

unesp

UNIVERSIDADE ESTADUAL PAULISTA
FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DISSERTAÇÃO DE MESTRADO
EM
ENGENHARIA ELÉTRICA

1210001051



Nº 16

SIMULAÇÃO DA ESTABILIDADE TRANSITÓRIA DE
SISTEMAS DE ENERGIA ELÉTRICA UTILIZANDO
COMPUTADORES VETORIAIS
ALUNO: ALESSANDRO RUIZ BASSO
ORIENTADOR: ANTONIO PADILHA FELTRIN

Te.1051

C613

UNIVERSIDADE ESTADUAL PAULISTA – UNESP

FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA – FEIS

DEPARTAMENTO DE ENGENHARIA ELÉTRICA – DEE

**SIMULAÇÃO DA ESTABILIDADE TRANSITÓRIA DE SISTEMAS
DE ENERGIA ELÉTRICA UTILIZANDO COMPUTADORES
VETORIAIS**

ALESSANDRO RUIZ BASSO

1210001051



Orientador:

Antonio Padilha Feltrin

Proc. 040/98 NED 010

UNESP - "CAMPUS" DE ILHA SOLTEIRA SERVIÇO DE ENGENHARIA E DOCUMENTAÇÃO	
DATA DE ENTRADA 01.04.98	DATA DE TCMBO 30.04.98
REGISTRO Deia	TCMBO 1051
ALCUNHA Deia R\$ 10,00 Aut	CLASSIFICAÇÃO B322A

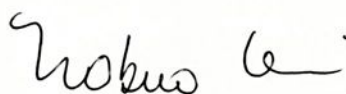
Dissertação apresentada à Faculdade de Engenharia de Ilha Solteira / FEIS / UNESP / como parte dos requisitos exigidos para a obtenção do título de MESTRE EM ENGENHARIA ELÉTRICA.

Ilha Solteira, novembro de 1997

“SIMULAÇÃO DA ESTABILIDADE TRANSITÓRIA DE SISTEMAS DE ENERGIA ELÉTRICA UTILIZANDO COMPUTADORES VETORIAIS”

Alessandro Ruiz Basso

DISSERTAÇÃO SUBMETIDA À FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA - UNESP
COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO TÍTULO DE
MESTRE EM ENGENHARIA ELÉTRICA (ME).



Nobuo Oki, D.Sc. - Coordenador

COMISSÃO EXAMINADORA:



Antonio Padilha Feltrin, D.Sc. - Orientador



André Luiz Morelato França, D.Sc.



Carlos Roberto Minussi, D.Sc.

Ilha Solteira – SP, 21 de novembro de 1997.



A Deus e a minha família.



AGRADECIMENTOS

- Aos professores Antonio Padilha Feltrin e Carlos Roberto Minussi pela orientação e estímulo;
- A todos os colegas do curso de pós-graduação;
- A todos os funcionários do DEE (Departamento de Engenharia Elétrica) da FEIS (Faculdade de Engenharia de Ilha Solteira);
- Ao CESUP (Centro Nacional de Supercomputação) da UFRGS (Universidade Federal do Rio Grande do Sul) que possibilitou o uso do computador CRAY Y-MP 2E/232;
- À CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) e à FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) pelo apoio financeiro.

RESUMO

Neste trabalho, desenvolve-se um programa para a análise da estabilidade transitória de sistemas de energia elétrica adaptado para ser executado em ambiente computacional dotado de processamento vetorial. O problema é abordado usando técnica de simulação passo a passo, através da qual as equações que descrevem o sistema são resolvidas ao longo do tempo. Utilizando-se esta técnica, o problema da análise da estabilidade transitória resume-se, basicamente, na resolução de um conjunto de equações algébricas e um conjunto de equações diferenciais ordinárias não-lineares. A etapa do problema que oferece maior dificuldade à vetorização é a resolução das equações algébricas. Portanto, é dado maior enfoque à resolução destas equações, propondo-se algumas metodologias que otimizam o uso do processamento vetorial. O programa para a análise da estabilidade transitória, implementado com as metodologias propostas para a resolução das equações algébricas, mostra que o tempo de CPU é reduzido significativamente, justificando a busca de novas metodologias que otimizam o processamento vetorial, viabilizando, assim, o seu emprego nos centros de controle em tempo real de sistemas de energia elétrica. Como forma de ilustração das metodologias propostas apresentam-se simulações considerando sistemas de energia elétrica compostos por 9, 68, 118, 320, 725 e 1729 barras.

ABSTRACT

This thesis deals with transient stability analysis on vector computer, based on time-domain simulation. This approach requires the solution of a set of differential equations in conjunction of another set of algebraic equations. The solution of the differential equations on vector computers is implemented straightforwardly. On the other hand, the set of algebraic equations presents a scalar and sequential set of tasks, and the solution of these equations on vector computers has required much more investigations to speedup the simulation. In this work, the main focus was to present some methods to solve faster the set of algebraic equations using vector processing. The results were taken from real life power systems with 9, 68, 118, 320, 725 and 1729 buses and have shown that this implementation has potential to allow analysis of on-line transient stability assessment.



SIMBOLOGIA

Y	=	matriz de admitância nodal (esparsa complexa);
I	=	vetor de injeção líquida de correntes nodais;
V	=	vetor de tensões nodais;
B'	=	matriz esparsa real;
P	=	vetor de injeção líquida de potência nodal;
θ	=	vetor dos ângulos das tensões nodais;
r_{ik}	=	resistência série da linha de transmissão que liga o nó "i" ao nó "k";
x_{ik}	=	reatância série da linha de transmissão que liga o nó "i" ao nó "k";
b_{ik}^{sh}	=	susceptância <i>shunt</i> da linha de transmissão que liga o nó "i" ao nó "k";
Y_c	=	admitância representativa das cargas do sistema;
V_c	=	tensão da barra de carga;
V_c^0	=	tensão inicial da barra de carga (caso base);
P_c^0	=	injeção de potência ativa na barra de carga (caso base);
Q_c^0	=	injeção de potência reativa na barra de carga (caso base);
I_c	=	corrente da barra de carga;
I_c^0	=	corrente inicial da barra de carga (caso base);
E	=	módulo da tensão interna do gerador síncrono;
δ_0	=	ângulo da tensão interna do gerador síncrono antes da ocorrência da contingência
	=	ângulo mecânico do rotor antes da ocorrência da contingência;

- x'_d = reatância transitória de eixo direto da máquina síncrona;
 P_g = injeção de potência ativa na barra de geração;
 Q_g = injeção de potência reativa na barra de geração;
 V_g = tensão terminal da barra de geração;
 α = ângulo da tensão terminal da barra de geração;
 ω = velocidade angular do eixo do gerador;
 ω_s = velocidade angular síncrona
 = $2 \pi f_o$;
 f_o = frequência nominal do sistema de energia elétrica;
 H = constante de inércia da máquina;
 P_m = potência mecânica aplicada ao eixo do gerador;
 P_e = potência elétrica de entreferro;
 δ = posição angular do rotor em relação ao eixo da referência síncrona;
 A = matriz esparsa (maioria dos elementos são nulos);
 x = vetor solução desejado;
 b = vetor independente conhecido;
 Δt = passo de integração.

ÍNDICE

CAPÍTULO I

INTRODUÇÃO	1
------------	---

CAPÍTULO II

ESTUDO DA ESTABILIDADE EM SISTEMAS DE ENERGIA ELÉTRICA

2.1 INTRODUÇÃO	4
2.2 DEFINIÇÃO GERAL DE ESTABILIDADE	5
2.3 ESTUDO DA ESTABILIDADE TRANSITÓRIA	6
2.4 MODELAGEM DO SISTEMA DE ENERGIA ELÉTRICA	7
2.4.1 MODELAGEM DAS LINHAS DE TRANSMISSÃO	7
2.4.2 MODELAGEM DOS TRANSFORMADORES	8
2.4.3 MODELAGEM DAS CARGAS	10
2.4.3.1 IMPEDÂNCIA CONSTANTE	10
2.4.3.2 CORRENTE CONSTANTE	10
2.4.3.3 POTÊNCIA CONSTANTE	11
2.4.4 MODELAGEM DOS GERADORES SÍNCRONOS	11
2.5 SOLUÇÃO DO PROBLEMA DE ESTABILIDADE TRANSITÓRIA	13
2.5.1 SOLUÇÃO POR MÉTODOS DIRETOS E SOLUÇÃO PASSO A PASSO AO LONGO DO TEMPO	13
2.5.2 ESQUEMAS DE SOLUÇÃO	14
2.5.3 RESOLUÇÃO DAS EQUAÇÕES ALGÉBRICAS E DAS EQUAÇÕES DIFERENCIAIS	17
2.5.4 SOLUÇÃO DO PROBLEMA DA ESTABILIDADE TRANSITÓRIA UTILIZANDO COMPUTADORES DE ALTO DESEMPENHO	19

CAPÍTULO III

RESOLUÇÃO DAS EQUAÇÕES ALGÉBRICAS

3.1 INTRODUÇÃO	20
3.2 SOLUÇÃO UTILIZANDO SUBSTITUIÇÕES	21
3.3 GRAFO DOS CAMINHOS DE FATORAÇÃO (GCF)	30
3.4 ORDENAÇÕES	31
3.4.1 ORDENAÇÃO <i>MINIMUM DEGREE</i> (MD)	32
3.4.2 ORDENAÇÃO <i>MINIMUM LENGTH</i> (ML)	33
3.4.3 ORDENAÇÃO <i>MINIMUM DEGREE</i> <i>MINIMUM LENGTH</i> (MDML)	33
3.4.4 ORDENAÇÃO <i>MINIMUM LENGTH</i> <i>MINIMUM DEGREE</i> (MLMD)	34
3.5 SOLUÇÃO UTILIZANDO O MÉTODO DA MATRIZ <i>W</i>	35
3.5.1 PARTICIONAMENTO DA MATRIZ <i>W</i>	36
3.5.2 PROCESSAMENTO VETORIAL USANDO O MÉTODO DA MATRIZ <i>W</i>	39
3.5.3 MATRIZ <i>W</i> COM VETORES CHEIOS	44
3.6 SOLUÇÃO UTILIZANDO O MÉTODO DBSA	51
3.6.1 MÉTODO DBSA COM VETORES CHEIOS	54
3.7 MÉTODO DOS VETORES ESPARSOS	57
3.8 MÉTODO DA MATRIZ <i>W</i> E DBSA COM VETORES ESPARSOS	58
3.9 DESEMPENHO DAS METODOLOGIAS APRESENTADAS	63

CAPÍTULO IV

PROGRAMA PARA A ANÁLISE DA ESTABILIDADE TRANSITÓRIA EM AMBIENTE VETORIAL

4.1 INTRODUÇÃO	68
4.2 REPRESENTAÇÃO DOS GERADORES SÍNCRONOS ATRAVÉS DO MODELO CLÁSSICO	69



4.3 DESCRIÇÃO DO PROGRAMA COMPUTACIONAL	71
4.4 RESULTADOS	75
4.5 DESEMPENHO DO PROGRAMA NO COMPUTADOR VETORIAL	84

CAPÍTULO V

CONCLUSÃO	88
-----------	----

REFERÊNCIAS BIBLIOGRÁFICAS	91
----------------------------	----

ANEXO I

DADOS DO SISTEMA DA REFERÊNCIA [07]
(68 BARRAS E 16 GERADORES)

CAPÍTULO I

INTRODUÇÃO

Neste trabalho foi enfocada a simulação da estabilidade transitória de sistemas de energia elétrica em um computador vetorial. Basicamente, o problema de cálculo da estabilidade, via simulação passo a passo, envolve a solução de um conjunto de equações algébricas e outro de equações diferenciais. Atualmente destaca-se o método direto, via decomposição LDU, para a solução das equações algébricas e o método trapezoidal na solução das equações diferenciais. Estes métodos de resolução são os escolhidos neste trabalho.

Na simulação usando processamento vetorial, a resolução das equações diferenciais pode ser vetorizada diretamente sem grandes dificuldades. Por outro lado, a solução do conjunto de equações algébricas é um procedimento predominantemente seqüencial, e que exige uma preparação e organização de dados modificada para fim de obter desempenho utilizando-se processamento vetorial. Neste trabalho foi dado um enfoque especial à solução das equações algébricas via método direto.

O computador utilizado foi o CRAY Y-MP2E/232 da Universidade Federal do Rio Grande do Sul. A utilização foi realizada remotamente utilizando a rede *internet*. Isto mostrou também que atualmente grandes empresas de energia elétrica não necessitam despendar grandes somas de dinheiro na compra de computadores de alto desempenho. Para muitas utilizações o aluguel de tempo de simulação é bastante viável.

O computador CRAY da Universidade Federal do Rio Grande do Sul possui as seguintes características:



Número de processadores	: 2;
Tamanho da memória	: 32 Mwords;
Seções de Memória	: 4;
Barras de Memória	: 32;
Ciclo do relógio	: 6 nanossegundos;
Velocidade máxima por CPU	: 333 Mflops;
Número de discos	: 8;
Espaço total em disco	: 16 Gbytes.

No processamento escalar as instruções são executadas uma a uma na ordem em que aparecem. No processamento vetorial as instruções dentro de um laço de programa (*loop*) são executadas em grupo, utilizando registradores vetoriais. Estes grupos são, por exemplo, um vetor ou uma coluna de uma matriz.

Este trabalho está organizado da seguinte forma:

- No capítulo II aborda-se, de uma forma geral, o estudo sobre a estabilidade de sistemas de energia elétrica, no qual se define o que é estabilidade, como ela pode ser classificada e quais os métodos utilizados para sua análise. É apresentada, em linhas gerais, uma modelagem dos principais componentes dos sistemas de energia elétrica, como geradores, linhas de transmissão, transformadores e cargas. Verifica-se que a modelagem do sistema resulta em dois conjuntos de equações: um conjunto de equações algébricas e um conjunto de equações diferenciais. Discute-se como estes conjuntos de equações podem interagir e como transformar as equações diferenciais em equações algébricas.

- O capítulo III é inteiramente dedicado à resolução das equações algébricas em ambiente dotado de processamento vetorial, visto que a resolução destas equações é a etapa que oferece maiores dificuldades à vetorização. São implementadas várias metodologias para resolução das equações algébricas, sendo algumas propostas na



literatura e outras propostas neste trabalho; compara-se quantitativamente os seus respectivos desempenhos.

- No capítulo IV é discutida a implementação do programa para a análise da estabilidade transitória. Este capítulo tem como objetivo aplicar algumas metodologias desenvolvidas no capítulo III e verificar os respectivos tempos de processamento. O cálculo da estabilidade transitória é o problema adotado pelo fato de ser necessário resolver muitas vezes um conjunto de equações algébricas.

- O capítulo V é dedicado à conclusão e às sugestões para futuros trabalhos.



CAPÍTULO II

ESTUDO DA ESTABILIDADE EM SISTEMAS DE ENERGIA ELÉTRICA

2.1 INTRODUÇÃO

Com a finalidade de suprir a crescente demanda de energia elétrica – com quantidade e qualidade apropriadas e menor custo possível – os sistemas de energia elétrica têm-se tornado cada vez maiores e mais complexos, aumentando-se assim, sua capacidade de geração, de transmissão e de distribuição.

Para que um sistema de energia elétrica forneça energia com qualidade, um dos requisitos que deve ser contemplado é a permanência do sistema em operação o maior tempo possível, e no caso da ocorrência de contingências no sistema, os seus efeitos devem ser limitados a uma menor região.

Portanto, torna-se necessário conhecer o comportamento do sistema de energia elétrica diante de várias situações, ou seja, durante sua operação normal, como também durante a ocorrência de algum imprevisto no mesmo. O conhecimento do comportamento do sistema é fundamental para que se possa operá-lo de maneira confiável e otimizada, dentro de uma margem de segurança, durante sua operação normal. Na presença de alguma contingência, torna-se necessário prever a reação do sistema, a fim de instalar e ajustar equipamentos de proteção que possam limitar os



efeitos da perturbação numa menor região possível, preservando-se assim a integridade do restante do sistema.

Atualmente, a análise do comportamento dinâmico do sistema é feito através de simulação digital, e como os sistemas têm-se tornado cada vez maiores e mais complexos, as simulações podem apresentar bons desempenhos em termos de tempos de CPU se realizadas em ambientes computacionais adequados.

2.2 DEFINIÇÃO GERAL DE ESTABILIDADE

A análise do comportamento dinâmico do sistema de energia elétrica, devido a ocorrência de uma contingência, é definido como o estudo de estabilidade. Este estudo tem sido classificado de diversas maneiras. A referência [28] classifica a estabilidade de acordo com o tempo de simulação após o defeito; ou seja, para tempos de simulação de até 8 segundos, 5 minutos e 20 minutos, denomina-se estabilidade de curto, médio e longo prazo, respectivamente.

O estudo de estabilidade de curto prazo é definido por muitos autores como o estudo de estabilidade transitória, cujo principal objetivo é a análise do comportamento do sistema devido as grandes perturbações, tais como: curto-circuito em linhas de transmissão, perda repentina de geração, etc. A análise da estabilidade do sistema de médio e longo prazo é conhecida como a análise de estabilidade dinâmica, e estuda o comportamento do sistema em consequência de pequenas perturbações, como por exemplo, aumento da carga do sistema.

O enfoque dado neste trabalho foi o estudo de estabilidade transitória através de simulação digital, adaptando-se o programa para a implementação em ambiente dotado de processamento vetorial.



2.3 ESTUDO DA ESTABILIDADE TRANSITÓRIA

Na análise da estabilidade transitória (ou de curto prazo), observa-se as primeiras oscilações dos rotores das máquinas síncronas frente a uma determinada perturbação. O estudo da estabilidade transitória de sistemas de energia elétrica tem como principal objetivo, determinar se os geradores síncronos, que compõem o sistema, permanecerão em sincronismo devido a ocorrência de uma contingência do tipo: curto-circuito, perda de geração, etc. Durante a ocorrência destes eventos, o equilíbrio entre a potência mecânica de entrada das máquinas síncronas e a potência elétrica de saída é drasticamente rompido, principalmente nos geradores mais próximos eletricamente do defeito. Desta forma, os rotores das máquinas síncronas sofrerão diferentes acelerações e poderão sair do estado de equilíbrio.

A referência [02] define estabilidade transitória da seguinte forma: “Se a resposta oscilatória de um sistema durante o período transitório, após uma perturbação, é amortecida e o sistema estabelece num tempo finito uma nova condição de operação estável, diz-se que o sistema é estável. Caso contrário, é considerado instável”.

Em outras palavras, caso as oscilações relativas dos diversos rotores síncronos tenderem a se amortecer, diz-se que o sistema é estável em relação a perturbação ocorrida. Porém, se a oscilação de um ou mais rotores síncronos se afastar das demais, o sistema será considerado instável. Cabe lembrar que um ponto de operação é considerado estável ou instável, dependendo das características da perturbação, ou seja, da localização e do tipo do defeito, do tempo de eliminação da falta, do tipo de atuação dos equipamentos de proteção para a eliminação do defeito, entre outros fatores.

Via de regra, para sistemas de grande porte, o estudo do comportamento dinâmico é realizado através de simulação e requer a utilização de modelos não-lineares. Durante o período transitório a modelagem do sistema pode ser simplificada, eliminando-se a representação dos transitórios elétricos da rede de transmissão, que são extremamente rápidos e desprezando-se, também, os efeitos dos controles suplementares dos geradores síncronos, que são lentos se comparados ao período em estudo.



2.4 MODELAGEM DO SISTEMA DE ENERGIA ELÉTRICA

Os componentes do sistema de energia elétrica podem ser modelados utilizando-se vários níveis de detalhamento, segundo os objetivos da análise que se pretende realizar.

2.4.1 MODELAGEM DAS LINHAS DE TRANSMISSÃO

As linhas de transmissão são modeladas segundo o modelo equivalente π , representado na figura 2.1.

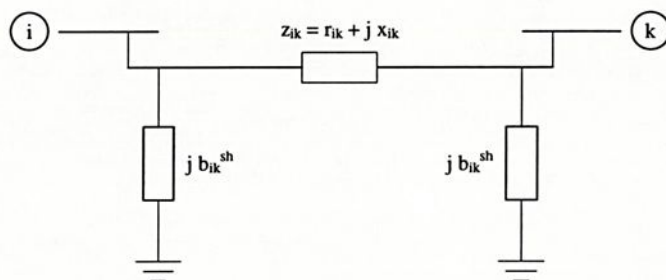


Figura 2.1 – Modelo equivalente π da linha de transmissão que liga o nó genérico “i” ao nó genérico “k”.

A admitância série da linha de transmissão é dada por:

$$y_{ik} = (z_{ik})^{-1} = (r_{ik} + j x_{ik})^{-1} = (r_{ik} / (r_{ik}^2 + x_{ik}^2)) - j (x_{ik} / (r_{ik}^2 + x_{ik}^2)) \quad (2.1)$$

2.4.2 MODELAGEM DOS TRANSFORMADORES

Os transformadores são modelados conforme mostra a figura 2.2.

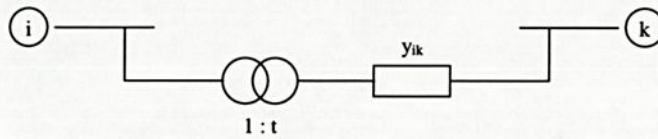


Figura 2.2 – Modelo do transformador que liga o nó genérico “i” ao nó genérico “k”.

Na figura 2.2, o transformador é representado por uma admitância série y_{ik} e um autotransformador ideal com relação de transformação 1:t.

Na modelagem de um transformador em fase, “t” é um número real ($t = a$); para um transformador defasador, “t” é um número complexo ($t = a e^{j\varphi}$), e para um transformador defasador puro $t = e^{j\varphi}$.

No caso de se trabalhar com um transformador em fase, pode-se representar o transformador por um modelo equivalente π [21], como ilustrado na figura 2.3.

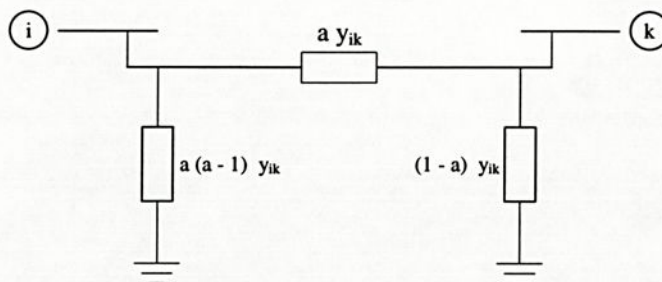


Figura 2.3 – Modelo equivalente π do transformador em fase que
liga o nó genérico “i” ao nó genérico “k”.

Portanto, a rede de transmissão (linhas e transformadores) é modelada através das equações que descrevem o problema de fluxo de potência. Durante o período transitório não se verifica completamente o regime senoidal que se encontra durante o regime estático, por isso, a modelagem da rede de transmissão através de equações lineares constitui uma aproximação. Caso esta aproximação não fosse adotada, a modelagem do sistema de transmissão deveria ser feita através de equações diferenciais o que aumentaria significativamente o esforço computacional sem acarretar melhorias significativas nos resultados, uma vez que os transitórios do sistema de transmissão são extremamente rápidos.

Desta forma, representa-se a rede de transmissão através da matriz admitância Y , conforme mostra a equação 2.2.

$$I = Y V \quad (2.2)$$

A matriz Y , quando desconsideradas as admitâncias mútuas, é formada da seguinte forma: na diagonal tem-se a soma de todas as admitâncias ligadas ao respectivo nó, e fora da diagonal, tem-se o negativo da admitância série que liga os nós correspondentes. Com isso, a matriz Y é altamente esparsa, pois só existem elementos diferentes de zero fora da diagonal da matriz, quando existir uma ligação física entre os nós correspondentes. Em todas as posições da diagonal existem elementos diferentes de zero.

A representação da rede de transmissão por linhas de transmissão e transformadores em fase leva a uma matriz Y simétrica. Caso sejam modelados transformadores defasadores, por exemplo, pode-se obter uma matriz assimétrica em valores, mas simétrica em estrutura.

A matriz Y permanece constante durante a análise da estabilidade transitória, exceto nos instantes em que ocorre alguma mudança topológica na rede de transmissão, como por exemplo, desligamento e religamento de uma linha de transmissão.



2.4.3 MODELAGEM DAS CARGAS

As cargas dos sistemas de energia elétrica são modeladas, basicamente, através dos modelos de impedância constante, corrente constante ou potência constante, ou ainda, como uma combinação destes três modelos.

2.4.3.1 IMPEDÂNCIA CONSTANTE

Neste modelo, a carga é descrita, segundo a referência [02], através de uma admitância constante para a terra da seguinte forma:

$$Y_c = P_c^0 / |V_c^0|^2 - j(Q_c^0 / |V_c^0|^2) \quad (2.3)$$

Todos os dados necessários para o cálculo de Y_c provêm de um estudo inicial de fluxo de potência (caso base).

Através deste modelo, as cargas são representadas por admitâncias que são incluídas na diagonal principal da matriz Y (equação 2.2).

2.4.3.2 CORRENTE CONSTANTE

No modelo de corrente constante, a carga é representada através de uma fonte de corrente constante para a terra, cujo valor é dado por:

$$I_c^0 = ((P_c^0 + jQ_c^0) / V_c^0)^* \quad (2.4)$$



Semelhante à modelagem anterior, todos os dados necessários para o cálculo de I_c^o têm como origem um estudo prévio de fluxo de potência. Os valores das correntes de carga são incluídos nas respectivas posições do vetor I da equação 2.2.

