = g_i - \sum_{j=i+1}^{N} b_{i,j} \cdot X_j \qquad i = N, N-1, N-2 \ldots 2.1.$$

2.2. Visualização do método de eliminação de Gauss

Para visualizar o método de eliminação de Gauss |2|, no que ele tem de interesse para sistema de potência, considera-se o diagrama unifilar de um circuito linear, mostrado na figura 2.1. Correntes elétricas I são injetadas nos nós e deseja-se as tensões V, destes nós em relação a um nó escolhido como referência, em geral, o neutro.

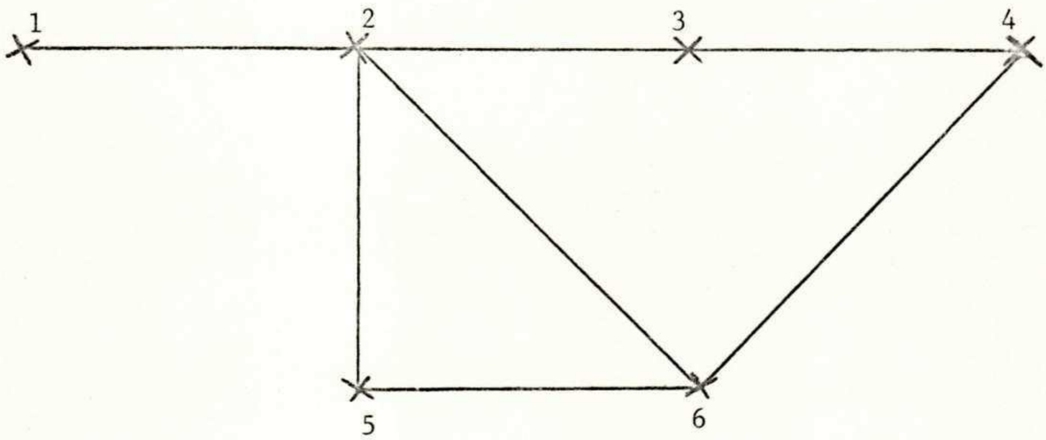

Figura 2.1.

A forma normalizada das seis equações de barra é:

$$
\begin{aligned}
&Y_{1,1} V_1 + Y_{1,2} V_2 &&= I_1 \\
&Y_{2,1} V_1 + Y_{2,2} V_2 + Y_{2,3} V_3 + Y_{2,5} V_5 + Y_{2,6} V_6 &&= I_2 \\
&Y_{3,2} V_2 + Y_{3,3} V_3 + Y_{3,4} V_4 &&= I_3 \quad (2.8.) \\
&Y_{4,3} V_3 + Y_{4,4} V_4 + Y_{4,6} V_6 &&= I_4 \\
&Y_{5,2} V_2 + Y_{5,5} V_5 + Y_{5,6} V_6 &&= I_5 \\
&Y_{6,2} V_2 + Y_{6,4} V_4 + Y_{6,5} V_5 + Y_{6,6} V_6 &&= I_6
\end{aligned}
$$

As admitâncias indicadas com índices repetidos são iguais a soma de todas as admitâncias que terminam na barra identificada pelos índices. As demais admitâncias são iguais a soma com sinal negativo, de todas as admitâncias ligadas diretamente as barras identificadas pelos índices.

Normalizando-se a primeira equação, do sistema de equações (2.8.), e eliminando-se V_1 nas demais equações, obtém-se o sistema de equações (2.9.).

$$\begin{array}{llllll}
Y^{(1)}_{2,2}V_2 + & Y^{(1)}_{2,3}V_3 + & & Y^{(1)}_{2,5}V_5 + & Y^{(1)}_{2,6}V_6 & = I^{(1)}_2 \\
Y^{(1)}_{3,2}V_2 + & Y^{(1)}_{3,3}V_3 + & Y^{(1)}_{3,4}V_4 & & & = I^{(1)}_3 \\
& Y^{(1)}_{4,3}V_3 + & Y^{(1)}_{4,4}V_4 + & & Y^{(1)}_{4,6}V_6 & = I^{(1)}_4 \\
Y^{(1)}_{5,2}V_2 + & & & Y^{(1)}_{5,5}V_5 + & Y^{(1)}_{5,6}V_6 & = I^{(1)}_5 \\
Y^{(1)}_{6,2}V_2 & & Y^{(1)}_{6,4}V_4 + & Y^{(1)}_{6,5}V_5 + & Y^{(1)}_{6,6}V_6 & = I^{(1)}_6
\end{array} \quad (2.9.)$$

Na primeira etapa do processo de eliminação, nenhum novo elemento foi introduzido. O sistema de equações (2.9.) representa o diagrama unifilar da figura 2.2.

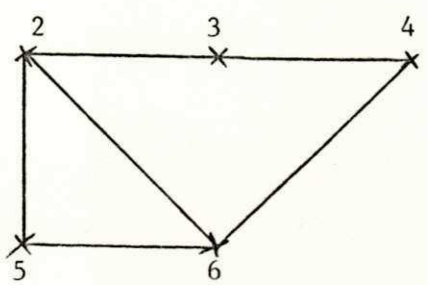

Figura 2.2.

Normalizando-se a primeira equação, do sistema de equação (2.9.), e eliminando-se V_2 nas demais equações, obtém-se:

$$\begin{aligned}
V_2^{(2)} + Y_{2,3}^{(2)} V_3 + \phantom{Y_{3,4}^{(2)} V_4 +} Y_{2,5}^{(2)} V_5 + Y_{2,6}^{(2)} V_6 &= I_2^{(2)} \\
Y_{3,3}^{(2)} V_3 + Y_{3,4}^{(2)} V_4 + Y_{3,5}^{(2)} V_5 + Y_{3,6}^{(2)} V_6 &= I_3^{(2)} \\
Y_{4,3}^{(2)} V_3 + Y_{4,4}^{(2)} V_4 + \phantom{Y_{3,5}^{(2)} V_5 +} Y_{4,6}^{(2)} V_6 &= I_4^{(2)} \qquad (2.10.) \\
Y_{5,3}^{(2)} V_3 + \phantom{Y_{3,4}^{(2)} V_4 +} Y_{5,5}^{(2)} V_5 + Y_{5,6}^{(2)} V_6 &= I_5^{(2)} \\
Y_{6,3}^{(2)} V_3 + Y_{6,4}^{(2)} V_4 + Y_{6,5}^{(2)} V_5 + Y_{6,6}^{(2)} V_6 &= I_6^{(2)}
\end{aligned}$$

Nesta etapa do processo de eliminação, foram introduzidos quatro novos elementos: $Y_{3,5}^{(2)}$, $Y_{3,6}^{(2)}$, $Y_{5,3}^{(2)}$, $Y_{6,3}^{(2)}$. O sistema de equação (2.10.) representa o diagrama unifilar da figura 2.3., onde as linhas tracejadas indicam as novas ligações introduzidas.

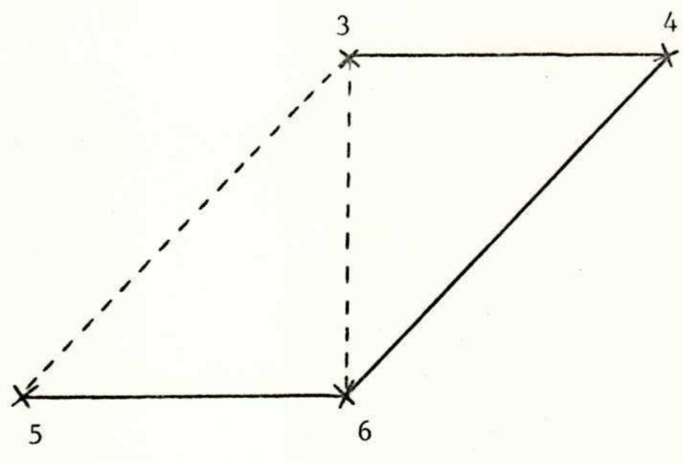

Figura 2.3.

Na terceira etapa do processo de eliminação, normaliza-se a segunda equação, do sistema de equações (2.10.), e elimina-se V_3 nas demais equações, obtendo-se:

$$\begin{aligned}
V_3 + Y^{(3)}_{3,4}\, V_4 + Y^{(3)}_{3,5}\, V_5 + Y^{(3)}_{3,6}\, V_6 &= I^{(3)}_3 \\
Y^{(3)}_{4,4}\, V_4 + Y^{(3)}_{4,5}\, V_5 + Y^{(3)}_{4,6}\, V_6 &= I^{(3)}_4 \\
Y^{(3)}_{5,4}\, V_4 + Y^{(3)}_{5,5}\, V_5 + Y^{(3)}_{5,6}\, V_6 &= I^{(3)}_5 \\
Y^{(3)}_{6,4}\, V_4 + Y^{(3)}_{6,5}\, V_5 + Y^{(3)}_{6,6}\, V_6 &= I^{(3)}_6
\end{aligned} \qquad (2.11.)$$

Nesta etapa do processo de eliminação, foram introduzidos dois novos elementos, $Y^{(3)}_{4,5}$ e $Y^{(3)}_{5,4}$. O sistema de equações (2.11.) representa o diagrama unifilar da figura 2.4., onde a nova ligação introduzida é indicada pela linha tracejada.

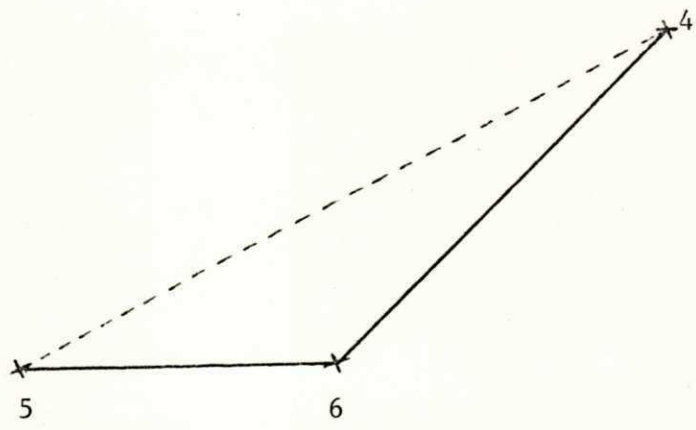

Figura 2.4.

A partir das figuras 2.2, 2.3 e 2.4., observa-se que a eliminação de uma barra i de um diagrama unifilar, consiste no seguinte:

a) Retirar do sistema a barra i, e todas as suas ligações com as outras barras.

b) Sejam K e L duas barras ligadas a barra i. Caso as mesmas não possuam ligações entre si, esta ligação deve ser introduzida.

2.3. Necessidade de uma ordenação

Entende-se por ordenação, a determinação de uma ordem de eliminação, que seguida na triangularização atende objetivos pré-estabelecidos.

A utilização de esquemas de ordenação na solução de sistemas de equações lineares, deve-se a dois aspectos distintos |3|.

O primeiro provém dos erros de arredondamento, gerados na triangularização, os quais podem falsear a solução de sistemas de equações lineares. Neste caso, usa-se esquemas de ordenação capazes de selecionar a ordem de eliminação das variáveis do sistema de equações lineares, visando minimizar o acúmulo de erros numéricos de arredondamentos.

O segundo aspecto da necessidade de ordenar diz respeito a grandes matrizes esparsas diagonalmente dominantes. Neste caso, usa-se esquemas computacionais aonde não se armazena nem opera com os zeros da matriz. Torna-se importante o aparecimento de elementos não nulos em posições antes ocupadas por zero, durante o processo de triangularização. Estes novos elementos serão usados nas operações subsequentes, aumentando o tempo de processamento. O objetivo da ordenação passa a ser a minimização do esforço computacional. No entanto, como é muito difícil traduzir matematicamente este objetivo, cuja solução não é geral, procura-se uma solução sub-ótima minimizando-se o aparecimento de novos elementos. Portanto, o problema de minimizar esforços computacio nais, transforma-se em selecionar a ordem de eliminação das variáveis de um sistema de equações lineares, que seguida na triangularização introduza o menor número possível de novas ligações.

No caso do sistema da figura (2.1.), com a numeração indicada, foram introduzidos três novos elementos no processo de triangularização. Com a numeração mostrada na figura (2.5.) é introduzido apenas um novo elemento no processo de triangularização.

O tempo de execução da solução dos sistemas de equações lineares, inclui o tempo gasto pelo esquema de ordenação. Portanto, o mesmo deve ser compatível com a aplicação, pois não há interesse em obter-se uma ordem de eliminação se o tempo gasto no processamento do esquema de ordenação é elevado, comparado ao tempo gasto na solução com uma ordenação aleatória.

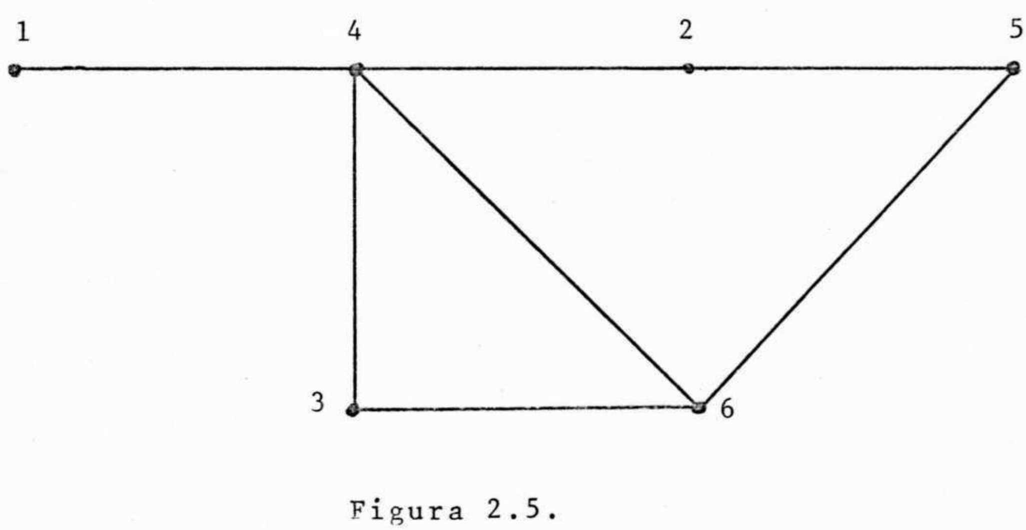

Figura 2.5.

CAPÍTULO III

ESQUEMAS DE ORDENAÇÃO PROPOSTOS POR TINNEY E SASSON

Com o objetivo de minimizar o aparecimento de novos elementos, durante o processo de triangularização, apresenta-se os três esquemas de ordenação propostos por TINNEY |4|, e o proposto por SASSON |5|.

O diagrama unifilar mostrado na figura (3.1.) será utilizado para exemplificar os esquemas de ordenação.

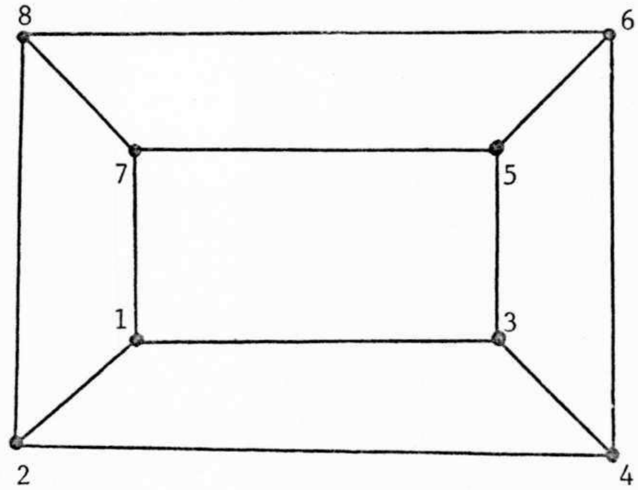

Figura 3.1.

3.1. Esquema - I de TINNEY

"Ordenar as barras do sistema, na ordem crescente do número de ligações que possui cada barra. Caso duas ou mais barras apresentem o mesmo número de ligações a escolha entre elas é arbitrária".

