

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**UM ESTUDO DA SIMULAÇÃO DA DINÂMICA DE  
SISTEMAS DE ENERGIA ELÉTRICA USANDO  
COMPUTADORES PARALELOS DE ALTO  
DESEMPENHO**

‘ Dissertação submetida como parte dos requisitos para a  
obtenção do grau de Mestre em Engenharia Elétrica ‘

**José Luiz Lima de Azevedo**

Florianópolis, março de 1996.

**UM ESTUDO DA SIMULAÇÃO DA DINÂMICA DE SISTEMAS  
DE ENERGIA ELÉTRICA USANDO COMPUTADORES  
PARALELOS DE ALTO DESEMPENHO**

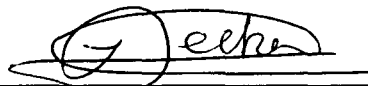
**José Luiz Lima de Azevedo**

Esta dissertação foi julgada adequada para a obtenção do título de

**Mestre em Engenharia Elétrica**

**Área de concentração em Planejamento  
de Sistemas de Energia Elétrica**

e aprovada em sua forma final pelo programa de Pós-Graduação



Prof. Ildemar Cassana Decker, D.Sc.  
Orientador



Prof. Enio Valmor Kassick, Dr.  
Coordenador do Curso de Pós-Graduação  
em Engenharia Elétrica

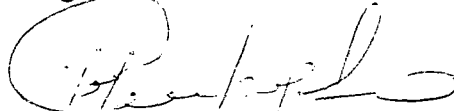
**Banca Examinadora :**



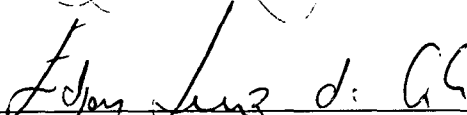
Prof. Ildemar Cassana Decker, D.Sc.  
Presidente



Prof. Aguinaldo Silveira e Silva, Ph.D.



Prof. Jorge Mário Campagnolo, D.Sc.



Prof. Edson Luiz da Silva, D.Eng.

Dedicada a minha esposa Rosane pelo amor,  
carinho e compreensão demonstrados neste  
meu longo período de ausência.

À memória de minha avó e mãe Cenira,  
que mesmo nos tendo deixado durante  
a realização deste trabalho sabe que,  
sem o seu sacrifício e dedicação,  
ele não teria sido possível.

À memória de meu sogro e amigo Otacilio,  
que mesmo tendo nos deixado durante  
a realização deste trabalho, sabe que  
permitiu que eu alcançasse o meu  
tesouro mais precioso : sua filha  
e minha esposa Rosane

## **AGRADECIMENTOS**

Ao professor, orientador e amigo Ildemar Cassana Decker, pelas longas horas de dedicação, orientação e apoio, sem as quais este trabalho não teria sido concluído.

A todos os professores do Curso de Pós-Graduação em Engenharia Elétrica da UFSC que me transmitiram os seus conhecimentos, em especial aos professores Ildemar, Aguinaldo, Simões e Hans.

Ao colega Maximiliano Orfali, pelo seu apoio quando de minha chegada a Florianópolis, e, principalmente pela sua sincera amizade demonstrada neste longo período em que convivemos juntos.

Ao colega Alessandro Manzoni, pela sua amizade e pelos longas discussões que tivemos a respeito de nossos trabalhos, que certamente serviram para enriquecer o conteúdo de ambos.

Ao colega Marcelo Felsky e família, pela amizade que sempre demonstraram ao longo da elaboração deste trabalho.

Ao colega Nixon Savaris, pelo seu apoio e amizade dentro do Labplan.

Aos meus cunhados Daniel e Nilza e a minha sobrinha Daniela, pelo apoio e carinho ministrados a minha esposa, durante a minha ausência de casa.

A minha amiga Luciane Ferreira e família, pelo apoio, carinho e amizade.

Aos colegas Elvis e André do NPD da UFSC, pelo seu apoio ao longo deste trabalho.

Ao pessoal anônimo do LNCC, pelo apoio e ajuda na solução de inúmeros problemas ocorridos durante os testes computacionais lá realizados.

A todos os professores, funcionários, bolsistas e colegas de mestrado/doutorado do Labplan e Labspot, pelas horas de convívio dentro e fora da universidade, cujas lembranças guardarei para sempre.

A Fundação Universidade de Rio Grande (FURG) e ao CNPq, pelo seu apoio financeiro.

# SUMÁRIO

	Página
Lista de Figuras	xiv
Lista de Tabelas	xvii
Nomenclatura	xix
Operadores	xxv
Abreviaturas	xxvi
Resumo	xxix
Abstract	xxx
<b>Capítulo 1 - Introdução</b>	<b>1</b>
1.1. A Utilização de Métodos Iterativos	2
1.2. Técnicas de Paralelização	3
1.3. Aspectos Principais do Trabalho	5
1.4. Estrutura do Trabalho	6
<b>Capítulo 2 - Modelagem Matemática e Técnicas de Solução</b>	<b>7</b>
2.1. Introdução	7
2.2. Modelagem Matemática	7
2.3. Estudos de Estabilidade	10

---

2.4. Esquemas de Solução	11
2.4.1. Os Métodos de Integração	12
2.4.2. Esquemas Básicos de Solução	13
2.5. O Esquema Alternado Implícito ( AI )	14
2.5.1. Formas de Solução da Rede	17
2.6. O Esquema Simultâneo Implícito - SI	18
2.6.1. Resolução da Equação Jacobiana	20
2.7. A SDSEE em Computadores Paralelos	22
2.7.1. Paralelização no Espaço	23
2.7.2. Paralelização no Tempo	23
2.7.3. Paralelização Simultânea no Espaço e no Tempo ( PET )	24
2.7.4. Relaxação da Forma de Onda ( RFO )	24
2.8. Comentários	25
<b>Capítulo 3 - Métodos do Tipo Gradiente Conjugado Pré-Condicionados</b>	<b>26</b>
3.1. Introdução	26
3.2. O Método Gradiente Conjugado	26
3.3. O Método Gradiente Conjugado Pré-Condicionado	30
3.4. Técnicas de Pré-condicionamento	33
3.4.1. Pré-condicionamento diagonal	33



---

3.4.2. Pré-condicionamento por Fatoração LU Incompleta	34
3.4.3. Pré-condicionamento Bloco-diagonal	34
3.4.4. Pré-condicionamento por Série Truncada de Neumann	35
3.4.5. Pré-condicionamento por Série Truncada de Maclaurin	35
3.4.6. Pré-condicionamento Polinomial	37
3.4.7. Pré-condicionamento de Dois Níveis	37
3.5. Métodos Tipo Gradiente Conjugado	38
3.6. Aplicação de Métodos do Tipo Gradiente Conjugado a Sistemas de Energia Elétrica	39
3.7. Comentários	40
<b>Capítulo 4 - Decomposição de Sistemas de Energia Elétrica</b>	<b>42</b>
4.1. Introdução	42
4.2. Técnicas de Decomposição	42
4.3. A Metodologia do Algoritmo DECOSEE	43
4.3.1. Princípios Básicos	43
4.3.2. O <i>Ranking</i> Nodal	44
4.3.3. A Escolha dos Nós Sementes	45
4.3.4. A Formação dos Subsistemas	45
4.3.5. O Algoritmo DECOSEE	49
4.4. Um Exemplo de Aplicação	50