2.4.3.3 POTÊNCIA CONSTANTE

Neste caso, o valor da potência da carga é mantido constante, independente da tensão nodal. Com isso, a relação entre a corrente e a tensão da barra de carga passa a ser dada por:

$$I_c = ((P_c^o + jQ_c^o) / V_c)^* \quad (2.5)$$

Isto faz com que, utilizando-se o modelo de potência constante para representar as cargas do sistema, a equação 2.2 passa a exigir um processo iterativo de solução, pois torna-se necessário encontrar valores para a corrente e a tensão nas barras de carga que satisfaçam as equações 2.5 e 2.2.

2.4.4 MODELAGEM DOS GERADORES SÍNCRONOS

Os geradores síncronos são modelados segundo [34] através de quatro modelos diferentes que são apresentados resumidamente a seguir:

Modelo I: Também conhecido como modelo clássico. Neste modelo a magnitude do fluxo do campo magnético é mantida constante. São desprezadas todas as correntes de amortecimento e seus efeitos, bem como a saliência transitória da máquina síncrona. Este modelo requer a mínima quantidade de dados, basicamente se faz necessário a reatância transitória de eixo direto (x'_d) e a constante de inércia da máquina (H).



Modelando-se o gerador síncrono através do modelo clássico, são geradas, para cada máquina, duas equações diferenciais de primeira ordem.

O modelo clássico, devido a sua simplicidade, é usado para a análise da estabilidade transitória, ou seja, para a análise das primeiras oscilações da máquina síncrona após uma dada perturbação. Para uma análise do comportamento da máquina além das primeiras oscilações, o modelo clássico deve ser usado com reservas, ou adota-se outro modelo que represente melhor a variação do fluxo magnético e as correntes de amortecimento, entre outros fatores.

Modelo II: Este é o modelo mais simples quando se representa os efeitos da variação do campo magnético no comportamento da máquina, mas despreza todos os efeitos das correntes de amortecimento. O modelo II requer mais cálculos do que o modelo I, mas ele é adequado para uma variedade de estudos, incluindo uma extensa análise de estabilidade dinâmica. São necessários seis parâmetros da máquina para representar o gerador através do modelo II, que por sua vez gera, para cada máquina, três equações diferenciais de primeira ordem.

Modelo III: Inclui o efeito da variação do campo magnético e representa os efeitos das correntes de amortecimento do eixo em quadratura. Este modelo é adequado para representar uma máquina de rotor cilíndrico; já as máquinas de pólos salientes necessitam de um modelo mais elaborado quando se deseja representar os efeitos das correntes de amortecimento. Para modelar a máquina síncrona, utilizando-se o modelo III, são necessários oito parâmetros diferentes do gerador. A utilização deste modelo, resulta, para cada máquina, quatro equações diferenciais de primeira ordem e três equações algébricas.

Modelo IV: Este é o modelo mais elaborado dentre os quatro modelos e inclui o efeito da variação do campo magnético, os efeitos das correntes de amortecimento dos eixos direto e em quadratura e a saturação da máquina síncrona. Por ser um modelo mais completo, exige maior quantidade de cálculos e de dados para a sua utilização (dez parâmetros da máquina). A modelagem de cada gerador, utilizando-se o modelo IV, gera cinco equações diferenciais de primeira ordem e cinco equações algébricas.



2.5 SOLUÇÃO DO PROBLEMA DE ESTABILIDADE TRANSITÓRIA

A solução do problema de estabilidade transitória consiste, basicamente, em determinar o comportamento dos rotores das máquinas síncronas frente a uma perturbação no sistema. Em outras palavras, consiste em resolver um conjunto de equações algébricas e diferenciais.

Dependendo da modelagem utilizada, as equações algébricas representam a rede de transmissão, os estatores das máquinas síncronas e as cargas. As equações diferenciais representam, normalmente, o comportamento dinâmico das partes girantes dos geradores síncronos e os efeitos da variação do campo magnético e das correntes de amortecimento, bem como os efeitos dos controles suplementares do gerador.

A seguir, passa-se a analisar algumas formas de resolução das equações algébricas e diferenciais, bem como, a maneira como elas interagem.

2.5.1 SOLUÇÃO POR MÉTODOS DIRETOS E SOLUÇÃO POR SIMULAÇÃO PASSO A PASSO AO LONGO DO TEMPO

A análise da estabilidade transitória de sistemas de energia elétrica utilizando métodos diretos é feita, de maneira geral, através do cálculo da energia transitória do sistema computada no final do período em estudo e comparada com a energia transitória limite que garante a estabilidade do sistema conforme discutido por [14] e [27]. A diferença entre estas energias é usada como indicação da estabilidade do sistema, ou seja, a análise através dos métodos diretos fornece um resultado quantitativo/qualitativo da estabilidade do sistema, indicando-se, de certa forma, o quanto o sistema é estável ou instável em relação a perturbação ocorrida.



O estudo dos métodos diretos de solução do problema de estabilidade transitória é uma área que tem estimulado o interesse de vários pesquisadores nos últimos anos, e pode ser encontrada em [14]. As desvantagens que ainda persistem na aplicação dos métodos diretos é a dificuldade de utilizar os diversos modelos que representam os vários componentes do sistema, além do fato de não serem ainda totalmente confiáveis para aplicações industriais.

Ao contrário dos métodos diretos, o estudo da estabilidade de sistemas de energia elétrica através de simulação passo a passo, consiste em resolver as equações que descrevem o comportamento do sistema no domínio do tempo, ou seja, através de simulação, obtêm-se as variáveis de interesse, como os ângulos dos rotores dos geradores síncronos. A análise da estabilidade transitória, nestes casos, é feita através de gráficos que descrevem o comportamento das variáveis de interesse em função do tempo.

O método de simulação passo a passo ao longo do tempo é um método convencional e ainda padrão utilizado na análise de sistemas de energia elétrica. Este método tem como atrativo a precisão dos resultados e a possibilidade de implementação de modelos sofisticados para descrever o comportamento dos componentes do sistema. A simulação em função do tempo foi o método adotado neste trabalho para a análise da estabilidade transitória.

2.5.2 ESQUEMAS DE SOLUÇÃO

Durante o processo de solução passo a passo, as equações diferenciais são transformadas, por meio de integração numérica, em equações algébricas. Estas transformações serão detalhadas no próximo item. Portanto, tem-se dois conjuntos de equações algébricas.

Existem basicamente duas formas de solução para as equações algébricas e diferenciais (transformadas em equações algébricas). Quando os dois conjuntos de



equações são resolvidos separadamente durante cada passo de integração, alternando-se as respectivas soluções. Este esquema de resolução é chamado alternado, como mostrado na figura 2.4. Porém, quando as equações algébricas e diferenciais são reunidas, formando-se assim um único conjunto de equações algébricas, que são resolvidas ao mesmo tempo durante cada passo de integração, tem-se o esquema de resolução simultâneo.

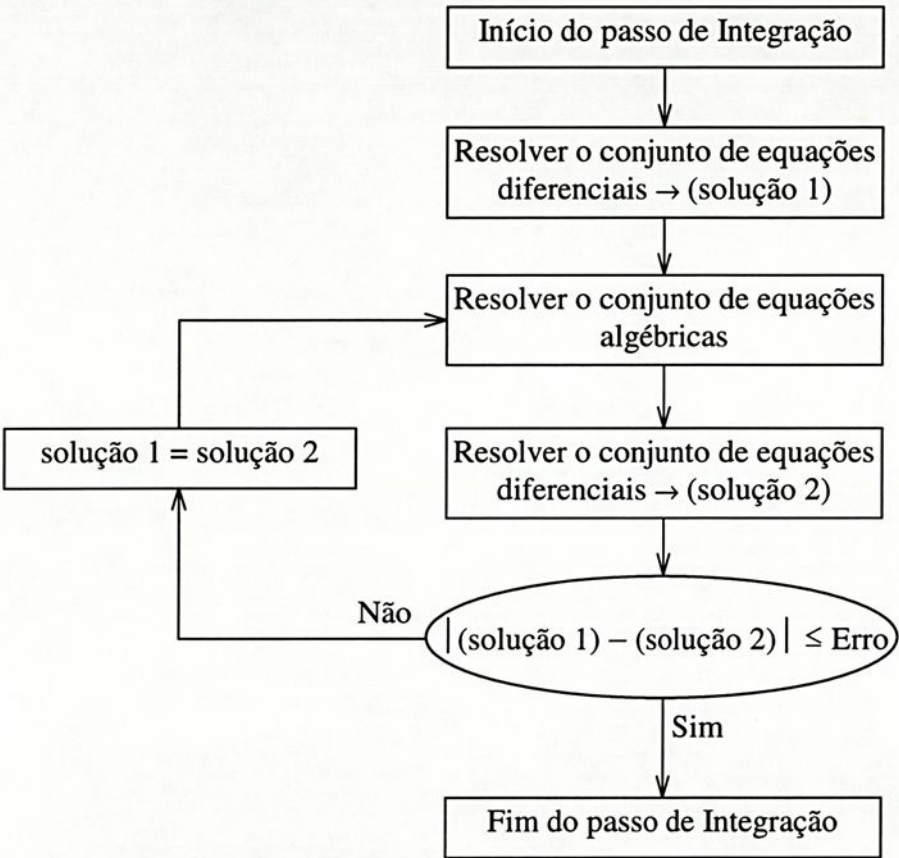


Figura 2.4 – Estrutura geral de um esquema alternado.

Observa-se pela figura 2.4 que a precisão da solução ou o critério de parada está relacionado com a convergência da resolução das equações diferenciais, ou seja, quando a diferença entre duas soluções consecutivas das equações diferenciais for menor do que uma tolerância pré-estabelecida, tem-se o final do passo de integração.

Dependendo de como as equações diferenciais são transformadas em equações algébricas, ou seja, se são utilizadas fórmulas de integração explícita como o método de



Euler simples, tem-se o esquema alternado explícito. Caso sejam utilizadas fórmulas de integração implícita, como o método trapezoidal, tem-se o esquema alternado implícito. A referência [11] aponta o método de integração trapezoidal como o mais indicado por sua estabilidade numérica e simplicidade.

No esquema alternado implícito, quando os sistemas de equações algébricas e diferenciais são resolvidos sem nenhuma atualização das variáveis comuns aos dois sistemas (variáveis de interface), durante cada passo de integração, ocorrerão erros no processo de solução. Estes erros são chamados de erros de interface, conforme discutido por [08], [22] e [24]. Este esquema de resolução assim definido, é conhecido como esquema alternado implícito normal.

Os erros de interface podem ser eliminados do esquema alternado implícito, se a resolução das equações algébricas e diferenciais forem entrelaçadas, ou seja, durante um mesmo passo de integração, faz-se uma iteração no sistema algébrico, para então fazer uma iteração no sistema diferencial. Isto faz com que ao final de um passo de integração as variáveis de interface estejam atualizadas. O esquema alternado assim apresentado é conhecido como esquema alternado implícito entrelaçado e tem como principal vantagem eliminar os erros de interface.

O principal atrativo do esquema alternado é o fato de que como as equações algébricas e diferenciais são resolvidas separadamente, as mesmas podem ser resolvidas por métodos diferentes, adaptando-se melhor as soluções para o ambiente computacional utilizado. Adicionalmente, a alteração dos modelos dos componentes do sistema, que acarreta a modificação, inclusão ou a retirada de equações de um dos conjuntos de equações, não afeta a resolução do outro. Isto confere ao esquema alternado uma grande flexibilidade de programação. Já o esquema simultâneo oferece como atrativo o fato de eliminar os erros de interface intrinsecamente, pois, neste esquema, as equações algébricas e diferenciais são reunidas em um único sistema e todas as variáveis são resolvidas ao mesmo tempo, mas o esquema simultâneo apresenta pouca versatilidade no tratamento das equações.



A referência [22] faz uma análise quantitativa entre os esquemas simultâneo e os esquemas alternado implícito normal e entrelaçado, e conclui que o último esquema é o mais recomendado para o cálculo da estabilidade transitória de sistemas de energia elétrica, motivo pelo qual foi o esquema utilizado neste trabalho.

2.5.3 RESOLUÇÃO DAS EQUAÇÕES ALGÉBRICAS E DAS EQUAÇÕES DIFERENCIAIS

Neste trabalho, foram implementadas subrotinas tradicionalmente utilizadas para a resolução das equações algébricas, e sugeriram-se alguns modelos híbridos para a resolução de tais equações, com o objetivo de otimizar o uso do processamento vetorial. Esta parte recebeu uma atenção especial e será abordada em detalhes no capítulo 3.

As equações diferenciais não oferecem maiores dificuldades para o processamento vetorial e foram resolvidas utilizando-se o método trapezoidal de integração, que pode ser entendido através do seguinte exemplo genérico.

Considere a seguinte equação:

$$\frac{\partial y}{\partial x} = f(x, y) \quad (2.6)$$

da qual se deseja obter a solução no ponto x_n a partir de sua solução conhecida no ponto x_{n-1} .

Integrando-se a equação 2.6 no intervalo x_{n-1} a x_n , obtém-se:

$$y_n - y_{n-1} = \int_{x_{n-1}}^{x_n} f(x, y) \partial x \quad (2.7)$$



Com base na figura 2.5, a integral da equação 2.7 pode ser aproximada por:

$$\int_{x_{n-1}}^{x_n} f(x, y) dx = \left[f(x_{n-1}, y_{n-1}) + f(x_n, y_n) \right] \frac{h}{2} \quad (2.8)$$

sendo: $h = x_n - x_{n-1}$.

Desta forma, a equação 2.7 torna-se:

$$y_n - y_{n-1} = \left[f(x_{n-1}, y_{n-1}) + f(x_n, y_n) \right] \frac{h}{2} \quad (2.9)$$

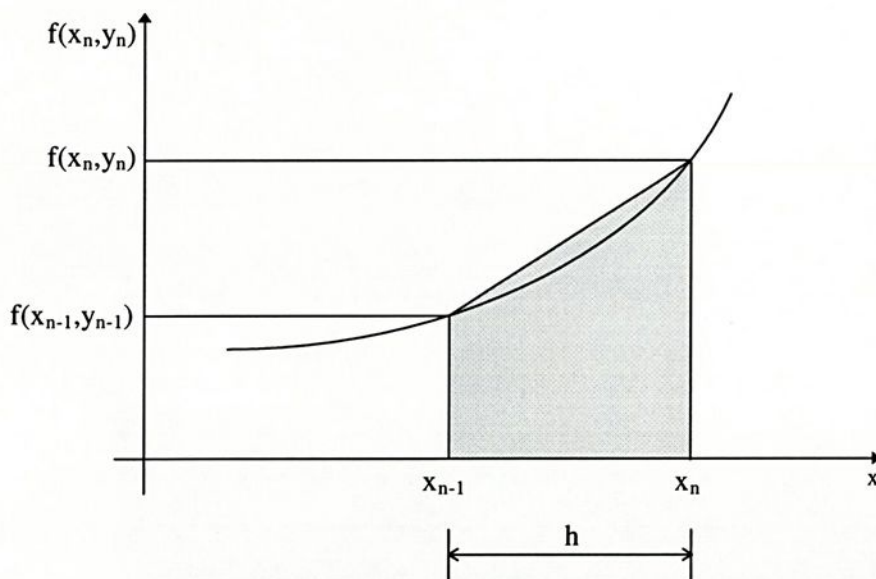


Figura 2.5 – Regra do Trapézio.

A equação 2.9, resultante da aplicação do método trapezoidal, é uma equação implícita e algébrica. Assim, com o uso do método trapezoidal um sistema de equações diferenciais pode ser transformado em um sistema algébrico.

2.5.4 SOLUÇÕES DO PROBLEMA DA ESTABILIDADE TRANSITÓRIA UTILIZANDO COMPUTADORES DE ALTO DESEMPENHO

A busca de soluções rápidas para o cálculo da estabilidade transitória tem recebido grande atenção, e várias implementações para solução deste problema têm surgido usando computadores paralelos [9, 10, 17, 18, 19]. Estas implementações têm sido realizadas em diferentes ambientes com multiprocessadores e tem-se conseguido uma solução rápida, porém, o algoritmo é concebido para uma máquina específica exigindo grandes alterações para implementá-lo em um computador diferente.

Recentemente têm surgido algumas propostas utilizando sistemas computacionais distribuídos, basicamente um grupo de heterogêneos computadores conectados através de rápidas redes de computadores [6, 25].

Estas investigações têm buscado, na verdade, realizar a análise da estabilidade transitória em tempo real. Uma diferente proposta de simulação, em tempo real, tem sido apresentada por [12] usando redes neurais, na qual se realiza *a priori* uma classificação das contingências mais severas.

Estas propostas têm levado a soluções mais rápidas e algumas destas têm como desvantagens o alto custo do sistema computacional. Atualmente, com os computadores conectados através da rede *internet*, é possível transpor esta barreira através do aluguel de horas de simulação em um computador que pode estar localizado a centenas de quilômetros da sala de simulação da empresa de energia elétrica. Este é o caminho adotado neste trabalho.



CAPÍTULO III

RESOLUÇÃO DAS EQUAÇÕES ALGÉBRICAS

3.1 INTRODUÇÃO

Neste capítulo, tratar-se-á da resolução de um conjunto de equações algébricas esparsas, resultantes da modelagem da parte estática do sistema de energia elétrica (rede de transmissão, cargas e estatores das máquinas síncronas), e que podem ser representadas na forma geral dada por:

$$A x = b \quad (3.1)$$

Serão apresentadas algumas propostas de solução usando-se métodos diretos em computadores dotados de processamento vetorial. Esta parte tem por objetivo ser uma das contribuições desta dissertação. Os melhores métodos de resolução das equações algébricas esparsas, através de processamento vetorial serão aplicados no estudo de estabilidade transitória de sistemas de energia elétrica, como descrito no capítulo IV.

Tem sido um desafio para os pesquisadores em sistemas de energia elétrica obter um bom desempenho com processamento vetorial e/ou paralelo no estágio de solução do conjunto de equações lineares esparsas. A necessidade de se obter repetidas soluções de grandes sistemas de equações ($A x = b$) e a tendência de redução de custos dos sistemas computacionais, têm estimulado muitas pesquisas neste campo [31]. Muitos trabalhos foram publicados tentando resolver este problema utilizando computadores

com processamento paralelo, entre eles: [23], [20] e [26], e outros utilizando computadores com processamento vetorial [15], [16], [03] e [32].

A solução da equação 3.1 pode ser obtida, basicamente, através do uso de métodos diretos ou métodos iterativos. Nos métodos diretos, como por exemplo, o método da decomposição LDU (vide [21]), a solução do sistema é obtida através da aplicação de transformações lineares. Nos métodos iterativos, parte-se de uma solução inicial arbitrária e através de um processo iterativo, caminha-se para a solução do sistema.

Os métodos diretos têm como atrativo, a sua confiabilidade, ou seja, a sua maior estabilidade numérica. Os métodos iterativos podem apresentar bons resultados quando da sua implementação em computadores de alto desempenho, como por exemplo, computadores dotados de processamento vetorial, pois utilizam muitas operações matriciais [05] que podem ser decompostas em tarefas independentes. O procedimento adotado neste trabalho, foi a utilização de métodos diretos, procurando-se adaptá-los para o uso em ambientes computacionais que possuam processamento vetorial, aumentando-se assim, o nível de vetorização de tais métodos (operações realizadas vetorialmente, ou seja, operações contidas num determinado laço do programa (*loop*), que possam ser realizadas ao mesmo tempo).

3.2 SOLUÇÃO UTILIZANDO SUBSTITUIÇÕES

Neste item, será descrito o processo de solução da equação 3.1, através do método direto de solução conhecido como método das substituições ou método de eliminação de Gauss, conforme tratado por [21]. Este método foi desenvolvido para computadores dotados de processamento serial, contudo o seu entendimento é primordial para que se possa criar métodos híbridos que possuam a característica de otimizar a solução, quando esta é obtida utilizando-se processamento vetorial.

De uma forma geral, a solução do sistema 3.1, utilizando-se o método de eliminação de Gauss, dá-se através de combinações lineares efetuadas sobre as equações



que compõem o sistema. Pode-se dividir o método de eliminação de Gauss em três etapas, descritas a seguir: na primeira etapa, zeram-se os elementos do triângulo inferior da matriz A ; na segunda, normaliza-se a diagonal principal e na terceira etapa, zeram-se os elementos do triângulo superior. Aplicando-se as mesmas operações sobre o vetor independente b , no final do processo, tem-se o vetor solução x .

As combinações lineares utilizadas no método de eliminação de Gauss, podem ser vistas como a premultiplicação da matriz A por matrizes elementares, mostradas de uma forma geral na figura 3.1.

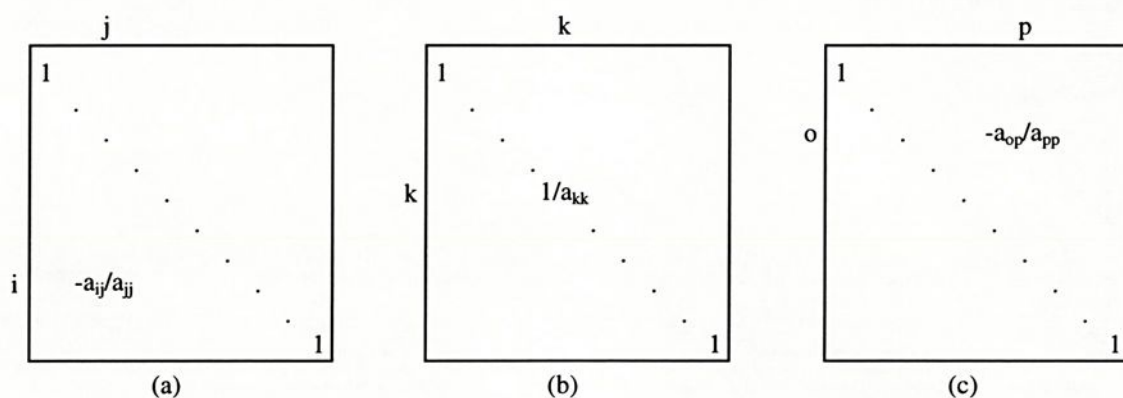


Figura 3.1 – (a) matriz elementar para eliminar um elemento do triângulo inferior, (b) matriz elementar para normalizar um elemento da diagonal e (c) matriz elementar para eliminar um elemento do triângulo superior.

Pode-se verificar que premultiplicando-se a matriz A pela matriz elementar da figura 3.1(a), torna-se zero o elemento que está na linha i e na coluna j da matriz A . Repetindo-se este processo, utilizando-se a matriz elementar da figura 3.1(b), normaliza-se o elemento que está na linha k e na coluna k da matriz A . Premultiplicando-se, agora, a matriz A pela matriz elementar da figura 3.1(c), zera-se o elemento que está na linha o e na coluna p da matriz A . Desta forma, mostra-se como o processo de eliminação de Gauss pode ser abordado como a premultiplicação da matriz

A por uma sequência de matrizes elementares. No final deste processo, a matriz A é transformada na matriz identidade I .

Para obter o vetor solução x , deve-se premultiplicar o vetor b com as mesmas matrizes elementares utilizadas sobre a matriz A , como mostram as equações 3.2 e 3.3.

$$T_q (T_{q-1} (... (T_3 (T_2 (T_1 A)))))) = I \quad (3.2)$$

$$T_q (T_{q-1} (... (T_3 (T_2 (T_1 b)))))) = x \quad (3.3)$$

sendo: T_i = matriz elementar genérica;

I = matriz identidade;

q = número de matrizes elementares necessárias para transformar a matriz A na matriz identidade I .

Analisando-se a equação 3.2, pode-se escrever a inversa da matriz A como:

$$A^{-1} = T_q T_{q-1} ... T_3 T_2 T_1 \quad (3.4)$$

Portanto, a matriz A pode ser expressa da seguinte forma:

$$A = T_1^{-1} T_2^{-1} T_3^{-1} ... T_{q-1}^{-1} T_q^{-1} \quad (3.5)$$

A diferença entre a matriz elementar T_i e a sua inversa T_i^{-1} é apenas o sinal contrário do elemento fora da diagonal quando se tratar de matrizes elementares do tipo apresentado nas figuras 3.1(a) e 3.1(c). Quando a matriz elementar tiver a forma apresentada na figura 3.1(b), a sua inversa é obtida pela simples inversão do elemento diferente do valor unitário da diagonal. Portanto, o cálculo das matrizes elementares inversas se faz de forma imediata.

A equação 3.5 pode ser reescrita como:

$$A = [L_1 L_2 ... L_r] [D_1 D_2 ... D_n] [U_1 U_2 ... U_s] \quad (3.6)$$

sendo: r = número de matrizes elementares necessárias para zerar o triângulo inferior da matriz A ;



n = ordem da matriz A ;

s = número de matrizes elementares necessárias para zerar o triângulo superior da matriz A . Se a matriz A for simétrica, tem-se: $r = s$;

L_i , D_i e U_i são o inverso das matrizes elementares que zeram o triângulo inferior, normalizam a diagonal e zeram o triângulo superior da matriz A , respectivamente. Como visto anteriormente, elas preservam a forma das matrizes elementares.

Sejam L_i e L_j duas matrizes elementares genéricas. Considere que no processo de eliminação de Gauss L_i zerou um coeficiente de A antes de L_j , portanto, realizando-se o produto $L_i \times L_j$, necessariamente, nesta ordem, pode-se verificar que:

$$L_i \times L_j = L_i + L_j - I \quad (3.7)$$

A equação 3.7 indica que o produto $L_i \times L_j$ é dado pela superposição das matrizes elementares L_i e L_j . Pode-se estender este raciocínio e concluir que a multiplicação das matrizes elementares L_i , obedecendo-se a ordem em que aparecem na equação 3.6, origina uma matriz triangular inferior, chamada matriz L , obtida pela superposição de todas as matrizes elementares L_i .