Para o diagrama unifilar da figura (3.1.), a aplicação deste critério pode resultar na numeração das barras mostrada na figura (3.2.). São criados dez novos elementos durante o processo de triangularização.

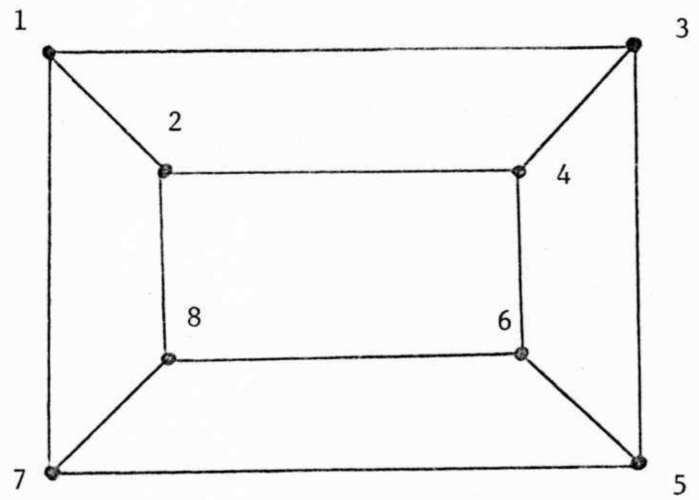

Figura 3.2.

As primeiras barras serão as que possuem apenas uma ligação(se houver). A eliminação destas barras reduz o sistema de uma ligação por barra eliminada, e não introduz nenhuma nova ligação.

Prossegue-se as etapas de eliminações tendo por base o sistema original, sem se levar em consideração as alterações na topologia do sistema durante o processo de eliminação das barras. Apesar disto, as ordens de eliminações selecionadas são em geral melhores do que ordens de eliminações aleatórias.

Os dados necessários para o funcionamento deste esquema de ordenação é a descrição da topologia do sistema, que geralmente é feita através de uma lista do número de barras conectadas a cada barra do sistema

A simplicidade de concepção e o pequeno tempo de execução favorecem a implementação deste esquema de ordenação.

3.2. Esquema - II de TINNEY

"Ordenar as barras do sistema de modo que, a cada etapa da triangularização, a barra a ser eliminada seja a que apresentar o menor número de ligações. Se duas ou mais barras apresentarem o mesmo número de ligações a escolha entre elas é arbitrária".

Renumerando as barras do diagrama unifilar da figura (3.1.) segundo este esquema de ordenação, pode-se obter a numeração mostrada na figura (3.3.). São criados oito novos elementos durante o processo de triangularização.

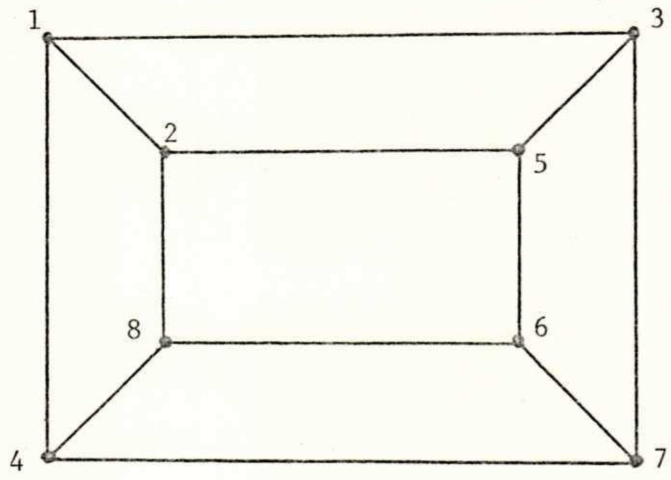

Figura 3.3.

Os dados necessários para a implementação deste esquema de ordenação, são o número de barras conectadas a cada barra do sistema. Este número de barras varia a cada etapa da triangularização, por isso, faz-se necessário a simulação das etapas da triangularização.

3.3. Esquema - III de TINNEY

"Ordenar as barras do sistema de modo que, a cada etapa da triangularização, a barra a ser eliminada seja a que introduzir o menor número de novas ligações na topologia do sistema. Se mais de uma barra for equivalente, a escolha entre elas é arbitrária".

Renumerando as barras do diagrama unifilar mostrado na figura (3.1.) segundo este esquema de ordenação, pode-se obter a numeração mostrada na figura (3.4.). São criados seis novos elementos durante o processo de triangularização.

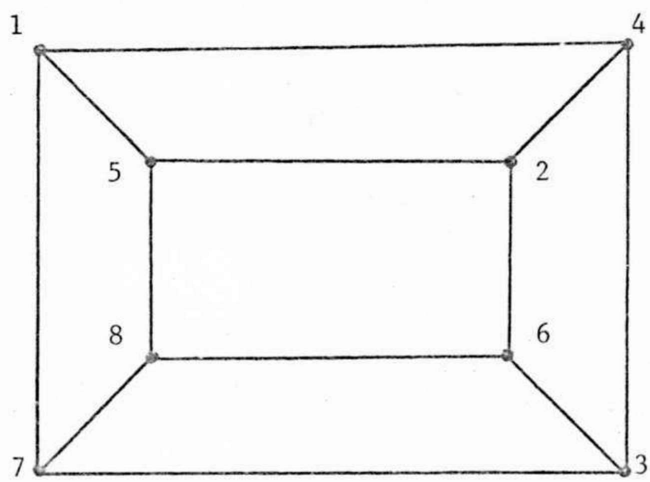

Figura 3.4.

Em cada etapa da triangularização, simula-se a eliminação de todas as barras do sistema, sendo selecionada aquela que introduzir o menor número de novas ligações. Como o esquema-II, este também é um esquema de ordenação dinâmico, ou seja, após a eliminação da barra selecionada a cada etapa, a topologia do sistema deve ser modificada devido a eliminação desta barra.

As vantagens comparativas entre os três esquemas de ordenações apresentados dependem da topologia e do tamanho do sistema. No caso de aplicações de Sistemas de Potência, o esquema-I é muito simples e rápido; o esquema-II é melhor a ponto de justificar o tempo de execução adicional; o esquema-III não parece ser suficientemente melhor do que o esquema-II para justificar o tempo de execução adicional.

3.4. Esquema de SASSON

O esquema de ordenação proposto por SASSON, tem por base a idéia de programação dinâmica. Para descrição do mesmo faz-se necessário enunciar as definições abaixo:

DEFINIÇÃO I:

VALÊNCIA de uma barra pertencente a um dado sistema, é o número de novas ligações que são introduzidas entre as barras remanescentes, caso esta barra seja eliminada.

DEFINIÇÃO II:

VALÊNCIA DA ORDENAÇÃO das barras de um dado sistema, é o número total de novas ligações introduzi das, durante o processo de eliminação das barras, segundo a ordem de eliminação estabelecida por esta ordenação.

DEFINIÇÃO III:

ESTÁGIO, indica a etapa do algorítmo de ordenação.

DEFINIÇÃO IV:

ESTADO, indica as barras que podem ser eliminadas num dado estágio.

O sistema da figura (3.5.) será utilizado para descrição do esquema de ordenação proposto por SASSON.

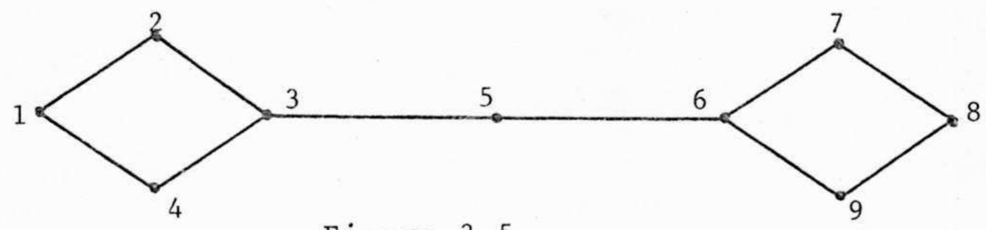

Figura 3.5.

No primeiro estágio calcula-se a valência de cada barra do sistema. A tabela (3.1.) apresenta a valência das barras do sistema da figura (3.5.)

BARRA	1	2	3	4	5	6	7	8	9
VALÊNCIA	1	1	3	1	1	3	1	1	1

Tabela 3.1

No primeiro estado do segundo estágio supõe-se a eliminação da barra-1. Simula-se a eliminação de cada barra do sistema no primeiro estágio e da barra-1 no segundo. A ordem de eliminação que tenha a valência da ordenação mínima é a selecionada. A tabela (3.2.) mostra as ordens de eliminações simuladas e a respectiva valência da ordenação.

ESTÁGIO-I	1º ESTADO DO ESTÁGIO-II (nº da barra)	VALÊNCIA DA ORDENAÇÃO
1	1	-
2	1	1
3	1	3
4	1	1
5	1	2
6	1	4
7	1	2
8	1	2
9	1	2

Tabela 3.2.

As ordens de eliminações (2.1) e (4.1) têm valência mínima. A escolha entre elas é arbitrária. Considera-se a ordem (2.1) como a selecionada.

As demais barras do sistema devem ser consideradas como candidatas a eliminação no segundo estágio. A tabela (3.3.) mostra a ordem de eliminação selecionada para cada estado do segundo estágio e a sua valência da ordenação.

ESTADOS DO ESTÁGIO-II	ORDEM DE ELIMINAÇÃO	VALÊNCIA DA ORDENAÇÃO
1	2.1	1
2	1.2	1
3	1.3	3
4	1.4	1
5	1.5	2
6	8.6	3
7	8.7	1
8	7.9	1
9	8.9	1

Tabela 3.3.

No primeiro estado do terceiro estágio supõe-se a eliminação da barra-1. Simula-se a eliminação da barra-1 como a terceira a ser eliminada nas ordens de eliminações da tabela (3.3). A ordem de eliminação com a valência da ordenação mínima é a selecionada. A tabela (3.4) mostra as ordens de eliminações e as respectivas valências da ordenação destas simulações.

ORDENS DE ELIMINAÇÕES	1º ESTADO DO ESTÁGIO-III	VALÊNCIA DA ORDENAÇÃO
2.1	1	-
1.2	1	-
1.3	1	-
1.4	1	-
1.5	1	-
8.6	1	4
8.7	1	2
7.9	1	2
8.9	1	2

Tabela 3.4.

As ordens de eliminações (8,7,1), (7,9,1) e (8,9,1) têm valência da ordenação mínima. Considere a ordem de eliminação (8,7,1) como a selecionada, no primeiro estado.

A tabela (3.5.) mostra a ordem de eliminação para cada estado do terceiro estágio e a sua valência da ordenação.

ESTADOS DO ESTÁGIO-III	ORDEM DE ELIMINAÇÃO	VALÊNCIA DA ORDENAÇÃO
1	8,7,1	2
2	1,4,2	1
3	1,2,3	2
4	1,2,4	1
5	1,2,5	2
6	7,9,6	2
7	8,9,7	1
8	7,9,8	1
9	8,7,9	1

Tabela 3.5.

O procedimento apresentado até o terceiro estágio é repetido até o nono estágio. A tabela (3.6) mostra as ordens de eliminações de cada estado do quarto estágio e a tabela (3.7.) apresenta os estados do nono estágio.

ESTADO	ORDENS DE ELIMINAÇÃO	VALÊNCIA DA ORDENAÇÃO
1	7 - 9 - 8 - 1	2
2	7 - 8 - 9 - 2	2
3	1 - 4 - 2 - 3	1
4	7 - 9 - 8 - 4	2
5	1 - 2 - 4 - 5	2
6	7 - 9 - 8 - 6	1
7	1 - 4 - 2 - 7	2
8	1 - 4 - 2 - 8	2
9	1 - 4 - 2 - 9	2

Tabela 3.6.

ESTADO	ORDENS DE ELIMINAÇÃO	VALÊNCIA DA ORDENAÇÃO
1	7-8-9-2-5-3-6-4-1	2
2	7-9-8-6-1-4-5-3-2	2
3	7-9-8-6-1-4-2-5-3	2
4	7-9-8-6-1-5-3-2-4	3
5	8-9-7-6-4-2-1-3-5	2
6	1-4-2-3-5-8-9-7-6	2
7	1-4-2-3-5-8-9-6-7	2
8	1-4-2-3-5-7-6-9-8	2
9	1-4-2-3-5-8-7-6-9	2

Tabela 3.7.

Após a execução do nono estágio obteve-se oito ordens de eliminações com a valência da ordenação mínima, igual a dois. A escolha entre elas é arbitrária.

Renumerando o diagrama unifilar mostrado na figura (3.1.) segundo este esquema de ordenação, obtém-se a numeração mostrada na figura (3.6.). São criados seis novos elementos durante o processo da triangularização.

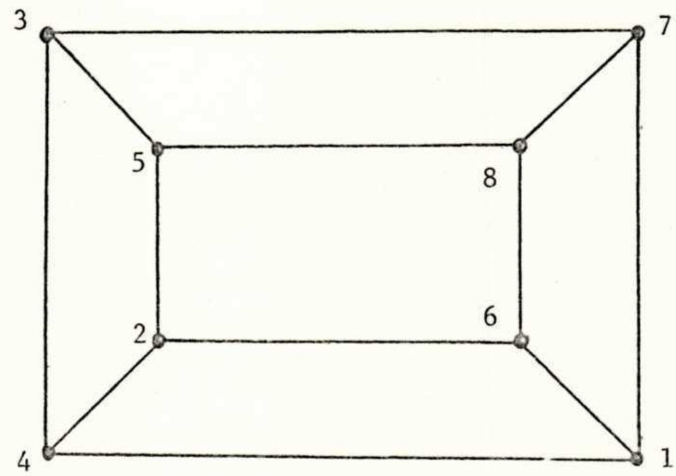

Figura 3.6.

3.5. <u>Exemplos</u>:

Os diagramas unifilares mostrados nas figuras (3.7.) e (3.8.), serão utilizados para exemplificar os esquemas de ordenação apresentados neste capítulo.

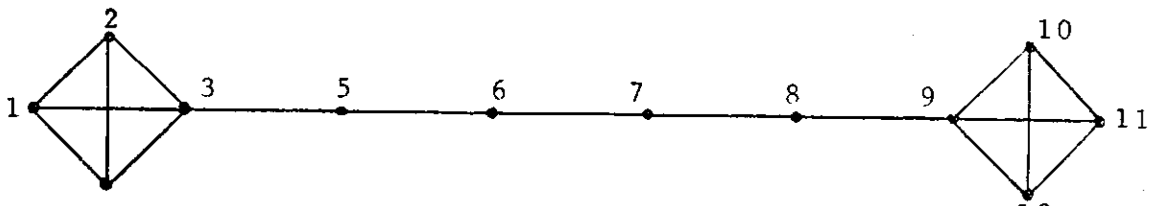

Figura 3.7. - Sistema de 12 barras

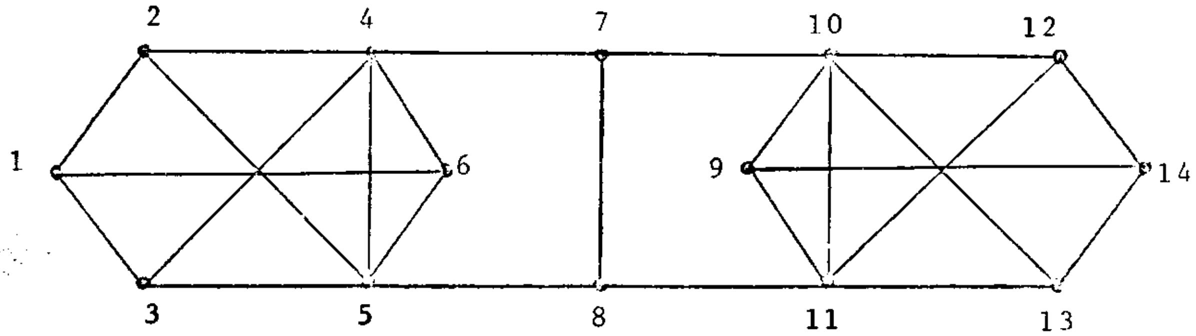

Figura 3.8. - Sistema com 14 barras

<u>Exemplo-1</u>:

A figura (3.7.) apresenta o diagrama unifilar do exemplo-1. Na tabela (3.8.) estão as ordens de eliminações obtidas pelos esquemas de ordenação propostos por TINNEY.