---

4.5. Comentários	53
<b>Capítulo 5 - Avaliação dos Métodos de Pré-condicionamento</b>	<b>54</b>
5.1. Introdução	54
5.2. Sistemas de Energia Elétrica Utilizados na Avaliação	54
5.3. Análise de Autovalores	55
5.3.1. Matriz Admitância de Barras	55
5.3.2. Matriz $R$ do Pré-condicionador por Série Truncada de Maclaurin - PCS	58
5.3.3. Comentários	61
5.3.4. O Método da Compensação	61
5.4. Análise pelo Número de Condição	63
5.4.1. Sistema de 9 barras	64
5.4.2. Sistema de 45 barras	65
5.4.3. Sistema de 118 barras	65
5.4.4. Sistema de 340 barras	66
5.4.5. Sistema de 616 barras	67
5.4.6. Sistema de 9 barras pós-falta compensado	67
5.4.7. Comentários	67
5.5. Análise do Comportamento de Convergência	68
5.5.1. Comentários	70

---

5.5.2. Comportamento dos Resíduos do Método GCP	70
5.5.3. O Método COGC	73
5.5.4. O Método Bi-CGSTAB	73
5.6. Comentários Finais	74
<b>Capítulo 6 - Plataformas Computacionais Paralelas</b>	<b>76</b>
6.1. Introdução	76
6.2. Conceitos Importantes em Processamento Paralelo	76
6.2.1. Paralelismo e <i>Pipelining</i>	77
6.2.2. <i>Speed-up</i>	77
6.2.3. A Lei de Amdahl	78
6.2.4. A Lei de Gustafson-Barsis	79
6.2.5. As Arquiteturas Computacionais	79
6.2.6. Latência	81
6.2.7. Granularidade	81
6.2.8. Modelos de Programação	82
6.2.9. Transparência	83
6.2.10. Portabilidade	83
6.2.11. Escalabilidade	83
6.3. Plataformas Utilizadas nas Simulações	83

---

6.3.1. Comunicação Internodal no IBM/SP	85
6.3.2. Tipos de Processamento no IBM/SP	87
6.4. Plataformas Usadas na Comparação de Desempenho	88
6.5. Plataformas Utilizadas em Outros Trabalhos	89
6.6. Síntese das Características Principais	89
6.7. Comentários	91
<b>Capítulo 7 - Implementações Computacionais</b>	<b>92</b>
7.1. Introdução	92
7.2. Algoritmos Implementados	92
7.2.1. O Algoritmo Paralelo AEI com GC	93
7.2.2. O Algoritmo Sequencial AEI com GC	98
7.2.3. O Algoritmo Sequencial AEI convencional	98
7.3. Comentários	99
<b>Capítulo 8 - Avaliação de Desempenho</b>	<b>100</b>
8.1. Introdução	100
8.2. Objetivos	100
8.3. Sistemas Elétricos Utilizados nas Simulações	101
8.4. Parâmetros das Simulações	102

---

8.5. Resultados Numéricos	102
8.5.1. O Referencial Adotado	103
8.5.2. Eficiência e <i>Speed-up</i>	105
8.5.3. Algoritmos que Utilizam o Método GC	106
8.5.3.1. A Decomposição dos Sistemas	106
8.5.3.2. Resultados das Simulações	107
8.6. Avaliação dos Resultados	110
8.6.1. A Comunicação entre os Processadores no IBM/SP1 e SP2	111
8.6.2. <i>Speed-ups</i>	113
8.6.3. A Granularidade	114
8.6.4. A Evolução das Máquinas	115
8.6.5. O Paralelismo como Alternativa	116
8.7. Comentários	117
<b>Capítulo 9 - Conclusões</b>	119
9.1. Sugestões para Trabalhos Futuros	122
<b>Bibliografia</b>	124

# LISTA DE FIGURAS

	Página
Figura 2.1 - Principais Malhas de Controle de um SEE	8
Figura 2.2 - Estrutura detalhada das equações do modelo matemático do SEE	9
Figura 2.3 - Blocos $2 \times 2$ da matriz $Y$ expandida	18
Figura 2.4 - Ilustração da Técnica de Paralelização no Espaço	23
Figura 2.5 - Ilustração da Técnica de Paralelização no Tempo	24
Figura 4.1 - Agregação de nós aos subsistemas - Etapa Inicial	46
Figura 4.2 - Agregação de nós aos subsistemas - Etapa Avançada	47
Figura 4.3 - Metodologia do programa DECOSEE	48
Figura 4.4 - Decomposição do sistema de 118 barras em 10 subsistemas parciais	50
Figura 4.5 - Decomposição do sistema de 118 barras em 2 e 4 subsistemas	52
Figura 5.1 - Autovalores das matrizes $Y$ do sistema de 9 barras	56
Figura 5.2 - Autovalores das matrizes $Y$ do sistema de 45 barras	56
Figura 5.3 - Autovalores das matrizes $Y$ do sistema de 118 barras	56
Figura 5.4 - Autovalores das matrizes $Y$ do sistema de 340 barras	57
Figura 5.5 - Autovalores das matrizes $Y$ do sistema de 616 barras	57
Figura 5.6 - Autovalores das matrizes $R$ do sistema de 9 barras	58
Figura 5.7 - Autovalores das matrizes $R$ pré-falta e sob-falta do sistema de 45 barras	58

Figura 5.8 - Autovalores da matriz $R$ pós-falta do sistema de 45 barras	59
Figura 5.9 - Autovalores das matrizes $R$ do sistema de 118 barras	59
Figura 5.10 - Autovalores da matriz $R$ pré-falta do sistema 340 barras	59
Figura 5.11 - Autovalores das matrizes $R$ sob-falta e pós-falta do sistema de 340 barras	60
Figura 5.12 - Autovalores das matrizes $R$ pré-falta e sob-falta do sistema de 616 barras	60
Figura 5.13 - Autovalores da matriz $R$ pós-falta do sistema de 616 barras	61
Figura 5.14 - Aplicação do Método da Compensação	62
Figura 5.15 - Autovalores do sistema 9 barras pós-falta compensado	63
Figura 5.16 - Números de condição para o sistema de 9 barras	64
Figura 5.17 - Números de condição para o sistema de 45 barras	65
Figura 5.18 - Números de condição para o sistema de 118 barras	66
Figura 5.19 - Números de condição para o sistema de 340 barras	66
Figura 5.20 - Números de condição para o sistema de 9 barras pós-falta compensado	67
Figura 5.21 - Comportamento da ortogonalidade dos vetores de resíduo do sistema de 118 barras	71
Figura 5.22 -Evolução da norma do resíduo no sistema de 118 barras	71
Figura 5.23 - Comportamento da ortogonalidade dos vetores de resíduo do sistema de 616 barras com a utilização do PCS 1 termo	72
Figura 5.24 - Comportamento da ortogonalidade dos vetores de resíduo do sistema de 616 barras com a utilização do PCB 2 blocos	72
Figura 6.1 - Modelo de Programação SPMD	82
Figura 6.2 - Modelo de Programação Mestre/Escravo	83

---

Figura 7.1 - Fluxograma do algoritmo AEI com GC paralelo - 1 passo de integração	96
Figura 7.2 - Fluxograma do algoritmo GC para resolver $I(E^k, V^k) = Y.V^{k+1}$	97
Figura 8.1 - Tempos de CPU dos algoritmos utilizados x Dimensão do SEE	110
Figura 8.2 - Parcelas percentuais do tempo de CPU no IBM/SP1	111
Figura 8.3 - Capacidade de comunicação do IBM/SP1 e IBM/SP2	112
Figura 8.4 - <i>Speed-ups</i> $S_1$ obtidos em várias máquinas paralelas	113
Figura 8.5 - <i>Speed-ups</i> $S_2$ obtidos em várias máquinas paralelas	113