De forma análoga, pode-se concluir que a multiplicação das matrizes elementares D_i e U_i , na mesma ordem em que aparecem na equação 3.6, origina uma matriz diagonal D e uma matriz triangular superior U , dadas pela superposição das matrizes elementares D_i e U_i , respectivamente.

Portanto, pode-se representar a matriz A como:

$$A = L D U \quad (3.8)$$

Esta metodologia que decompõem a matriz A em três matrizes elementares é conhecida como decomposição LDU. Portanto, a equação 3.1 pode ser reescrita da seguinte forma:



$$L D U x = b \quad (3.9)$$

O processo de solução é tradicionalmente dividido em três etapas, como descrito a seguir:

$$L z = b \quad \text{SF (Substituição Forward)} \quad (3.10)$$

$$D y = z \quad \text{SD (Substituição Diagonal)} \quad (3.11)$$

$$U x = y \quad \text{SB (Substituição Backward)} \quad (3.12)$$

No processo tradicional de solução, as substituições são realizadas uma a uma, na mesma ordem em que aparecem na decomposição da matriz A , sobre o vetor b e no final do processo tem-se o valor do vetor solução x .

Em sistemas de energia elétrica, a matriz A é do tipo admitância e apresenta um grau de esparsidade muitas vezes superior a 98%. Assim, a utilização de técnicas de esparsidade para solução de problemas do tipo $A x = b$ ganha muita importância por trazer vantagens de armazenamento de dados e velocidade de cálculos. Estas técnicas ganharam importância a partir do trabalho de [30].

A referência [35] apresenta uma forma de armazenar somente os elementos não nulos da matriz A em vetores apontadores, bem como um algoritmo para resolver o sistema 3.1 utilizando estes vetores. Cabe lembrar que o uso dos vetores apontadores acarreta economia de espaço de memória e que as operações relativas ao processo de solução são desenvolvidas utilizando-se os próprios vetores apontadores, sendo os mesmos definidos a seguir:

CE – Armazena os elementos não nulos da matriz A e os elementos adicionais (elementos *fill-ins*) que aparecerão no decorrer do processo de eliminação de Gauss;

LCOL – Contém na posição i a informação de onde se encontra o primeiro elemento não nulo da coluna i no vetor CE;



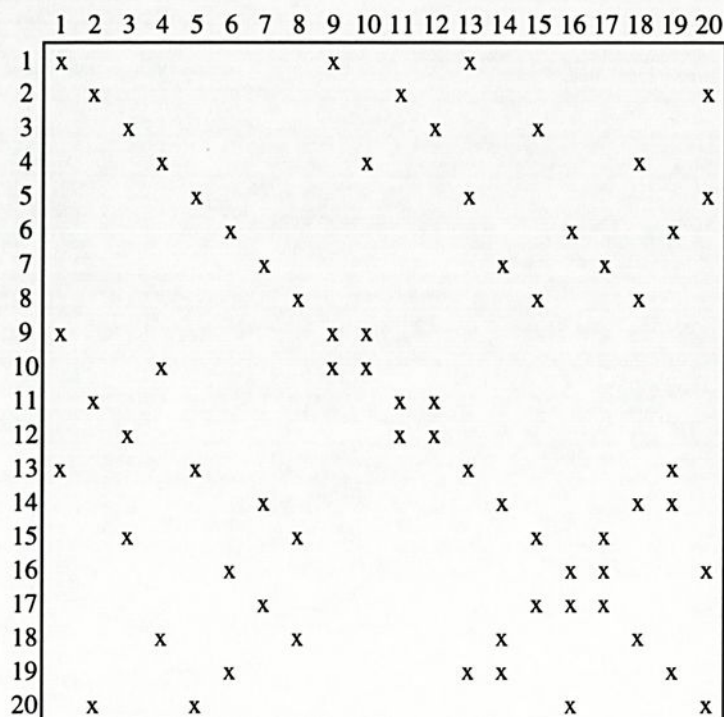


Figura 3.3 – Matriz A.

Resolver o sistema de equações lineares esparsas dadas pela equação 3.1, empregando-se técnicas de substituição ou decomposição LDU, envolve três etapas principais: *ordenação* – que corresponde a obtenção da ordem de pivoteamento dos nós da rede; *fatoração* – que consiste em obter os fatores triangulares através da eliminação de Gauss; e *solução direta* – que compreende na obtenção do vetor solução.

Tomando-se, por exemplo, o sistema apresentado na figura 3.2, tem-se a matriz L mostrada na figura 3.4, resultado da fatoração da matriz A . A ordem em que aparecem os nós na figura 3.4 será melhor detalhada, quando se tratar da etapa de ordenação dos nós da rede.

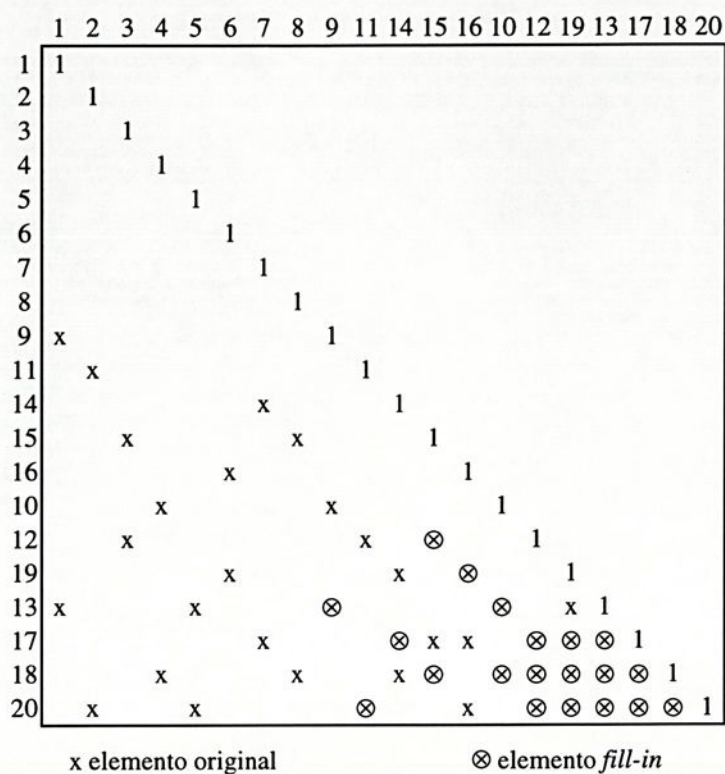


Figura 3.4 – Matriz L .

Na tabela 3.1, a seguir, são mostradas as operações de substituições da primeira etapa do processo de solução (SF), para a matriz da figura 3.4. A coluna “tarefa”, indica a seqüência em que devem ser feitas as substituições.

Tabela 3.1 – Operações na SF utilizando decomposição LDU.

Tarefa	Operação	Tarefa	Operação
1	$b_9 = b_9 - L_{9,1} \cdot b_1$	23	$b_{18} = b_{18} - L_{18,14} \cdot b_{14}$
2	$b_{13} = b_{13} - L_{13,1} \cdot b_1$	24	$b_{12} = b_{12} - L_{12,15} \cdot b_{15}$
3	$b_{11} = b_{11} - L_{11,2} \cdot b_2$	25	$b_{17} = b_{17} - L_{17,15} \cdot b_{15}$
4	$b_{20} = b_{20} - L_{20,2} \cdot b_2$	26	$b_{18} = b_{18} - L_{18,15} \cdot b_{15}$
5	$b_{15} = b_{15} - L_{15,3} \cdot b_3$	27	$b_{19} = b_{19} - L_{19,16} \cdot b_{16}$
6	$b_{12} = b_{12} - L_{12,3} \cdot b_3$	28	$b_{17} = b_{17} - L_{17,16} \cdot b_{16}$
7	$b_{10} = b_{10} - L_{10,4} \cdot b_4$	29	$b_{20} = b_{20} - L_{20,16} \cdot b_{16}$
8	$b_{18} = b_{18} - L_{18,4} \cdot b_4$	30	$b_{13} = b_{13} - L_{13,10} \cdot b_{10}$
9	$b_{13} = b_{13} - L_{13,5} \cdot b_5$	31	$b_{18} = b_{18} - L_{18,10} \cdot b_{10}$
10	$b_{20} = b_{20} - L_{20,5} \cdot b_5$	32	$b_{17} = b_{17} - L_{17,12} \cdot b_{12}$
11	$b_{16} = b_{16} - L_{16,6} \cdot b_6$	33	$b_{18} = b_{18} - L_{18,12} \cdot b_{12}$
12	$b_{19} = b_{19} - L_{19,6} \cdot b_6$	34	$b_{20} = b_{20} - L_{20,12} \cdot b_{12}$
13	$b_{14} = b_{14} - L_{14,7} \cdot b_7$	35	$b_{13} = b_{13} - L_{13,19} \cdot b_{19}$
14	$b_{17} = b_{17} - L_{17,7} \cdot b_7$	36	$b_{17} = b_{17} - L_{17,19} \cdot b_{19}$
15	$b_{15} = b_{15} - L_{15,8} \cdot b_8$	37	$b_{18} = b_{18} - L_{18,19} \cdot b_{19}$
16	$b_{18} = b_{18} - L_{18,8} \cdot b_8$	38	$b_{20} = b_{20} - L_{20,19} \cdot b_{19}$
17	$b_{10} = b_{10} - L_{10,9} \cdot b_9$	39	$b_{17} = b_{17} - L_{17,13} \cdot b_{13}$
18	$b_{13} = b_{13} - L_{13,9} \cdot b_9$	40	$b_{18} = b_{18} - L_{18,13} \cdot b_{13}$
19	$b_{12} = b_{12} - L_{12,11} \cdot b_{11}$	41	$b_{20} = b_{20} - L_{20,13} \cdot b_{13}$
20	$b_{20} = b_{20} - L_{20,11} \cdot b_{11}$	42	$b_{18} = b_{18} - L_{18,17} \cdot b_{17}$
21	$b_{19} = b_{19} - L_{19,14} \cdot b_{14}$	43	$b_{20} = b_{20} - L_{20,17} \cdot b_{17}$
22	$b_{17} = b_{17} - L_{17,14} \cdot b_{14}$	44	$b_{20} = b_{20} - L_{20,18} \cdot b_{18}$

Nota-se que durante o processo tradicional de solução, como apresentado acima, as operações de adição e multiplicação são realizadas uma a uma, começando pela tarefa 1 e finalizando na tarefa de número 44, ou seja, a solução como foi apresentada é essencialmente serial. Por isso, para que se possa explorar o desempenho do processamento vetorial, deve-se verificar a possibilidade de agrupar tarefas que possam ser realizadas ao mesmo tempo.

3.3 GRAFO DOS CAMINHOS DE FATORAÇÃO (GCF)

O GCF [29] é uma ferramenta importante que permite, entre outros aspectos, determinar uma seqüência de operações que não necessita ser realizada serialmente durante o processo de solução, portanto, o uso do GCF é muito útil quando se deseja adaptar subrotinas para serem utilizadas em computadores dotados de processamento paralelo ou vetorial.

Através do GCF se estabelece a interdependência na ordem de fatoração das colunas da matriz L durante a eliminação de Gauss. O GCF é obtido, para o caso de uma matriz simétrica, através da matriz L , ilustrada na figura 3.4, onde escolhe-se uma coluna i qualquer e encontra-se a posição da linha do primeiro elemento diferente de zero abaixo da diagonal; esta posição refere-se a coluna j que será imediatamente alterada, após a fatoração da coluna i . Repete-se este processo, para todos os nós, até o último nó da matriz L (nó raiz do GCF).

Para a matriz L da figura 3.4, obtém-se o GCF como mostrado na figura 3.5.

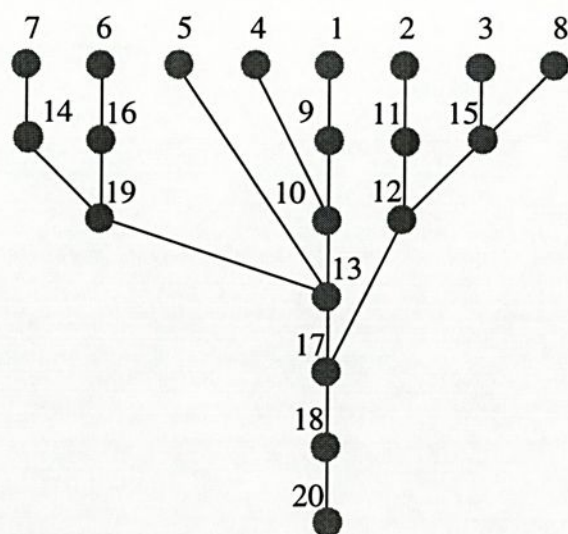


Figura 3.5 – Grafo dos Caminhos de Fatoração.



A profundidade de cada nó no GCF é o número de nós que o antecede no grafo. Para programar o GCF é necessário implementar apenas o vetor NEXT, definido a seguir:

NEXT – O valor de NEXT(i) informa o nó que está imediatamente abaixo do nó i no GCF. Quando NEXT(i) for nulo, significa que o nó i é o último nó do GCF.

3.4 ORDENAÇÕES

Ordenar os nós de uma rede de energia elétrica corresponde a ordenar as linhas/colunas da matriz resultante da modelagem do sistema. Os critérios de ordenação foram criados, visando diminuir o aparecimento de novos elementos diferentes de zero durante a fatoração da matriz A (*fill-ins*).

Inicialmente, foram propostos três esquemas de ordenação [30]:

- 1) **Mínimo grau estático** [Tinney 1]. Os nós são ordenados de acordo com o menor grau (número de nós vizinhos) da ligação original da rede. Nós vizinhos são aqueles conectados diretamente ao nó em questão.
- 2) **Mínimo grau dinâmico** [Tinney 2]. Aplica-se o esquema 1 para o primeiro nó a ser ordenado. Após a ordenação de um nó altera-se a configuração da rede supondo sua eliminação, atualiza-se o grau dos nós vizinhos, e volta-se a aplicar o esquema 1.
- 3) **Mínimo número de *fill-ins*** [Tinney 3]. Em cada estágio de ordenação simula-se a eliminação de todos os nós. A configuração que gerar menos *fill-ins* será a escolhida.

O esquema 2 tem sido o preferido por sua simplicidade e pelo reduzido número de *fill-ins* gerados, número não muito diferente do gerado pelo esquema 3, que exige muito mais processamento. Já o esquema 1 gera muito mais *fill-ins*, sendo por isso pouco útil.



Recentemente outros critérios de ordenação foram propostos buscando explorar certas características dos sistemas, a fim de otimizar o uso do processamento paralelo e/ou vetorial e para otimizar o uso dos vetores esparsos. Alguns destes critérios serão detalhados nos itens seguintes.

3.4.1 ORDENAÇÃO *MINIMUM DEGREE* (MD)

O esquema Tinney 2 de ordenação, também conhecido como ordenação MD (*Minimum Degree*), baseia-se no grau de cada nó do sistema atualizado em cada passo da fatoração da matriz A . Para implementar a ordenação MD é necessário criar um vetor NOZE, como definido a seguir:

NOZE – indica o grau (número de nós vizinhos) de cada nó na rede.

A ordenação MD pode ser melhor entendida pelo seguinte algoritmo:

ALGORITMO MD

- 1) Faça $k=1$;
- 2) Escolha o nó i com $\text{noze}(i)$ mínimo. Faça $\text{nseq}(k) = i$;
- 3) Se $k = \text{nb}$, fim (nb é o número de nós da rede);
- 4) Para cada nó j vizinho de i , faça $\text{noze}(j) = \text{noze}(j) - 1$;
- 5) Para cada par de nós (m,n) vizinhos de i , mas não vizinhos um do outro, crie uma nova ligação entre m e n . Faça $\text{noze}(m) = \text{noze}(m) + 1$ e $\text{noze}(n) = \text{noze}(n) + 1$;
- 6) Faça $k = k + 1$ e volte ao passo 2.



3.4.2 ORDENAÇÃO *MINIMUM LENGTH* (ML) [04]

Esta ordenação tem por objetivo reduzir a profundidade do GCF, ou seja, o critério de escolha de cada nó durante o processo de fatoração leva ao estabelecimento de um CGF com profundidade reduzida, o que constitui uma característica desejável para o processamento paralelo e/ou vetorial. Para implementar a ordenação ML, é necessário criar o vetor NPROF. Este vetor contém a profundidade de cada nó no GCF, sendo iniciado com valor nulo.

A seguir é apresentado o algoritmo da ordenação ML.

ALGORITMO ML

- 1) Faça $k=1$;
- 2) Escolha um nó i com $nprof(i)$ mínimo. Faça $nseq(k) = i$;
- 3) Se $k = nb$, fim;
- 4) Para cada nó j vizinho de i com $nprof(j) \leq nprof(i)$, faça $nprof(j) = nprof(i) + 1$;
- 5) Para cada par de nós (m,n) vizinhos de i , mas não vizinhos um do outro, crie uma nova ligação entre m e n ;
- 6) Faça $k = k + 1$ e volte ao passo 2.

3.4.3 ORDENAÇÃO *MINIMUM DEGREE MINIMUM LENGTH* (MDML) [04]

Nesta ordenação, aplica-se o critério da mínima profundidade do GCF para escolher qual o nó a ser fatorado quando há empate no critério de mínimo grau.

A ordenação MDML pode ser melhor entendida pelo algoritmo a seguir:



ALGORITMO MDML

- 1) Faça $k = 1$;
- 2) Escolha um nó i com $n\text{grau}(i)$ mínimo. Caso mais de um nó satisfazer esta condição, opte pelo nó com $n\text{prof}(i)$ menor. Faça $n\text{seq}(k) = i$;
- 3) Se $k = nb$, fim;
- 4) Para cada nó j vizinho de i faça $\text{noze}(j) = \text{noze}(j) - 1$ e se $n\text{prof}(j) \leq n\text{prof}(i)$, então faça $n\text{prof}(j) = n\text{prof}(i) + 1$;
- 5) Para cada par de nós (m,n) vizinhos de i , mas não vizinhos um do outro, crie uma nova ligação entre m e n . Faça $\text{noze}(m) = \text{noze}(m) + 1$ e $\text{noze}(n) = \text{noze}(n) + 1$;
- 6) Faça $k = k + 1$ e volte ao passo 2.

3.4.4 ORDENAÇÃO *MINIMUM LENGTH MINIMUM DEGREE* (MLMD) [04]

A diferença entre esta ordenação e a anterior está na ordem dos critérios de escolha do nó que será fatorado. Ou seja, aplica-se o critério de mínimo grau quando há empate no critério de mínima profundidade, como visto no algoritmo a seguir:

ALGORITMO MLMD

- 1) Faça $k = 1$;
- 2) Escolha um nó i com $n\text{prof}(i)$ mínimo. Caso mais de um nó satisfazer esta condição, opte pelo nó com $\text{noze}(i)$ menor. Faça $n\text{seq}(k) = i$;
- 3) Se $k = nb$, fim;
- 4) Para cada nó j vizinho de i faça $\text{noze}(j) = \text{noze}(j) - 1$ e se $n\text{prof}(j) \leq n\text{prof}(i)$, então faça $n\text{prof}(j) = n\text{prof}(i) + 1$;
- 5) Para cada par de nós (m,n) vizinhos de i , mas não vizinhos um do outro, crie uma nova ligação entre m e n . Faça $\text{noze}(m) = \text{noze}(m) + 1$ e $\text{noze}(n) = \text{noze}(n) + 1$;
- 6) Faça $k = k + 1$ e volte ao passo 2.



Durante o processo de solução das equações algébricas em computadores dotados de processamento paralelo e/ou vetorial é desejável que se trabalhe com poucos elementos *fill-ins* e que o GCF possua profundidade reduzida.

A ordenação MD gera poucos elementos *fill-ins*, mas o GCF resultante possui uma profundidade grande. Já a ordenação ML possui um comportamento inverso, pois ela gera um GCF com profundidade reduzida, porém, gera muitos *fill-ins*. Isto acontece porque as duas ordenações possuem apenas um critério de escolha dos nós, ou seja, ou se considera o número de *fill-ins* gerado (ordenação MD), ou se considera a profundidade do GCF (ordenação ML).

Por isso, a ordenação que apresenta um melhor desempenho, quando da utilização do processamento paralelo e/ou vetorial, é aquela que utiliza os critérios do menor número de *fill-ins* e a menor profundidade no GCF. Em outras palavras, as ordenações MDML e MLMD são as ordenações que se destacam, motivo pelo qual elas serão utilizadas para a obtenção dos resultados.

3.5 SOLUÇÃO UTILIZANDO O MÉTODO DA MATRIZ W

A solução da equação 3.1, utilizando-se o método das substituições, como descrito na seção 3.2, é obtida pela aplicação de operações que possuem uma determinada ordem de precedência, ou seja, o processo de solução, assim definido, é um processo serial.

Quando se utiliza um computador que possua processamento paralelo e/ou vetorial, deve-se adaptar os algoritmos de solução para que sua utilização seja otimizada em tais ambientes. A referência [13] introduziu o método da matriz W , que tem por objetivo eliminar e/ou reduzir as relações de precedência das operações elementares (multiplicação e adição) durante o processo de solução.

A principal idéia do emprego do método da matriz W (também conhecido como método dos fatores inversos), é utilizar operações matriciais. A matriz W é obtida



invertendo-se a matriz L ; de forma análoga, a matriz W^U é resultado da inversão da matriz U , como mostram as equações 3.13 e 3.14:

$$W = L^{-1} \quad (3.13)$$

$$W^U = U^{-1} \quad (3.14)$$

Portanto, o processo de solução da equação 3.1, utilizando-se o método da matriz W , é dado pelas três equações abaixo:

$$z = W b \quad (3.15)$$

$$y = D^{-1} z \quad (3.16)$$

$$x = W^U y \quad (3.17)$$

3.5.1 PARTICIONAMENTO DA MATRIZ W

Analisando-se as equações 3.15, 3.16 e 3.17, verifica-se que a solução é obtida através de operações matriciais, o que favorece o desempenho do processamento vetorial. Porém, obter a matriz W simplesmente pela inversão da matriz L , como mostrado na equação 3.13, gera elevado número de novos elementos adicionais (*W-fills*), fazendo-se com que a matriz W seja bem menos esparsa do que a matriz L . Este fato pode tornar o uso dos fatores inversos proibitivo, uma vez que assim tem-se de trabalhar com muito mais elementos.

Com o propósito de evitar o aparecimento de elevado número de *W-fills*, a referência [01] propôs três algoritmos que particionam a matriz W , como descritos a seguir:

- **Algoritmo PA1.** Faz-se o particionamento agrupando-se nós que possuam a mesma profundidade no GCF. Nenhuma operação aritmética extra é necessária e nenhum novo elemento é gerado.



- **Algoritmo PA2.** Usa o princípio do algoritmo PA1 mais uma função para avaliar nós de profundidades diferentes que poderiam ser agrupados sem gerar novos elementos. Nenhum elemento adicional é gerado, mas são necessárias algumas operações de multiplicação e adição para se obter as partições.
- **Algoritmo PA3.** Usa todos os critérios do algoritmo PA2 mais uma função que permite a geração de uma porcentagem preestabelecida de novos elementos. Dependendo desta porcentagem, muitos cálculos adicionais são necessários.

O algoritmo de particionamento utilizado neste trabalho foi o algoritmo PA1, por sua simplicidade de programação e por gerar as primeiras partições com vários elementos. Os algoritmos PA2 e PA3 exigem cálculos adicionais com o objetivo de aumentar o tamanho de todas as partições. Contudo, para o processamento vetorial este aumento é de fundamental importância para as últimas partições, pois estas contêm poucos elementos a serem processados.

Utilizando-se a matriz L da figura 3.4, gerou-se a matriz W particionada, como ilustra a figura 3.6.

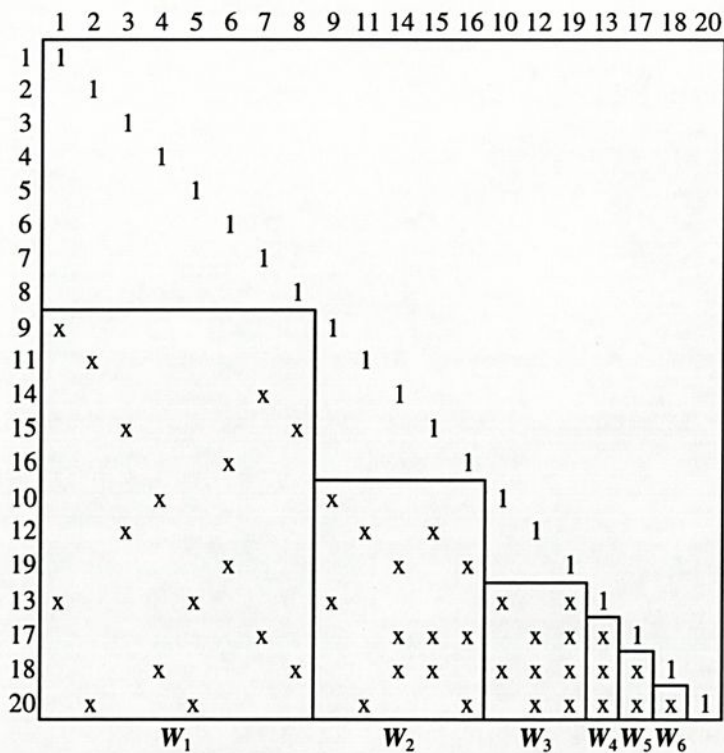


Figura 3.6 – Matriz W particionada segundo o algoritmo PA1.