ESQUEMAS DE ORDENAÇÃO	ORDENS DE ELIMINAÇÕES SELECIONADAS	VALÊNCIA DA ORDENAÇÃO
ESQUEMA-I	8-7-6-5-12-11-10-4-2-1-9-3	4
ESQUEMA-II	8-7-6-5-12-11-10-9-3-4-2-1	4
ESQUEMA-III	12-11-10-9-8-7-6-5-3-4-2-1	zero

Tabela (3.8.) - Exemplo-1 - Esquemas de TINNEY

O esquema de SASSON selecionou três ordens de eliminações, com valência nula. As ordens de eliminação são:

2, 1, 10, 4, 3, 11, 5, 12, 6, 7, 8, 9

1, 4, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12

2, 1, 10, 4, 3, 11, 5, 12, 6, 9, 7, 8

Exemplo-2:

A figura (3.7.) apresenta o diagrama unifilar do exemplo-2. Na tabela (3.9.) estão as ordens de eliminação obtidas pelos esquemas de ordenação propostos por TINNEY.

ESQUEMAS DE ORDENAÇÃO	ORDENS DE ELIMINAÇÕES SELECIONADAS	VALÊNCIA DA ORDENAÇÃO
ESQUEMA-I	14-13-12-9-8-7-6-3-2-1-11-10-5-4	11
ESQUEMA-II	14-8-6-3-1-2-4-5-7-11-13-12-10-9	8
ESQUEMA-III	13-14-12-9-11-10-8-7-6-5-4-3-2-1	8

Tabela (3.9.) - Exemplo-2 - Esquemas de TINNEY

O esquema de ordenação proposto por SASSON selecionou a ordem de eliminação apresentada abaixo, com valência seis.

12-9-14-13-11-10-2-3-1-6-5-4-8-7

Os resultados da aplicação do esquema de ordenação proposto por SASSON foram retirados da referência-5. Os resultados da aplicação dos esquemas TINNEY foram obtidos com um programa de simulação digital de sua autoria, anexo à referência-4.

CAPÍTULO IV

ESQUEMA DE ORDENAÇÃO ITERATIVA

Neste capítulo apresenta-se um esquema de ordenação iterativa, a partir do esquema de ordenação Tinney-III, que procura obter uma ordem de eliminação num tempo de processamento razoável.

4.1. - Algorítmo

Um processo de eliminação de n variáveis é visto como um processo iterativo, executado para obter a valência da ordenação de uma ordem de eliminação.

Para descrição do algorítmo deste esquema de ordenação, considere-se um sistema com n barras.

No primeiro estágio da primeira iteração a barra-1 é eliminada. Após a eliminação desta barra atualiza-se o sistema. Os demais estágios da primeira iteração são executados de acordo com o esquema de ordenação Tinney-III, a fim de que seja estabelecida a ordem de eliminação das n barras do sistema.

No primeiro estágio da segunda iteração a barra-2 é eliminada. Após a eliminação desta barra atualiza-se o sistema. Os demais estágios da segunda iteração são executados de acordo com o esquema de ordenação Tinney-III.

O procedimento apresentado para as duas primeiras iterações é seguido até a n-ésima iteração.

A ordem de eliminação selecionada será a que possuir valência da ordenação mínima, dentre as n ordens de eliminações indicadas pelas n iterações. Se duas ou mais iterações indicarem ordens de eliminações com valência da ordenação mínima, a escolha entre elas é arbitrária.

O esquema de ordenação apresentado pode ser descrito pelas seguintes etapas:

(1) Fixar a barra a ser eliminada no primeiro estágio.

(2) Executar os demais estágios de acordo com o esquema de ordenação Tinney-III.

(3) Retornar a etapa(1) até sejam executadas as n iterações.

4.2. - Exemplo demonstrativo

O sistema da figura (4.1.) é utilizado no exemplo demosntrativo do esquema de ordenação iterativa. A numeração das barras deste sistema foi feita de maneira aleatória.

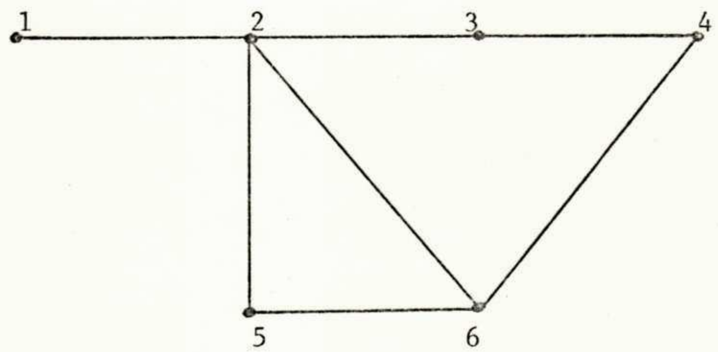

Figura 4.1. - Sistema do exemplo demonstrativo

No primeiro estágio da primeira iteração a barra-1 é eliminada. A eliminação da barra-1 no sistema da figura(4.1.) não introduz nenhum novo elemento. O sistema atualizado após a eliminação da barra-1 é mostrado na figura (4.2.).

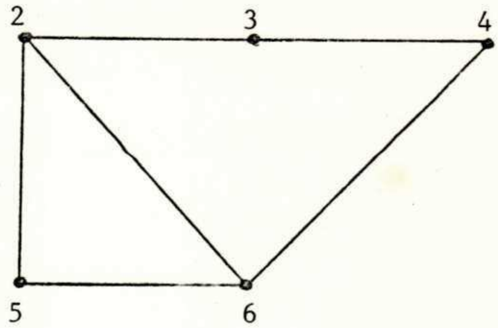

Figura 4.2. - Sistema após o primeiro estágio da primeira iteração

No segundo estágio da primeira iteração, será eliminada a barra de menor valência do sistema da figura (4.2.). A valência de cada barra neste estágio é apresentada na tabela (4.1.).

BARRA	VALÊNCIA DA BARRA
2	2
3	1
4	1
5	0
6	2

Tabela 4.1. - Valências das barras no segundo estágio da primeira iteração

A barra-5 é eliminada no segundo estágio, pois tem valência mínima. A figura (4.3.) mostra o sistema após o segundo estágio.

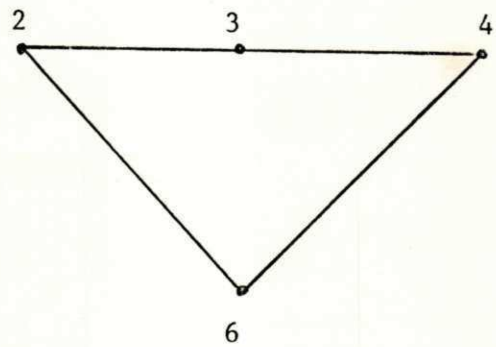

Figura 4.3. - Sistema após o segundo estágio da primeira iteração

No terceiro estágio da primeira iteração será eliminada a barra de menor valência do sistema da figura (4.3.). Todas as barras deste sistema têm valência um. A escolha entre elas é arbitrária. Considere a barra-2 como a eliminada neste estágio. A figura (4.4.) mostra o sistema após o terceiro estágio. A linha tracejada indica o novo elemento introduzido.

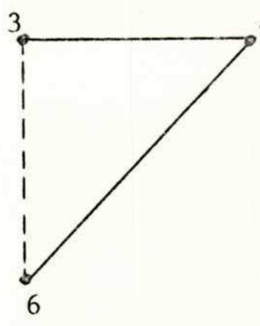

Figura 4.4. - Sistema após o terceiro estágio da primeira iteração

No quarto estágio da primeira iteração será eliminada a barra de menor valência do sistema da figura (4.4.). Todas as barras deste sistema têm valência nula. Considere-se a barra-2 como a eliminada neste estágio. A figura (4.5.) mostra o sistema após o quarto estágio.

Figura 4.5. - Sistema após o quarto estágio da primeira iteração

Como o sistema da figura (4.5.) possui apenas duas barras, a ordem de eliminação pode ser qualquer, sem que nenhum novo elemento seja introduzido. Considere-se a barra-4 como a eliminada no quinto estágio e a barra-6 no sexto estágio da primeira iteração.

Foi executada a primeira iteração do esquema de ordenação iterativa. A tabela (4.2.) mostra a ordem de eliminação obtida. A valência da ordenação é um, e o novo elemento foi introduzido no terceiro estágio.

ESTÁGIO	BARRA ELIMINADA
1	1
2	5
3	2
4	3
5	4
6	6

Tabela 4.2. - Ordem de eliminação da primeira iteração

No primeiro estágio da segunda iteração, a barra-2 é eliminada. A eliminação da barra-2 no sistema da figura (4.1.) introduz cinco novos elementos, (1-3), (1-5), (1-6), (3-5) e (3-6). Como a valência da barra-2 é maior do que a valência da ordenação da primeira iteração, não são executados os demais estágios da segunda iteração. A não execução dos demais estágios desta iteração é denominada de truncamento.

No primeiro estágio da terceira iteração a barra-3 é eliminada. A eliminação da barra-3 no sistema da figura (4.1.) introduz um novo elemento. Os demais estágios desta iteração não são executados pois, a valência da barra-3 é igual a valência da ordenação da primeira iteração.

No primeiro estágio da quarta iteração a barra-4 é a eliminada. A eliminação da barra-4 no sistema da figura (4.1.) introduz um novo elemento. Os demais estágios desta iteração não são executados pois, a valência da barra-4 é igual a valência da ordenação da primeira iteração.

No primeiro estágio da quinta iteração a barra-5 é eliminada. A eliminação da barra-5 no sistema da figura (4.1.) não introduz nenhum novo elemento. A figura (4.6.) mostra o sistema após o primeiro estágio da quinta iteração.

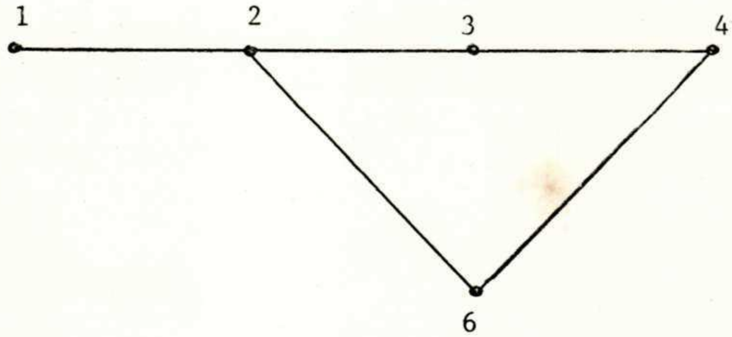

Figura 4.6. - Sistema após o primeiro estágio
da quinta iteração

No segundo estágio da quinta iteração será eliminada a barra de menor valência do sistema da figura (4.6.). A barra-1 é eliminada. A figura (4.7.) mostra o sistema após este estágio.

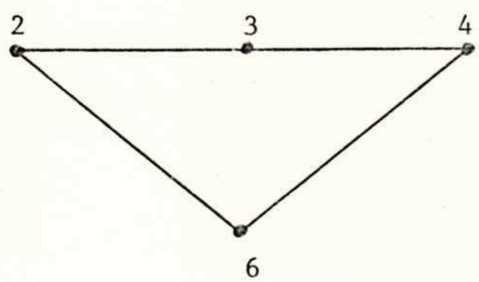

Figura 4.7. - Sistema após o segundo estágio
da quinta iteração

No terceiro estágio da quinta iteração será eliminada a barra de menor valência do sistema da figura (4.7.). Todas as barras possuem valência um. Considere-se a barra-2 como a eliminada neste estágio. Os demais estágios desta iteração não são executados pois, a soma das valências das barras eliminadas no primeiro, segundo e terceiro estágios da quinta iteração, é igual a valência da ordenação da primeira iteração.

No primeiro estágio da sexta iteração a barra-6 é a eliminada. A eliminação da barra-6 no sistema da figura (4.1.) introduz dois novos elementos. Os demais estágios desta iteração não são executados pois, a valência da barra-6 é maior do que a valência da ordenação da primeira iteração.

A ordem de eliminação da primeira iteração é a resultante desejada.

A partir da segunda iteração, todos os estágios de uma iteração só serão executados se nesta iteração for obtida uma ordem de eliminação, com valência da ordenação menor do que as obtidas em todas as iterações já executadas.

4.3. - Aspectos computacionais

Descreve-se neste parágrafo o arranjo computacional do esquema de ordenação iterativa. A figura (4.8.) mostra o diagrama de blocos do esquema de ordenação iterativa.

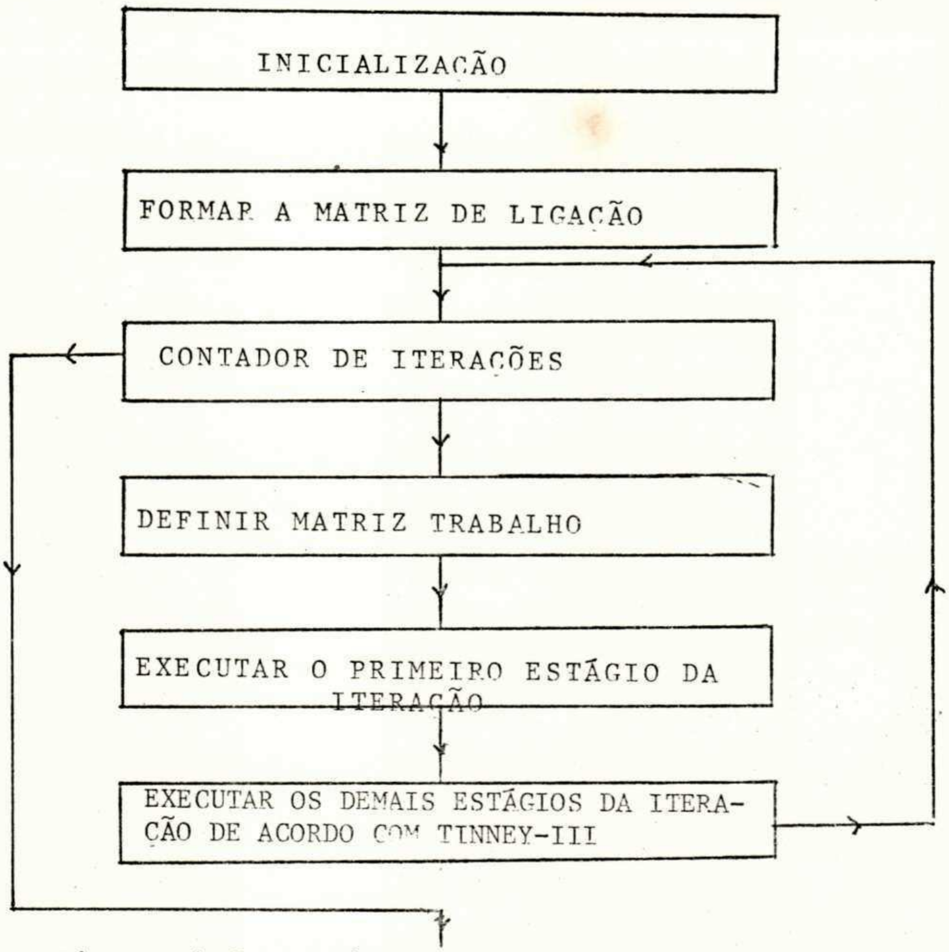

Figura 4.8. - Diagrama de blocos do esquema de ordenação iterativa

O arranjo computacional pode ser descrito pelas seguintes etapas:

1 - Formar uma matriz inteira, capaz de indicar a topologia do sistema original. Denomine-se de MATRIZ DE LIGAÇÃO esta matriz.