# LISTA DE TABELAS

	Página
Tabela 2.1 - Esquemas de solução	14
Tabela 4.1 - Decomposição do sistema de 118 barras em 10 subsistemas parciais	50
Tabela 4.2 - Decomposição do sistema de 118 barras em 2 subsistemas finais	51
Tabela 4.3 - Decomposição do sistema de 118 barras em 4 subsistemas finais	51
Tabela 5.1 - Parâmetros dos SEE utilizados na avaliação dos pré-condicionadores	54
Tabela 5.2 - Contingências consideradas nos SEE	55
Tabela 5.3 - Resultados das simulações para avaliação da convergência dos pré-condicionadores	69
Tabela 6.1 - Características de várias máquinas paralelas	90
Tabela 8.1 - Parâmetros dos SEE utilizados nas simulações	101
Tabela 8.2 - Número de iterações para o algoritmo AEI convencional sequencial (a)	103
Tabela 8.3 - Número de iterações para o algoritmo AEI convencional sequencial (b)	104
Tabela 8.4 - Tempos de CPU para o algoritmo AEI convencional sequencial (a)	104
Tabela 8.5 - Tempos de CPU para o algoritmo AEI convencional sequencial (b)	105
Tabela 8.6 - Parâmetros das decomposições	107
Tabela 8.7 - Resultados dos algoritmos AEI com GC no IBM/SP1	108
Tabela 8.8 - Resultados dos algoritmos AEI com GC no IBM/SP2	109
Tabela 8.9 - Dados para cálculo do desempenho das máquinas relativo ao NCP I	114

---

Tabela 8.10 - Desempenho das máquinas relativo ao NCP I ( em pu. )	115
Tabela 8.11 - Tempos de CPU para o AEI convencional x Dimensão do SEE	116

## NOMENCLATURA

- $P_m$  - potência mecânica fornecida ao eixo do gerador (pu.)
- $P_e$  - potência elétrica fornecida pelo gerador (pu.)
- $\delta$  - posição angular do rotor com relação a uma referência síncrona (rad.)
- $\delta_0$  - posição angular do rotor em regime permanente (rad.)
- $I_d$  - componente de eixo direto da corrente de estator da máquina síncrona (pu.)
- $I_q$  - componente de eixo quadratura da corrente de estator da máquina síncrona (pu.)
- $E_d''$  - componente de eixo direto da tensão subtransitória interna da máquina síncrona (pu.)
- $E_q''$  - componente de eixo quadratura da tensão subtransitória interna da máquina (pu.)
- $E_d'$  - componente de eixo direto da tensão transitória interna da máquina síncrona (pu.)
- $E_q'$  - componente de eixo quadratura da tensão transitória interna da máquina (pu.)
- $V_d$  - componente de eixo direto da tensão terminal da máquina síncrona (pu.)
- $V_q$  - componente de eixo quadratura da tensão terminal da máquina síncrona (pu.)
- $E_{fd}$  - tensão fornecida pelo sistema de excitação ao enrolamento de campo do gerador (pu.)
- $V_T$  - tensão terminal do gerador (pu.)
- $I_T$  - corrente complexa de estator do gerador (pu.)
- $Y$  - matriz de admitância nodal da rede elétrica
- $g$  - condutância elétrica (pu.)
- $b$  - susceptância elétrica

- $J$  - matriz jacobiana
- $k$  - contador de iterações
- $p^k$  - direção vetorial a ser seguida na iteração  $k$  ( método GC )
- $\alpha^k$  - valor do passo na iteração  $k$  ( método GC )
- $A$  - matriz quadrada genérica
- $r, \tilde{r}$  - resíduos do método GC
- $S$  - matriz da transformação de congruência ( método GC )
- $\lambda$  - autovalor
- $L$  - matriz triangular inferior
- $U$  - matriz triangular superior
- $M$  - matriz de pré-condicionamento do método GC
- $F$  - matriz de *fill-ins* originada na fatoração LU incompleta
- $P, Q$  - matrizes originadas da decomposição da matriz  $A$
- $A_{DIAG}$  - matriz originada da decomposição da matriz  $A$
- $A_{OFF}$  - matriz originada da decomposição da matriz  $A$
- $R$  - matriz auxiliar do PCS
- $I$  - matriz identidade
- $U$  - matriz originada da decomposição da matriz  $A$
- $\rho$  - raio espectral
- $y_{comp}$  - admitância de compensação (pu.)

- $I_{comp}$  - corrente de compensação (pu.)
- $V$  - tensão de barra (pu.)
- $\|r\|$  - norma do resíduo
- $\Omega_i$  - conjunto de todos os ramos conectados ao nó  $i$
- $\Omega_T$  - conjunto de todos os ramos do SEE
- $B_L$  - susceptância do ramo L
- $n_L$  - número de ramos contidos no SEE
- $T$  - período total de simulação (s.)
- $I_g$  - injeção de corrente na barra com gerador representado dinamicamente (pu.)
- $V_g$  - tensão na barra com gerador representado dinamicamente (pu.)
- $V_f$  - tensão em barra de fronteira (pu.)
- $G$  - teste de convergência global ( algoritmo AEI )
- $RR$  - escalar resultante dos subprodutos internos de cada processador
- $DYD$  - escalar resultante dos subprodutos internos de cada processador
- $d_f$  - subvetor de direções globais associado as barras de fronteira
- $S$  - speed-up
- $E$  - eficiência
- $q$  - número de processadores
- $H$  - constante de inércia da máquina (s.)
- $t$  - tempo (s.)

- $w$  - velocidade angular (rad./s.)
- $w_s$  - velocidade angular síncrona (rad./s.)
- $T'_{D_o}$  - constante de tempo transitória de eixo direto com o estator em aberto (s.)
- $T'_{Q_o}$  - constante de tempo transitória de eixo quadratura com o estator em aberto (s.)
- $T''_{D_o}$  - constante de tempo subtransitória de eixo direto com o estator em aberto (s.)
- $T''_{Q_o}$  - constante de tempo subtransitória de eixo quadratura com o estator em aberto (s.)
- $x_d$  - reatância em regime permanente de eixo direto (pu.)
- $x'_d$  - reatância em regime transitório de eixo direto (pu.)
- $x''_d$  - reatância em regime subtransitório de eixo direto (pu.)
- $x_q$  - reatância em regime permanente de eixo quadratura (pu.)
- $x'_q$  - reatância em regime transitório de eixo quadratura (pu.)
- $x''_q$  - reatância em regime subtransitório de eixo quadratura (pu.)
- $V_r$  - tensão de saída do regulador (pu.)
- $V_{r_{MIN}}$  - valor mínimo de  $V_r$  (pu.)
- $V_{r_{MAX}}$  - valor máximo de  $V_r$  (pu.)
- $V_{REF}$  - tensão de referência do regulador (pu.)
- $K_a$  - ganho do regulador
- $K_e$  - constante da excitatriz relacionada ao campo de auto-excitação
- $K_f$  - ganho do circuito de estabilização
- $S_e$  - função da saturação da excitatriz

- $T_a$  - constante de tempo do amplificador (s.)
- $T_e$  - constante de tempo da excitatriz (s.)
- $T_f$  - constante de tempo do circuito de estabilização do regulador (s.)
- $T_r$  - constante de tempo do filtro de entrada do regulador (s.)
- $K_c$  - fator de carga do retificador
- $T_b, T_c$  - constantes de tempo (s.)
- $V_{Re}$  - componente real da tensão terminal da máquina ( pu.)
- $V_{Im}$  - componente imaginária da tensão terminal da máquina ( pu.)
- $P_o$  - potência ativa da carga em regime permanente (pu.)
- $P_z$  - potência ativa associada a parcela de carga do tipo impedância constante (pu.)
- $P_I$  - potência ativa associada a parcela de carga do tipo corrente constante (pu.)
- $P_s$  - potência ativa associada a parcela de carga do tipo potência constante (pu.)
- $Q_o$  - potência reativa da carga em regime permanente (pu.)
- $Q_z$  - potência reativa associada a parcela de carga do tipo impedância constante (pu.)
- $Q_I$  - potência reativa associada a parcela de carga do tipo corrente constante (pu.)
- $Q_s$  - potência reativa associada a parcela de carga do tipo potência constante (pu.)
- $V_{Bo}$  - tensão na barra de carga em regime permanente (pu.)
- $V_B$  - tensão na barra de carga ao longo da simulação (pu.)
- $a_I$  - constante que especifica a porcentagem de potência constante para a carga ativa
- $b_I$  - constante que especifica a porcentagem de corrente constante para a carga ativa