A diferença entre a matriz W ilustrada na figura 3.6 e a matriz L ilustrada na figura 3.4 é apenas o sinal dos elementos não-nulos fora da diagonal. Portanto, nenhuma operação de ponto flutuante extra precisa ser realizada. A matriz W é formada através de multiplicações de matrizes elementares W_i , que por sua vez são o inverso das matrizes elementares L_i , e que se diferenciam apenas pelo sinal do elemento não nulo abaixo da diagonal principal.

Para melhor entendimento do algoritmo de particionamento PA1, sem perda de generalidade, considere que na formação da partição W_1 , realizou-se o produto $W_i \times W_j$, sendo ambas as matrizes pertencentes à primeira partição. Tais matrizes podem ser representadas de uma maneira genérica da seguinte forma:

$$W_i = \begin{array}{cc} & \begin{array}{cc} \text{nz} & \text{nr} \end{array} \\ \begin{array}{c} \text{nz} \\ \text{nr} \end{array} & \begin{array}{|cc|} \hline A_i & B_i \\ \hline C_i & D_i \\ \hline \end{array} \end{array}$$

sendo:

- nz = número de nós na profundidade zero do GCF;
- nr = número de nós restantes, ou seja, nós que não estão na profundidade zero;
- $A_i = I_{\text{nz}}$ = matriz identidade ($\text{nz} \times \text{nz}$);
- $B_i = 0$ = matriz nula;
- C_i = matriz que contém o elemento não nulo, fora da diagonal, da matriz W_i ;
- $D_i = I_{\text{nr}}$ = matriz identidade ($\text{nr} \times \text{nr}$).



Efetuada-se o produto $W_i \times W_j$, tem-se:

$$\begin{array}{|c|c|} \hline A & B \\ \hline C & D \\ \hline \end{array} = \begin{array}{|c|c|} \hline A_i & B_i \\ \hline C_i & D_i \\ \hline \end{array} \times \begin{array}{|c|c|} \hline A_j & B_j \\ \hline C_j & D_j \\ \hline \end{array}$$

$W_i \times W_j \qquad \qquad W_i \qquad \qquad W_j$

sendo:

$$A = A_i \times A_j + B_i \times C_j = Inz \times Inz + 0 \times C_j = Inz$$

$$B = A_i \times B_j + B_i \times D_j = Inz \times 0 + 0 \times Inr = 0$$

$$C = C_i \times A_j + D_i \times C_j = C_i \times Inz + Inr \times C_j = C_i + C_j$$

$$D = C_i \times B_j + D_i \times D_j = C_i \times 0 + Inr \times Inr = Inr$$

Portanto, o produto $W_i \times W_j$ (ambas matrizes elementares pertencentes a primeira partição) não gera elemento adicional, e a matriz resultante é a sobreposição das matrizes elementares. Este resultado pode ser estendido para as demais partições, e conclui-se que particionando a matriz W , segundo o algoritmo PA1, nenhum trabalho adicional é requerido para a formação das partições.

3.5.2 PROCESSAMENTO VETORIAL USANDO O MÉTODO DA MATRIZ W

A metodologia da matriz W foi inicialmente proposta para utilização em computadores dotados de processamento paralelo. A referência [15] aplicou o método



da matriz W para a resolução do problema de fluxo de potência em computadores com processamento vetorial, obtendo bons resultados. Naquela referência, é abordado o problema da recorrência ou dependência de dados durante a vetorização e uma forma de armazenar a matriz W adaptada ao processamento vetorial.

Para que o processamento vetorial apresente um desempenho satisfatório, é necessário armazenar os elementos das partições da matriz W de forma a obter vetores longos. Portanto, deve-se armazenar os elementos da matriz W de uma forma diferente daquela como foram armazenados os elementos da matriz L . Como visto anteriormente, os elementos da matriz L foram armazenados através de vetores apontadores que indicavam a seqüência em que deveriam ser feitas as operações de substituição. Este tipo de armazenamento não é atrativo quando se trabalha com processamento vetorial, pois leva a um processo de solução serial.

A fim de obter melhores resultados com o processamento vetorial é necessário agrupar todas as operações que possam ser realizadas simultaneamente em um vetor. Duas formas possíveis de agrupamento dos elementos das partições são: por pseudocolunas como ilustra a figura 3.7 e por pseudolinhas como ilustra a figura 3.8.



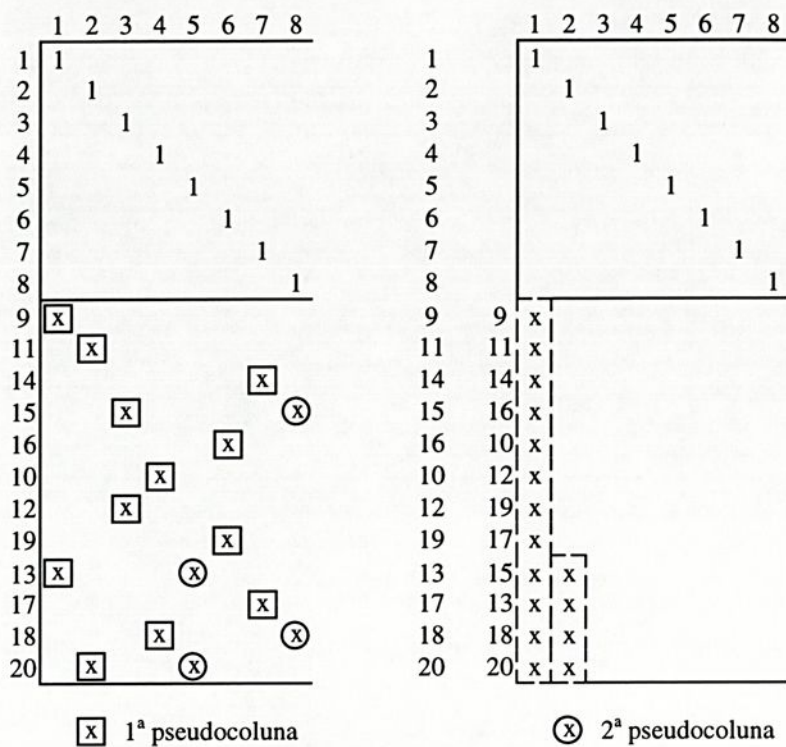


Figura 3.7 – Pseudocolunas.

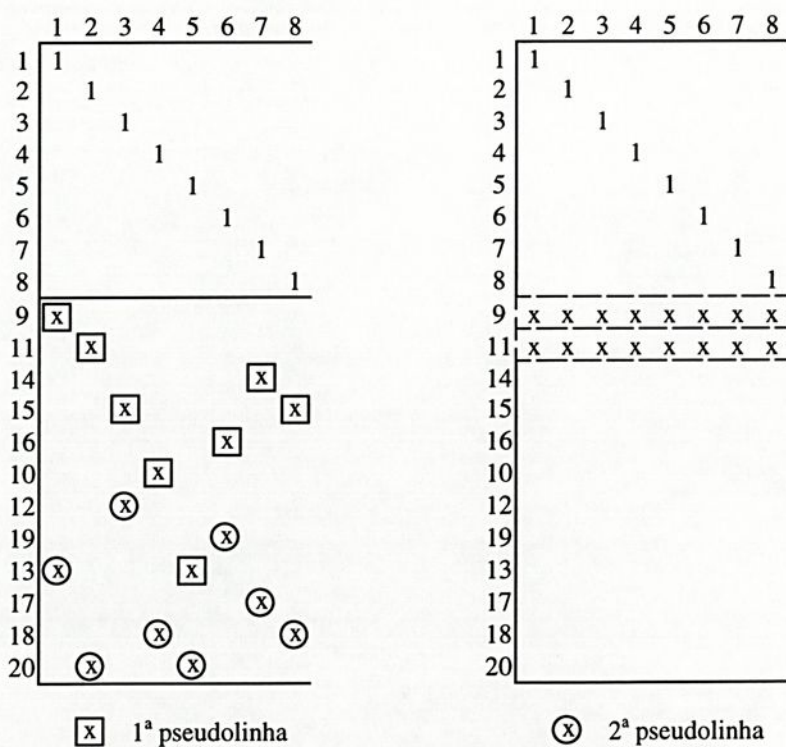


Figura 3.8 – Pseudolinhas.

As operações realizadas no processo de solução, utilizando o método da matriz W com os elementos armazenados por pseudocolunas, são operações independentes tanto na etapa SF como na SB e por isso podem ser realizadas todas de uma só vez através do processamento vetorial. Já as operações com os elementos armazenados por pseudolinhas apresentam problemas de dependência de dados e não podem ser realizadas todas de uma vez, como mostram as tarefas 3 e 8 da tabela 3.2, que especifica as operações elementares realizadas no processo de solução, quando do processamento dos elementos armazenados na primeira pseudolinha (figura 3.8).

Tabela 3.2 – Operações na SF com os elementos da primeira pseudolinha.

Tarefa	Operação	Tarefa	Operação
1	$b_9 = b_9 + W_{9,1} \cdot b_1$	5	$b_{13} = b_{13} + W_{13,5} \cdot b_5$
2	$b_{11} = b_{11} + W_{11,2} \cdot b_2$	6	$b_{16} = b_{16} + W_{16,6} \cdot b_6$
3	$b_{15} = b_{15} + W_{15,3} \cdot b_3$	7	$b_{14} = b_{14} + W_{14,7} \cdot b_7$
4	$b_{10} = b_{10} + W_{10,4} \cdot b_4$	8	$b_{15} = b_{15} + W_{15,8} \cdot b_8$

Logo, a solução do problema, utilizando-se o método da matriz W adaptada para o processamento vetorial, dar-se-á pelo particionamento através do algoritmo PA1, armazenando-se os elementos das partições através de pseudocolunas, como mostrado na figura 3.9.



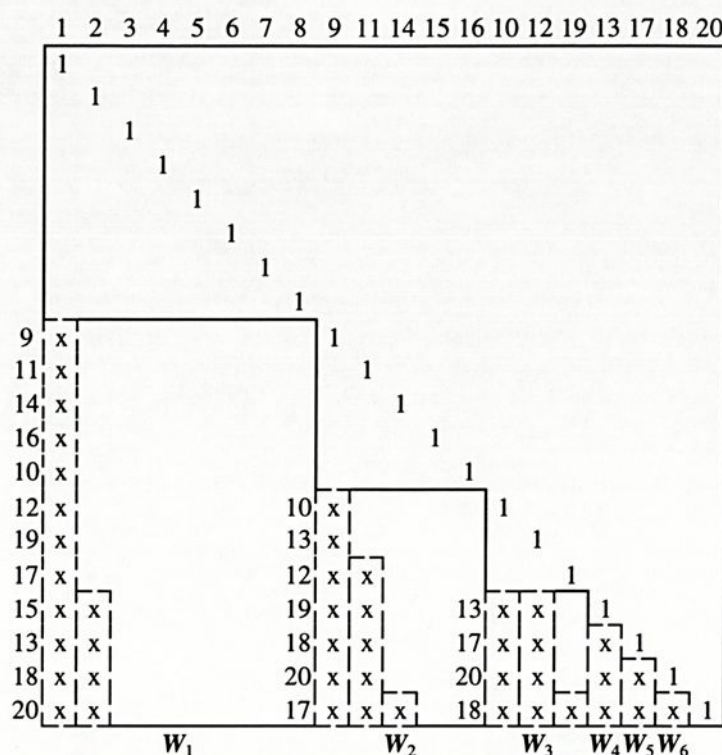


Figura 3.9 – Matriz W particionada segundo o algoritmo PA1 com os elementos armazenados através de pseudocolunas.

Esta técnica apresenta um bom desempenho vetorial para as primeiras partições, pois estas são partições grandes e, desta forma, obtêm-se vetores (pseudocolunas) longos. À medida que as partições tornam-se pequenas, verifica-se que o desempenho do processamento vetorial diminui.

Pensando-se em obter melhorias, foram testadas várias metodologias. Algumas destas metodologias foram introduzidas anteriormente e algumas delas são propostas deste trabalho. Convencionou-se identificar estas metodologias através da combinação de três letras, sendo a primeira relacionada com o método de resolução adotado, a segunda letra, com os vetores x e b e a terceira letra está relacionada com a forma como foram processadas as operações referentes ao últimos nós do GCF.

3.5.3 MATRIZ W COM VETORES CHEIOS

Considera-se, inicialmente, o caso em que os vetores x e b são vetores cheios, ou seja, todas as posições do vetor b são diferentes de zero e deseja-se conhecer o valor de todas os componentes do vetor x .

Metodologia WCN. Esta metodologia foi proposta por [15]. Ela utiliza o método da matriz W , particionada segundo o algoritmo PA1 e os elementos das partições são armazenados através de pseudocolunas. Em cada partição são agrupados nós pertencentes à mesma profundidade no GCF, desde os nós folhas até o nó raiz.

O processo de solução é então dado pela equação 3.18, como mostrado a seguir:

$$x = W_1^U \cdot W_2^U \dots W_n^U \cdot D^{-1} \cdot W_n \dots W_2 \cdot W_1 \cdot b \quad (3.18)$$

sendo que W_1 , W_2 e W_n representam, respectivamente, a primeira partição (nós folhas do GCF), a segunda partição e a última partição (nó raiz do GCF) da matriz W .

Para melhor entendimento da metodologia WCN, é apresentado, a seguir, um algoritmo que descreve, seqüencialmente, as etapas do processo de solução, ou seja, a substituição *forward* – SF, a substituição *diagonal* – SD e a substituição *backward* – SB. Estas etapas são agora realizadas com os elementos da matriz W armazenados em pseudocolunas.



ALGORITMO WCN

do $i = 1$, número de pseudocolunas da matriz W

do $j = \{\text{elementos da pseudocoluna } i\}$

$$b(ilj) = b(ilj) + W(ilj, icj) \times b(icj)$$

end do

end do

do $j = \{\text{elementos da matriz } D\}$

$$b(ilj) = b(ilj) \times D(ilj, ilj)$$

end do

do $i = \text{número de pseudocolunas da matriz } W^U, 1$

do $j = \{\text{elementos da pseudocoluna } i\}$

$$b(ilj) = b(ilj) + W^U(ilj, icj) \times b(icj)$$

end do

end do

sendo: ilj = índice da linha do elemento j armazenado em pseudocoluna;

icj = índice da coluna do elemento j armazenado em pseudocoluna.

Para a implementação do algoritmo WCN, é necessário armazenar os elementos das pseudocolunas das matrizes W e W^U em vetores, bem como os índices das linhas e das colunas de tais elementos.

Na tabela 3.3, a seguir, são mostradas as operações de substituições da primeira etapa do processo de solução (SF), aplicando-se a metodologia WCN, utilizando a matriz W da figura 3.9. As etapas SD e SB são processadas de forma semelhante. A coluna “tarefa” indica, agora, qual pseudocoluna (processada vetorialmente) pertence os elementos da matriz W .



Tabela 3.3 – Operações na SF utilizando a metodologia WCN.

Tarefa	Operação	Tarefa	Operação
1	$b_9 = b_9 + W_{9,1} \cdot b_1$	3	$b_{20} = b_{20} + W_{20,11} \cdot b_{11}$
1	$b_{11} = b_{11} + W_{11,2} \cdot b_2$	4	$b_{12} = b_{12} + W_{12,15} \cdot b_{15}$
1	$b_{14} = b_{14} + W_{14,7} \cdot b_7$	4	$b_{19} = b_{19} + W_{19,16} \cdot b_{16}$
1	$b_{15} = b_{15} + W_{15,3} \cdot b_3$	4	$b_{17} = b_{17} + W_{17,15} \cdot b_{15}$
1	$b_{16} = b_{16} + W_{16,6} \cdot b_6$	4	$b_{18} = b_{18} + W_{18,15} \cdot b_{15}$
1	$b_{10} = b_{10} + W_{10,4} \cdot b_4$	4	$b_{20} = b_{20} + W_{20,16} \cdot b_{16}$
1	$b_{12} = b_{12} + W_{12,3} \cdot b_3$	5	$b_{17} = b_{17} + W_{17,16} \cdot b_{16}$
1	$b_{19} = b_{19} + W_{19,6} \cdot b_6$	6	$b_{13} = b_{13} + W_{13,10} \cdot b_{10}$
1	$b_{13} = b_{13} + W_{13,1} \cdot b_1$	6	$b_{17} = b_{17} + W_{17,12} \cdot b_{12}$
1	$b_{17} = b_{17} + W_{17,7} \cdot b_7$	6	$b_{18} = b_{18} + W_{18,10} \cdot b_{10}$
1	$b_{18} = b_{18} + W_{18,4} \cdot b_4$	6	$b_{20} = b_{20} + W_{20,12} \cdot b_{12}$
1	$b_{20} = b_{20} + W_{20,2} \cdot b_2$	7	$b_{13} = b_{13} + W_{13,19} \cdot b_{19}$
2	$b_{15} = b_{15} + W_{15,8} \cdot b_8$	7	$b_{17} = b_{17} + W_{17,19} \cdot b_{19}$
2	$b_{13} = b_{13} + W_{13,5} \cdot b_5$	7	$b_{18} = b_{18} + W_{18,12} \cdot b_{12}$
2	$b_{18} = b_{18} + W_{18,8} \cdot b_8$	7	$b_{20} = b_{20} + W_{20,19} \cdot b_{19}$
2	$b_{20} = b_{20} + W_{20,5} \cdot b_5$	8	$b_{18} = b_{18} + W_{18,19} \cdot b_{19}$
3	$b_{10} = b_{10} + W_{10,9} \cdot b_9$	9	$b_{17} = b_{17} + W_{17,13} \cdot b_{13}$
3	$b_{12} = b_{12} + W_{12,11} \cdot b_{11}$	9	$b_{18} = b_{18} + W_{18,13} \cdot b_{13}$
3	$b_{19} = b_{19} + W_{19,14} \cdot b_{14}$	9	$b_{20} = b_{20} + W_{20,13} \cdot b_{13}$
3	$b_{13} = b_{13} + W_{13,9} \cdot b_9$	10	$b_{18} = b_{18} + W_{18,17} \cdot b_{17}$
3	$b_{17} = b_{17} + W_{17,14} \cdot b_{14}$	10	$b_{20} = b_{20} + W_{20,17} \cdot b_{17}$
3	$b_{18} = b_{18} + W_{18,14} \cdot b_{14}$	11	$b_{20} = b_{20} + W_{20,18} \cdot b_{18}$

Analisando-se a tabela 3.3, nota-se que a etapa SF pode ser processada em apenas 11 *do-loops*. As operações contidas nestes laços são realizadas vetorialmente sem nenhum problema de dependência de dados, diferente do que ocorre quando da implementação da solução através do método tradicional de substituições (decomposição LDU) realizada serialmente, sendo as operações processadas uma a uma numa determinada ordem (tabela 3.1). Esta comparação pode ser entendida como um argumento favorável e motivador para a busca de novas metodologias que explorem o processamento vetorial.

Metodologia WCW. Esta metodologia foi proposta por [16]. Consiste na aplicação da metodologia WCN para as primeiras partições da matriz W (partições grandes), sendo que as restantes são reunidas em uma única partição chamada W_{UP} . Todas as partições são armazenadas por pseudocolunas, inclusive a W_{UP} . A formação da última partição se dá através do GCF, no qual se escolhe uma profundidade qualquer, chamada profundidade de corte, de forma que os nós que estão abaixo desta profundidade são reunidos em W_{UP} . O objetivo desta metodologia é eliminar as relações de precedência das operações elementares (multiplicação e adição) existentes no processamento das últimas partições que são formadas por poucos elementos (partições pequenas). Quando se agrupa elementos do GCF com profundidades diferentes em uma única partição, pode-se gerar menor número de pseudocolunas e estas possuem maior comprimento, favorecendo assim o processamento vetorial. Deve-se destacar que na formação da partição W_{UP} são necessários cálculos adicionais e que aparecerão novos elementos (*W-fills*).

Aplicando-se a metodologia WCW, o processo de solução é representado pela equação 3.19, como mostrado a seguir:

$$x = W_1^U \cdot W_2^U \cdots W_p^U \cdot W_{UP}^U \cdot D^{-1} \cdot W_{UP} \cdot W_p \cdots W_2 \cdot W_1 \cdot b \quad (3.19)$$

sendo: $W_{UP} = W_n \cdot W_{n-1} \cdots W_{p+2} \cdot W_{p+1}$;

n = número de partições originais da matriz W ;

p = profundidade de corte do GCF para a formação de W_{UP} .

O algoritmo da metodologia WCW é idêntico ao algoritmo da metodologia WCN.

Metodologia WCA. Esta é uma metodologia proposta neste trabalho. Ela utiliza o método da matriz W particionada segundo o algoritmo PA1. Os elementos das primeiras partições (grandes) são armazenados em pseudocolunas, e os elementos das últimas



partições (pequenas) são agrupados em uma matriz cheia, chamada matriz A_{UP} . A utilização da matriz A_{UP} foi inicialmente proposta por [26], para o uso em computadores dotados de processamento paralelo.

Aplicando-se a metodologia WCA, o processo de solução pode ser representado por:

$$x = W_1^U \cdot W_2^U \dots W_p^U \cdot D^{*-1} \cdot A_{UP}^{-1} \cdot W_p \dots W_2 \cdot W_1 \cdot b \quad (3.20)$$

sendo: $D^* = D_1 \cdot D_2 \dots D_p$;

$$A_{UP}^{-1} = W_{p+1}^U \cdot W_{p+2}^U \dots W_n^U \cdot D_n^{-1} \dots D_{p+2}^{-1} \cdot D_{p+1}^{-1} \cdot W_n \dots W_{p+2} \cdot W_{p+1}.$$

Cabe destacar que a representação anterior da matriz A_{UP} é uma forma algébrica de expressar a matriz da última partição. Porém, a matriz A_{UP} , por facilidade de cálculos, foi obtida diretamente da topologia original da rede, ou seja, após realizada a etapa de ordenação, fatora-se os nós acima da profundidade de corte, escolhida no GCF. Os nós restantes formam a matriz A_{UP} com os dados originais do sistema de energia elétrica. Se por um lado economiza-se cálculos com a não fatoração dos nós que compõem a matriz A_{UP} , por outro, a obtenção da inversa de A_{UP} é uma etapa que exige cálculos adicionais.

Em estudos onde é utilizada a mesma matriz A ($A \cdot x = b$) para obtenção de várias soluções com diferentes valores do vetor b , a metodologia WCA pode ser vantajosa apesar de requerer cálculos adicionais, pois a A_{UP} permanece constante.

Para melhor entendimento da metodologias WCA, é apresentado a seguir um algoritmo que descreve as etapas SF, SD e SB do processo de solução.

ALGORITMO WCA

```

do  $i = 1$ , número de pseudocolunas da matriz  $W$ 
  do  $j = \{\text{elementos da pseudocoluna } i\}$ 
     $b(ilj) = b(ilj) + W(ilj, icj) \times b(icj)$ 
  end do
end do
do  $k = \{\text{índice das linhas dos nós da última partição}\}$ 
  do  $j = \{\text{índice das colunas dos nós da última partição}\}$ 
     $b(k) = b(k) + A_{UP}^{-1}(k, j) \times b^*(j)$ 
  end do
end do
do  $j = \{\text{elementos da matriz } D^*\}$ 
   $b(ilj) = b(ilj) \times D^*(ilj, ilj)$ 
end do
do  $i = \text{número de pseudocolunas da matriz } W^U, 1$ 
  do  $j = \{\text{elementos da pseudocoluna } i\}$ 
     $b(ilj) = b(ilj) + W^U(ilj, icj) \times b(icj)$ 
  end do
end do

```

sendo: ilj = índice da linha do elemento j armazenado em pseudocoluna;

icj = índice da coluna do elemento j armazenado em pseudocoluna.

b^* = posições do vetor b relativas aos nós da última partição.

Com o objetivo de esclarecer o algoritmo WCA, a figura 3.10 ilustra, como exemplo, a matriz W particionada, no qual as duas primeiras partições são armazenadas através de pseudocolunas e as restantes foram reunidas em uma matriz cheia. O vetor b foi também particionado apenas para melhor visualização da metodologia.

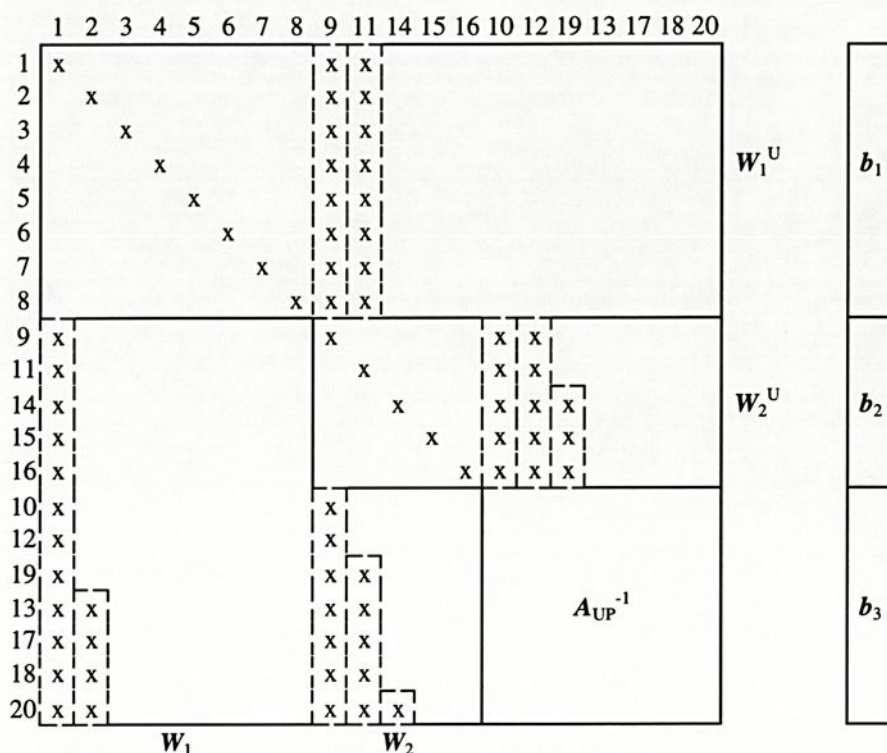


Figura 3.10 – Representação da metodologia WCA.