2 - Definir uma matriz inteira de dimensão nº de barra x nº de barras, que deve indicar os novos elementos gerados durante a simulação das eliminações das barras. Denomine-se MATRIZ TRABALHO esta matriz.

3 - Executar o primeiro estágio da iteração.

4 - Executar os demais estágios da iteração de acordo com o es-

quema de ordenação Tinney-III.

5 - Retornar a etapa-2 até que sejam executadas todas as iterações.

4.4. - Armazenamento compacto de matrizes

O armazenamento de uma matriz cheia na memória do computador, permite relacionar cada elemento da matriz a partir do número da linha e do número da coluna. Quando a matriz é esparsa, é possível uma economia de memória e tempo de processamento, se apenas os elementos não nulos são armazenados e processados. Para isto, é necessário usar tabelas de apontadores capazes de identificar cada elemento armazenado compactamente. O esquema de armazenamento deve estar relacionado com a forma de processamento usado.

O esquema de armazenamento compacto de matrizes apresentado neste parágrafo, utiliza dois vetores para armazenar os elementos não nulos da matriz na memória do computador. O primeiro vetor armazena os elementos diagonais da matriz, e o segundo, armazena os demais elementos não nulos.

Denomine-se DIAG o vetor que armazena os elementos diagonais. O armazenamento é feito de modo que o local de armazenamento no vetor DIAG corresponde ao índice do elemento diagonal, como mostra a tabela (4.3.).

DIAG(1) = A(1,1)
DIAG(2) = A(2,2)
DIAG(3) = A(3,3)
.
.
.
DIAG(K) = A(K,K)
.
.
.
DIAG(N) = A(N,N)

Tabela 4.3. - Armazenamento compacto dos elementos diagonais

Não há necessidade de apontadores para indicar a posição dos elementos armazenados no vetor DIAG. Sempre que necessário, estes elementos são obtidos diretamente.

O segundo vetor, armazena os elementos não diagonais diferentes de zero. O armazenamento é feito por linha da matriz. Nas primeiras posições são armazenados os elementos da primeira linha, em seguida os elementos da segunda linha, assim sucessivamente, até que por fim são armazenados os elementos da última linha da matriz. Denomine-se Y este vetor.

Para identificar os elementos armazenados no vetor Y, utiliza-se dois vetores apontadores. Denomine-se de ICOL e LINZ estes vetores.

ICOL(K): Indica a coluna do k-ésimo elemento do vetor Y, como mostra a tabela (4.4.). O vetor ICOL tem a mesma dimensão do vetor Y.

$$Y(K) = A(i,j) \Longrightarrow ICOL(K) = j$$

Tabela 4.4. - Formação do vetor ICOL

LINZ(I): Indica a posição, onde inicia o armazenamento dos elementos da i-ésima linha da matriz, no vetor Y.

A matriz esparsa mostrada na tabela (4.5.) é armazenada compactamente, segundo o esquema apresentado. As tabelas (4.6.) e (4.7.) mostram a matriz armazenada compactamente.

$$\begin{array}{cccccc}
A_{1,1} & A_{1,2} & & & & \\
A_{2,1} & A_{2,2} & A_{2,3} & & A_{2,5} & A_{2,6} \\
 & A_{3,2} & A_{3,3} & A_{3,4} & & \\
 & & A_{4,3} & A_{4,4} & & A_{4,6} \\
 & A_{5,2} & & & A_{5,5} & A_{5,6} \\
 & A_{6,2} & & A_{6,4} & A_{6,5} & A_{6,6}
\end{array}$$

Tabela 4.5. - Matriz esparsa

J	DIAG(J)	LINZ(J)
1	A(1.1)	1
2	A(2.2)	2
3	A(3.3)	6
4	A(4.4)	8
5	A(5.5)	10
6	A(6.6)	12

Tabela 4.6. - Armazenamento dos elementos diagonais

K	Y(K)	ICOL(K)
1	A(1.2)	2
2	A(2.1)	1
3	A(2.3)	3
4	A(2.5)	5
5	A(2.6)	6
6	A(3.2)	2
7	A(3.4)	4
8	A(4.3)	3
9	A(4.6)	6
10	A(5.2)	2
11	A(5.6)	6
12	A(6.2)	2
13	A(6.4)	4
14	A(6.5)	5

Tabela 4.7. - Armazenamento dos elementos não diagonais, e os vetores apontadores.

Considerando a matriz dos coeficientes na forma normalizada das equações de barra de um sistema, a existência do elemento a(i,j) indica que há uma ligação entre a i-ésima e a j-ésima barra do sistema. Sabendo-se as colunas dos elementos da i-ésima linha da matriz dos coeficientes, sabe-se as barras ligadas a i-ésima barra do sistema. Portanto, para indicar a topologia de um sistema, utilizando-se o esquema de armazenamento compacto de matrizes apresentado, necessita-se apenas dos vetores apontadores ICOL e LINZ.

A ordem de armazenamento das colunas dos elementos de cada linha da matriz no vetor ICOL, não é importante no processo de simulação da eliminação das barras de um sistema.

O esquema de ordenação apresentado, utiliza na simulação das eliminações das barras de um sistema, duas matrizes esparsas denominadas MATRIZ TRABALHO e MATRIZ DE LIGAÇÃO. A utilização do esquema de armazenamento compacto de matriz apresentado, não deu resultados satisfatórios quando utilizado no esquema de ordenação iterativa, devido a dois fatores: dificuldade de armazenar os novos elementos gerados durante a simulação das eliminações das barras e, a dificuldade em relacionar um elemento de uma matriz com os demais elementos da outra matriz.

Para suprir as dificuldades citadas, faz-se as seguintes modificações no esquema de armazenamento/processamento:

Utiliza-se apenas a MATRIZ TRABALHO nas simulações das eliminações das barras de um sistema, para evitar relacionar um elemento de uma matriz com os demais elementos da outra matriz. Esta modificação é introduzida no esquema de ordenação iterativa, fazendo-se a MATRIZ TRABALHO igual a MATRIZ DE LIGAÇÃO no início de cada iteração, como mostra o diagrama de blocos da figura (4.9.).

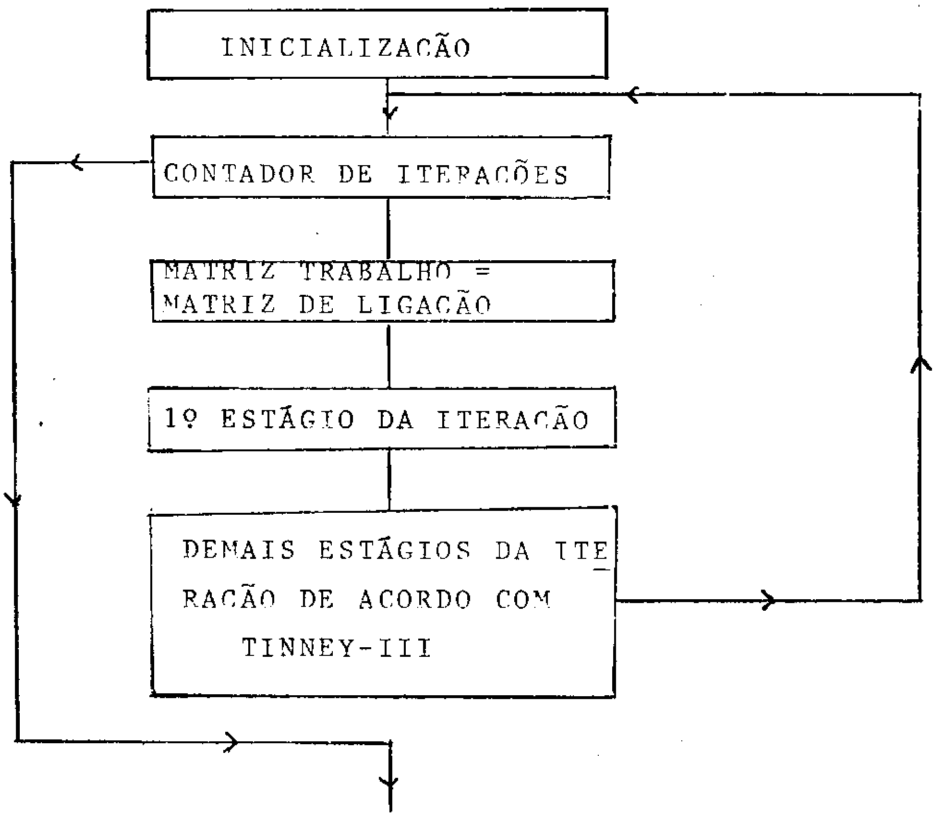

Fig.4.9. - Diagrama de blocos do esquema de ordenação iterativa

A fim de armazenar com facilidade os novos elementos gerados na simulação das eliminações das barras de um sistema, deixa-se, após o armazenamento das colunas dos elementos de cada linha da matriz, no vetor ICOL, algumas posições livres onde serão armazenadas as colunas dos novos elementos gerados em cada linha da matriz, como mostra a figura (4.10.). Devido a esta modificação no esquema de armazenamento compacto de matrizes, há necessidade da definição de um vetor apontador que indique o local onde é armazenado o último elemento de cada linha da matriz no vetor ICOL. Denomine-se IFIM este vetor.

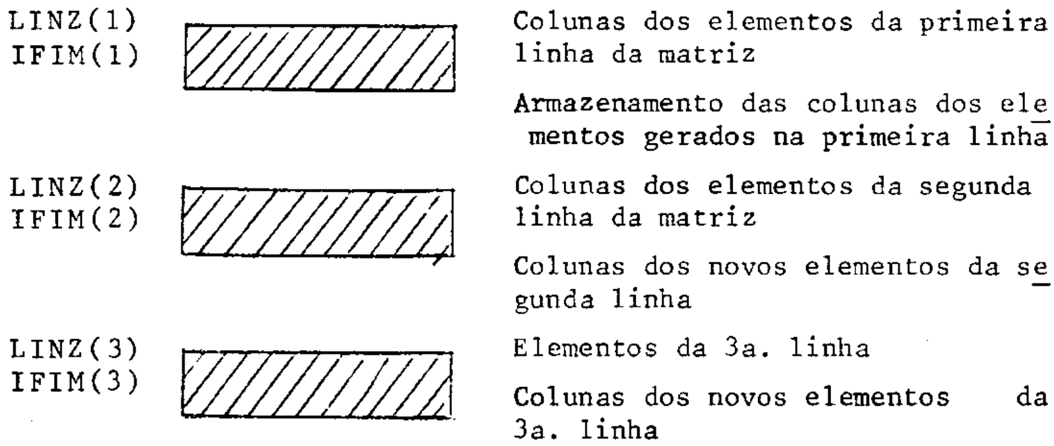

Figura 4.10. - Esquema armazenamento no VETOR ICOL

O aumento de memória, devido as posições livres para armazenar os novos elementos gerados, e a definição de um novo vetor apontador não é importante pois: O vetor IFIM é inteiro tendo dimensão igual ao número de barras do sistema; o espaço livre após o armazenamento das colunas de cada linha da matriz no vetor ICOL, é de, aproximadamente, 10% do número de barras do sistema, para grandes sistemas.

4.5. - Exemplos

Neste parágrafo são apresentados alguns resultados obtidos pelo esquema de ordenação iterativa. Em cada caso, é apresentado a ordem de eliminação resultante e a valência da ordenação.

Exemplo-1: Sistema da figura 4.11. A ordem de eliminação resultante foi 1,4,8,3,5,2,6,7, com valência da ordenação igual a dois.

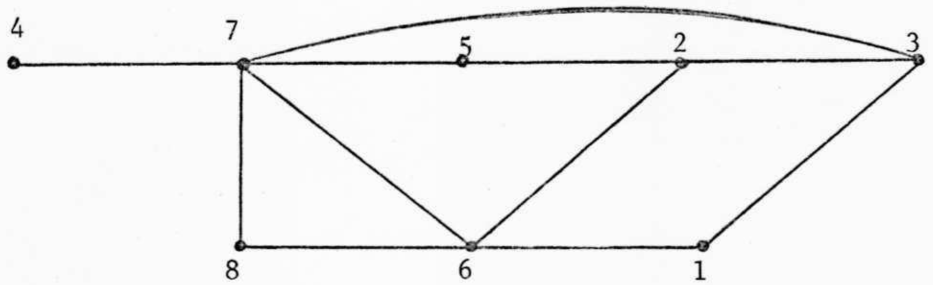

Figura 4.11. - Sistema do exemplo-1

Exemplo-2: Sistema da figura 4.12. A ordem de eliminação resultante foi 7,8,9,6,15,1,2,4,3, 11, 12,13,10,14,16,17 com valência da ordenação igual a três.

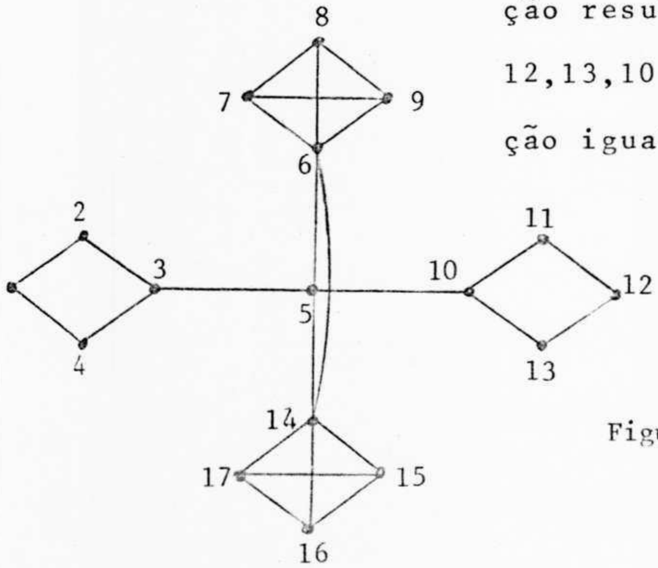

Figura 4.12. - Sistema do exemplo-2

4.6. - Truncamento N/3

A partir dos resultados obtidos com o esquema de ordenação iterativa, pode-se concluir que a iteração de menor valência da ordenação, é indicada executando-se somente os primeiros estágios de cada iteração. As soluções obtidas de 12 sistemas com configurações diferentes, pelo esquema de ordenação iterativa, levaram à conclusão que o número de estágios necessários a cada iteração para indicar a iteração de menor valência de ordenação, é, no máximo, o número de barras do sistema dividido por três. Dessa forma obtém-se a ordem de eliminação de n/3 barras do sistema. Em seguida, a ordem de eliminação das demais barras do sistema é estabelecida de acordo com um dos três esquemas de ordenação propostos por Tinney |3|.

4.7. - Exemplo Demonstrativo do Truncamento N/3

O sistema da figura (4.13.) é utilizado para mostrar truncamento em N/3. A numeração das barras deste sistema foi feita de maneira aleatória.

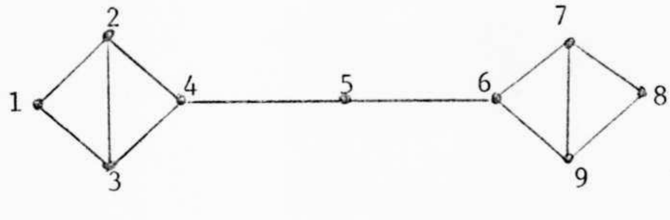

Figura 4.13. - Sistema do exemplo demonstrativo

O sistema da figura (4.13.) tem nove barras. De acordo com o procedimento apresentado, devem ser executados três estágios de cada iteração do esquema de ordenação iterativa. A tabela (4.8) mostra as ordens de eliminações e a valência da ordenação até o 3º estágio, obtida a cada iteração de acordo com o esquema de ordenação iterativa.

ITERAÇÃO	ESTÁGIOS 1º	2º	3º	VALÊNCIA DE ORDENAÇÃO
1	1	2	3	0
2	2	1	3	1
3	3	1	2	1
4	4	1	2	2
5	5	1	2	1
6	6	8	7	2
7	7	8	9	1
8	8	7	9	0
9	9	7	8	1

Tabela 4.8. - Barras eliminadas nos três primeiros estágios e a valência da ordenação até o terceiro estágio.