- $c_1$  - constante que especifica a porcentagem de impedância constante para a carga ativa
- $a_2$  - constante que especifica a porcentagem de potência constante para a carga reativa
- $b_2$  - constante que especifica a porcentagem de corrente constante para a carga reativa
- $c_2$  - constante que especifica a porcentagem de impedância constante para a carga reativa
- $Ra$  - resistência da armadura da máquina síncrona (pu.)
- $y_c$  - admitância da carga durante a simulação (pu.)
- $y_0$  - admitância da carga durante a simulação (pu.)
- $a_D$  - variável auxiliar no Esquema AEI
- $Fq$  - variável auxiliar no Esquema AEI
- $a_Q$  - variável auxiliar no Esquema AEI
- $Fd$  - variável auxiliar no Esquema AEI
- $a_{DT}$  - variável auxiliar no Esquema AEI
- $Fq_T$  - variável auxiliar no Esquema AEI
- $a_{QT}$  - variável auxiliar no Esquema AEI
- $Fd_T$  - variável auxiliar no Esquema AEI
- $f$  - fração de operações processadas sequencialmente dentro de um algoritmo
- $\kappa$  - número de condição



# OPERADORES

$d$  - indica derivada total

$\in$  - indica pertence a

$\Delta$  - indica variação incremental de uma grandeza

$\Sigma$  - indica somatório de elementos

$\dot{x}$  - indica derivada da variável  $x$  com relação ao tempo

$\partial$  - indica derivada parcial

## ABREVIATURAS

AEI	- Alternado Entrelaçado Implícito
AI	- Alternado Implícito
BCG	- <i>Biconjugated Gradient</i>
CC	- Capacidade de Comunicação
CGS	- <i>Conjugate Gradient Squared</i>
COGC	- <i>Conjugate Orthogonal Gradient Method</i>
Comun.	- Comunicação
conv.	- Convencional
CP	- Capacidade de Processamento
EADs	- Equações Algébricas e Diferenciais
EMS	- Sistema de Gerenciamento de Energia
EP	- Elemento de Processamento
FDLF	- <i>Fast Decoupled Load Flow</i>
FS	- <i>File Server</i>
GC	- Gradiente Conjugado
GCP	- Gradiente Conjugado Pré-Condicionado
GMRES	- <i>Generalized Orthogonal Gradient Method</i>
HPS	- <i>High Performance Switch</i>

---

Im	- Imaginário
I/O	- In / Out
KB	- Kilobytes
LNCC	- Laboratório Nacional de Computação Científica
LT	- Linha de Transmissão
MB	- Megabytes
MHz	- Megahertz
MIMD	- <i>Multiple Instruction Stream, Multiple Data Stream</i>
MISD	- <i>Multiple Instruction Stream, Single Data Stream</i>
MPL	- <i>Message Passing Library</i>
MPP	- <i>Massively Parallel Processor</i>
NADJ	- Grupo de nós adjacentes
NAGR	- Grupo de nós agregados
No.	- Número
NS	- Nó Semente
OC	- Opção de Compilação
Par.	- Paralelo
PCB	- Pré-condicionador Bloco-diagonal
PCD	- Pré-condicionador Diagonal
PCS	- Pré-condicionador por Série Truncada de Maclaurin

---

PET	- Paralelização no Espaço e no Tempo
pu.	- Por Unidade
RAM	- <i>Random Access Memory</i>
Re	- Real
RFO	- Relaxação da Forma de Onda
SDSEE	- Simulação da Dinâmica de Sistemas de Energia Elétrica
SEE	- Sistemas de Energia Elétrica
Seq.	- Sequencial
SI	- Simultâneo Implícito
SIMD	- <i>Single Instruction Stream, Multiple Data Stream</i>
SISD	- <i>Single Instruction Stream, Single Data Stream</i>
SPMD	- <i>Single Program Multiple Data</i>
SSIS	- Subsistema
SSUD	- Sul-Sudeste
T.	- Tempo
VDHN	- <i>Very Dishonest Newton</i>

## RESUMO

Neste trabalho realiza-se um estudo da simulação da dinâmica de sistemas de energia elétrica nas plataformas computacionais paralelas IBM/SP. São implementados três algoritmos de simulação onde o conjunto de equações algébricas e diferenciais não lineares que descreve a dinâmica do sistema é resolvido pelo Método Alternado Entrelaçado Implícito. O Método Gradiente Conjugado Pré-condicionado é utilizado em dois destes algoritmos, um sequencial e outro paralelo, na solução das equações associadas à rede elétrica. Devido à utilização do Método Gradiente Conjugado Pré-condicionado foi realizado um estudo sobre técnicas de pré-condicionamento e decomposição de redes elétricas. O terceiro algoritmo é sequencial e utiliza um método direto baseado na fatoração LU e substituição direta e inversa para a solução das equações da rede elétrica. Os dois métodos utilizados foram testados com sistemas elétricos de pequeno a grande porte e os resultados obtidos foram comparados em termos de *speed-ups*. As simulações com o algoritmo paralelo foram realizadas com a utilização de 2 e 4 processadores das plataformas IBM/SP, onde foi explorada a paralelização no espaço.

As plataformas paralelas utilizadas neste trabalho foram comparadas, quanto ao seu desempenho com as plataformas iPSC/860 e NCP I. Uma análise global de várias plataformas paralelas mostrou que as máquinas atuais, com relação as anteriores, tiveram um grande aumento da sua capacidade de processamento, não acompanhado, em mesma proporção, pelo aumento na capacidade de comunicação entre os seus processadores, o que dificulta a obtenção de bons índices de *speed-ups* nas máquinas atuais.

## ABSTRACT

The purpose of this work is to carry out a study on electric system dynamic simulation using IBM/SP parallel computers. Three algorithms were implemented to solve the set of non-linear, algebraic and differential equations, describing the system dynamic behaviour, which is solved by the Implicit Interlaced Alternated Method. The Pre-conditioned Conjugated Gradient Method is used in two of these algorithms, while one is sequential and the other is parallel, for solving the equations associated with the electric network. A study about both pre-conditioning techniques and network decompositions was made, due the utilization of the Pre-conditioned Conjugated Gradient Method. The third algorithm is a sequential one, which uses a direct method based on LU factoration, and forward and backward substitution, for solving the electric network equations. The two methods mentioned above were validated using electrical systems of several dimensions, and the obtained speed-ups were compared. The simulations on the parallel algorithm were performed using 2 and 4 IBM/SP processors, exploring the space parallelism.

The parallel machines used in this work were compared to the iPSC/860 and NCP I computers, in relation to their performances. A global analysis of several parallel machines showed that the machines currently used have a greater processing capacity then the older ones, despite their lower communication capacity, which makes it difficult to obtain higher speed-ups.

# CAPÍTULO 1

## INTRODUÇÃO

O objetivo deste trabalho é realizar um estudo da simulação da dinâmica de sistemas de energia elétrica ( SDSEE ) em computadores paralelos, especificamente nas plataformas computacionais paralelas IBM/SP.