O processo geral de solução da metodologia WCA é descrito a seguir, no final do processo, a solução estará no vetor b .

- Passo 1.** Multiplicar $W_1 \times b_1$ e o resultado atualiza b_2 e b_3 ;
- Passo 2.** Multiplicar $W_2 \times b_2$ e o resultado atualiza b_3 ;
- Passo 3.** Multiplicar $A_{UP}^{-1} \times b_3$ e o resultado atualiza b_3 ;
- Passo 4.** Multiplicar todos os elementos da diagonal por b_1 e b_2 e o resultado atualiza b_1 e b_2 ;
- Passo 5.** Multiplicar $W_2^U \times b_3$ e o resultado atualiza b_2 ;
- Passo 6.** Multiplicar $W_1^U \times (b_2 \text{ e } b_3)$ e o resultado atualiza b_1 .

A utilização da matriz A_{UP} (passo 3) é um processo totalmente vetorizável, no qual se observa a multiplicação de uma matriz por um vetor explicitamente. Neste passo, são resolvidas de uma só vez as etapas SF, SD e SB.

3.6 SOLUÇÃO UTILIZANDO O MÉTODO DBSA

A filosofia do método DBSA (*Dependency-Based Substitution Algorithms*) foi inicialmente introduzida em [33], cujo objetivo, assim como no método da matriz W , é obter uma longa seqüência de operações que possam ser processadas sem dependência de dados.

O método DBSA foi originalmente desenvolvido para ser usado em computadores dotados de processamento paralelo. Neste método, a matriz A (sistema 3.1) é fatorada em LDU, através do processo de eliminação de Gauss, como descrito na seção 3.2. Durante a etapa de fatoração e a etapa de solução, o método DBSA analisa quais destas operações podem ser processadas ao mesmo tempo, sem dependência de dados, e atribui a execução destas operações aos diferentes processadores. Utilizando-se os dados da tabela 3.1, a figura 3.11 ilustra um diagrama chamado grafo de tarefas que especifica a interdependência das operações elementares (adição e multiplicação) na etapa SF. Verifica-se na figura 3.11 que todas as operações elementares que estão no mesmo nível, podem ser processadas ao mesmo tempo.



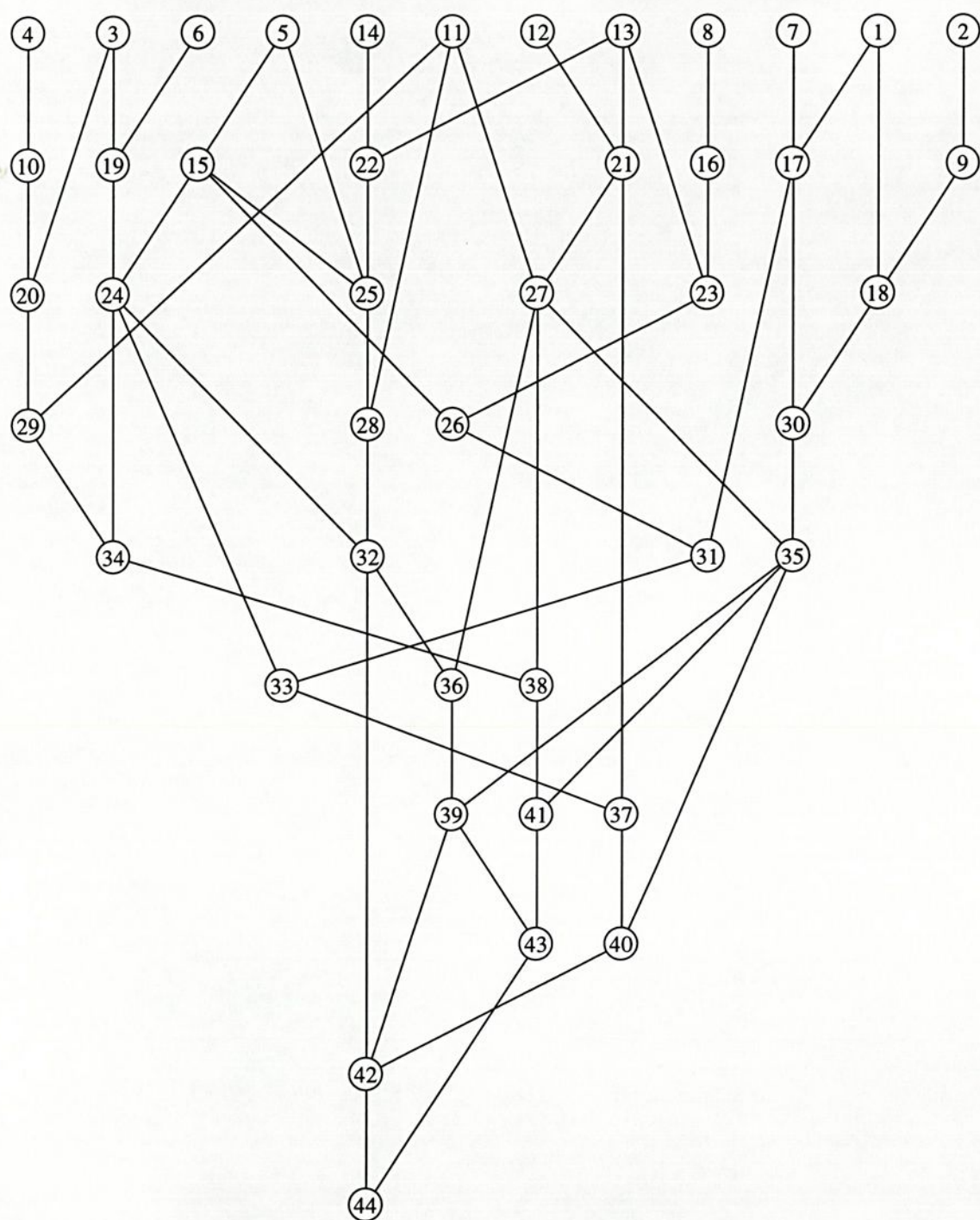


Figura 3.11 – Grafo das relações entre as tarefas da tabela 3.1.

A referência [32] aplicou o método DBSA em computadores dotados de processamento vetorial. Para obter bom desempenho neste ambiente computacional, é necessário armazenar os elementos das matrizes L , D e U em vetores, cujos componentes podem ser processados sem dependência de dados. Estes vetores são chamados de *slices*.

Tomando-se, como exemplo, a matriz L apresentada na figura 3.4, e aplicando-se a metodologia DBSA para processamento vetorial, mostra-se na figura 3.12, a seguir, a formação dos dois primeiros *slices*.

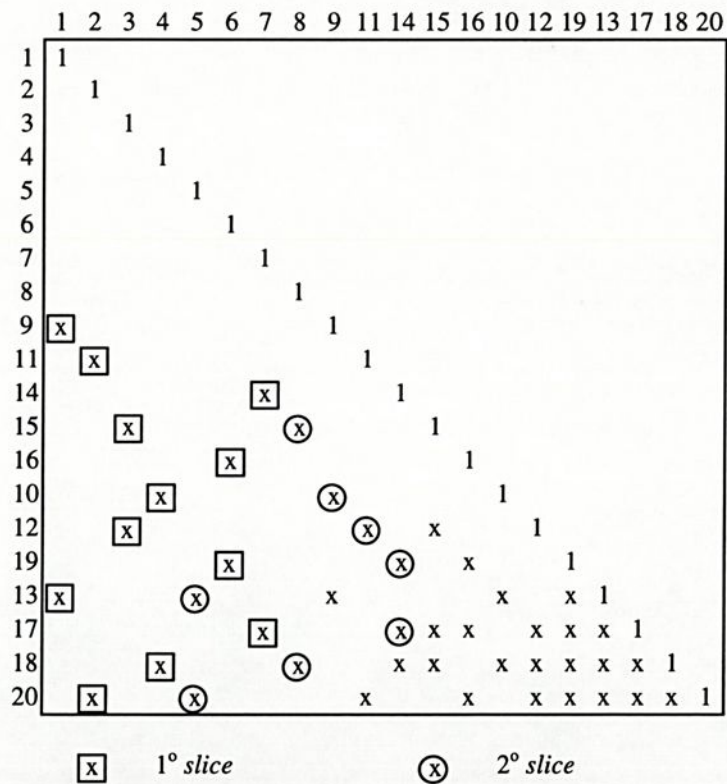


Figura 3.12 – Metodologia DBSA - Formação dos dois primeiros *slices*.

A matriz L utilizada na metodologia DBSA é semelhante a matriz W utilizada no método original da matriz W (metodologia WCN), salvo os sinais contrários dos elementos das matrizes. Desta forma, pode-se concluir que a formação das

pseudocolunas é análoga à formação dos *slices*, exceto pelo fato de que as pseudocolunas são formadas por elementos pertencentes à mesma partição. Os *slices* não possuem esta restrição.

Comparando-se o armazenamento dos elementos da matriz L , quando da aplicação do método DBSA (figura 3.12) com o armazenamento dos elementos da matriz W , pela aplicação do método original da matriz W (figuras 3.7 e 3.9), verifica-se que o segundo *slice* possui mais elementos do que a segunda pseudocoluna. Com isso, utilizando-se o método DBSA, trabalha-se com vetores mais longos e, portanto, o desempenho do processamento vetorial tende a ser maior.

A seguir, serão apresentadas algumas propostas de utilização do método DBSA para a resolução do sistema de equações lineares com processamento vetorial.

3.6.1 MÉTODO DBSA COM VETORES CHEIOS

Metodologia DCN. Originalmente proposta por [32]. Nesta metodologia, todos os elementos das matrizes L e U são armazenados em *slices*, cujas operações são processadas vetorialmente sem nenhuma dependência de dados. A seguir, é apresentado o algoritmo da metodologia DCN que descreve, seqüencialmente, as etapas SF, SD e SB do processo de solução.



ALGORITMO DCN

```
do  $i = 1$ , número de slices da matriz  $L$ 
  do  $j = \{\text{elementos do slice } i\}$ 
     $b(ilj) = b(ilj) - L(ilj, icj) \times b(icj)$ 
  end do
end do
do  $j = \{\text{elementos da matriz } D\}$ 
   $b(ilj) = b(ilj) \times D(ilj, ilj)$ 
end do
do  $i = \text{número de slices da matriz } U, 1$ 
  do  $j = \{\text{elementos do slices } i\}$ 
     $b(ilj) = b(ilj) - U(ilj, icj) \times b(icj)$ 
  end do
end do
```

sendo: ilj = índice da linha do elemento j armazenado em *slice*;

icj = índice da coluna do elemento j armazenado em *slice*.

Para a implementação do método DBSA, como se verifica no algoritmo DCN, é necessário armazenar os elementos das matrizes L e U em vetores (*slices*). É necessário armazenar, também, os índices das linhas e das colunas de tais elementos.

Metodologia DCA. Esta metodologia é proposta neste trabalho. Verifica-se que os *slices* que armazenam os elementos das últimas colunas das matrizes L e U são vetores pequenos e com isso o desempenho do processamento vetorial diminui. Com o intuito de resolver este problema, nesta metodologia, escolhe-se uma profundidade no GCF chamada profundidade de corte; os elementos das colunas das matrizes L e U que estão acima desta profundidade são agrupados em *slices*. Os elementos restantes, são



agrupados em uma matriz chamada A_{UP} . A seguir é apresentado o algoritmo que descreve as etapas SF, SD e SB do processo de solução, utilizando-se a metodologia DCA.

ALGORITMO DCA

```

do  $i = 1$ , número de slices da matriz  $L$ 
  do  $j = \{\text{elementos do slice } i\}$ 
    
$$b(ilj) = b(ilj) - L(ilj, icj) \times b(icj)$$

  end do
end do
do  $k = \{\text{índice das linhas dos nós da última partição}\}$ 
  do  $j = \{\text{índice das colunas dos nós da última partição}\}$ 
    
$$b(k) = b(k) + A_{UP}^{-1}(k, j) \times b^*(j)$$

  end do
end do
do  $j = \{\text{elementos da matriz } D^*\}$ 
  
$$b(ilj) = b(ilj) \times D^*(ilj, ilj)$$

end do
do  $i = \text{número de slices da matriz } U, 1$ 
  do  $j = \{\text{elementos do slices } i\}$ 
    
$$b(ilj) = b(ilj) - U(ilj, icj) \times b(icj)$$

  end do
end do

```

Cabe observar que a obtenção da matriz A_{UP} exige cálculos adicionais, mas a sua utilização é um processo totalmente vetorizável, cujas etapas SF, SD e SB são resolvidas todas de uma só vez.



3.7 MÉTODO DOS VETORES ESPARSOS

O método dos vetores esparsos é uma técnica usada na solução de sistemas de equações lineares ($Ax = b$) que explora a esparsidade dos vetores x e b . Este método foi desenvolvido por [29] e tem como objetivo obter uma economia computacional nas etapas SF e SB, calculando-se somente as operações estritamente necessárias para a obtenção da resposta do sistema. A esparsidade dos vetores ocorre em estudos de sistemas de energia elétrica, devido a modelagem dos componentes do sistema e/ou quando se está interessado em conhecer apenas um subconjunto da solução.

A eliminação das operações desnecessárias durante a etapa de solução é desenvolvida utilizando-se o GCF que indica, neste caso, como as posições dos vetores b e x são atualizadas durante o processo de solução. As substituições realizadas desta forma são conhecidas como *fast-forward* (FF) e *fast-backward* (FB).

A substituição FF calcula somente um subconjunto de colunas das matrizes L ou W (quando o vetor b é esparso), e FB necessita calcular somente um subconjunto de linhas das matrizes U ou W^U (quando um subconjunto de x é requerido). A figura 3.13 ilustra as operações necessárias na FF ou FB quando algumas posições de b e x são nulas. As linhas pontilhadas mostram os caminhos necessários para a FF, considerando-se que o vetor b possui elementos diferentes de zero somente nas posições 3, 7 e 8. Os mesmos caminhos são necessários para a FB se é desejado conhecer a solução correspondente às posições 3, 7 e 8.

A substituição diagonal (SD), também necessita ser realizada somente para as mesmas posições do vetor x processadas na FB, caso contrário, estar-se-ia realizando operações desnecessárias.



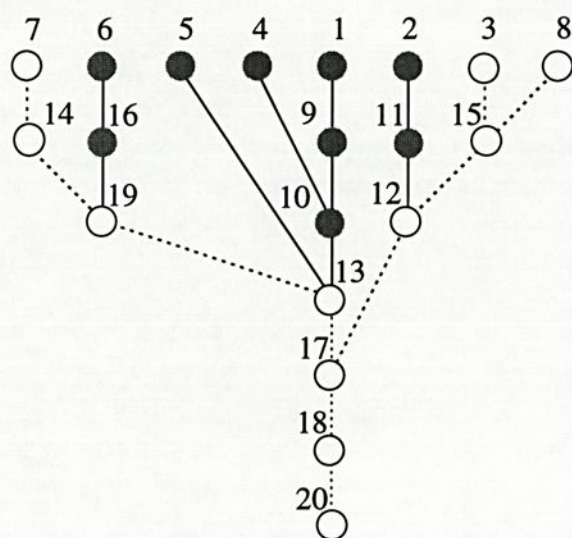


Figura 3.13 – Grafo dos Caminhos de Fatoração para FF e FB
utilizando-se a matriz da figura 3.4.

3.8 MÉTODO DA MATRIZ W E DBSA COM VETORES ESPARSOS

As quatro metodologias apresentadas neste item são metodologias propostas neste trabalho. São desenvolvidas para ambiente dotado de processamento vetorial e possuem a habilidade de explorar a esparsidade dos vetores x e b (sistema $Ax = b$). A diferença básica em relação as metodologias anteriores está na formação das pseudocolunas ou *slices*, que agora são obtidos aplicando-se o método dos vetores esparsos.

Metodologia WEN. Esta metodologia utiliza o método da matriz W , adaptada para o caso de se trabalhar com vetores esparsos. Agora são armazenados em pseudocolunas somente os elementos das colunas das matrizes W e W^U que são realmente necessárias para se implementar as etapas FF e FB.

Para efeito de ilustração, considere que para o sistema da figura 3.2 (sistema de 20 nós), somente as posições 3, 7 e 8 do vetor b são diferentes de zero e que se deseja conhecer o valor das mesmas posições do vetor x . A figura 3.13 ilustra os nós que

devem ser processados na FF e na FB. Portanto, utilizando-se a metodologia WEN, tem-se ilustrados na figura 3.14 os elementos da matriz W armazenados em pseudocolunas e os elementos da diagonal que são necessários para a etapa FF e substituição diagonal, respectivamente. Na etapa FB, os elementos da matriz W^U são armazenados utilizando-se o mesmo critério.

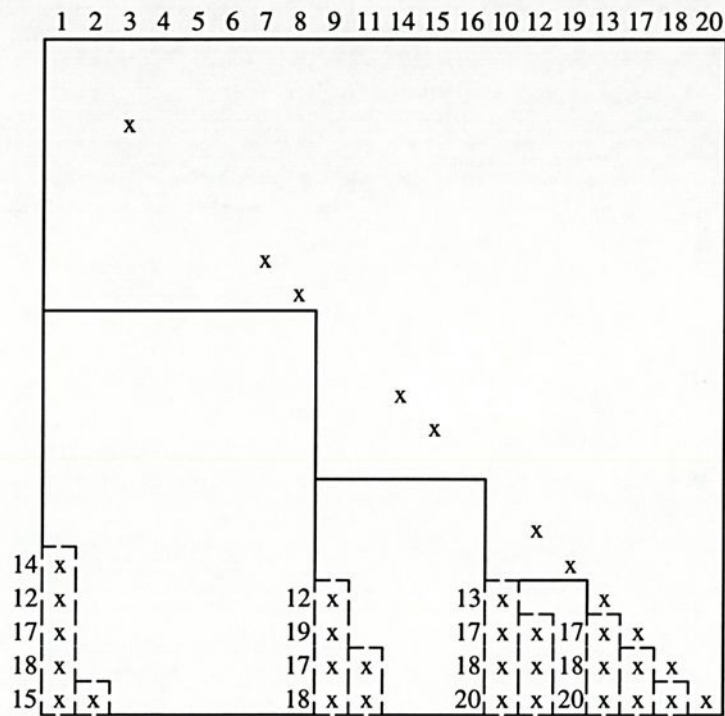


Figura 3.14 – Pseudocolunas da FF e elementos da diagonal.

O algoritmo que descreve a metodologia WEN é semelhante ao algoritmo da metodologia WCN.

Metodologia WEA. Esta metodologia aplica o método da matriz W e armazena em pseudocolunas os elementos das primeiras partições das matrizes W e W^U necessários para a implementação da FF e FB. As últimas partições são reunidas em uma matriz chamada A_{UP} . O objetivo desta metodologia é agrupar as últimas partições,

que são partições pequenas, de forma a melhorar o desempenho do processamento vetorial.

Com base na figura 3.13, que mostra os nós necessários para FF e FB, adotou-se armazenar, a título de exemplo, as duas primeiras partições da matriz W em pseudocolunas e as partições restantes foram reunidas na matriz A_{UP} , como ilustra a figura 3.15.

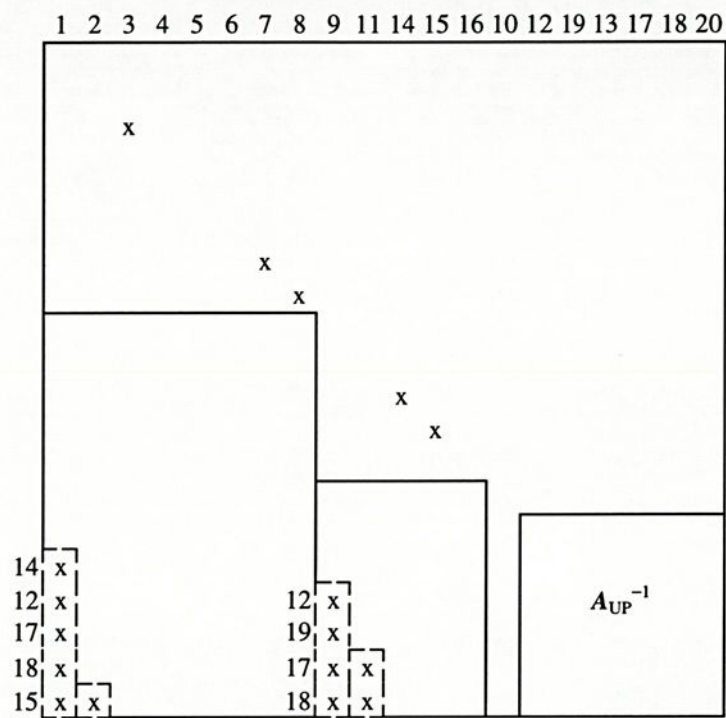


Figura 3.15 – Pseudocolunas da FF, elementos da diagonal e matriz A_{UP} .

O algoritmo desta metodologia é idêntico ao algoritmo da metodologia WCA.

Metodologia DEN. Nesta metodologia aplica-se o método DBSA, armazenando-se em *slices* os elementos das matrizes L e U , quando da aplicação do método dos vetores esparsos, ou seja, são armazenados somente os elementos necessários para a implementação da FF e da FB.

Para ilustrar o método DEN, considera-se que para os sistema da figura 3.2, somente as posições 3, 7 e 8 do vetor $\mathbf{b}$ são não nulas e deseja-se conhecer o valor das mesmas posições do vetor $\mathbf{x}$. A figura 3.16 ilustra os *slices* para a implementação da FF e os elementos da diagonal necessários para a etapa de substituição diagonal.

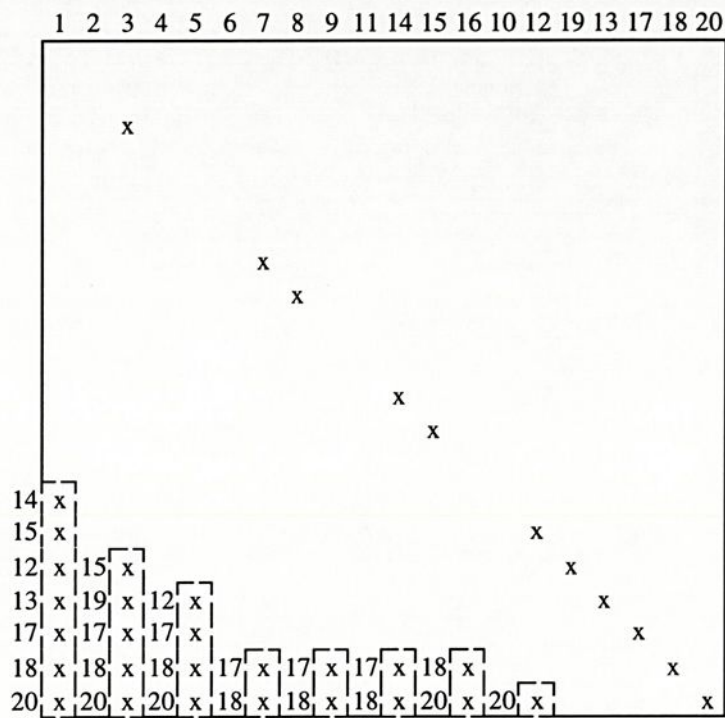


Figura 3.16 – Slices da FF e elementos da diagonal.

O algoritmo da metodologia DEN é o mesmo da metodologia DCN.

Metodologia DEA. Nesta metodologia aplica-se o método DBSA, juntamente com a técnica dos vetores esparsos para os primeiros elementos do GCF, que são armazenados em *slices*. Os nós restantes são reunidos em uma matriz chamada A_{UP} , cujo objetivo é obter uma melhoria no desempenho do processamento vetorial, pois assim, trabalha-se com vetores maiores.

Tomando-se o mesmo sistema de 20 nós (figura 3.2) e adotando-se a mesma esparsidade, utilizada nas metodologias anteriores, para os vetores x e b , a figura 3.17 ilustra os *slices* para a FF, os elementos da diagonal necessários para a solução do sistema e a matriz A_{UP} , que agrupa os nós destacados na figura 3.13 por círculos vazios, e que possuem uma profundidade no GCF maior ou igual a três, a qual neste exemplo, foi a profundidade de corte escolhida.