As ordens de eliminações da primeira e oitava iterações possuem valência de ordenação mínima. A escolha entre elas é arbitrária. Considere-se a primeira iteração como sendo a selecionada.

Utilizando o truncamento em N/3, as demais barras do sistema são ordenadas de acordo com um dos três esquemas de ordenações propostos por Tinney. Denomine-se:

ORDEM-1: quando é utilizado o TINNEY-I
ORDEM-2: quando é utilizado o TINNEY-II
ORDEM-3: quando é utilizado o TINNEY-III

Portanto, a partir do esquema de ordenação ITERATIVA são obtidos três esquemas de ordenações.

Para o sistema da figura(4.13) os esquemas de ordenações ORDEM-1, ORDEM-2 e ORDEM-3, obtêm as seguintes ordens de eliminações:

ORDEM-1: 1, 2, 3, 4, 5, 8, 6, 7, 9
ORDEM-2: 1, 2, 3, 4, 5, 6, 7, 8, 9
ORDEM-3: 1, 2, 3, 4, 5, 6, 7, 8, 9

4.8. - Aspectos computacionais do Truncamento N/3

Inicialmente determina-se a ordem de eliminação de um terço das barras do sistema, de acordo com o esquema de ordenação iterativa, com truncamento.

Determina-se a ordem de eliminação do restante das barras do sistema, de acordo com um dos três esquemas de ordenações propostos por Tinney |3|, que tem como dados iniciais, o sistema atualizado após a eliminação de um terço das barras, de acordo com o esquema de ordenação iterativa.

A seguir alguns exemplos:

Exemplo 1: Sistema de 9 barras da figura 4.14.

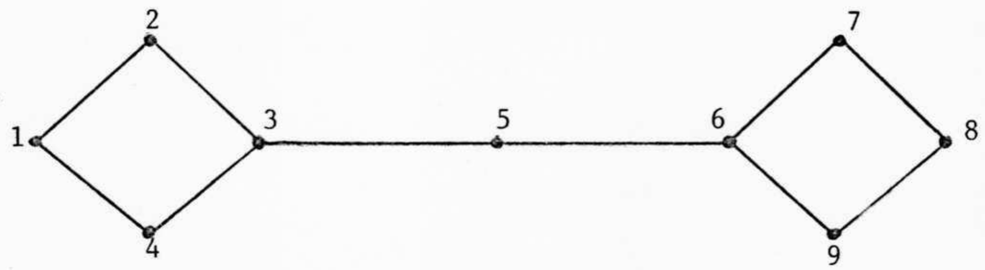

Fig. 4.14. - Sistema de 9 barras

As ordens de eliminações obtidas pelos esquemas de or-denação são:

ORDEM-1: 1, 2, 4, 3, 5, 7, 8, 9, 6

ORDEM-2: 1, 2, 4, 3, 5, 6, 7, 8, 9

ORDEM-3: 1, 2, 4, 3, 5, 6, 7, 8, 9

Exemplo 2: Sistema de 12 barras da figura 4.15.

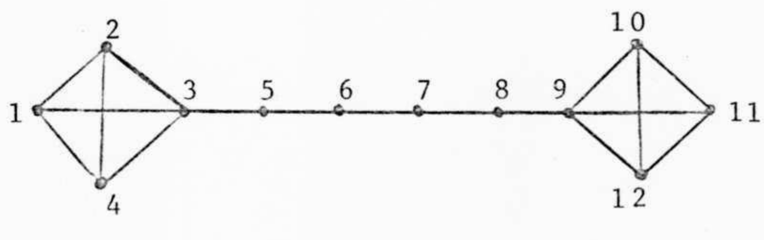

Fig. 4.15. - Sistema de 12 barras

As ordens de eliminações obtidas pelos esquemas de or-denação são:

ORDEM-1: 1, 2, 4, 3, 5, 6, 7, 8, 10, 11, 12, 9

ORDEM-2: 1, 2, 4, 3, 5, 6, 7, 8, 9, 10, 11, 12

ORDEM-3: 1, 2, 4, 3, 5, 6, 7, 8, 9, 10, 11, 12

Exemplo 3: Sistema de 15 barras da figura 4.16.

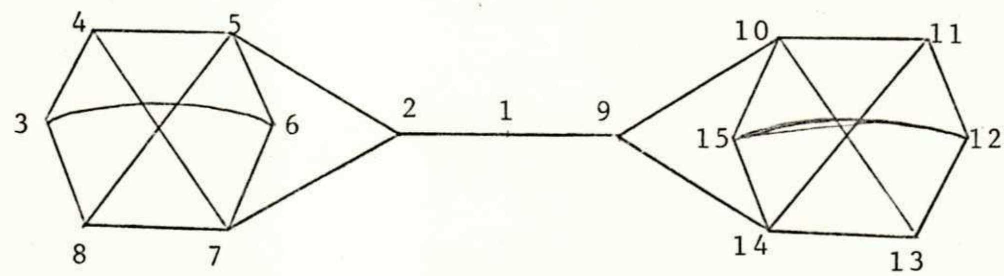

Fig. 4.16. - Sistema de 15 barras

As ordens de eliminações obtidas pelos esquemas de ordenações são:

ORDEM-1: 2,3,5,8,10,9,12,7,11,14,15,16,18,20,21,22,24,1,4,6,13,17,19,23

ORDEM-2: 2,3,5,8,10,9,12,7,11,1,6,14,16,15,18,13,17,4,20,22,19,21,23,24

ORDEM-3: 2,3,5,8,10,9,12,7,11,16,14,16,15,18,13,17,4,20,22,19,21,23,24

EXEMPLOS	ESQUEMAS DE ORDENAÇÕES					
	ORDEM 1 VALÊNCIA	TEMPO	ORDEM 2 VALÊNCIA	TEMPO	ORDEM 3 VALÊNCIA	TEMPO
Sistema de 9 barras	2	0,31s	2	0.32s	2	0.40s
Sistema de 12 barras	0	0.58s	0	0.59s	0	0.73s
Sistema de 15 barras	22	21.01s	17	21.03s	17	22.79s

Tabela 4.9. - Valência da ordenação e o tempo de processamento dos esquemas de ordenações.

A tabela (4.9) apresenta os resultados da valência e tempo de processamento para os três exemplos citados.

CAPÍTULO V

COMPARAÇÕES ENTRE OS ESQUEMAS DE ORDENAÇÕES

Neste capítulo, faz-se comparações entre os esquemas de ordenações apresentados, em termos do tempo de processamento, memória e valência da ordenação.

Os sistemas das figuras (5.1.) a (5.8.) serão utilizados nas comparações dos esquemas de ordenações apresentadas.

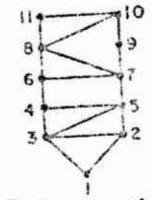

Fig. 5.1. - Sistema do exemplo-1

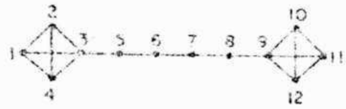

Fig. 5.2. - Sistema do exemplo-2

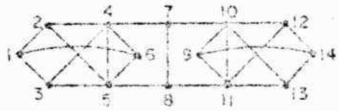

Fig. 5.3. - Sistema do exemplo-3

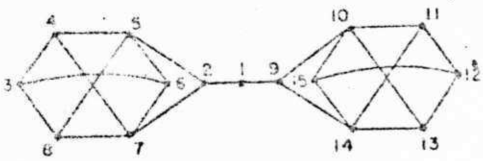

Fig. 5.4. - Sistema do exemplo-4

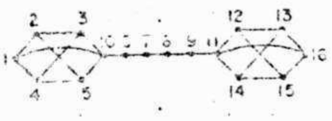

Fig. 5.5. - Sistema do exemplo-5

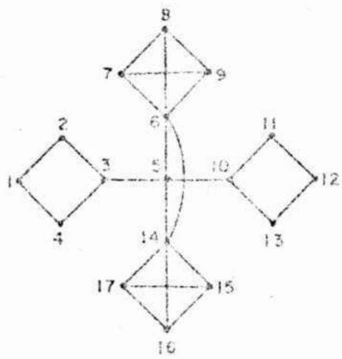

Fig. 5.6. - Sistema do exemplo-6

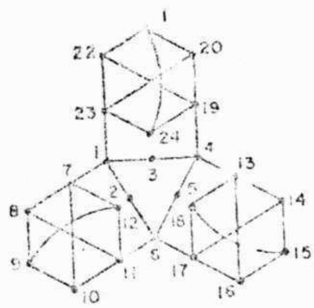

Fig. 5.7. - Sistema do exemplo-7

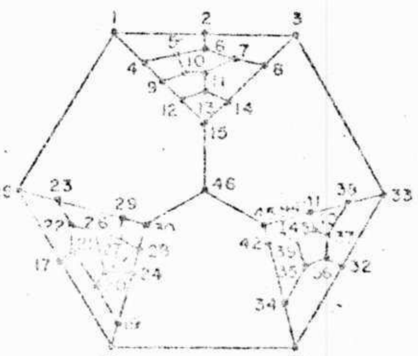

Fig. 5.8. - Sistema do exemplo-8

5.1. - Esquemas de ordenações ORDEM-1, ORDEM-2, ORDEM-3 e ITERAÇÃO

A tabela (5.1.) apresenta resultados da valência da ordenação e do tempo de processamento, para os sistemas das figuras (5.1.) a (5.8.). O tempo de processamento é expresso em segundos.

O esquema de ordenação ORDEM-3, indica uma ordem de eliminação com valência da ordenação menor ou igual, do que as obtidas pelos esquemas de ordenações ORDEM-1 e ORDEM-2.Este fato justifica sua aplicação, apesar do seu tempo de processamento ser um um pouco maior.

A memória requerida pelos esquemas de ordenações ORDEM-1, ORDEM-2 e ORDEM-3 é a mesma, pois eles têm a mesma estrutura de formação.

Os esquemas de ordenações ORDEM-3 e ITERATIVA, indicam ordens de eliminações com mesma valência da ordenação.Ambos são esquemas de ordenações dinâmicos. O tempo de processamento do esquema de ordenação ORDEM-3 é menor do que o do ITERATIVA, como é observado na tabela (5.1.).

A memória utilizada pelo esquema de ordenação ITERATIVA é menor do que a do esquema ORDEM-3. Para o sistema da figura(5.7.) a memória utilizada pelo esquema ITERATIVA é 6396 bytes, enquando o ORDEM-3 utiliza 14828 bytes.

O esquema de ordenação ORDEM-3 apresenta melhores resultados, apesar de utilizar mais memória do que o ITERATIVA.

5.2. - Esquema de ordenação ORDEM-3 e os esquemas Tinney |3|

A tabela (5.2.) apresenta resultados da valência da ordenação e do tempo de processamento, para os sistemas das figuras (5.1.) a (5.8.). O tempo de processamento é expresso em segundos.

EXEMPLO	NÚMERO DE BARRAS	NÚMERO DE LINHAS	ORDEM-1		ORDEM-2		ORDEM-3		ITERATIVA	
			VALÊNCIA	TEMPO	VALÊNCIA	TEMPO	VALÊNCIA	TEMPO	VALÊNCIA	TEMPO
1	11	17	2	0,53	2	0,56	2	0,69	2	1,48
2	12	17	0	0,58	0	0,58	0	0,73	0	0,77
3	14	25	6	1,30	6	1,53	6	1,78	6	7,70
4	15	24	8	2,04	6	2,08	6	2,59	6	8,28
5	16	23	7	1,55	6	1,72	6	2,11	6	8,74
6	17	25	4	1,41	3	1,51	2	1,94	2	11,27
7	24	39	22	21,14	17	22,03	17	23,21	17	62,95
8	46	69	89	81,03	81	86,03	78	89,37	78	637,5

Tabela 5.1. - Resultados da valência da ordenação e tempo de processamento dos esquemas de ordenações ORDEM-1, ORDEM-2, ORDEM-3 e ITERATIVA.

EXEMPLO	NÚMERO DE BARRAS	NÚMERO DE LINHAS	TINNEY-1		TINNEY-2		TINNEY-3		ORDEM-3	
			VALÊNCIA	TEMPO	VALÊNCIA	TEMPO	VALÊNCIA	TEMPO	VALÊNCIA	TEMPO
1	11	17	6	1,80	2	2,23	2	2,24	2	0,69
2	12	17	4	1,78	4	2,06	0	2,08	0	0,73
3	14	25	11	2,30	8	2,71	8	2,85	6	1,78
4	15	24	13	2,25	7	2,64	9	2,82	6	2,59
5	16	23	12	2,06	11	2,74	12	3,01	6	2,11
6	17	25	5	2,53	2	2,76	3	2,68	2	1,94
7	24	39	19	3,35	17	4,06	19	4,40	17	23,21
8	46	69	122	5,77	117	6,62	96	8,40	78	89,37

Tabela 5.2. - Resultados da valência da ordenação e tempo de processamento dos esquemas de ordenações apresentados por Tinney e o ORDEM-3.

O esquema de ordenação ORDEM-3, indica uma ordem de eliminação com valência da ordenação menor, ou igual, do que as obtidas pelos esquemas Tinney.

Para sistemas com no máximo 17 barras, o tempo de processamento do ORDEM-3 é menor do que o dos esquemas Tinney, devido aos truncamentos utilizados pelo ORDEM-3.

Para sistemas com mais de 17 barras, o tempo de processamento dos esquemas Tinney é menor do que o do esquema ORDEM-3, porém a valência da ordenação do ORDEM-3 é menor.

A memória requerida pelo esquema de ordenação ORDEM-3 é menor do que a dos esquemas Tinney.

Para o sistema da figura (5.6.) a memória requerida pelo esquema ORDEM-3 é 14.828 bytes, enquanto para os esquemas Tinney é 87.240 bytes.

5.3. - Esquema de ordenação ORDEM-3 e o apresentado por Sasson |5|

Não se dispõe do tempo de processamento do esquema de ordenação apresentado por Sasson portanto, não é possível compararmos o tempo de processamento.

A tabela (5.3.) apresenta os resultados da valência da ordenação, para os sistemas das figuras (5.1.) a (5.8.).

Exemplo	Número de Barras	Número de Linhas	VALÊNCIA	
			ORDEM-3	SASSON
1	11	17	2	2
2	12	17	0	0
3	14	25	6	6
4	15	24	6	6
5	16	23	6	6
6	17	25	2	2
7	24	39	17	17
8	46	69	78	81

Tabela (5.3.) - Valência da ordenação obtida pelos esquemas ORDEM-3 e SASSON.

Pode ser feito uma análise comparativa do número de operações do esquema de Sasson e o ORDEM-3, para isto considere-se um sistema com n barras. O esquema de Sasson executa no primeiro estágio n(n-1) comparações, no segundo estágio n(n-2) e no estágio (n-1) executa n comparações, portanto são executadas $\sum_{i=1}^{n-1} n(n-i)$ comparações nos n estados. O esquema ORDEM-3 executa no primeiro estágio de cada iteração (n-1) comparações, no segundo (n-2), e no estágio n/3 executa $(n-\frac{n}{3})$ comparações, portanto são executadas $\sum_{i=1}^{n/3} n(n-i)$ comparações. Nos estágios seguintes executa-se $\sum_{i=\frac{n}{3}+1}^{n-3}$ comparações. Assim, o número total de comparações executadas é $\sum_{i=1}^{n/3} n(n-i) + \sum_{i=\frac{n}{3}+1}^{n-3} (n-i)$ nos n estados pelo esquema ORDEM-3.

Caso seja importante selecionar uma ordem de eliminação com valência mínima dentre as obtidas, pode ser utilizado o esquema de ordenação ITERATIVA sem truncamento, ou o esquema apresentado por Sasson. A tabela (5.4.) mostra resultados comparativos, do número de ordenações com valência mínima, para os sistemas das figuras (5.1.) a (5.8.).