A modelagem matemática da dinâmica de um sistema de energia elétrica é feita por um conjunto de equações algébricas e diferenciais não lineares que pode reunir milhares de equações [3]. Salienta-se ainda que em estudos de estabilidade transitória e segurança dinâmica, várias simulações devem ser realizadas e considerando diferentes contingências aplicadas ao sistema. Isto contribui de forma substancial para o crescimento da dimensão do problema. Outros fatores também contribuem para este crescimento, como a interconexão de vários sistemas de transmissão e a sofisticação da modelagem dos componentes do sistema [4], onde uma única máquina síncrona, por exemplo, pode ser representada por 2 a 20 equações diferenciais [5]. Esta elevada quantidade de equações e dados no problema da SDSEE geralmente resulta em tempos de processamento superiores ao tempo real dos fenômenos simulados, diferença que se acentua com o crescimento da dimensão do sistema. Estas características dificultam a utilização de estudos de estabilidade na avaliação *on-line* da segurança dinâmica nos Sistemas de Gerenciamento de Energia ( EMS ). A necessidade de um processamento mais rápido aponta para o processamento paralelo como alternativa viável para o problema de SDSEE, onde os computadores multiprocessados possuem potencial para um grande desempenho computacional associado a um custo relativamente baixo. Outros problemas na área de sistemas de potência também buscam o processamento paralelo como alternativa para tempos de processamento menores, como por exemplo, a Análise de Curto-Circuitos, estudos de Transitórios Eletromagnéticos e a Análise de Confiabilidade.

A utilização do processamento paralelo na análise de sistemas de potência é motivada pela necessidade de menores tempos de CPU e não pela estrutura do problema em si. Excetuando-se aqueles procedimentos analíticos que requerem soluções repetidas, como Análise de Contingências, não há um paralelismo óbvio inerente a estrutura matemática de um problema de sistemas de

energia elétrica. Logo, para a análise e solução de um caso particular, é necessário que uma formulação paralela seja encontrada, possibilitando a implementação de um algoritmo paralelo ao problema. Este algoritmo será então implementado em uma máquina paralela definida, onde o seu desempenho e eficiência computacional dependerão da adequação do algoritmo a arquitetura paralela da máquina utilizada.

## 1.1. A Utilização de Métodos Iterativos

No desenvolvimento de novos algoritmos, a aplicação de métodos com características atrativas para o ambiente paralelo deve ser considerada. Os inexpressivos ganhos obtidos com a paralelização e/ou vetorização dos métodos diretos de resolução de sistemas lineares originados em problemas de sistemas de potência, têm motivado a investigação no sentido da utilização de métodos iterativos que sejam potencialmente promissores para soluções paralelas e/ou vetoriais [1]. Os métodos mais conhecidos são o método de Jacobi, Gauss-Seidel, Sobre-relaxação e Gradiente Conjugado. O método de Jacobi é inerentemente paralelo mas possui convergência lenta [28], praticamente inviabilizando a sua utilização em problemas envolvendo SEE. Os métodos de Gauss-Seidel e Sobre-relaxação, com características de convergência superiores as do método de Jacobi não possuem um bom potencial de paralelização. Além disso, os métodos de Sobre-relaxação utilizam coeficientes aceleradores de convergência, cujos valores devem ser ajustados de acordo com a experiência de utilização da metodologia [1]. Estes valores são difíceis de serem obtidos no caso de sistemas de grande porte.

Os métodos do tipo Gradiente Conjugado têm sido utilizados, com sucesso, na resolução paralela e vetorial de sistemas lineares esparsos [1], conforme demonstrado em [28,30,32,63,64,65,66]. Os métodos do tipo Gradiente Conjugado são derivados da família de métodos do subespaço de Krylov ou métodos tipo projetados [28,32], sendo o método Gradiente Conjugado (GC), de utilização restrita a sistemas simétricos, o mais popular. Para casos assimétricos outros métodos podem ser utilizados, dentre eles pode-se citar o GMRES ( *Generalized Minimal Residual* ) [32], o Gradiente Biconjugado ( BCG - *Biconjugate Gradient* ) [32], o CGS ( *Conjugate Gradient-Squared* ) [32] e o Bi-CGSTAB [31]. O maior esforço computacional nos algoritmos associados a métodos do tipo Gradiente Conjugado está no cálculo de produtos matriz-vetor, produtos internos e operações do tipo *linked-triad* ( vetor + escalar-vetor ). Todas essas operações são vantajosas para a implementação em ambiente paralelo, sendo esta uma das vantagens dos métodos do tipo Gradiente Conjugado quando utilizados em



algoritmos desenvolvidos para máquinas paralelas. Os métodos do tipo GC devem trabalhar associados a algum tipo de pré-condicionamento para que a sua convergência seja acelerada.

Estes aspectos motivaram Decker, em [1], a propor a utilização do método GC, em algoritmos paralelos de SDSEE, na resolução das equações que descrevem o comportamento da rede elétrica, quando é empregado o Esquema Alternado Entrelaçado Implícito na resolução do sistema algébrico-diferencial que descreve o comportamento dinâmico do SEE em estudo, ou na solução do sistema linear jacobiano, quando da utilização do Esquema Simultâneo. O processamento paralelo na SDSEE, com o emprego do método GC, é novamente enfatizado por Decker et alli em [9,10,11,43].

Outros trabalhos são citados na literatura com a utilização do método GC em problemas de sistemas de energia elétrica. Alvarado, em [44], propõe a utilização do método Gradiente Conjugado Pré-Condicionado ( GCP ) na resolução do sistema linear jacobiano do fluxo de potência originado da aplicação do método de Newton-Raphson. Em [45] são apresentados vários estudos com a aplicação do método GCP a sistemas lineares oriundos de matrizes esparsas de sistemas de potência. Alvarado, em [67,68], estuda várias técnicas de pré-condicionamento a serem aplicadas no método GC para a solução de problemas de sistemas de potência. Galiana, em [36], apresenta um estudo clássico em que aplica o método GC na resolução do problema de fluxo de carga DC e desacoplado rápido, assim como em análise de contingências. Neste trabalho é mostrado que os métodos do tipo GC apresentam menores tempos de CPU que os métodos diretos na solução de sistemas algébricos lineares de muito grande porte . Hennenberg, em [42], propõe uma modificação no algoritmo padrão de resolução do sistema linear jacobiano com o método GCP, e que traz grande redução no tempo de CPU desta metodologia. Alvarado, recentemente, vem realizando estudos sobre pré-condicionadores adequados para o processamento paralelo do método GCP [67].

M.A.Pai et alli, em [70], utilizando um Convex C3800 e um IBM RS6000/340, compara os métodos GMRES e GC na solução de sistemas lineares de sistemas de potência. Em [71], é realizado um estudo sobre a performance e escalabilidade do método GC em máquinas paralelas.