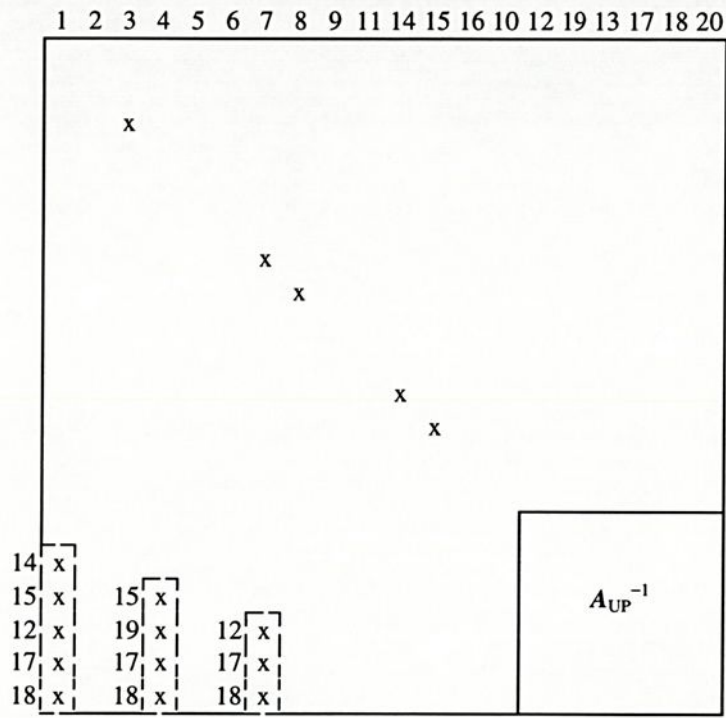


Figura 3.17 – Slices da FF, elementos da diagonal e matriz A_{UP} .

O algoritmo que descreve esta metodologia é coincidente com o algoritmo DCA.



3.9 DESEMPENHO DAS METODOLOGIAS APRESENTADAS

O principal objetivo da análise do desempenho das metodologias apresentadas é identificar qual proporciona os melhores resultados quando se utiliza processamento vetorial.

As metodologias foram implementadas no computador CRAY Y-MP2E/232 da Universidade Federal do Rio Grande do Sul. Utilizaram-se os mesmos níveis de otimização durante a compilação das metodologias (compilador Fortran 77). A diferença entre as metodologias no modo serial (ou escalar) e vetorial é dada pelo uso de diretivas de compilação dentro do próprio programa que desabilitam a vetorização no modo escalar, ou informam ao compilador que não existe dependência de dados no laço *do-loop* subsequente, para o caso vetorial.

Com o intuito de analisar o desempenho das metodologias, foram estudados dois tipos de problemas, dados a seguir, sendo o primeiro um sistema com números complexos e o segundo com números reais:

$$I = Y V \quad (3.21)$$

$$P = B' \theta \quad (3.22)$$

Para as metodologias que trabalham com vetores esparsos, foi considerado o caso em que os vetores I e P contêm elementos diferentes de zero somente nas posições associadas às barras geradoras. Foi também considerada a mesma esparsidade para os vetores V e θ , ou seja, desejou-se conhecer somente os valores das posições associadas às barras de geração.

As metodologias foram testadas utilizando-se quatro sistemas de energia elétrica, apresentados a seguir: sistema IEEE-118 barras, e três configurações do sistema interligado sul-sudeste brasileiro, cujas principais características são descritas na tabela 3.4.



Tabela 3.4 – Características dos Sistemas Utilizados.

	Sistemas			
	IEEE-118	BR-320	BR-725	BR-1729
Barras	118	320	725	1729
Linhas	176	470	935	2154
Geradores	53	44	92	156

Com o intuito de apresentar as metodologias de uma forma mais clara, construiu-se a tabela 3.5, na qual se encontram as principais características de cada metodologia.

Tabela 3.5 – Características das Metodologias Apresentadas.

Denominação	Método Utilizado	Vetores b/x	W_{UP}	A_{UP}^{-1}
WCN	Matriz W	Cheios	Não	Não
WCW	Matriz W	Cheios	Sim	Não
WCA	Matriz W	Cheios	Não	Sim
DCN	DBSA	Cheios	Não	Não
DCA	DBSA	Cheios	Não	Sim
WEN	Matriz W	Esparsos	Não	Não
WEA	Matriz W	Esparsos	Não	Sim
DEN	DBSA	Esparsos	Não	Não
DEA	DBSA	Esparsos	Não	Sim

Como discutido anteriormente, na seção 3.4, concluiu-se que as ordenações MDML e MLMD são as mais indicadas para o processamento vetorial. Com o objetivo de determinar a ordenação que apresenta melhor desempenho, a tabela 3.6 mostra os



tempos de CPU utilizados na etapa de solução da equação $I = YV$ para o sistema IEEE-118 barras, utilizando-se algumas metodologias que trabalham com vetores cheios (WCW e WCA) e esparsos (WEA), apresentadas anteriormente. Nesta tabela, “np” indica a profundidade de corte no GCF para o cálculo da matriz W_{UP} (WCW) e A_{UP} (WCA e WEA). Destaca-se que os demais sistemas de energia elétrica testados apresentam comportamento semelhante.

Tabela 3.6 – Tempo de CPU (seg x 10^{-4}) para o sistema IEEE-118 nós.

np	MDML			MLMD		
	Metod. WCW	Metod. WCA	Metod. WEA	Metod. WCW	Metod. WCA	Metod. WEA
0	3,62	3,86	1,72	3,62	3,86	1,85
1	2,02	1,66	1,16	2,09	1,53	1,10
2	1,45	0,85	0,70	1,49	0,83	0,64
3	1,23	0,77	0,61	1,27	0,81	0,62
4	1,14	0,75	0,63	1,21	0,89	0,70
5	1,09	0,81	0,65	1,30	1,04	0,84
6	1,08	0,88	0,70	1,37	1,18	0,98
7	1,13	0,95	0,78	1,45	1,31	1,08
8	1,13	1,01	0,84	1,51	1,41	1,17
9	1,17	1,12	0,92	1,58	1,50	1,25
10	1,23	1,19	0,98	1,63	1,60	1,31
11	1,22	1,22	1,01	1,65	1,62	1,36
12	1,23	1,24	1,04	1,67	1,66	1,40
13	–	–	–	1,67	1,69	1,42

A tabela 3.6 mostra melhores resultados com o uso da ordenação MDML. Esta ordenação mantém o desempenho com as outras três redes simuladas, motivo pela qual ela é utilizada na sequência dos trabalhos. Comparando-se o desempenho das metodologias WCW e WCA, verifica-se que reunir as últimas partições em W_{UP} é menos eficiente do que agrupá-las em A_{UP} , por isso, na continuação dos trabalhos, adotou-se utilizar metodologias que não agrupem a última partição em W_{UP} .

Para analisar a eficiência das metodologias, foi utilizado um ganho definido como: o tempo computacional de uma metodologia escalar (WCN no modo escalar) dividido



pelo tempo computacional de cada uma das referidas metodologias no modo vetorial. Isto permite comparar as vantagens do uso do processamento vetorial em relação ao processamento escalar. A figura 3.18 mostra este ganho para as diferentes metodologias utilizando-se o sistema brasileiro de 1729 barras. Cabe destacar, também, que os demais sistemas de energia elétrica testados apresentam um comportamento semelhante.

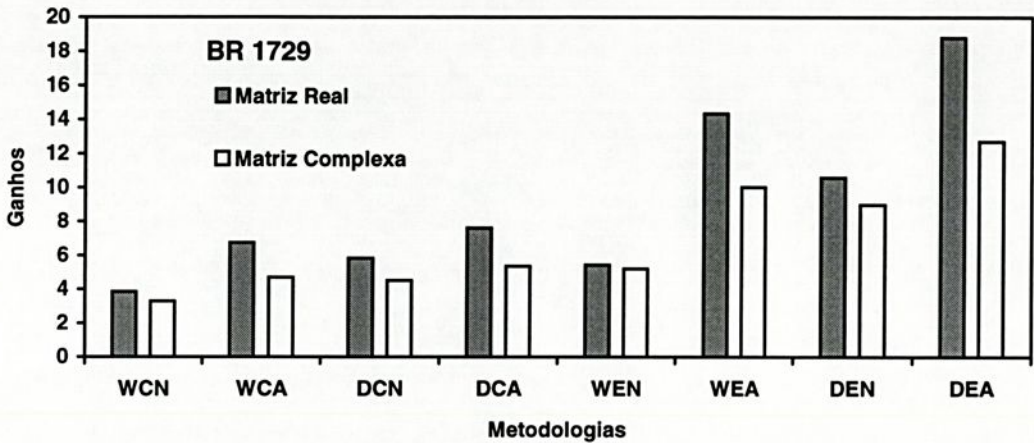


Figura 3.18 – Ganho das Metodologias.

Com o objetivo de comparar as melhores metodologias que explorem a esparsidade dos vetores no modo vetorial em relação ao modo escalar, apresenta-se na tabela 3.7, o ganho das metodologias WEA e DEA, definido agora, como sendo o tempo de CPU da metodologia WEN no modo escalar, pelo tempo de CPU das metodologias WEA e DEA no modo vetorial.



Tabela 3.7 – Ganho das Metodologias WEA e DEA.

	IEEE-118	BR-320	BR-725	BR-1729
Matriz Real ($P = B' \theta$)				
WEA	4,72	6,52	5,36	5,69
DEA	5,24	7,10	6,83	7,33
Matriz Complexa ($I = YV$)				
WEA	3,21	3,86	3,40	3,73
DEA	3,72	4,62	4,37	4,77

Nota-se pelos resultados anteriores, que a exploração da esparsidade dos vetores torna-se vantajosa em relação ao tempo de processamento. Nota-se também que a metodologia DBSA, agrupando-se os últimos nós na matriz A_{UP} , apresenta o melhor desempenho. Portanto, na implementação do programa de estabilidade, descrito no próximo capítulo, será utilizada a metodologia DEA, proposta neste trabalho, para a resolução de um sistema de equações lineares esparsas.

Deve ser salientado que na obtenção dos tempos de CPU utilizados no cálculo dos ganhos anteriores, não foram considerados os tempos relativos ao armazenamento dos elementos em pseudocolunas ou *slices*, nem o tempo para a formação das matrizes W_{UP} e A_{UP} . Estas etapas serão consideradas no próximo capítulo, quando da implementação do programa para a análise da estabilidade transitória.



CAPÍTULO IV

PROGRAMA PARA A ANÁLISE DA ESTABILIDADE TRANSITÓRIA EM AMBIENTE VETORIAL

4.1 INTRODUÇÃO

Com o objetivo de verificar os desempenhos das metodologias discutidas no capítulo III, implementou-se um programa de cálculo da estabilidade transitória em um computador vetorial. Isto permite avaliar todas as etapas do processo de cálculo, como leitura e impressão de dados, ordenação, preparação em vetores, fatoração e solução direta.

Desta forma, foi levada em consideração uma implementação sem grandes sofisticacões, mas que permite avaliar o desempenho vetorial dos conjuntos diferencial e algébrico. Como já discutido, a etapa de vetorização das equações algébricas é a que apresenta maior dificuldade. A resolução destas equações foi abordada no capítulo III, no qual foram apresentadas várias metodologias e comparados os seus respectivos desempenhos, sendo que a metodologia DEA se mostrou a mais indicada para a resolução das equações algébricas esparsas em ambiente vetorial.

Portanto, na implementação do programa de estabilidade transitória, descrito neste capítulo, foi utilizada a metodologia DEA, e, para efeito de comparação, as

metodologias DEN e a apresentada na referência [35], na resolução das equações algébricas.

Como o principal objetivo deste trabalho é sugerir meios para se otimizar o uso do processamento vetorial, a modelagem adotada para representar os componentes do sistema foi a mais simples possível, ou seja, as linhas de transmissão e os transformadores (no caso, os transformadores em fase) foram modelados utilizando-se o modelo π , as cargas foram representadas como admitância constante e os geradores foram modelados segundo o modelo clássico, cujas equações são mostradas a seguir.

4.2 REPRESENTAÇÃO DOS GERADORES SÍNCRONOS ATRAVÉS DO MODELO CLÁSSICO

A representação dos geradores síncronos através do modelo clássico resume-se a uma fonte de tensão constante atrás da reatância transitória de eixo direto, como ilustrado na figura 4.1.

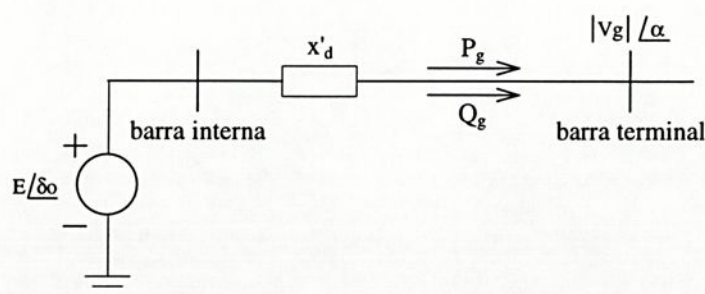


Figura 4.1 – Representação do gerador síncrono.

Os dados utilizados para se determinar a tensão interna do gerador síncrono são provenientes de um estudo prévio de fluxo de potência. As reatâncias transitórias das

máquinas síncronas (transformadas em admitâncias) são incluídas na diagonal principal da matriz Y (equação 2.2).

De acordo com a referência [02], adotou-se o seguinte procedimento para calcular a tensão interna da máquina síncrona: primeiro considerou-se a tensão da barra terminal como uma referência temporária, obtendo-se:

$$E \angle \delta' = (|V_g| + Q_g \cdot x'd / |V_g|) + j(P_g \cdot x'd / |V_g|) \quad (4.1)$$

Para calcular o valor real do ângulo da tensão interna do gerador, antes da ocorrência da perturbação, basta adicionar o ângulo α ao ângulo δ' , ou seja ([02]):

$$\delta_0 = \delta' + \alpha \quad (4.2)$$

O comportamento dinâmico das partes girantes das máquinas síncronas (rotor, turbina, etc), quando da utilização do modelo clássico, é descrito pelas equações diferenciais a seguir:

$$\dot{\omega} = \frac{\omega_s}{2H} (P_m - P_e) \quad (4.3)$$

$$\dot{\delta} = \omega - \omega_s \quad (4.4)$$

Neste trabalho, não se considerou o efeito do regulador de velocidade das máquinas síncronas, portanto, a potência mecânica permanecerá constante durante todo o período de estudo, e seu valor será o mesmo da potência elétrica de entreferro antes da ocorrência do defeito, ou seja, o seu valor é determinado por um estudo inicial de fluxo de potência.

Aplicando-se a fórmula trapezoidal de integração numérica às equações 4.3 e 4.4, que descrevem o comportamento dinâmico das partes girantes das máquinas síncronas, obtêm-se as seguintes equações algébricas ([24]):

$$\omega(t) = -\frac{\omega_s \Delta t}{4H} P_e(t) + \omega(t - \Delta t) + \frac{\omega_s \Delta t}{4H} [2 P_m - P_e(t - \Delta t)] \quad (4.5)$$

$$\delta(t) = \frac{\Delta t}{2} \omega(t) + \delta(t - \Delta t) + \frac{\Delta t}{2} [\omega(t - \Delta t) - 2\omega_s] \quad (4.6)$$

4.3 DESCRIÇÃO DO PROGRAMA COMPUTACIONAL

O programa implementado para a análise da estabilidade transitória de sistemas de energia elétrica é apresentado, de uma forma geral, na figura 4.2.

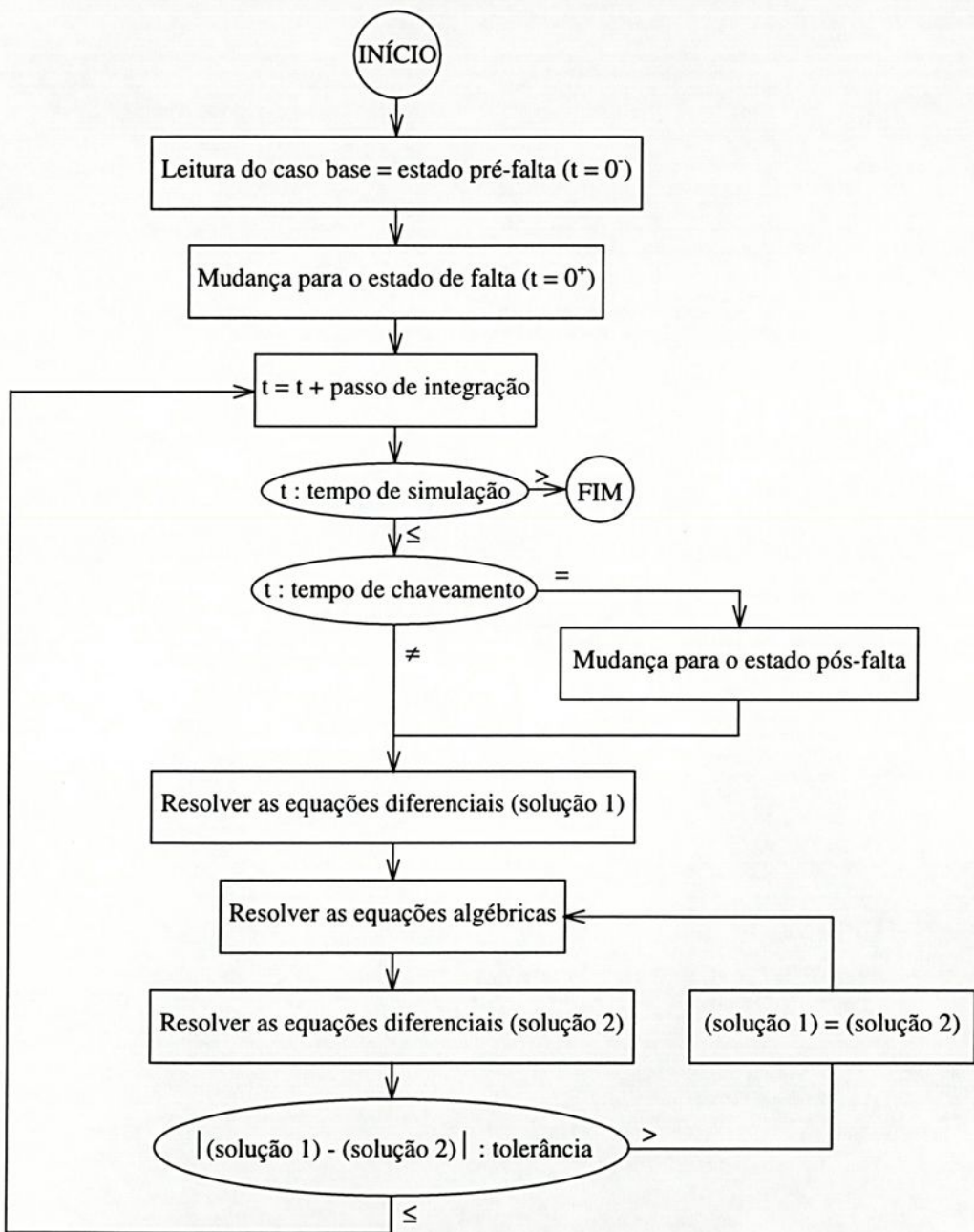


Figura 4.2 – Fluxograma simplificado do programa para a análise da estabilidade transitória.



Referente à figura 4.2, a leitura do caso base é a leitura dos dados do sistema em operação normal, ou seja, são lidos a configuração do sistema e o perfil de geração e carga para um dado instante de operação. Estes dados são provenientes de um estudo prévio de fluxo de potência. São lidos, também, os dados referentes à contingência a ser estudada que, neste trabalho, foi adotado como sendo um curto-circuito trifásico em uma linha de transmissão, próximo a um de seus terminais; portanto são lidos os terminais da linha em falta, e especificado o terminal mais próximo do defeito, juntamente com o tempo de duração da falta, ou seja, o tempo que o sistema de proteção leva para desligar a linha com defeito. Não foi considerado neste trabalho o religamento da linha de transmissão em falta.

A obtenção do estado de falta e do estado pós-falta é feita logo após a leitura do caso base. Nestas etapas, a configuração do sistema é mudada através da alteração na matriz Y , que por sua vez é fatorada e armazenada segundo a metodologia DEA, DEN e a metodologia apresentada por [35]. Portanto, quando se deseja mudar a configuração do sistema de um estado para outro, basta fazer a leitura do estado desejado, previamente armazenado.

Na resolução das equações algébricas, segundo a modelagem adotada neste trabalho, tem-se a seguinte representação detalhada da parte estática do sistema:

$$\begin{bmatrix} I_g \\ 0 \end{bmatrix} = \begin{bmatrix} Y_{gg} & Y_{gr} \\ Y_{rg} & Y_{rr} \end{bmatrix} \begin{bmatrix} E \\ V_r \end{bmatrix} \quad (4.7)$$

sendo: g = índice das barras internas de geração (figura 4.1);

r = índice das barras restantes.

A equação 4.7 representa o sistema na forma $I = Y V$, na qual a matriz Y e os vetores I e V estão particionados, abrangendo inclusive as barras internas de geração, para melhor detalhamento do processo de solução das equações algébricas.

Devido a modelagem simplificada utilizada, neste trabalho, pode-se reduzir o sistema às barras internas de geração. Porém, este procedimento não foi adotado para não destruir a esparsidade no processo de solução (um dos aspectos explorado no trabalho), e para que a solução apresentada fosse facilmente adaptada para ser utilizada em um caso geral, pois caso se deseje, por exemplo, conhecer a tensão em uma barra que não seja de geração ou, caso fosse adotada uma modelagem diferente para as cargas, a redução do sistema às barras internas de geração não seria aplicável.

Portando, com base na equação 4.7, pode-se escrever:

$$I_g = Y_{gg} E + Y_{gr} V_r \quad (4.8)$$

$$0 = Y_{rg} E + Y_{rr} V_r \quad (4.9)$$

A equação 4.9 pode ser reescrita da seguinte forma:

$$I = Y_{rr} V_r \quad (4.10)$$

sendo: $I = -Y_{rg} E$.

O vetor E representa a tensão da barra interna de geração das máquinas síncronas, cujo módulo permanece constante durante o estudo da estabilidade transitória, pois não foi considerado o efeito do regulador de tensão. O ângulo da tensão interna foi considerado coincidente com o ângulo mecânico do rotor, e a sua variação indicará se o sistema é estável ou não perante à contingência estudada.

Na equação 4.10, I é um vetor que possui elementos diferentes de zero somente nas posições relativas às barras de geração. Como o maior interesse na resolução da equação 4.10 é determinar o valor da tensão terminal das barras de geração, pode-se desenvolver uma solução direcionada para a obtenção do valor das posições do vetor V_r (relativas às barras geradoras).

Conhecido o vetor V_r , retorna-se à equação 4.8 e calcula-se o valor do vetor I_g . Com os vetores V_r e I_g , calcula-se o novo valor da potência elétrica de saída dos geradores síncronos. A potência elétrica, atualizada, é utilizada nas equações



diferenciais que descrevem o comportamento dinâmico das partes girantes das máquinas síncronas, dadas pelas equações 4.5 e 4.6. O processo se repete segundo o fluxograma ilustrado na figura 4.2.

Foram implementadas duas versões do programa para a análise da estabilidade transitória. Em cada versão utilizou-se a metodologia DEA, DEN e a apresentada em [35] para a resolução das equações algébricas.

A primeira versão do programa é a ilustrada na figura 4.2 e fornece como saída um arquivo de dados que contém a variação dos ângulos dos rotores dos geradores síncronos em função do tempo. Este arquivo de dados é então lido pelo programa *MATLAB® for Windows* o qual plota as curvas que mostram a variação dos ângulos dos rotores. Estas curvas estão ilustradas no próximo item, e a análise das mesmas determina se o sistema é estável ou não.

A segunda versão do programa tem como objetivo determinar soluções para diversos casos. Um simples teste realizado foi adotar um tempo de chaveamento inicial nulo e simular o comportamento do sistema perante a contingência estipulada, verificando se o sistema é estável. Caso o sistema continue estável, aumenta-se o tempo de chaveamento e repete-se a simulação. Este processo continua até que seja determinado o tempo de chaveamento para o qual o sistema se torne instável, obtendo-se, desta maneira, uma quantidade razoável de estudos a realizar.

Cabe destacar que para as duas versões do programa, a obtenção do estado de falta e pós-falta é realizada somente uma vez e que estes estados são armazenados para serem utilizados quando necessários durante a execução do programa.



4.4 RESULTADOS

Os primeiros resultados foram obtidos com o sistema de 9 barras e 3 geradores da referência [02], ilustrado na figura 4.3.

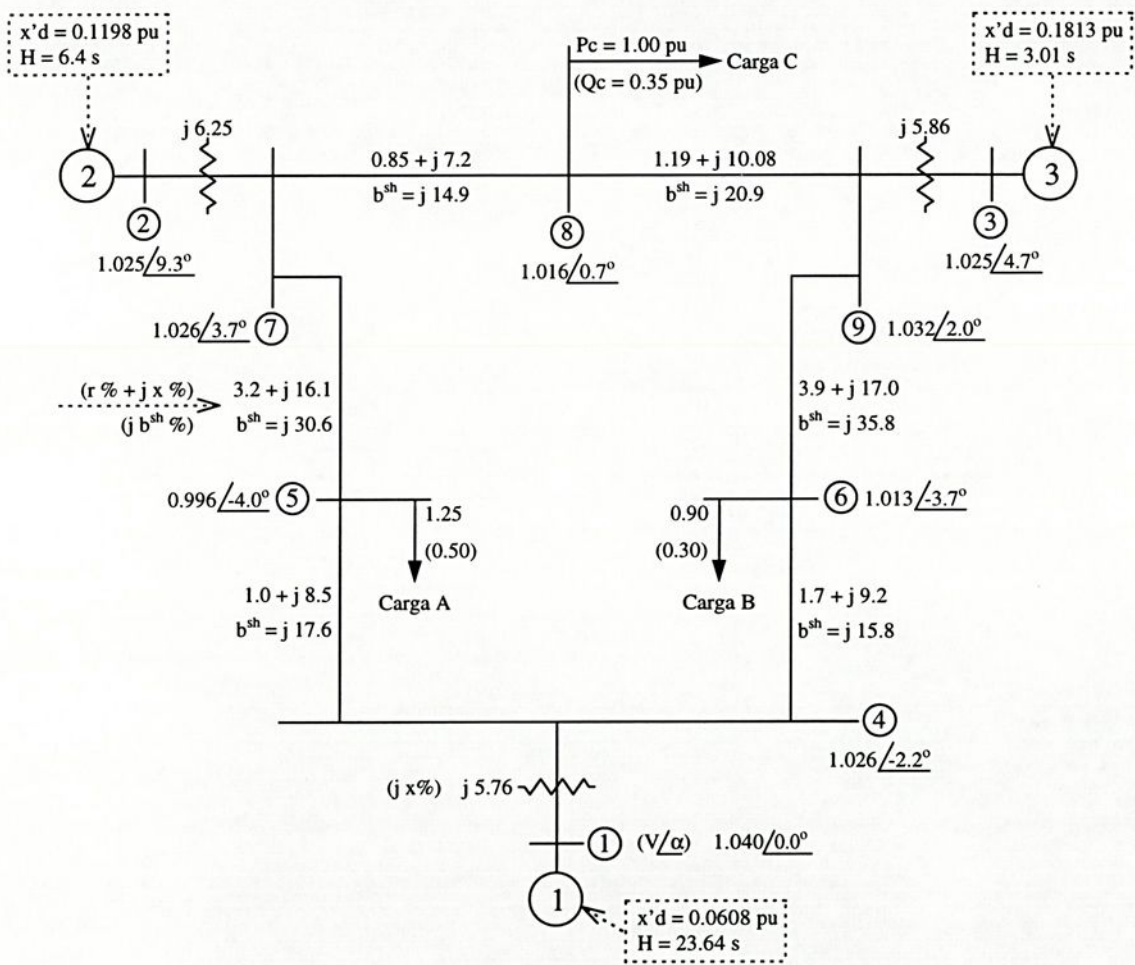


Figura 4.3 – Sistema de 9 barras da referência [02].