Exemplo	Número de Barras	Número de Linhas	NÚMERO DE ORDENAÇÕES	
			ITERATIVA	SASSON
1	11	17	5	4
2	12	17	6	3
3	14	25	6	4
4	15	24	6	4
5	16	23	10	5
6	17	25	12	6
7	24	39	12	5
8	46	69	2	2

Tabela 5.4. - Resultados do número de ordenações com valência mínima, obtidas pelos esquemas ITERATIVA e SASSON

O esquema de ordenação ITERATIVA indica um número maior ou igual, de ordenações com valência mínima, comparado ao esquema apresentado por Sasson, como pode ser observado na tabela (5.4.).

CONCLUSÕES

A escolha de um esquema de ordenação depende do tipo de problema. O tempo de processamento de um esquema de ordenação deve ser compatível com a aplicação. Não há interesse em usar um bom esquema de ordenação, se o seu tempo de processamento é alto, comparado ao tempo gasto na solução do problema com uma ordenação aleatória.

Os esquemas de ordenações ORDEM-2, ORDEM-3, ITERATIVA, SASSON, o esquema-II de Tinney e o esquema-III de Tinney são esquemas de ordenações dinâmicas. Os esquemas de ordenações ORDEM-1 e o esquema-1 de Tinney são esquemas de ordenações estáticas.

Os esquemas de ordenações dinâmicas ORDEM-3, ITERATIVA, SASSON e o esquema-III de Tinney são capazes de indicar a valência da ordenação da ordem de eliminação selecionada e, quais serão os novos elementos gerados na decomposição/triangularização de Gauss.

A ordem de eliminação ótima, que teoricamente é a procurada, não tem aplicação viável, devido ao tempo de processamento necessário.

Na solução de sistemas de equações lineares, normalmente encontrados em estudos de fluxo de cargas, curto-circuito e estabilidade, o esquema de ordenação ORDEM-3 é o que apresenta melhores resultados, se o sistema tem menos de 20 barras. Para sistemas com mais de 20 barras, a escolha de um esquema de ordenação depende da aplicação. Neste caso, o esquema de ordenação ORDEM-3 será vantajoso quando deseja-se obter várias soluções de um sistema de equações lineares, para diferentes vetores independentes, como ocorre em estudos de fluxo de cargas, curto-circuito e estabilidade, quando o número de iterações da solução do sistema de equações lineares é alto, comparado ao tempo gasto na execução do esquema de ordenação.

Quando se trabalha com um único sistema, é vantajoso utilizar o esquema de ordenação ORDEM-3, mesmo se o número de iterações da solução do sistema de equações lineares for baixo. Neste caso, ordena-se o sistema uma única vez, em seguida, renumera-se as barras do sistema segundo a ordenação proposta pelo esquema ORDEM-3, para o estudo a ser feito.

Se o sistema é simétrico, como o mostrado na figura (7.1), os esquemas de ordenações dinâmicos apresentados neste trabalho, indicam ordens de eliminações com mesma valência da ordenação.

Quando é importante selecionar uma ordem de eliminação, dentre as obtidas com valência mínima, deve ser utilizado o esquema de ordenação ITERATIVA sem truncamento.

APÊNDICE

São apresentados os programas dos esquemas de ordenações propostos por Tinney |4|, o do esquema ITERATIVA, e o do ORDEM-3. O programa dos esquemas Tinney são da referência |4|.

1. - Variáveis utilizadas no programa do esquema de ordenação ITERATIVA e, no ORDEM-3.

SB(K)	: Barra de saída,da k-ésima linha do sistema.
EB(K)	: Barra de chegada,da k-ésima linha do sistema.
KCOP(J)	: Coluna do j-ésimo elemento da matriz.
INÍCIO(I)	: Posição onde é armazenado o primeiro elemento, da i-ésima linha da matriz.
IFIM(I)	: Posição onde é armazenada o último elemento,da i-ésima linha da matriz.
NVESO(L)	: Indica a L-ésima barra a ser eliminada.
NVA	: Valência da ordenação

NB : Número de barras do sistema

NL : Número de linhas do sistema

2. - Composição dos programas:

2.1. - Esquema de ordenação ITERATIVA:

O programa deste esquema de ordenação, é composto apenas pelo programa principal.

Os dados de entrada são: NB, NL, e uma tabela contendo a barra de saída e a barra de chegada, de todas as linhas do sistema.

Considere o sistema da figura(7.1.). A tabela (7.1.) mostra os dados de entrada deste sistema.

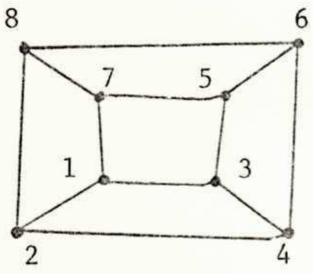

Figura 7.1. - Sistema com 8 barras e 12 linhas.

K	SB(K)	EB(K)
1	1	2
2	1	3
3	1	7
4	2	4
5	2	8
6	3	4
7	3	5
8	4	6
9	5	6
10	5	7
11	6	8
12	7	8

Tabela 7.1. - Dados de entrada do sistema da figura (7.1.).

Para o sistema da figura (7.1.), NB = 8 e NL = 12.

Os dados de saída são: a valência da ordenação selecionada, e uma lista indicando a ordem de eliminação selecionada. Para o sistema a figura (7.1) tem-se:

VALÊNCIA DA ORDENAÇÃO ABAIXO 6

1	1
2	4
3	5
4	3
5	2
6	6
7	7
8	8

2.2. - Esquema de ordenação ORDEM-3

O programa deste esquema de ordenação é composto por um programa principal, que comanda a execução de uma subrotina denominada ORDEM.

O programa principal estabelece a ordem de eliminação de um terço das barras do sistema. A subrotina ORDEM estabelece a ordem de eliminação das demais barras do sistema.