## 1.2. Técnicas de Paralelização

As técnicas de paralelização utilizadas nos algoritmos paralelos de SDSEE encontrados na literatura exploram basicamente a paralelização no tempo, a paralelização no espaço, a

paralelização no espaço e no tempo e a relaxação da forma de onda. Na paralelização no espaço, o conjunto de equações algébricas a diferenças e originalmente algébricas é dividido entre os processadores a cada passo de integração, e todos os processadores cooperam para a solução deste conjunto de equações. Este é o processo de paralelização mais natural [1]. Esta técnica é citada em [19,39] onde o algoritmo VDHN ( *Very Dishonest Newton* ) é paralelizado e implementado nos computadores iPSC/2 ( hipercúbico com memória distribuída ) e Alliant FX/8 ( com memória compartilhada ), sendo realizadas simulações da dinâmica no sistema IEEE 118 e US. Midwestern com 662 barras, tendo-se conseguido um máximo *speed-up* de 5.61 para 32 processadores. Em [19] é também proposta uma versão paralela do método dos fatores W ( *Sparse Matrix Inverse Factors* [38] ), cujo algoritmo foi testado no computador Symmetry, de memória compartilhada, nos mesmos sistemas citados acima, tendo apresentado resultados inferiores aos do algoritmo VDHN paralelo. Em [60] uma outra versão deste método foi implementada no computador paralelo PP/286, desenvolvido pela CPQd/Telebrás, onde foram obtidos altos valores de *speed-ups*, porém, além da máquina ser do tipo híbrida, possui uma granularidade extremamente baixa. Em máquinas mais modernas e do tipo memória distribuída estes índices não se repetiram [80]. Em [61] é apresentado um simulador digital em tempo real da dinâmica de um SEE, baseado em uma configuração massivamente paralela do computador nCUBE 2, onde cada processador simulava uma barra do SEE analisado, ocasionando, com isso, um forte desbalanceamento de carga entre os processadores. Uma simulação de 10 segundos do sistema IEEE 118 foi feita em pouco menos de 10 segundos em 118 processadores do nCUBE 2. Em [39] foram simulados 1.12 segundos deste mesmo sistema, em 1.0 segundo de CPU, usando-se o método SOR-Newton ( *Successive Overrelaxation - Newton* ) no computador Alliant FX/8 com a utilização de somente oito processadores. Decker et alli, em [1,9,10,11], apresentam algoritmos de SDSEE, que utilizam a paralelização no espaço e o Esquema AEI.

Na técnica de paralelização no tempo vários passos de integração, onde cada um está associado a um conjunto de equações, são resolvidos simultaneamente pelos vários processadores. O primeiro trabalho na área da SDSEE com a utilização desta técnica é de Alvarado [20], que fez um estudo genérico para a resolução de equações diferenciais ordinárias lineares e não lineares. Esta técnica é utilizada também em [1,9,10,11] onde é apresentado um algoritmo de SDSEE, com a utilização do Método Simultâneo Implícito ( SI ), que explora o paralelismo no tempo.

A paralelização no espaço e no tempo foi desenvolvida por LaScala et alli [21,22], que utilizaram os métodos Gauss-Jacobi ( substituições sucessivas ) [21] e o Gauss-Jacobi-Block-

Newton [22] na solução do sistema de equações algébricas não lineares gerado neste tipo de técnica. Nesta técnica vários passos de integração são resolvidos simultaneamente, sendo ainda explorado o paralelismo no espaço de cada passo.

A aplicação do método da relaxação da forma de onda ( RFO ), em sua forma clássica, ao problema da SDSEE, foi explorada em vários trabalhos de Crow e Ilic [3,24,25,26,27]. A premissa básica do método da RFO é a decomposição do sistemas de EDAs em subsistemas fracamente acoplados que são resolvidos separadamente por um certo número de passos de integração ( janela de integração ), os resultados são então intercambiados entre os subsistemas ( pelas fronteiras ), e para cada nova janela o processo é repetido, até a convergência final. A efetividade deste método é dependente da fase de decomposição do sistema em subsistemas, visto que esta decomposição afeta a velocidade de convergência dos algoritmos e o balanceamento de carga entre os processadores.

Outros métodos têm sido propostos na literatura. Em [1] são citados os Métodos do Tipo Newton Relaxado. Tylavsky, em [62], motivado pela tendência futura do processamento de alto desempenho fixar-se no processamento distribuído e no processamento paralelo-vetorial, propõe a metodologia *Vectorization Across Equations Sets* ( VAES ), testada em um computador Alliant FX/8 e Cray Y/MP.

### 1.3. Aspectos Principais do Trabalho

O estudo da simulação da dinâmica de sistemas de energia elétrica nas plataformas computacionais paralelas IBM/SP, apresentado neste trabalho, continua a linha de investigação iniciada por Decker em [1]. Foi utilizado o Esquema Alternado Entrelaçado Implícito e a técnica de paralelização no espaço. Os algoritmos sequenciais e paralelos desenvolvidos em [1] foram adaptados e implementados nas plataformas IBM/SP1 do Laboratório Nacional de Computação Científica - LNCC ( Rio de Janeiro - RJ ) e IBM/SP2 da Universidade Federal de Santa Catarina - UFSC. O método GCP foi também utilizado, sendo analisadas várias técnicas de pré-condicionamento. Um estudo sobre técnicas de decomposição de sistemas de energia elétrica em subsistemas menores é apresentado, tendo sido desenvolvido e implementado um algoritmo de decomposição denominado DECOSEE, que apresenta modificações na técnica de decomposição baseada no critério do ranking nodal proposta em [52,54]. Foram realizadas simulações da dinâmica em sistemas de muito grande porte que foram gerados artificialmente a partir do sistema

sul-sudeste de 1916 barras ( SSUD 2000 ). Uma avaliação de desempenho dos algoritmos implementados e das plataformas paralelas utilizadas neste trabalho e em outros trabalhos citados na literatura [1,4,19,67,69,77,78] é apresentada. Um perfil da evolução das máquinas paralelas associado a perspectiva de obtenção de *speed-ups* nas máquinas atuais e uma avaliação da utilização de algoritmos sequenciais em sistemas de grande porte são também apresentados.

## 1.4. Estrutura do Trabalho

O trabalho está organizado da seguinte forma : o Capítulo 2 apresenta a modelagem matemática utilizada neste trabalho, os esquemas básicos de solução do conjunto de equações algébrico-diferencial não linear que descreve a dinâmica de um sistema de energia elétrica e as técnicas de paralelização utilizadas nos algoritmos de simulação. O Capítulo 3 é todo dedicado aos Métodos Gradiente Conjugado, onde é apresentado o estado da arte referente a esta família de métodos, com ênfase a sua aplicação em sistemas de energia elétrica. São também apresentadas neste capítulo várias técnicas de pré-condicionamento para o método GCP, onde três delas são escolhidas para uma segunda etapa de avaliação. O Capítulo 4 apresenta diversas técnicas de decomposição e uma nova proposta para a decomposição de redes elétricas em subredes de menor porte, baseada no critério do *ranking* nodal proposto em [52,54]. Um detalhado estudo sobre pré-condicionadores é apresentado no Capítulo 5. No Capítulo 6 são apresentados vários conceitos importantes em processamento paralelo e as plataformas computacionais paralelas IBM/SP utilizadas neste trabalho e outras máquinas usadas em outros trabalhos. Aspectos de *software* e *hardware* das máquinas IBM/SP são também descritos neste capítulo. No Capítulo 7 são apresentados os algoritmos sequenciais e paralelos de simulação da dinâmica de sistemas de energia elétrica implementados neste trabalho. A análise dos resultados obtidos, desempenho dos algoritmos e comparativo entre várias plataformas paralelas são apresentados no Capítulo 8. As conclusões finais do trabalho e sugestões para trabalhos futuros são apresentadas no Capítulo 9. A bibliografia utilizada encerra o presente trabalho.

# CAPÍTULO 2

## MODELAGEM MATEMÁTICA E TÉCNICAS DE SOLUÇÃO

### 2.1. Introdução

Este capítulo visa estabelecer um referencial, em termos de modelagem, nomenclatura e técnicas de solução do problema da simulação da dinâmica de sistemas de energia elétrica, para os demais capítulos que compõem este trabalho. São abordados a descrição matemática da dinâmica dos sistemas de energia elétrica, os objetivos e tipos de estudos de estabilidade, os esquemas básicos de solução sequencial, o Esquema Alternado Entrelaçado Implícito, usado neste trabalho, as formas de solução da rede elétrica e os aspectos conceituais das principais técnicas de paralelização deste problema encontradas na literatura.