A seguir serão apresentados alguns dados referentes à contingência simulada e às duas versões do programa.



Contingência: (curto-circuito trifásico)

Localização: linha de transmissão 5–7, próximo ao terminal 7.

Primeira versão do programa (1 caso)

tempo final de simulação: 2,0 [s];

passo de integração: 0,005 [s];

tempo de chaveamento: especificado nos exemplos apresentados a seguir;

tolerância ou precisão do ângulo dos geradores: 0,001 [rad] ou 0,057 [°];

tolerância ou precisão da velocidade dos geradores: 0,01 [rad/s].

Segunda versão do programa (vários casos)

tempo final de simulação: 2,0 [s];

passo de integração: 0,003 [s];

variação do tempo de chaveamento: 0,005 [s];

tolerância ou precisão do ângulo dos geradores: 0,001 [rad] ou 0,057 [°];

tolerância ou precisão da velocidade dos geradores: 0,01 [rad/s].

A segunda versão do programa utiliza o seguinte critério para determinar se o sistema é estável ou instável. Caso o módulo da diferença do rotor de algum gerador síncrono em relação ao centro dos ângulos (δ_{coa} , definido na equação 4.11), durante a simulação, for maior do que 5,23 [rad] ou 300 [°], o sistema é considerado instável.

$$\delta_{coa} = \frac{\sum_{i=1}^{n.de\ geradores} H(i) \cdot \delta(i)}{\sum_{i=1}^{n.de\ geradores} H(i)} \quad (4.11)$$

Com base neste critério, a segunda versão do programa determina o intervalo de tempo que contém o tempo crítico de chaveamento para a contingência apresentada, aplicada ao sistema de nove barras, ou seja:



- o sistema é estável para tempo de chaveamento = 0,160 [s];
- o sistema é instável para tempo de chaveamento = 0,165 [s].

A seguir, são apresentadas algumas curvas obtidas com a primeira versão do programa, para simulações com diferentes tempos de chaveamento, referentes à contingência aplicada ao sistema de nove barras, apresentada anteriormente.

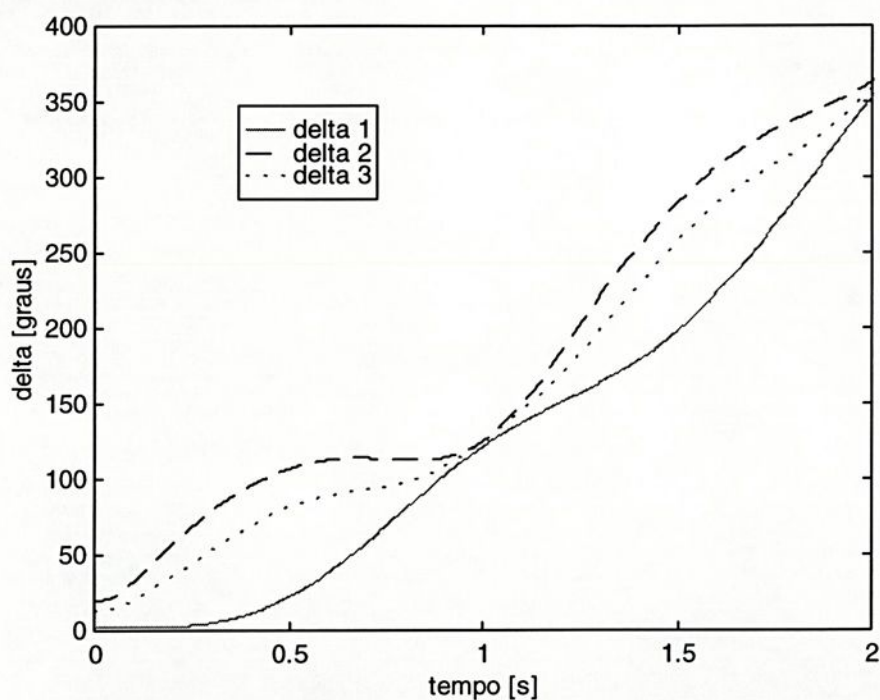


Figura 4.4 – Ângulos dos geradores 1, 2 e 3 para um tempo de chaveamento de 0,083 [s] (caso estável).

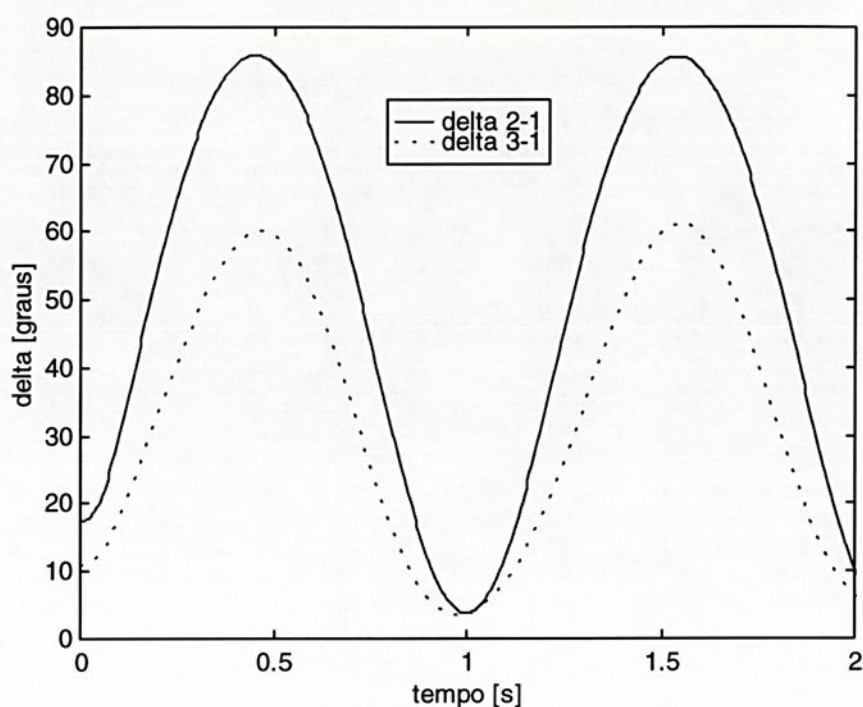


Figura 4.5 – Ângulos dos geradores 2 e 3 em relação ao gerador 1 para um tempo de chaveamento de 0,083 [s] (caso estável).

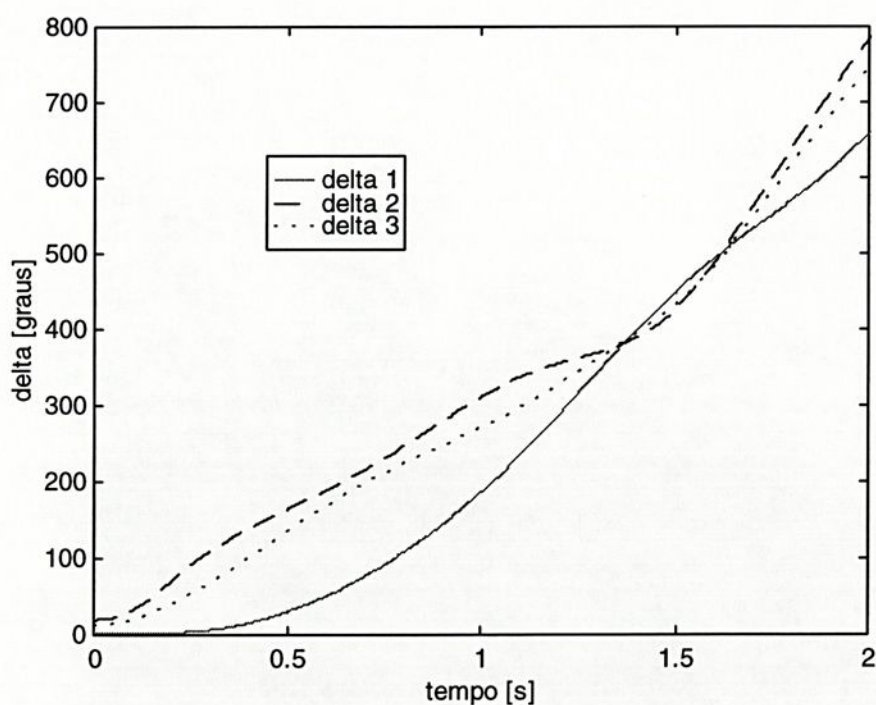


Figura 4.6 – Ângulos dos geradores 1, 2 e 3 para um tempo de chaveamento de 0,160 [s] (caso estável).

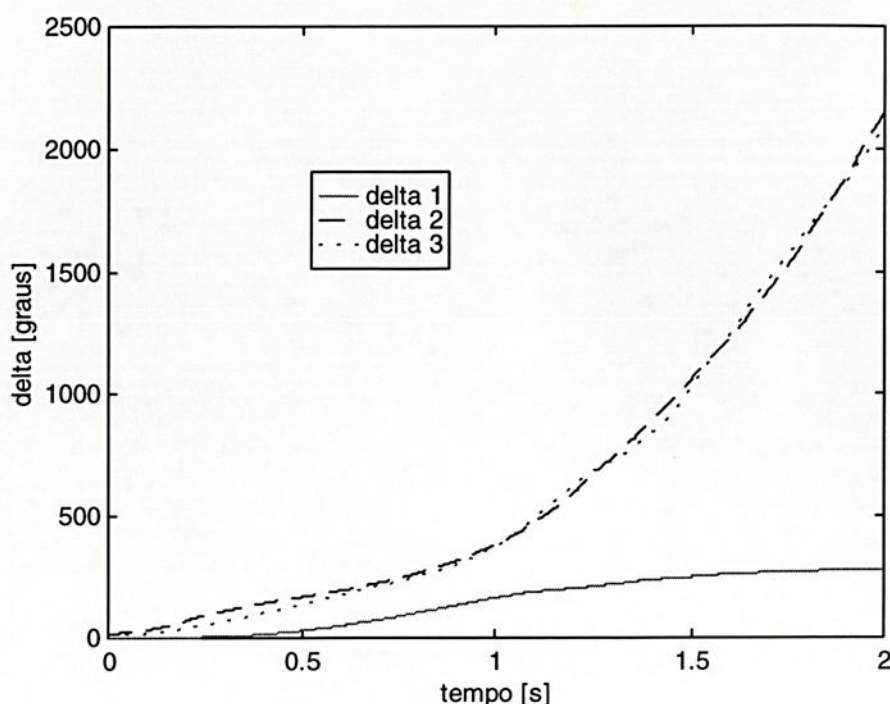


Figura 4.7 – Ângulos dos geradores 1, 2 e 3 para um tempo de chaveamento de 0,165 [s] (caso instável).

Outro sistema estudado foi o sistema apresentado na referência [07]. Este sistema possui 68 barras e 16 geradores, cujos dados se encontram no anexo I.

A seguir são apresentados os dados referentes à contingência estudada e algumas informações sobre as versões do programa utilizado para a análise da estabilidade transitória no sistema de 68 barras.

Contingência: (curto-circuito trifásico)

Localização: linha de transmissão 28–29, próximo ao terminal 29.

Primeira versão do programa (1 caso)

tempo final de simulação: 3,5 [s];

passo de integração: 0,005 [s];

tempo de chaveamento: especificado nos exemplos apresentados a seguir;

tolerância ou precisão do ângulo dos geradores: 0,001 [rad] ou 0,057 [°];

tolerância ou precisão da velocidade dos geradores: 0,01 [rad/s].

Segunda versão do programa (vários casos)

tempo final de simulação: 3,5 [s];

passo de integração: 0,003 [s];

variação do tempo de chaveamento: 0,005 [s];

tolerância ou precisão do ângulo dos geradores: 0,001 [rad] ou 0,057 [°];

tolerância ou precisão da velocidade dos geradores: 0,01 [rad/s].

A segunda versão do programa aplicada ao sistema de 68 barras determina o seguinte intervalo de tempo que contém o tempo crítico.

- o sistema é estável para tempo de chaveamento = 0,065 [s];
- o sistema é instável para tempo de chaveamento = 0,070 [s].

A seguir são apresentadas algumas curvas obtidas com a primeira versão do programa, para simulações com os tempos de chaveamento que determinam o intervalo que contém o tempo crítico, dado anteriormente, referentes à contingência aplicada ao sistema de 68 barras.



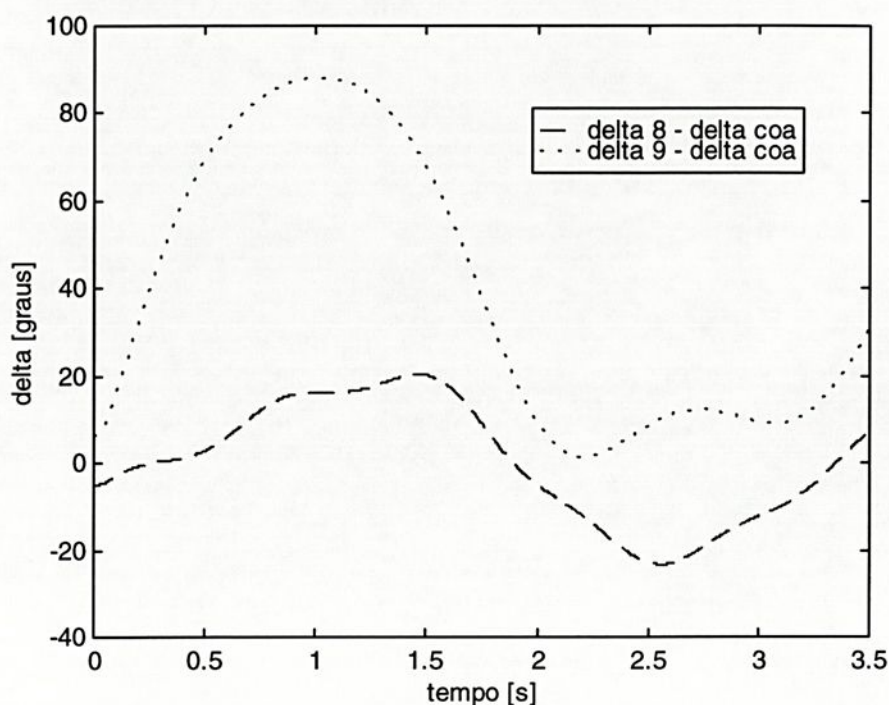


Figura 4.8 – Ângulos dos geradores 8 e 9 (barras 60 e 61), com relação ao centro dos ângulos (δ coa), para um tempo de chaveamento de 0,065 [s] (caso estável).

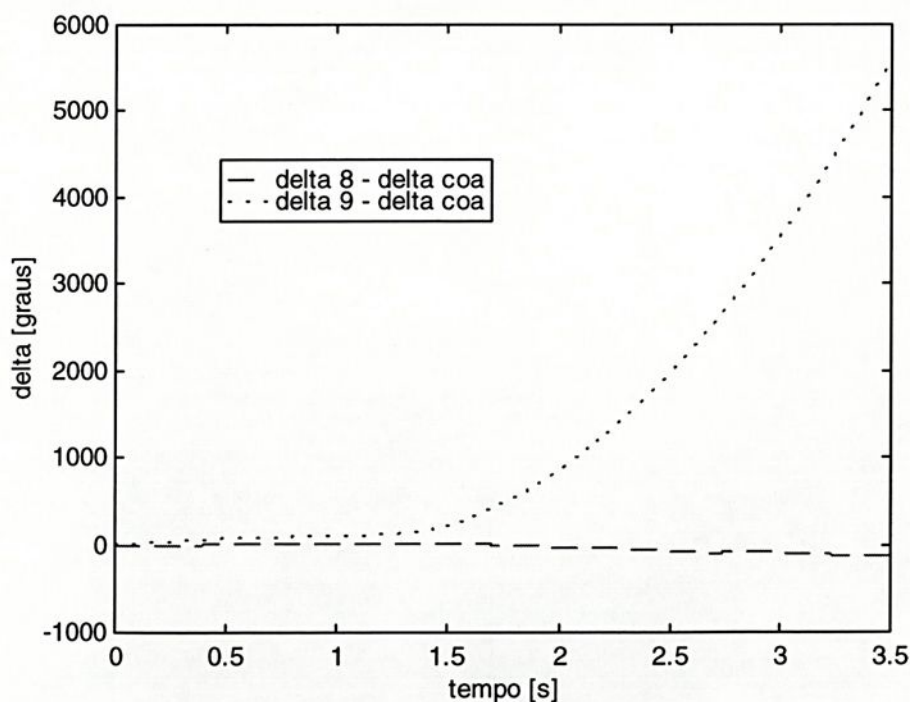


Figura 4.9 – Ângulos dos geradores 8 e 9 (barras 60 e 61), com relação ao centro dos ângulos (δ coa), para um tempo de chaveamento de 0,070 [s] (caso instável).

Também foi realizado um estudo da estabilidade transitória do sistema sul-sudeste brasileiro que possui 320 barras, 470 linhas e 44 geradores. Os dados relativos à contingência aplicada e às duas versões do programa são mostrados a seguir.

Contingência: (curto-circuito trifásico)

Localização: linha de transmissão 402–403, próximo ao terminal 403.

A linha de transmissão especificada liga as usinas hidrelétricas de Ilha Solteira e Água Vermelha, e a falta simulada ocorre próximo à usina de Água Vermelha.

Primeira versão do programa (1 caso)

tempo final de simulação: 3,0 [s];

passo de integração: 0,005 [s];

tempo de chaveamento: especificado nos exemplos apresentados a seguir;

tolerância ou precisão do ângulo dos geradores: 0,001 [rad] ou 0,057 [°];

tolerância ou precisão da velocidade dos geradores: 0,01 [rad/s].

Segunda versão do programa (vários casos)

tempo final de simulação: 3,0 [s];

passo de integração: 0,003 [s];

variação do tempo de chaveamento: 0,005 [s];

tolerância ou precisão do ângulo dos geradores: 0,001 [rad] ou 0,057 [°];

tolerância ou precisão da velocidade dos geradores: 0,01 [rad/s].

Aplicando-se a segunda versão do programa ao sistema sul-sudeste de 320 barras, determina-se o seguinte intervalo de tempo que contém o tempo crítico.

- o sistema é estável para tempo de chaveamento = 0,290 [s];
- o sistema é instável para tempo de chaveamento = 0,295 [s].

A seguir são apresentadas algumas curvas obtidas com a primeira versão do programa para alguns tempos de chaveamento.



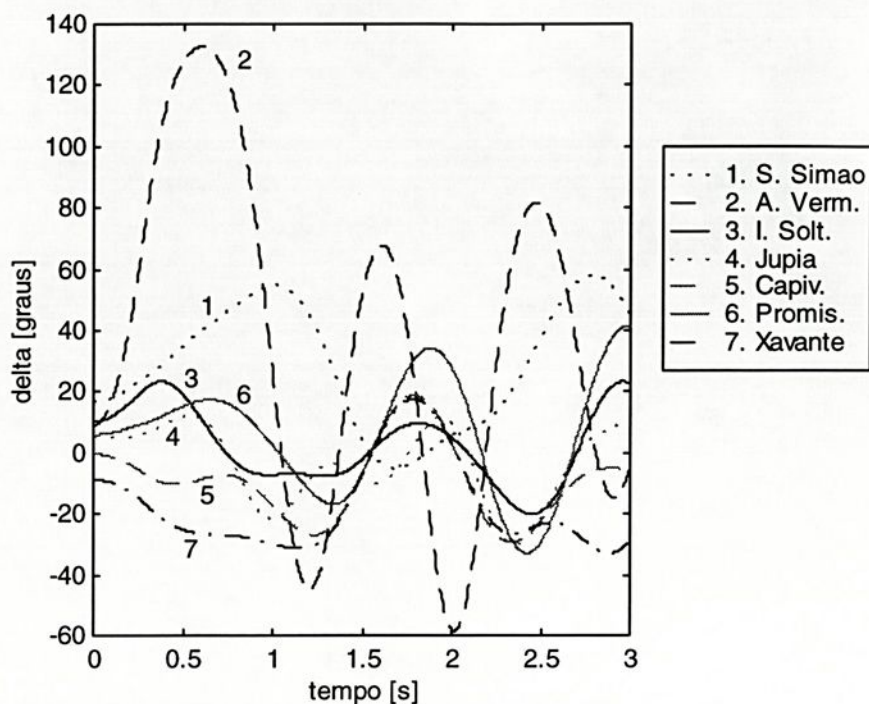


Figura 4.10 – Ângulos de alguns geradores do sistema de 320 barras, com relação ao centro dos ângulos (δ_{coa}), para um tempo de chaveamento de 0,290 [s] (caso estável).

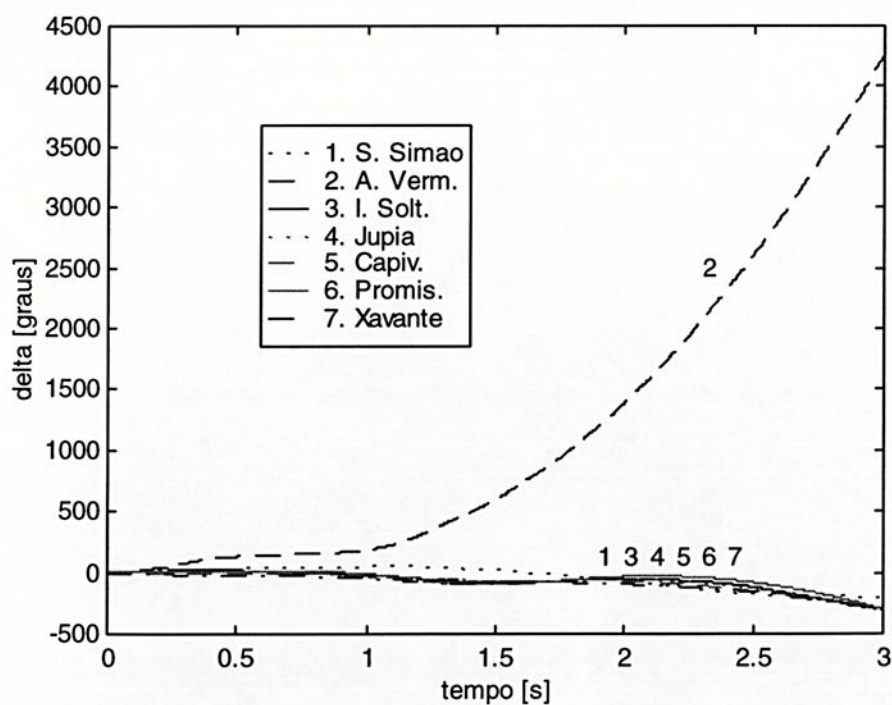


Figura 4.11 – Ângulos de alguns geradores do sistema de 320 barras, com relação ao centro dos ângulos (δ_{coa}), para um tempo de chaveamento de 0,295 [s] (caso instável).

Analisando-se a figura 4.10, verifica-se que o sistema é estável, mas a usina de Água Vermelha sofre grandes oscilações. A figura 4.11 ilustra a instabilidade do sistema, uma vez que as máquinas da usina de Água Vermelha perdem o sincronismo.

4.5 DESEMPENHO DO PROGRAMA NO COMPUTADOR VETORIAL

Para verificar o desempenho do programa para a análise da estabilidade transitória no computador CRAY Y-MP2E/232, foram realizados vários testes com o sistema sudeste brasileiro com 320 barras, 470 linhas e 44 geradores. As cargas foram modeladas como admitância constante e os geradores através do modelo clássico. Esta modelagem básica já permite uma boa avaliação do desempenho de um programa de cálculo de estabilidade transitória com processamento vetorial.

As tabelas 4.1 e 4.2 mostram o desempenho do programa no computador CRAY, utilizando-se a forma de armazenamento e a programação da solução das equações algébricas como descrito em [35], considerando vetores cheios. A tabela 4.1 ilustra os resultados com processamento escalar e a tabela 4.2 com processamento vetorial. Nestas tabelas foram simulados 59 casos usando-se a segunda versão do programa.