Os dados de entrada e saída deste esquema de ordenação, são os mesmos do esquema de ordenação ITERATIVA.

```
$JOB              ITERATIVA,TIME=9,PAGES=50
      COMMON NT(50),MT(50,50),NSOL(50),NVESO(50),
     *LINZ(200),ICOL(500),SB(99),EB(99)
      INTEGER R,W,SB,EB
      R=5
      W=6
      READ(R,100)NB,NL,NVN
  100 FORMAT(3I10)
      WRITE(W,36)
   36 FORMAT('    NUMERO DE BARRAS - NUMERO DE LINHAS - NVN')
      WRITE(W,110) NB,NL,NVN
  110 FORMAT(I10,I20,I13)
          DO  88 KZA=1,NB
          NSOL(KZA)=0
          LINZ(KZA)=1
   88     NVESO(KZA)=0
      DO  4 KC=1,NL
      READ(R,40) SB(KC),EB(KC)
   40 FORMAT(2I10)
          I=SB(KC)
          IF(I-NB) 1000,1010,1010
 1000     IX=SB(KC)+1
          DO  1020 IA=IX,NB
 1020     LINZ(IA)=LINZ(IA)+1
 1010     CONTINUE
          J=EB(KC)
          IF(J-NB)1030,4,4
 1030     IP=EB(KC)+1
          DO  1040 IC=IP,NB
 1040     LINZ(IC)=LINZ(IC)+1
    4 CONTINUE
          DO  1050 K=1,NL
          I=SB(K)
          J=EB(K)
          K1=LINZ(I)
          K2=LINZ(J)
          ICOL(K1)=J
          ICOL(K2)=I
          LINZ(I)=LINZ(I)+1
          LINZ(J)=LINZ(J)+1
 1050     CONTINUE
      NB1=NB+1
      NB2=NB-1
      DO  1  K=1,NB
      KP=1
      NV=0
      DO  3  KD=1,NB1
      DO  14 KE=1,NB1
      MT(KD,KE)=0
   14 CONTINUE
    3 CONTINUE
      DO  6  KC=2,NB
      MPB1=LINZ(KC-1)
      MPB2=LINZ(KC)-1
        DO  7 KA=MPB1,MPB2
        K1=ICOL(KA)
        MT(KC,K1)=1
    7   CONTINUE
    6  CONTINUE
          MPB1=1
```

```
                MPB2=LIN7(1)-1
                DO 72 KM=MPB1,MPB2
                K1=ICOL(KM)
C------------------------------
C------------------------------
                MT(1,K1)=1
  72      CONTINUE
          DO 1021 KK=1,NB
          M1=MT(K,KK)
          IF(M1) 1021,1021,1110
 1110 MP=KK+1
          DO 1022 KL=MP,NB1
            M2=MT(K,KL)
            IF(M2) 1022,1022,1070
 1070       M3=MT(KK,KL)
            IF(M3) 1080,1080,1022
 1080       NV=NV+1
            MT(KK,KL)=1
            MT(KL,KK)=1
 1022       CONTINUE
 1021 CONTINUE
C------------------------------
          DO 18 MA=1,NB
   18 NSOL(MA)=0
          NSOL(1)=K
          NV1=NV
          NV=0
          IF(NV1-NVN)9,1,1
    9 CONTINUE
  300 KP=KP+1
          KP1=KP-1
          KPA=1
          DO 200 KF=1,NB
          DO 10 LA=1,NB1
   10 NT(LA)=0
          DO 210 KAM=1,KP1
          IS=NSOL(KAM)
          NT(IS)=-1
          IF(IS-KF) 210,200,210
  210 CONTINUE
C    ++++++++++++++++++++
C  TRUNCAMENTO DAS ULTIMAS DUAS BARRAS
C    ++++++++++++++++++++
C    ++++++++++++++++++++
             IF(KP-NB2) 350,360,370
 360         NSOL(KP)=KF
             GO TO 9
 370         NSOL(KP)=KF
             GO TO 131
 350         CONTINUE
C------------------------------
          MT(KF,KF)=0
C-----------------------------------
C        VALENCIA DO ESTAGIO
C-----------------------------------
          DO 1066 KAO=1,NB
          M1=MB(KF,KAO)
          IF(M1) 1066,1066,1067
 1067 M1=NT(KAO)
          IF(M1) 1066,1068,1068
```

```
 1068 L=KAO+1
      DO 1069 KO=L,NB1
      M2=MT(KF,KO)
      IF(M2) 1069,1069,1075
 1075 M2=MT(KO)
      IF(M2) 1061,1071,1071
 1061 GO TO 1069
 1071 M3=MT(KAO,KO)
      IF(M3) 1072,1072,1069
 1072 NV=NV+1
 1069 CONTINUE
 1066 CONTINUE
C-----------------------------
C++++++++++++++++++++++++++++
C    CONTROLE DO PROGRAMA
      NV=NV1+NV
      IF(KPA-1)  401,401,400
  401 NV2=NV
      NSOL(KP)=KF
      KPA=KPA+1
      NV=0
      GO TO 200
  400 CONTINUE
      IF(NV-NV2) 402,403,403
  402 NV2=NV
      NSOL(KP)=KF
  403 NV=0
  200 CONTINUE
      NV1=NV2
      IF(NV1-NVN) 131,1,1
  131 CONTINUE
      IF(KP-NB) 310,380,380
  380 CONTINUE
      WRITE (W,180)NV1
  180 FORMAT(///,5X, 28HVALENCIA DA ORDENACAO ABAIXO,I4)
      DO 130 KJ=1,NB
      WRITE(W,133) KJ,NSOL(KJ)
  133 FORMAT(2I10)
  130 CONTINUE
      NVN=NV1
      GO TO 1
  310 CONTINUE
C ++++++++++++++++++++
      KAP=KP-1
      DO 410 KA5=1,KAP
      IS=NSOL(KA5)
  410 MT(IS)=-1
      KBA=NSOL(KP)
      MT(KBA,KBA)=0
C-----------------------------
C +++    INTRODUCAO DOS NOVOS ELEMENTOS
C-----------------------------
      DO 404 KAA=1,NB
      KB1=MT(KAA)
         IF(KB1) 404,444,444
  444 NC1=MT(KBA,KAA)
      IF(NC1)404,404,405
  405 MT(KAA)=2
  404 CONTINUE
      DO 415 KAF=1,NB
```

```
      N2=NT(KAF)-1
      IF(N2) 415,415,409
  409 CONTINUE
      L=KAF+1
      DO 411 KAG=L,NB1
      N3=NT(KAG)-1
         IF(N3) 411,411,418
  418 MT(KAF,KAG)=1
       MT(KAG,KAF)=1
      MT(KAG,NB1)=0
      MT(KAF,NB1)=0
      MT(KAG,KAG)=0
  411 CONTINUE
  415 CONTINUE
      GO TO 9
    1 CONTINUE
      STOP
      END
```

```
        $ENTRY
 NUMERO DE BARRAS - NUMERO DE LINHAS - NVN
      8                      12            7


   VALENCIA DA ORDENACAO ABAIXO    6
        1          1
        2          4
        3          5
        4          3
        5          2
        6          6
        7          7
        8          8

CORE USAGE        OBJECT CODE=    10496 BYTES,ARRAY AREA=    14192 BYTES,TOTAL AR

DIAGNOSTICS         NUMBER OF ERRORS=        0, NUMBER OF WARNINGS=        0, NUM

COMPILE TIME=     1.62 SEC,EXECUTION TIME=     1.05 SEC,  WATFIV - JUL 1973 V1
```

```
$JOB              ORDEM3,TIME=9,PAGES=20
       COMMON NVESO(50),KCOP(1500),KFIM(50),INICIO(50),
      *NVN,R,W,LNB,NB,NB1,NB2
       DIMENSION NT(50),NSOL(50),LINZ(50),SB(99),EB(99),
      *ICOL(200),IFIM(50),ICOP(3000)
       INTEGER R,W,SB,EB
       R=5
       W=6
       READ(R,100) NB,NL,NVN
  100  FORMAT(3I10)
       WRITE(W,36)
   36  FORMAT('    NUMERO DE BARRAS - NUMERO DE LINHAS - NVN')
       WRITE(W,110)NB,NL,NVN
  110  FORMAT(I10,I20,I13)
       DO 88 KZA=1,NB
       NSOL(KZA)=0
       NVESO(KZA)=0
   88  LINZ(KZA)=1
       DO 4 KC=1,NL
       READ(R,40) SB(KC),EB(KC)
   40  FORMAT(2I10)
       I=SB(KC)
       IF(I-NB) 1000,1010,1010
 1000  IX=I+1
       DO 1020 IA=IX,NB
 1020  LINZ(IA)=LINZ(IA)+1
 1010  CONTINUE
       J=EB(KC)
       IF(J-NB) 1030,4,4
 1030  IP=J+1
       DO 1040 IC=IP,NB
 1040  LINZ(IC)=LINZ(IC)+1
 4     CONTINUE
       DO 1050 K=1,NL
       I=SB(K)
       J=EB(K)
       K1=LINZ(I)
       K2=LINZ(J)
       ICOL(K1)=J
       ICOL(K2)=I
       LINZ(I)=LINZ(I)+1
       LINZ(J)=LINZ(J)+1
 1050 CONTINUE
       NB1=NB+1
       NB2=NB-1
       KNB=NB
       INICIO(1)=1
       DO 30 KZ=1,NB2
       KZ1=KZ+1
       INICIO(KZ1)=LINZ(KZ)+KNB*KZ
   30  CONTINUE
C+++++++++++++++++++++++++++++
       LNB=NB/3
C+++++++++++++++++++++++++++++++++++++++
       DO 1 K=1,NB
       KP=1
       NV=0
       KMO=INICIO(1)
       MPB1=1
       MPB2=LINZ(1)-1
```

```
      DO 66 KO=MPB1,MPB2
      ICOP(KMO)=ICOL(KO)
      KMO=KMO+1
   66 CONTINUE
      IFIM(1)=KMO-1
      DO 361 II=2,NB
      MPB1=LINZ(II-1)
      MPB2=LINZ(II)-1
      KMO=INICIO(II)
      DO 39 IL=MPB1,MPB2
      ICOP(KMO)=ICOL(IL)
      KMO=KMO+1
   39 CONTINUE
      IFIM(II)=KMO-1
  361 CONTINUE
C-------------------------------
C+++++   VALENCIA NO PRIMEIRA ESTADO
C-------------------------------
      MPB1=INICIO(K)
      MPB2=IFIM(K)
 1510 CONTINUE
       KL=MPB1
      L1=ICOP(KL)
      MP1=INICIO(L1)
      MP2=IFIM(L1)
      KK=MPB1
      IF(KK-MPB2)555,2023,2023
  555 KK=KK+1
      K1=ICOP(KK)
      DO 2022 LM=MP1,MP2
       K2=ICOP(LM)
      IF(K2-K1)2022,72,2022
   72 GO TO 2063
 2022 CONTINUE
      NV=NV+1
      K10=ICOP(KL)
      K11=ICOP(KK)
      IFIM(K10)=IFIM(K10)+1
      IFIM(K11)=IFIM(K11)+1
      KA1=IFIM(K10)
      KA2=IFIM(K11)
      ICOP(KA1)=K11
      ICOP(KA2)=K10
      MPB2=IFIM(K)
      MP2=IFIM(L1)
 2063 CONTINUE
      IF(KK-MPB2)555,2023,2023
 2023 CONTINUE
      IF(MPB1-MPB2)1530,1520,1520
 1530 MPB1=MPB1+1
      GO TO 1510
 1520 CONTINUE
C-------------------------------
      DO 18 MA=1,NB
   18 NSOL(MA)=0
      NSOL(1)=K
      NV1=NV
      NV=0
       IF(NV1-NVN)9,1,1
    9 CONTINUE
```

```
  200 KP=KP+1
      KP1=KP-1
       KPA=1
       DO  200 KF=1,NB
      DO 10 LA=1,NB1
   10 NT(LA)=0
      DO 210 KAM=1,KP1
      IS=NSOL(KAM)
      NT(IS)=-1
      IF(IS-KF)210,200,210
  210 CONTINUE
C++++++++++++++++++++++++++++++
C +++++  VALENCIA DO ESTADO
C++++++++++++++++++++++++++++++
      MPB1=INICIO(KF)
      MPB2=IFIM(KF)
      DO 5040 KAA=MPB1,MPB2
      K1=ICOP(KAA)
      KA1=NT(K1)
      IF(KA1)5030,5045,5045
 5045 CONTINUE
      MP1=INICIO(K1)
      MP2=IFIM(K1)
      KA=KAA
      DO 5080 KAB=KA,MPB2
      IF(KA-KAB)5120,5080,5120
 5120 CONTINUE
      K1=ICOP(KAB)
      KA2=NT(K1)
      IF(KA2)5080,5090,5090
 5090 CONTINUE
      DO 5100 LM=MP1,MP2
      K2=ICOP(LM)
      KA2=NT(K2)
      IF(KA2) 5100,5110,5110
 5110 CONTINUE
      IF(K2-K1) 5100,720,5100
  720 GO TO 5080
 5100 CONTINUE
      NV=NV+1
 5080 CONTINUE
 5030 CONTINUE
 5040 CONTINUE
C----------------------------------------
C+ CONTROLE DO PROGRAMA
C----------------------------------------
      NV=NV1+NV
      IF(KPA-1)401,401,400
  401 NV2=NV
      NSOL(KP)=KF
      KPA=KPA+1
      NV=0
      GO TO 300
  400 CONTINUE
      IF(NV-NV2)402,403,403
  402 NV2=NV
      NSOL(KP)=KF
  403 NV=0
  200 CONTINUE
      NV1=NV2
```

```
C-----------------------------------------
        IF(MVI-NVN) 131,1,1
C-----------------------------------------
  131 CONTINUE
C----------------------------
C+++ INTRODUCAO DOS NOVOS ELEMENTOS
C----------------------------
      KAP=KP-1
      DO 410 KA5=1,KAP
      IS=NSOL(KA5)
  410 NT(IS)=-1
      KBA=NSOL(KP)
      MPB1=INICIO(KBA)
      MPB2=IFIM(KBA)
 1560 CONTINUE
      KL=MPB1
      L1=ICOP(KL)
      KA1=NT(L1)
      IF(KA1)6030,6040,6040
 6040 CONTINUE
      MP1=INICIO(L1)
      MP2=IFIM(L1)
      KK=MPB1
      IF(KK-MPB2)6120,6030,6030
 6120 KK=KK+1
      K1=ICOP(KK)
      KA2=NT(K1)
      IF(KA2)6080,6090,6090
 6090 CONTINUE
      DO 6100 LM=MP1,MP2
      K2=ICOP(LM)
      KA2=NT(K2)
      IF(KA2)6100,6110,6110
 6110 CONTINUE
      IF(K2-K1)6100,722,6100
  722 GO TO 6080
 6100 CONTINUE
      K10=ICOP(KL)
      K11=ICOP(KK)
      IFIM(K10)=IFIM(K10)+1
      IFIM(K11)=IFIM(K11)+1
      KA1=IFIM(K10)
      KA2=IFIM(K11)
      ICOP(KA1)=K11
      ICOP(KA2)=K10
      MPB2=IFIM(KBA)
      MP2=IFIM(L1)
 6080 CONTINUE
      IF(KK-MPB2)6120,6030,6030
 6030 CONTINUE
      IF(MPB1-MPB2)1580,1570,1570
 1580 MPB1=MPB1+1
      GO TO 1560
 1570 CONTINUE
C-----------------------------------------
      IF(KP-LNB)310,380,380
  380 CONTINUE
      DO 130 KJ=1,LNB
       NVESO(KJ)=NSOL(KJ)
  130 CONTINUE
```

```
          DO 11 II=1,NB
          K1=INICIO(II)
          K2=IFIM(II)
          DO 12 IJ=K1,K2
          KCOP(IJ)=ICOP(IJ)
   12     CONTINUE
   11     CONTINUE
          DO 25 IME=1,NB
          KFIM(IME)=IFIM(IME)
   25     CONTINUE
         NVN=NV1
         GO TO 1
  310    CONTINUE
         GO TO 9
  1      CONTINUE
C++++++++++++++++++++++++++++++++
          CALL ORDE3
C ++++++++++++++++++++++++++++++++++++
          WRITE(W,180) NVN
   180    FORMAT(///,5X,28HVALENCIA DA ORDENACAO ABAIXO,I4)
          DO 132 KJ=1,NB
  133     FORMAT(2I10)
          WRITE(W,133) KJ,NVESO(KJ)
   132    CONTINUE
          STOP
          END

          SUBROUTINE ORDE3
          COMMON NVESO(50),KCOP(1500),KFIM(50),INICIO(50),
       *NVN,R,W,LNB,NB,NB1,NB2
          DIMENSION NT(50)
          INTEGER R,W
          NV=0
          KP=LNB
   9      CONTINUE
  300     KP=KP+1
         KP1=KP-1
         KPA=1
          DO  200 KF=1,NB
          DO 10 LA=1,NB1
          NT(LA)=0
   10     CONTINUE
          DO 210 KAM=1,KP1
         IS=NVESO(KAM)
         NT(IS)=-1
          IF(IS-KF) 210,200,210
  210     CONTINUE
C++++++++++++++++++++++++++++
C----TRUNCAMENTO DAS DUAS ULTIMAS BARRAS
C++++++++++++++++++++++++++++
          IF(KP-NB2) 350,360,370
  360     NVESO(KP)=KF
          GO TO 9
  370     NVESO(KP)=KF
          GO TO 380
  350     CONTINUE
C-------------------------------------
C---VALENCIA DO PASSO
C++++++++++++++++++++++++++++++++++++ +
         MPB1=INICIO(KF)
```

```
      MPB2=KFIM(KF)
      DO 5040 KAA=MPB1,MPB2
      K1=KCOP(KAA)
      KA1=NT(K1)
      IF(KA1) 5030,5045,5045
 5045 CONTINUE
      MP1=INICIO(K1)
      MP2=KFIM(K1)
      KA=KAA
      DO 5080 KAB=KA,MPB2
      IF(KA-KAB) 5120,5080,5120
 5120 CONTINUE
      K1=KCOP(KAB)
      KA2=NT(K1)
      IF(KA2) 5080,5090,5090
 5090 CONTINUE
      DO 5100 LM=MP1,MP2
      K2=KCOP(LM)
      KA2=NT(K2)
      IF(KA2) 5100,5110,5110
 5110 CONTINUE
      IF(K2-K1) 5100,720,5100
 720  GO TO 5080
 5100 CONTINUE
      NV=NV+1
 5080 CONTINUE
 5030 CONTINUE
 5040 CONTINUE
C----------------------------
C++++CONTROLE DO PROGRAMA
C----------------------------
      NV=NVM+NV
      IF(KPA-1) 401,401,400
 401  NV2=NV
      NVESO(KP)=KF
      KPA=KPA+1
      NV=0
      GO TO 200
 400  CONTINUE
      IF(NV-NV2) 402,403,403
 402  NV2=NV
      NVESO(KP)=KF
 403  NV=0
 200  CONTINUE
      NVM=NV2
      IF(KP-NB) 310,380,380
 310  CONTINUE
      KAP=KP-1
      DO  410 KA5=1,KAP
      IS=NVESO(KA5)
 410  NT(IS)=-1
      KBA=NVESO(KP)
C-----------------------------------
C++++ INTRODUCAO DOS NOVOS ELEMENTOS
C--------------------------------- --
      MPB1=INICIO(KBA)
      MPB2=KFIM(KBA)
 1560 CONTINUE
      KL=MPB1
      N1=KCOP(KL)
```

```
      KA1=NT(N1)
      IF(KA1) 6030,6040,6040
 6040 CONTINUE
      MP1=INICIO(N1)
      MP2=KFIM(N1)
      KK=MPB1
      IF(KK-MPB2) 6120,6030,6030
 6120 KK=KK+1
      K1=KCOP(KK)
      KA2=NT(K1)
      IF(KA2) 6080,6090,6090
 6090 CONTINUE
      DO  6100 LM=MP1,MP2
      K2=KCOP(LM)
      KA2=NT(K2)
      IF(KA2) 6100,6110,6110
 6110 CONTINUE
      IF(K2-K1) 6100,722,6100
 722  GO TO  6080
 6100 CONTINUE
      K10=KCOP(KL)
      K11=KCOP(KK)
      KFIM(K10)=KFIM(K10)+1
      KFIM(K11)=KFIM(K11)+1
      KA1=KFIM(K10)
      KA2=KFIM(K11)
      KCOP(KA1)=K11
      KCOP(KA2)=K10
      MPB2=KFIM(K3A)
      MP2=KFIM(N1)
 6080 CONTINUE
      IF(KK-MPB2) 6120,6030,6030
 6030 CONTINUE
      IF(MPB1-MPB2) 1580,1570,1570
 1580 MPB1=MPB1+1
      GO TO 1560
 1570 GO TO 9
  330 CONTINUE
      RETURN
      END

    $ENTRY
 NUMERO DE BARRAS - NUMERO DE LINHAS - NVN
      9                    10            9


   VALENCIA DA ORDENACAO ABAIXO    2
        1          1
        2          2
        3          4
        4          3
        5          5
        6          6
        7          7
        8          8
        9          9

CORE USAGE        OBJECT CODE=    15440 BYTES,ARRAY AREA=    21220 BYTES,TOTAL A
```

```
$JOB              TINNEY,TIME=9,PAGES=50
$NOEXT
         INTEGER R,W
         COMMON NTOT,NEXT,KOWNT(1500),LOC(1500),KOLUM(5000),KORDER(5000),
        1NORDER(1500),X1,INDEX(99),ICH1(1500),ICH2(1500),LAST,ISCHM,
        2IOFFD,TD,JBRR,R,W,NNTOT
         R=5
         W=6
         READ(5,100) NTOT,NL,NVN
   100   FORMAT(3I10)
         JBR=2*NL
            DO 135 IK=1,JBR
   135      KOLUM(IK)=0
         NNTOT=NTOT+1
         JBRR=JBR+1
         NEXT=JBRR+1
         K=2
         KOW=1
         LOC(2)=2
         DO 106 I=2,JBRR
         READ(R,102)M,KOLUM(I)
         WRITE(W,102)M,KOLUM(I)
   102   FORMAT(2I10)
         N=M+1
         IF(N.EQ.K)GO TO 104
         KOWNT(K)=KOW
         KOW=1
         K=K+1
         KORDER(I-1)=1
         LOC(K)=I
    104 KOW=KOW+1
    106 KORDER(I)=I+1
         KOWNT(NNTOT)=KOW
         KORDER(JBRR)=1
         DO 108 I=JBRR,4999
    108 KORDER(I+1)=I+2
         KORDER(5000)=1
C -----------------------------
         ISCHM=2
C -----------------------------
         IOFFD=1
         CALL NUMBER
         WRITE(W,132)NTOT,IOFFD-1
    132 FORMAT(1X,I6,I6)
         WRITE(W,134) (K-1,NORDER(K)-1,K=2,NNTOT)
    134 FORMAT(1X,2I6)
         STOP
         END

         SUBROUTINE NUMBER
         INTEGER R,W
         COMMON NTOT,NEXT,KOWNT(1500),LOC(1500),KOLUM(5000),KORDER(5000),
        1NORDER(1500),X1,INDEX(99),ICH1(1500),ICH2(1500),LAST,ISCHM,
        2IOFFD,TD,JBRR,R,W,NNTOT
         TD=0.0
         LAST=5000
    110 DO 120 JI=2,99
         I=JI-1
    120 INDEX(I)=I
   130   NTOTT=NNTOT
```

```
  140 I=1
 3000 I=I+1
      IF(I.GT.NNTOT) GO TO 3100
      IF(ISCHM.EQ.3) GO TO 1510
  150 J=KOUNT(I)
      IF(J.EQ.1) GO TO 170
  155 K=INDEX(J+1)
      ICH1(I)=K
      ICH2(K)=I
      INDEX(J+1)=I
      ICH2(I)=1
      GO TO 130
  170 NORDER(I)=NTOTT
      NTOTT=NTOTT-1
  180 GO TO 3000
 3100 CONTINUE
  190 X1=.0
      X2=.0
      X3=.0
      NCN=1
      NELIM=2
  200 CONTINUE
      IF(INDEX(NCN+1).NE.1) GO TO 220
  210 NCN=NCN+1
      GO TO 200
  220 I=INDEX(NCN+1)
      IF(NELIM.LE.6)CALL DUMP
      WRITE(W,221)I-1
  221 FORMAT(I23)
  222 J=ICH1(I)
      INDEX(NCN+1)=J
      ICH2(J)=1
      NORDER(I)=NELIM
  230 IF(NTOTT.GT.NELIM)GO TO 240
  234 RETURN
  240 NELIM=NELIM+1
      IF(ISCHM.EQ.1)GO TO 200
  250 IF(LOC(I).EQ.1) GO TO 200
  260 IST=LOC(I)
      JST=IST
      GO TO 290
  270 JST=KORDER(JST)
  280 IF(JST.NE.1) GO TO 290
      IF(ISCHM.EQ.3) GO TO 1110
      GO TO 200
  290 IB=IST
      IOFFD=IOFFD+1
      J=KOLUM(JST)+1
      JB=LOC(J)
      ICOM=0
      JBS=1
  300 IF(JB.EQ.1) GO TO 200
      WRITE(W,291)J-1
  291 FORMAT(I26)
  310 IF(KOLUM(IB)-KOLUM(JB)) 360,320,500
  320 IF(KORDER(IB).EQ.1) GO TO 430
  330 IB=KORDER(IB)
  340 IF(KORDER(JB).EQ.1) GO TO 700
  350 JBS=JB
      JB=KORDER(JB)
```

```
      GO TO 310
  360 IF(KOLUM(IB)+1.EQ.J) GO TO 410
  370 KOLUM(NEXT)=KOLUM(IB)
      MEXT=NEXT
      IF(NEXT.EQ.LAST) GO TO 375
      NEXT=KORDER(NEXT)
      KORDER(MEXT)=JB
      JBT=JBS
      JBS=MEXT
      ICON=ICON+1
      GO TO 380
  375 PRINT, '  SAIU PELO COMANDO  375'
      RETURN
  380 IF(JBT.EQ.1) GO TO 400
  390 KORDER(JBT)=MEXT
      GO TO 410
  400 LOC(J)=MEXT
  410 IB=KORDER(IB)
  420 IF(IB.NE.1) GO TO 310
  430 IF(KOLUM(JB)+1-I)440,470,790
  440 JBS=JB
      JB=KORDER(JB)
  450 IF(JB.NE.1) GO TO 430
  460 PRINT, 'SAIU PELO COMANDO 460'
      RETURN
  470 KORDER(LAST)=JB
      LAST=JB
      IF(JBS.EQ.1) GO TO 490
  480 KORDER(JBS)=KORDER(JB)
      GO TO 790
  490 LOC(J)=KORDER(JB)
      GO TO 790
  500 IF(KOLUM(JB)+1-I)510,540,520
  510 JBS=JB
      JB=KORDER(JB)
      GO TO 310
  520 IF(KORDER(JB).EQ.1) GO TO 700
  530 JBS=JB
      JB=KORDER(JB)
      GO TO 310
  540 IF(JBS.EQ.1)GO TO 580
  550 KORDER(JBS)=KORDER(JB)
      KORDER(LAST)=JB
      LAST=JB
  560 IF(KORDER(JB).NE.1) GO TO 575
  570 JB=JBS
      GO TO 700
  575 JB=KORDER(JB)
      GO TO 310
  580 KORDER(LAST)=JB
      LAST=JB
      IF(KORDER(JB).EQ.1)GO TO 600
  590 LOC(J)=KORDER(JB)
      JB=KORDER(JB)
      GO TO 310
  600 IF(KOLUM(IB)+1.EQ.J) GO TO 620
  610 LOC(J)=NEXT
      JB=NEXT
      MEXT=NEXT
      IF(NEXT.EQ.LAST) GO TO 615
```

```
      NEXT=KORDER(NEXT)
      ICON=ICON+1
      KOLUM(JB)=KOLUM(IB)
      GO TO 650
  615 PRINT, ' SAIU PELO COMANDO 615'
      RETURN
  620 IB=KORDER(IB)
  630 IF(IB.NE.1) GO TO 610
  640 LOC(J)=1
      GO TO 790
  650 CONTINUE
  660 IF(KORDER(IB).EQ.1) GO TO 690
  670 IB=KORDER(IB)
      KOLUM(NEXT)=KOLUM(IB)
      GO TO 680
  675 PRINT,'  SAIU PELO COMANDO 675'
      RETURN
  680 MEXT=NEXT
      IF(NEXT.EQ.LAST) GO TO 675
      NEXT=KORDER(NEXT)
      ICON=ICON+1
      GO TO 650
  690 KORDER(MEXT)=1
      GO TO 790
  700 IF(KOLUM(IB)+1-J) 710,760,780
 710  KOLUM(NEXT)=KOLUM(IB)
      KORDER(JB)=NEXT
  720 MEXT=NEXT
      IF(NEXT.EQ.LAST) GO TO 725
      NEXT=KORDER(NEXT)
      ICON=ICON+1
      IB=KORDER(IB)
      GO TO 730
  725 PRINT,  'SAIU PELO COMANDO 725'
      RETURN
  730 IF(KOLUM(IB)+1-J)  750,650,740
  740 PRINT, ' SAIU PELO COMANDO 740'
      RETURN
  750 KOLUM(NEXT)=KOLUM(IB)
      GO TO 720
  760 IF(KORDER(IB).EQ.1)     GO TO 790
  770 IB=KORDER(IB)
  780 KOLUM(NEXT)=KOLUM(IB)
      KORDER(JB)=NEXT
      GO TO 680
  790 IF(ISCHM.NE.2) GO TO 270
      IF(KOWNT(J).EQ.1) GO TO 270
  800 IF(ICON-1) 810,270,830
  810 IF(KOWNT(J)+ICON.GE.NCN+1) GO TO 830
  820 NCN=KOWNT(J)+ICON-1
  830 IF(ICH2(J).EQ.1)GO TO 850
  840 L=ICH2(J)
      K=ICH1(J)
      ICH1(L)=K
      ICH2(K)=L
      M=KOWNT(J)+ICON-1
      GO TO 860
  850 K=ICH1(J)
      ICH2(K)=1
      M=KOWNT(J)
```

```
      INDEX(M+1)=K
      M=M+ICON-1
  860 KOWNT(J)=M
      K=INDEX(M+1)
      INDEX(M+1)=J
      ICH1(J)=K
      ICH2(K)=J
      ICH2(J)=1
      GO TO 270
 1110 JS=LOC(I)
      GO TO 1130
 1120 JS=KORDER(JS)
 1130 ICON=1
 1140 IF(JS.EQ.1) GO TO 200
 1150 J=KOLUM(JS)+1
      IF(KOWNT(J).LT.1)GO TO 1120
      KS=LOC(J)
      LS=KS
      GO TO 1170
 1160 KS=KORDER(KS)
 1170 IF(KS.NE.1) GO TO 1190
 1180 IICON=ICON-1
      ICON=IICON/2+1
      GO TO 1410
 1190 K=KOLUM(KS)+1
      JB=LS
      KB=LOC(K)
 1200 IF(KOLUM(JB)-KOLUM(KB)) 1210,1250,1270
 1210 IF(KOLUM(JB)+1.EQ.K) GO TO 1230
 1220 ICON=ICON+1
 1230 JB=KORDER(JB)
 1240 IF(JB.NE.1) GO TO 1200
      GO TO 1160
 1250 IF(KORDER(JB).EQ.1) GO TO 1160
 1260 JB=KORDER(JB)
 1270 IF(KORDER(KB).EQ.1) GO TO 1290
 1280 KB=KORDER(KB)
      GO TO 1200
 1290 IF(KOLUM(JB)+1.EQ.K) GO TO 1310
 1300 ICON=ICON+1
 1310 IF(KORDER(JB).EQ.1) GO TO 1160
 1320 JB=KORDER(JB)
      GO TO 1290
 1410 IF(KOWNT(J)-ICON) 1440,1120,1420
 1420 IF(NCN.LE.ICON) GO TO 1440
 1430 NCN=ICON
 1440 IF(ICH2(J).EQ.1) GO TO 1460
 1450 L=ICH2(J)
      K=ICH1(J)
      ICH1(L)=K
      ICH2(K)=L
      M=ICON
      GO TO 1470
 1460 K=ICH1(J)
      ICH2(K)=1
      M=KOWNT(J)
      INDEX(M+1)=K
      M=ICON
 1470 KOWNT(J)=M
      K=INDEX(M+1)
```

```
      INDEX(M+I)=J
      ICH1(J)=K
      ICH2(K)=J
      ICH2(J)=1
      GO TO 1120
1510  ICON=1
      IF(KOWNT(I).LT.1) GO TO 170
      KS=LOC(I)
      LS=KS
1520  IF(KS.NE.1) GO TO 1540
1530  IICON=ICON-1
      J=IICON/2+1
      KOWNT(I)=J
      GO TO 155
1540  K=KOLUM(KS)+1
      JB=LS
      KB=LOC(K)
1550  IF(KOLUM(JB)-KOLUM(KB)) 1600,1560,1580
1560  IF(KORDER(JB).EQ.1) GO TO 1680
1570  JB=KORDER(JB)
1580  IF(KORDER(KB).EQ.1) GO TO 1640
1590  KB=KORDER(KB)
      GO TO 1550
1600  IF(KOLUM(JB)+1.EQ.K) GO TO 1620
1610  ICON=ICON+1
1620  JB=KORDER(JB)
1630  IF(JB.NE.1) GO TO 1550
      GO TO 1680
1640  IF(KOLUM(JB)+1.EQ.K) GO TO 1660
1650  ICON=ICON+1
1660  IF(KORDER(JB).EQ.1) GO TO 1680
1670  JB=KORDER(JB)
      GO TO 1640
1680  KS=KORDER(KS)
      GO TO 1520
      END

      SUBROUTINE DUMP
      INTEGER R,W
      COMMON NTOT,NEXT,KOWNT(1500),LOC(1500),KOLUM(5000),KORDER(5000),
     1NORDER(1500),X1,INDEX(99),ICH1(1500),ICH2(1500),LAST,ISCHM,
     2IDEED,TO,JBRR,R,W,NNTOT
      DIMENSION N1(200),N2(2000),N3(2000)
      LF=NTOT-1
      DO 100 I=2,LF
      N1(I)=1
100   IF(INDEX(I).NE.1) N1(I)=INDEX(I)
      DO 200 I=2,NNTOT
      N2(I)=1
      N3(I)=1
      IF(ICH2(I).NE.1) N2(I)=ICH2(I)
200   IF(ICH1(I).NE.1) N3(I)=ICH1(I)
      WRITE(W,302)(N1(I)-1,N2(I)-1,N3(I)-1,KOWNT(I)-1,LOC(I)-1,I-1,
     *KOLUM(I),KORDER(I)-1,I=2,LF)
302   FORMAT(20X,8I3)
      LG=LF+1
      WRITE(W,304)(N2(I)-1,N3(I)-1,KOWNT(I)-1,LOC(I)-1,I-1,KOLUM(I),
     *KORDER(I)-1,I=LG,NNTOT)
304   FORMAT(23X,7I3)
      IX=NNTOT+1
```

```
      WRITE(M,306) (I-1,KOLUM(I),KORDER,I)-1,I=IX,NEXT)
  306 FORMAT(35X,3I3)
      RETURN
      END

   $ENTRY
  1          2
  1          3
  1          7
  2          1
  2          4
  2          8
  3          1
  3          4
  3          5
  4          2
  4          3
  4          6
  5          3
  5          6
  5          7
  6          4
  6          5
  6          8
  7          1
  7          5
  7          8
  8          2
  8          6
  8          7
                  0  2  0  3  1  1  2  2
                  0  3  1  3  4  2  3  3
                  0  4  2  3  7  3  7  0
                  8  5  3  3 10  4  1  5
                  0  6  4  3 13  5  4  6
                  0  7  5  3 16  6  8  0
                     8  6  3 19  7  1  8
                     0  7  3 22  8  4  9
                                 9  5  0
                                10  2 11
                                11  3 12
                                12  6  0
                                13  3 14
                                14  6 15
                                15  7  0
                                16  4 17
                                17  5 18
                                18  8  0
                                19  1 20
                                20  5 21
                                21  8  0
                                22  2 23
                                23  6 24
                                24  7  0
                                25UUU 26
                  8
                     2
                     6
                     7
                  0  3  0  3  1  1  2  2
```