### 2.2. Modelagem Matemática

Em estudos de simulação da dinâmica os principais componentes do sistema de energia elétrica modelados são :

- Sistema de transmissão
- Cargas
- Unidades geradoras

A rede de transmissão é descrita por equações algébricas não lineares, apresentadas na forma matricial, onde a matriz  $Y$  de admitância de barras da rede tem um acentuado grau de esparsidade. As cargas são, em geral, modeladas algebricamente [8] como funções exponenciais ou polinomiais da tensão da barra onde estão conectadas, e, ocasionalmente, também em função da frequência. Na situação em que as cargas são modeladas como potência constante suas equações são não lineares, quando modeladas como impedância ou corrente constante suas equações são lineares.

O estator da máquina síncrona é representado [8] por uma impedância onde a reatância tem componentes nos eixos  $d$  e  $q$ . O equacionamento do estator é algébrico se seus transitórios forem desprezados. O rotor da máquina síncrona pode ser representado [8] por vários modelos de circuitos equivalentes nos eixos  $d$  e  $q$ , os quais são representados por um conjunto de equações diferenciais de primeira ordem. Estas equações são escritas diretamente em termos de parâmetros da máquina (resistência, reatâncias síncronas, transitórias e subtransitórias e constantes de tempo). As equações diferenciais eletromecânicas do rotor representam [8] a ação do torque acelerante sobre a máquina. A turbina e outros elementos de uma unidade de geração também são descritos por equações diferenciais.

As malhas de controle também são modeladas por equações diferenciais, e, geralmente, são incluídos limitadores que resultam em descontinuidades nas equações diferenciais. As principais malhas de controle de um sistema de energia elétrica são mostradas na Figura 2.1 a seguir [1], onde estão destacados os blocos considerados na modelagem usada neste trabalho :

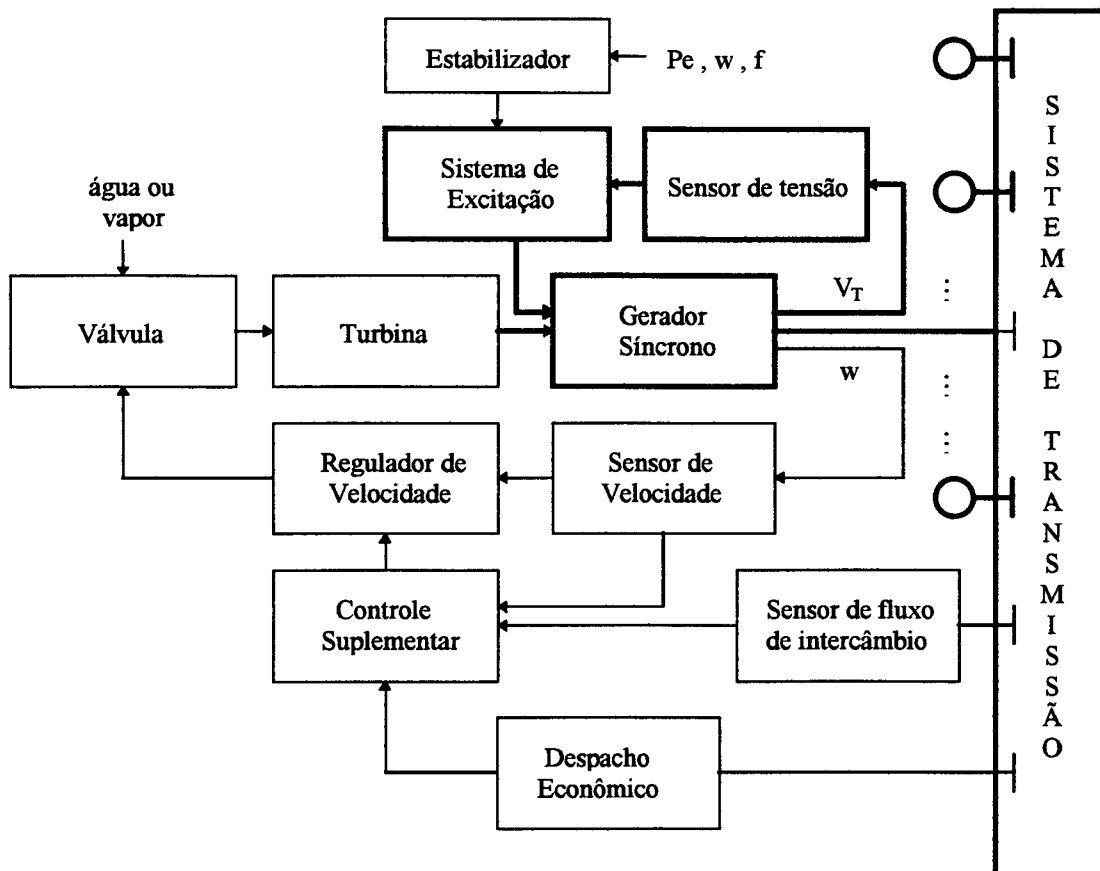


Figura 2.1 - Principais malhas de controle de um SEE

A Figura 2.2, a seguir, adaptada de [1,12], mostra a estrutura do modelo matemático do sistema de energia elétrica adotado neste trabalho :