Tabela 4.1 – Método Zollenkopf ([35]) na forma escalar.

	Com entrada/saída de dados Tempo [s]	Sem entrada/saída de dados Tempo [s]	Processo iterativo (algébrico/difer.) Tempo [s]
1 caso	1,539	1,273	1,230
59 casos	117,894	117,755	117,711



Tabela 4.2 – Método Zollenkopf ([35]) na forma vetorial.

	Com entrada/saída de dados Tempo [s]	Sem entrada/saída de dados Tempo [s]	Processo iterativo (algébrico/difer.) Tempo [s]
1 caso	1,154	0,975	0,936
59 casos	88,839	88,736	88,695

Uma comparação entre as tabelas 4.1 e 4.2 mostra que a programação utilizada em [35] não é adequada para processamento vetorial, pois o ganho da solução vetorial em relação a escalar é baixo. Isto justifica o desenvolvimento de novas metodologias que têm como objetivo otimizar o uso do processamento vetorial.

As tabelas 4.3 e 4.4 mostram o desempenho do programa no computador CRAY, utilizando-se a forma de armazenamento e a programação da solução das equações algébricas segundo a metodologia DEA, considerando vetores esparsos; tanto na FF como na FB as posições diferentes de zero são as das barras de geração. A tabela 4.3 ilustra os resultados com processamento escalar e a tabela 4.4 com processamento vetorial.

Tabela 4.3 – Método DEA na forma escalar.

	Com entrada/saída de dados Tempo [s]	Sem entrada/saída de dados Tempo [s]	Processo iterativo (algébrico/difer.) Tempo [s]
1 caso	1,015	0,752	0,654
59 casos	63,589	63,449	63,350



Tabela 4.4 – Método DEA na forma vetorial.

	Com entrada/saída de dados Tempo [s]	Sem entrada/saída de dados Tempo [s]	Processo iterativo (algébrico/difer.) Tempo [s]
1 caso	0,378	0,196	0,135
59 casos	13,236	13,133	13,071

As tabelas 4.5 e 4.6 mostram o desempenho da metodologia DEN para realizar as mesmas simulações feitas nas tabelas 4.3 e 4.4.

Tabela 4.5 – Método DEN na forma escalar.

	Com entrada/saída de dados Tempo [s]	Sem entrada/saída de dados Tempo [s]	Processo iterativo (algébrico/difer.) Tempo [s]
1 caso	1,064	0,800	0,673
59 casos	65,299	65,159	65,030

Tabela 4.6 – Método DEN na forma vetorial.

	Com entrada/saída de dados Tempo [s]	Sem entrada/saída de dados Tempo [s]	Processo iterativo (algébrico/difer.) Tempo [s]
1 caso	0,484	0,306	0,188
59 casos	18,084	17,980	17,861

Comparando-se as tabelas 4.1 a 4.6, conclui-se que a solução vetorial utilizando-se a metodologia DEA produz ganhos significativos, principalmente quando o número de casos a simular for muito grande. Pode-se concluir também que o desempenho da

metodologia DEA aplicada ao programa para a análise da estabilidade transitória de sistemas de energia elétrica justifica o esforço computacional extra na preparação dos *slices* e da matriz da última partição.



CAPÍTULO V

CONCLUSÃO

Neste trabalho, implementou-se um programa computacional que simula o comportamento dos sistemas de energia elétrica nos primeiros instantes após a ocorrência de uma contingência severa. O programa foi elaborado para otimizar o tempo de CPU quando da sua execução em ambiente dotado de processamento vetorial.

A técnica utilizada para implementar o programa foi a simulação passo a passo, na qual foram resolvidas as equações que descrevem o comportamento do sistema no domínio do tempo. O problema foi abordado considerando o esquema alternado implícito, em que as equações diferenciais resultantes da modelagem da parte dinâmica do sistema foram transformadas em equações algébricas, através do método trapezoidal de integração.

Basicamente, o programa de estabilidade implementado pode ser dividido nas seguintes etapas:

- aquisição de dados (leitura dos dados do sistema);
- preparação dos dados (organização dos dados para serem processados vetorialmente durante o período de duração da contingência e após a eliminação da mesma);
- resolução das equações algébricas (modelagem da parte estática do sistema);
- resolução das equações diferenciais (modelagem da parte dinâmica do sistema).

A parte do programa que oferece maiores dificuldades à vetorização é a resolução das equações algébricas. Por isso, foi dada uma especial atenção à resolução destas equações. Primeiramente, foram estudadas algumas metodologias já existentes e depois sugeriram-se algumas formas de melhorias para a resolução das equações algébricas em ambiente vetorial.

As primeiras metodologias implementadas basearam-se no método da matriz W particionada, onde se conseguiu bom desempenho para as primeiras partições, mas o processamento vetorial quase não existia para as últimas partições. Com a finalidade de trabalhar com vetores longos também nas últimas partições procurou-se agrupar estas em uma matriz cheia.

Implementaram-se, também, metodologias baseadas no método DBSA, que se mostraram mais eficientes do que as baseadas no método da matriz W por produzirem vetores mais longos e, portanto, exigiram menos “*do loops*” durante a solução.

Todas estas metodologias foram testadas considerando-se vetores cheios e também esparsos. Vários testes foram realizados, utilizando-se sistemas de energia elétrica reais, e permitiram concluir que a melhor solução utiliza o método DBSA e agrupa os últimos nós em uma matriz cheia.

Após esta etapa, buscou-se implementar as melhores soluções do problema $Ax = b$, em um programa de cálculo da estabilidade transitória. O programa de estabilidade foi desenvolvido utilizando-se os modelos mais simplificados, pois o principal objetivo foi observar a influência da entrada e saída de dados, preparação para processamento vetorial, etc, no desempenho das metodologias testadas para solução de $Ax = b$.

A implementação vetorial mostrou que o cálculo da estabilidade transitória pode ser feito de uma maneira rápida e isto pode permitir um ganho bastante significativo nas simulações para planejamento e operação não em tempo real, bem como pode tornar possível obtenção de resultados para análises em tempo real. Algumas melhorias podem ser realizadas no programa implementado, entre elas sugerem-se:



- melhorar a formação das pseudocolunas e dos *slices*;
- implementar técnicas de refatoração parcial de matrizes, para permitir uma simulação mais rápida de uma série de casos distintos;
- incorporar modelos mais detalhados para representar geradores e cargas.



REFERÊNCIAS BIBLIOGRÁFICAS

- [01] ALVARADO, F. L., YU, D. C., BETANCOURT, R. Partitioned sparse A^{-1} methods. *IEEE Transactions on Power Systems*, v. 5, n. 2, p. 452–459, May 1990.
- [02] ANDERSON, P. M., FOUAD, A. A. Power system control and stability. *IEEE Press*, 1994.
- [03] AYKANAT, C., OZGU, O., GUVEN, N. Algorithms for efficient vectorization of repeated sparse power system. *IEEE Transactions on Power Systems*, v. 10, n.1, p. 448–456, Feb. 1995.
- [04] BETANCOURT, R. An heuristic ordering algorithm for partial matrix refactorization. *IEEE Transactions on Power Systems*, v. 3, n. 3, p. 1181–1187, Aug. 1988.
- [05] BORGES, C. L. T., COUTINHO, A. L. G. A., FALCÃO, D. M. Solução do fluxo de potência em ambiente vetorial usando o método do gradiente bi-conjugado estabilizado. *11^a Congresso Brasileiro de Automática*, v. 1, p. 161–166, set. 1996.
- [06] CASTRO, M., MORELATO, A. Implementations of an on-line transient stability simulations engine on distributed computing environments. *V SEPOPE*, Recife, Maio 1996.
- [07] CHOW, J. H., editor. Time-scale modelling of dynamic networks with applications to power systems. *Lecture Notes in Control and Information Sciences*. Springer-Verlag, 1982. Series editors BALAKRISHNAN, A. V. and THOMAS, M.

- [08] COSTA, V. F. Análise de estabilidade transitória em sistemas de energia elétrica. Campinas (SP), Faculdade de Engenharia de Campinas, Tese (mestrado) – FEC/UNICAMP, 1981.
- [09] CROW, M. L., ILIÇ, M. The parallel implementation of the waveform relaxation method for transient stability simulation. *IEEE Transactions on Power Systems*, v. 5, p. 922-932, Aug. 1990.
- [10] DECKER, I. C., FALCÃO, D. M., KASZKUREWICZ, E. Parallel implementation of a power system dynamic simulation methodology using the conjugate gradient method. *Power Industry Computer Applications Conference (PICA)*, p. 245-252, May 1991.
- [11] DOMMEL, H.W., SATO, N. Fast transient stability solutions. *IEEE Transactions on Power Apparatus Systems*, PAS -91 (4), p. 1643-1650, July/Aug. 1972.
- [12] EL-SHARKAWI, M. A., MANSOUR, Y., VAAHEDI, E. Large scale dynamic security screening and ranking using neural networks. *V SEPOPE*, Recife, Maio 1996.
- [13] ENNS, M. K., TINNEY, W. F., ALVARADO, F. L. Sparse matrix inverse factors. *IEEE Transactions on Power Systems*, v. 5, n. 2, p. 466-473, May 1990.
- [14] FOUAD, A. A., VITTAL, V. Power system transient stability analysis using the transient energy function. *Prentice-Hall, Inc., Englewood Cliffs, New Jersey USA*, 1992.
- [15] GRANELLI, G. P., MONTAGNA, M., PASINI, G. L., MARANNINO, P. A W -matrix based fast decoupled load flow for contingency studies on vector computers. *IEEE Transactions on Power Systems*, v. 8, n. 3, p. 946-953, Aug. 1993.
- [16] HUANG, H. S., LU, C. N. Efficient storage scheme and algorithms for W -matrix vector multiplication on vector computers. *IEEE Transactions on Power Systems*, v. 9, n. 2, p. 1083-1094, May 1994.



- [17] LASCALA, M., BOSE, A., TYLAVSKY, D. J., CHAI, J. S. A highly parallel method for transient stability analysis. *IEEE Transactions on Power Systems*, v. 5, p. 1439-1446, Nov. 1990.
- [18] LASCALA, M., BRUCOLI, M., TORELLI, F., TROVATO, M. A gauss-jacobi-block-newton method for parallel transient stability analysis. *IEEE Transactions on Power Systems*, v. 5, p. 1168-1177, Nov. 1990.
- [19] LASCALA, M., SBLENDORIS, G., BOSE, A., WU, J. Comparison of algorithms for transient stability simulations on shared and distributed memory multiprocessors. *IEEE PES Winter Meeting*, Jan. 1996.
- [20] LIN, S. L., VAN NESS, J. E. Parallel solution of sparse algebraic equations. *IEEE Transactions on Power Systems*, v. 9, n. 2, p. 743-799, May 1994.
- [21] MONTICELLI, A. J. Fluxo de carga em redes de energia elétrica. São Paulo: Edgard Blücher Ltda, 1983.
- [22] MORELATO, A. Simulação da dinâmica de curto prazo em sistemas de energia elétrica: comportamento do esquema alternado implícito. *5^o Congresso Brasileiro de Automática / 1^o Congresso Latino Americano de Automática*, p. 172-176, 1984.
- [23] MORELATO, A., AMANO, M., KOKAI, Y. Combining direct and inverse factors for solving sparse network equations in parallel. *IEEE Transactions on Power Systems*, v. 9, n. 4, p. 1942-1948, Nov. 1994.
- [24] PADILHA, A. Cálculo da estabilidade transitória em sistemas de energia elétrica utilizando esquema simultâneo implícito. Campinas (SP), Faculdade de Engenharia de Campinas, Tese (mestrado) – FEC/UNICAMP, Mar. 1986.
- [25] PADILHA, A., DAG, H., ALVARADO, F. A Parallel virtual machine approach to transient stability analysis. *28th North American Power Symposium MIT*, Nov. 1996.



- [26] PADILHA, A., MORELATO, A. A W -matrix methodology for solving sparse network equations on multiprocessor computers. *IEEE Transactions on Power Systems*, v. 7, n. 3, p. 1023–1030, Aug. 1992.
- [27] PAI, M. A. Power system stability: Analysis by the direct method of Lyapunov. *North Holland Control Series*, 1981.
- [28] STOTT, B. Power system dynamic response calculations. *Proceedings of the IEEE*, v. 67, n. 2, p. 219–241, Feb. 1979.
- [29] TINNEY, W. F., BRANDWAIN, V., CHAN, S. M. Sparse vector methods. *IEEE Transactions on Power Apparatus and Systems*, v. 104, n. 2, p. 295–301, Feb. 1985.
- [30] TINNEY, W. F., WALKER, J. W. Direct solutions of sparse network equations by optimally ordered triangular factorization. *Proceedings of the IEEE*, v. 55, p. 1801–1819, 1967.
- [31] TYLAVSKY, D. J., BOSE, A., ALVARADO, F. L., BETANCOURT, R., CLEMENTS, K., HEYDT, G., HUANG, G., ILIC, M., LASCALA, M., PAI, M., POTTLE, C., TALUKDAR, S., VAN NESS, J., WU, F. Parallel processing in power system computation. *IEEE Transactions on Power Systems*, v. 7, n. 2, p. 629–637, May 1992.
- [32] VUONG, G. T., CHAHINE, R., GRANELLI, G. P., MONTAGNA, M. Dependency-based algorithms for vector processing of sparse matrix forward/backward substitutions. *IEEE Transactions on Power Systems*, v. 11, n. 1, p. 198–204, Feb. 1996.
- [33] WING, O., HUANG, J. W. A computational model of parallel solution of linear equations. *IEEE Transactions on Computers*, v. 29, n. 7, p. 632–638, July 1980.
- [34] YOUNG, C. C. The synchronous machine. *IEEE Tutorial course. Modern concepts of power system dynamics*, p. 11 – 24, New York, May 1970.



- [35] ZOLLENKOPF, K. Bi-factorisation basic computational algorithm and programming techniques. In Reid, J. K. (ed.) *Large Sparse Sets of Linear Equations*, Academic Press, p. 75–96, 1971.

N.B. Para a realização desta dissertação, assim como do texto, foram utilizados os *software Windows'95*, editor *Word 7*, compilador *Fortran*, *Matlab* e suas principais *toolboxes*, versões para microcomputador. Todos estes *software* são aquisições oficiais da Unesp ou do Curso de Pós-Graduação em Engenharia Elétrica - FEIS - Unesp. Foi utilizado, também, o compilador *Fortran* do computador CRAY.



ANEXO I

DADOS DO SISTEMA DA REFERÊNCIA [07]

(68 BARRAS E 16 GERADORES)



Dados de barra (caso base)

Núm.	=	número da barra.
Tipo	=	tipo da barra (0 = carga, 1 = geração e 2 = referência).
Tensão	=	tensão da barra dada em pu.
Ângulo	=	ângulo da barra dado em graus.
P carga	=	potência ativa da carga dada em pu.
Q carga	=	potência reativa da carga dada em pu.
P gera.	=	potência ativa de geração dada em pu.
Q gera.	=	potência reativa de geração dada em pu.

Núm.	Tipo	Tensão	Ângulo	P carga	Q carga	P gera.	Q gera.
1	0	1,0590	6,61	2,53	1,19	0,00	0,00
2	0	1,0520	8,43	0,00	0,00	0,00	0,00
3	0	1,0330	5,43	3,22	0,02	0,00	0,00
4	0	1,0060	4,31	5,00	1,84	0,00	0,00
5	0	1,0070	5,25	0,00	0,00	0,00	0,00
6	0	1,0090	5,93	0,00	0,00	0,00	0,00
7	0	0,9990	3,66	2,34	0,84	0,00	0,00
8	0	0,9990	3,12	5,22	1,77	0,00	0,00
9	0	1,0390	2,58	1,04	1,25	0,00	0,00
10	0	1,0180	8,45	0,00	0,00	0,00	0,00
11	0	1,0140	7,60	0,00	0,00	0,00	0,00
12	0	1,0550	7,62	0,09	0,88	0,00	0,00
13	0	1,0160	7,78	0,00	0,00	0,00	0,00
14	0	1,0130	6,24	0,00	0,00	0,00	0,00
15	0	1,0170	6,14	3,20	1,53	0,00	0,00
16	0	1,0330	7,68	3,29	0,32	0,00	0,00
17	0	1,0360	6,59	0,00	0,00	0,00	0,00
18	0	1,0340	5,72	1,58	0,30	0,00	0,00
19	0	1,0500	12,27	0,00	0,00	0,00	0,00
20	0	0,9900	10,84	6,80	1,03	0,00	0,00
21	0	1,0330	10,31	2,74	1,15	0,00	0,00
22	0	1,0500	15,00	0,00	0,00	0,00	0,00
23	0	1,0450	14,71	2,48	0,85	0,00	0,00
24	0	1,0390	7,85	3,09	-0,92	0,00	0,00
25	0	1,0600	9,70	2,24	0,47	0,00	0,00
26	0	1,0560	8,20	1,39	0,17	0,00	0,00

27	0	1,0430	6,31	2,81	0,76	0,00	0,00
28	0	1,0520	11,33	2,06	0,28	0,00	0,00
29	0	1,0510	13,97	2,84	0,27	0,00	0,00
30	0	1,0540	6,07	0,00	0,00	0,00	0,00
31	0	1,0570	8,63	0,00	0,00	0,00	0,00
32	0	1,0510	10,96	0,00	0,00	0,00	0,00
33	0	1,0560	7,47	1,12	0,00	0,00	0,00
34	0	1,0650	2,54	0,00	0,00	0,00	0,00
35	0	1,0140	2,53	0,00	0,00	0,00	0,00
36	0	1,0420	-0,85	1,02	-0,19	0,00	0,00
37	0	1,0290	-6,80	60,00	3,00	0,00	0,00
38	0	1,0560	8,68	0,00	0,00	0,00	0,00
39	0	1,0060	-8,44	2,67	0,13	0,00	0,00
40	0	1,0680	15,22	0,66	0,24	0,00	0,00
41	0	0,9990	44,49	10,00	2,50	0,00	0,00
42	0	0,9990	38,92	11,50	2,50	0,00	0,00
43	0	1,0150	-7,61	0,00	0,00	0,00	0,00
44	0	1,0140	-7,64	2,68	0,05	0,00	0,00
45	0	1,0180	2,53	2,08	0,21	0,00	0,00
46	0	1,0320	9,65	1,51	0,28	0,00	0,00
47	0	1,0740	7,36	2,03	0,33	0,00	0,00
48	0	1,0770	9,28	2,41	0,02	0,00	0,00
49	0	1,0120	12,88	1,64	0,29	0,00	0,00
50	0	1,0120	19,33	1,00	-1,47	0,00	0,00
51	0	1,0220	6,52	3,37	-1,22	0,00	0,00
52	0	0,9930	38,59	24,70	1,23	0,00	0,00
53	1	1,0450	10,85	0,00	0,00	2,50	1,15
54	1	0,9800	14,41	0,00	0,00	5,45	1,84
55	1	0,9830	16,44	0,00	0,00	6,50	1,99
56	1	0,9970	17,49	0,00	0,00	6,32	1,07
57	1	1,0110	16,01	0,00	0,00	5,05	1,62
58	1	1,0500	20,34	0,00	0,00	7,00	2,20
59	1	1,0630	22,56	0,00	0,00	5,60	0,99
60	1	1,0300	16,45	0,00	0,00	5,40	0,00
61	1	1,0250	20,79	0,00	0,00	8,00	0,05
62	1	1,0100	15,90	0,00	0,00	5,00	0,06
63	1	1,0000	18,35	0,00	0,00	10,00	-0,17
64	1	1,0160	4,86	0,00	0,00	13,50	2,49
65	2	1,0110	0,00	0,00	0,00	35,91	8,75
66	1	1,0000	46,02	0,00	0,00	17,85	0,62
67	1	1,0000	39,78	0,00	0,00	10,00	0,67
68	1	1,0000	45,53	0,00	0,00	40,00	4,60

Dados das linhas de transmissão e dos transformadores (caso base)

Nó inic.	=	nó inicial da linha de transmissão ou do transformador.
Nó final	=	nó final da linha de transmissão ou do transformador.
Resist.	=	resistência série dada em porcentagem.
Reat.	=	reatância indutiva série dada em porcentagem.
Suscep.	=	susceptância capacitiva paralela total dada em porcentagem.
Rel. transf.	=	relação de transformação do transformador em fase.

Nó inic.	Nó final	Resist.	Reat.	Suscep.	Rel. transf.
1	2	0,35	4,11	69,87	0,0000
1	30	0,08	0,74	48,00	0,0000
2	3	0,13	1,51	25,72	0,0000
2	25	0,70	0,86	14,60	0,0000
2	53	0,00	1,81	0,00	1,0250
3	4	0,13	2,13	22,14	0,0000
3	18	0,11	1,33	21,38	0,0000
4	5	0,08	1,28	13,42	0,0000
4	14	0,08	1,29	13,82	0,0000
5	6	0,02	0,26	4,34	0,0000
5	8	0,08	1,12	14,76	0,0000
6	7	0,06	0,92	11,30	0,0000
6	11	0,07	0,82	13,89	0,0000
6	54	0,00	2,50	0,00	1,0700
7	8	0,04	0,46	7,80	0,0000
8	9	0,23	3,63	38,04	0,0000
9	30	0,19	1,83	29,00	0,0000
10	11	0,04	0,43	7,29	0,0000
10	13	0,04	0,43	7,29	0,0000
10	55	0,00	2,00	0,00	1,0700
12	11	0,16	4,35	0,00	1,0600
12	13	0,16	4,35	0,00	1,0600
13	14	0,09	1,01	17,23	0,0000
14	15	0,18	2,17	36,60	0,0000
15	16	0,09	0,94	17,10	0,0000
16	17	0,07	0,89	13,42	0,0000
16	19	0,16	1,95	30,40	0,0000
16	21	0,08	1,35	25,48	0,0000
16	24	0,03	0,59	6,80	0,0000
17	18	0,07	0,82	13,19	0,0000

17	27	0,13	1,73	32,16	0,0000
19	20	0,07	1,38	0,00	1,0600
19	56	0,07	1,42	0,00	1,0700
20	57	0,09	1,80	0,00	1,0090
21	22	0,08	1,40	25,65	0,0000
22	23	0,06	0,96	18,46	0,0000
22	58	0,00	1,43	0,00	1,0250
23	24	0,22	3,50	36,10	0,0000
23	59	0,05	2,72	0,00	0,0000
25	26	0,32	3,23	53,10	0,0000
25	60	0,06	2,32	0,00	1,0250
26	27	0,14	1,47	23,96	0,0000
26	28	0,43	4,74	78,02	0,0000
26	29	0,57	6,25	102,90	0,0000
28	29	0,14	1,51	24,90	0,0000
29	61	0,08	1,56	0,00	1,0250
9	30	0,19	1,83	29,00	0,0000
9	36	0,22	1,96	34,00	0,0000
9	36	0,22	1,96	34,00	0,0000
36	37	0,05	0,45	32,00	0,0000
34	36	0,33	1,11	145,00	0,0000
35	34	0,01	0,74	0,00	0,9460
33	34	0,11	1,57	20,20	0,0000
32	33	0,08	0,99	16,80	0,0000
30	31	0,13	1,87	33,30	0,0000
30	32	0,24	2,88	48,80	0,0000
1	31	0,16	1,63	25,00	0,0000
31	38	0,11	1,47	24,70	0,0000
33	38	0,36	4,44	69,30	0,0000
38	46	0,22	2,84	43,00	0,0000
46	49	0,18	2,74	27,00	0,0000
1	47	0,13	1,88	131,00	0,0000
47	48	0,25	2,68	40,00	0,0000
47	48	0,25	2,68	40,00	0,0000
48	40	0,20	2,20	128,00	0,0000
35	45	0,07	1,75	139,00	0,0000
37	43	0,05	2,76	0,00	0,0000
43	44	0,01	0,11	0,00	0,0000
44	45	0,25	7,30	0,00	0,0000
39	44	0,00	4,11	0,00	0,0000
39	45	0,00	8,39	0,00	0,0000
45	51	0,04	1,05	72,00	0,0000
50	52	0,12	2,88	206,00	0,0000
50	51	0,09	2,21	162,00	0,0000
49	52	0,76	11,41	116,00	0,0000
52	42	0,40	6,00	225,00	0,0000
42	41	0,40	6,00	225,00	0,0000



41	40	0,60	8,40	315,00	0,0000
31	62	0,00	2,60	0,00	1,0400
32	63	0,00	1,30	0,00	1,0400
36	64	0,00	0,75	0,00	1,0400
37	65	0,00	0,33	0,00	1,0400
41	66	0,00	0,15	0,00	1,0000
42	67	0,00	0,15	0,00	1,0000
52	68	0,00	0,30	0,00	1,0000
1	27	3,20	32,00	41,00	0,0000

Dados dos geradores

- Núm. = número da barra de geração.
- x'_d = reatância transitória de eixo direto dada em pu.
- H = constante de inércia da máquina síncrona dada em segundos.

Núm.	x'_d	H
53	0,03100	42,00000
54	0,06970	30,20000
55	0,05310	35,80000
56	0,04360	28,60000
57	0,06600	26,00000
58	0,05000	34,80000
59	0,04900	26,40000
60	0,05700	24,30000
61	0,05700	34,50000
62	0,04570	31,00000
63	0,01800	28,20000
64	0,03100	92,30000
65	0,00550	248,00000
66	0,00290	300,00000
67	0,00290	300,00000
68	0,00710	225,00000

DEPARTAMENTO DE ENGENHARIA ELÉTRICA
AV. BRASIL, 56 - CAIXA POSTAL 31
15378-000 ILHA SOLTEIRA - SP