0	6	0	4	4	2	3	3
0	4	1	3	7	3	7	0
5	5	3	3	10	4	1	5
7	0	4	3	13	5	4	25
0	7	2	4	27	6	8	18
	0	6	4	19	7	1	8
	0	7	3	22	8	4	9
					9	5	0
					10	2	11
					11	3	12
					12	6	0
					13	3	14
					14	6	15
					15	7	0
					16	4	17
					17	5	28
					18	8	21
					19	1	29
					20	5	30
					21	8	0
					22	2	23
					23	6	24
					24	7	0
					25	6	26
					26	7	0
					27	2	16
					28	7	0
					29	2	20
					30	6	0
					31UUU		32
5							
	3						
	6						
	7						
0	4	0	3	1	1	2	2
0	6	0	4	4	2	3	3
0	0	7	4	7	3	7	0
4	0	1	3	10	4	1	5
3	0	4	3	13	5	4	25
0	7	2	4	27	6	8	18
	3	6	4	19	7	1	8
	0	7	3	22	8	4	31
					9	5	17
					10	2	11
					11	3	12
					12	6	0
					13	3	14
					14	6	15
					15	7	0
					16	4	28
					17	5	20
					18	8	21
					19	1	29
					20	5	30
					21	8	9
					22	2	23
					23	6	24
					24	7	0
					25	6	26
					26	7	0

```
                      27   2  33
                      28   7   0
                      29   2  34
                      30   6   0
                      31   6  32
                      32   7   0
                      33   3  16
                      34   3  30
                      35UUU   36
   4
       2
       3
       6
   0   6   0   3   1   1   2   2
   0   7   0   4   4   2   3   3
   0   0   7   4   7   3   7   0
   6   0   1   3  10   4   1  35
   3   0   4   3  13   5   4   8
   0   0   1   3  27   6   8  18
       3   2   4  19   7   1  36
       0   7   3  22   8   4  16
                       9   5  17
                      10   2  11
                      11   3  12
                      12   6   0
                      13   3  14
                      14   6  15
                      15   7   0
                      16   4  28
                      17   5  20
                      18   8  21
                      19   1  29
                      20   5   5
                      21   8   9
                      22   2  23
                      23   6  24
                      24   7   0
                      25   6  26
                      26   7   0
                      27   2  33
                      28   7   0
                      29   2  34
                      30   6   0
                      31   6  32
                      32   7   0
                      33   3  28
                      34   3  30
                      35   3  25
                      36   2  31
                      37UUU   38
   6
       2
       3
       7
   0   2   0   3   1   1   2   2
   0   3   1   3   4   2   3   3
   0   7   2   3   7   3   7   0
   7   0   1   3  10   4   1  35
   0   0   4   3  13   5   4   8
   0   0   1   3  27   6   8  18
```

```
                                   0   3   3 19   7   1 36
                                   0   7   3 22   8   4 16
                                                  9   5 17
                                                 10   2 11
                                                 11   3 12
                                                 12   6  0
                                                 13   3 14
                                                 14   6 15
                                                 15   7  0
                                                 16   4 25
                                                 17   5 20
                                                 18   8 21
                                                 19   1 29
                                                 20   5  5
                                                 21   8  9
                                                 22   2 23
                                                 23   6 24
                                                 24   7  0
                                                 25   6 31
                                                 26   7  0
                                                 27   2 33
                                                 28   7  0
                                                 29   2 34
                                                 30   6  0
                                                 31   6 30
                                                 32   7  0
                                                 33   3 28
                                                 34   3  0
                                                 35   3 26
                                                 36   2 32
                                                 37UUU 38
                              7
                                  1
                                  2
                                  3
                              3
                                  1
                                  2
                              2
                                  1
                              1
        8       18
        1        8
        2        7
        3        6
        4        3
        5        2
        6        4
        7        5
        8        1

CORE USAGE        OBJECT CODE=    16896 BYTES,ARRAY AREA=    37240 BYTES,TOTAL AR

DIAGNOSTICS          NUMBER OF ERRORS=         0, NUMBER OF WARNINGS=         0, NUM

COMPILE TIME=      3.17 SEC,EXECUTION TIME=      1.55 SEC,  WATFIV - JUL 1973 V1
```

BIBLIOGRAFIA

1. FOX L., "An introduction to numerical linear algebra", cla̲redon Press-Oxford, 1964.

2. D. X. Lima, "Fluxo de cargas", Universidade Federal de Pernambuco, 1981.

3. W. F. Tinney, J. W. Walker, "Direct solutions of sparse network Equations by Optimally Ordered Triangular Factorization", IEEE Proceedings, Vol.55, Nº 11, November, 1967.

4. W. F. Tinney, W. L. Powell, J. W. Wolker, "Programming os Sparsity-Directed Ordering Schemes", Bonneville Power Administration P.O. Box 3621 Portland,Oreon 97208.

5. G. Irisarri, S. F. Hodges, A. M. Sasson, "An Optimal Or̲deringer Algorithm for sparse Matrix Applications", IEEE Transactions on Power Apparatus and Systems, Vol.PASS-97, Nº 6, Nov/Dec.1978.

6. W. D. Stevenson, "Elements of Power System Analysis", Mac̲Graw-Hill Book Company, 1962.

7. G. W. Stagg e A. H. El-Abiad, "Computer Methods in Power System Analysis", McGraw-Hill Book Company,1968.

8. O. I. Elgerd, "Electric Energy Systems Theory: An Introduction", McGraw-Hill Book Company, 1971.