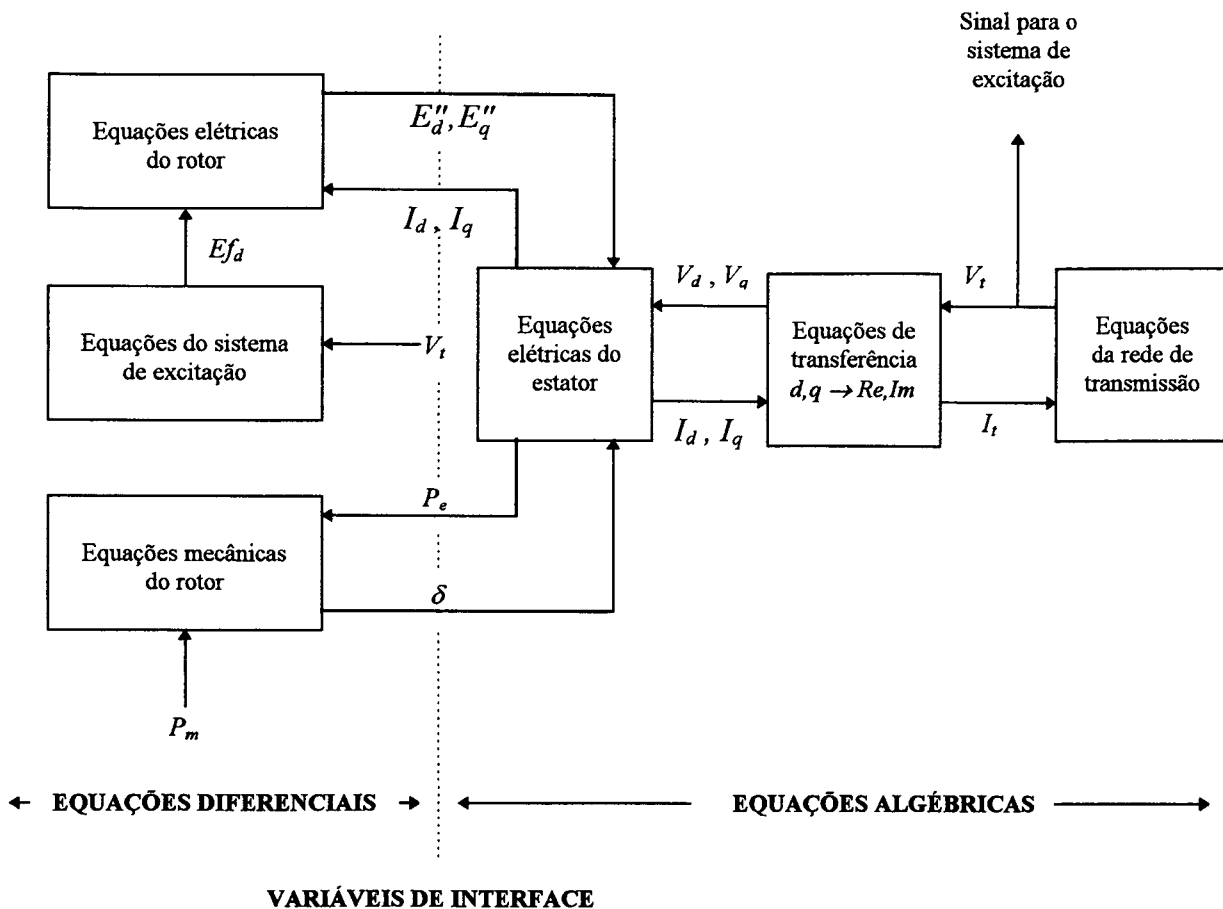


Figura 2.2 - Estrutura detalhada das equações do modelo matemático do SEE

onde :

$P_m$  e  $P_e$  - potência mecânica e elétrica, respectivamente, do gerador

$\delta$  - posição angular do rotor com relação a uma referência síncrona

$I_d, I_q$  - componentes de eixo direto e quadratura, respectivamente, da corrente de estator da máquina síncrona

$E_d'', E_q''$  - componentes de eixo direto e quadratura, respectivamente, da tensão subtransitória interna da máquina síncrona

$E_f$  - tensão fornecida pelo sistema de excitação ao enrolamento de campo do gerador

$V_d, V_q$  - componentes de eixo direto e quadratura, respectivamente, da tensão terminal da máquina síncrona

$V_t$  - tensão terminal do gerador

$I_t$  - corrente complexa de estator do gerador

As variáveis que aparecem tanto nas equações diferenciais como nas algébricas, mostradas na linha pontilhada da Figura 2.2, são chamadas **variáveis de interface**, sendo definidas por [1]:

$u$  - subconjunto do vetor de variáveis algébricas ( $I_d, I_q, P_e$  e  $V_t$ ) que aparecem nas equações diferenciais

$E$  - subconjunto do vetor de variáveis de estado ( $E_d'', E_q''$  e  $\delta$ ) que aparecem nas equações algébricas.

O conjunto de equações algébrico-diferencial que descreve a dinâmica de um sistema de energia elétrica é representado sinteticamente na seguinte forma:

$$\begin{cases} \dot{x} = f(x, z) \\ 0 = g(x, z) \end{cases} \quad (2.1)$$

onde  $x$  é o vetor de variáveis de estado,  $f$  é uma função vetorial que define as equações diferenciais,  $z$  é o vetor de variáveis das equações algébricas e  $g$  é a função vetorial que define as equações algébricas. Na modelagem usada neste trabalho, o conjunto de equações diferenciais está associado aos rotores das máquinas síncronas e ao sistema de excitação. O conjunto de equações algébricas está associado aos estatores das máquinas síncronas, à rede de transmissão e às cargas conectadas a esta rede, representadas por modelos estáticos. As simulações de defeitos e chaveamentos introduzem grandes discontinuidades nestas equações algébricas [8].

### 2.3. Estudos de Estabilidade

Os estudos de estabilidade em sistemas interligados de energia elétrica, após a ocorrência de uma grande perturbação, visam determinar, principalmente, se [12]:

- as máquinas elétricas do sistema serão capazes de permanecer em sincronismo após o SEE sofrer uma grande perturbação, como do tipo curto-circuito, chaveamento de linhas, perda de unidade geradora do sistema e outros, ou mesmo uma combinação de várias destas contingências.



- as oscilações de tensão causadas pelos mesmos motivos acima não podem deteriorar a qualidade de suprimento de energia elétrica do sistema

Estes estudos podem abranger diferentes períodos de tempo de análise do comportamento dinâmico do sistema, que são a base para classificá-los em [8] :

- **Estabilidade de curto prazo** : neste caso o comportamento dinâmico do sistema é observado após alguns segundos de ocorrência a perturbação, geralmente até o limite de 8 segundos. Também chamada de **estabilidade transitória**.

- **Estabilidade de médio prazo** : o SEE é analisado até 5 minutos após a perturbação.

- **Estabilidade de longo prazo** : o período de análise é bem maior, sendo comum análises de até 20 minutos após a perturbação.

A escolha do tipo de análise a ser realizada sobre o SEE determina o grau de modelagem a ser utilizado na SDSEE, visto que as constantes de tempo envolvidas com as diversas malhas de controle das máquinas do sistema possuem uma faixa de valores muito ampla, e, conforme o período de análise considerado após a perturbação, algumas malhas de controle poderão ser desprezadas. A classificação acima reflete a influência dos vários componentes do sistema no seu próprio comportamento dinâmico [12]. Pode-se citar, por exemplo, o estudo de estabilidade transitória, onde a malha de controle suplementar de carga-frequência, cujas constantes de tempo são da ordem de minutos [13], não é considerada face ao pequeno período de análise abrangido neste tipo de estudo. A modelagem e métodos abordados neste trabalho foram desenvolvidos para a análise de estabilidade transitória.

## 2.4. Esquemas de Solução

As equações ( 2.1 ) são resolvidas numericamente ao longo do período abrangido pela simulação a ser realizada, partindo-se de condições iniciais definidas pelo ponto de operação em regime permanente pré-perturbação. Este processo de cálculo passo-a-passo deve simular o comportamento dinâmico do SEE tão rápido quanto possível e deve reunir as seguintes características [8] :

- 1 • **Exatidão** : o processo não deverá ter grandes erros percentuais ( geralmente obtidos em função das medidas do ângulo do rotor das máquinas ).

**2 • Confiança** : no sentido da estabilidade numérica do método de integração adotado e da convergência de qualquer processo iterativo utilizado na simulação.

**3 • Economia** : que se tenha a menor necessidade possível de armazenamento de dados na memória do computador utilizado na simulação.

**4 • Flexibilidade** : facilidade com que modelos podem ser alterados ou acrescentados para a realização de novos estudos.

**5 • Facilidade de manutenção** : associado à flexibilidade, implica em algoritmos robustos, de códigos simplificados e de pouca e facilitada manutenção.

O tempo computacional de cada estudo dependerá do volume de processamento de cada passo de integração, do próprio valor utilizado como passo e do computador utilizado. As características 1, 2 e 5, mostradas acima, tendem a conflitar com o objetivo de soluções rápidas [8], o que torna difícil achar o ponto ótimo de trabalho para cada algoritmo.

Os esquemas de resolução das equações diferenciais e algébricas envolvidas na SDSEE são caracterizados pelo método de integração numérica aplicado às equações diferenciais, pelo método de solução das equações algébricas e no modo pelo qual os sistemas diferencial e algébrico interagem entre si ao longo da solução.

### 2.4.1. Os Métodos de Integração

Os métodos de integração podem ser classificados em **multipassos** ou de **passo simples**, ou seja, em relação ao número de passos necessários no instante  $t$ , para o cálculo do instante  $t + \Delta t$ , onde  $\Delta t$  é o passo de integração. Os métodos de passo simples utilizam informações apenas do passo  $t$  para o cálculo do passo  $t + \Delta t$ . Estes métodos são auto-inicializáveis, característica muito conveniente em presença de descontinuidades [8]. Os métodos baseados na Série de Taylor e os da família Runge-Kutta são exemplos típicos desta categoria. Já os métodos multipassos necessitam de dados de vários passos ( $t, t-\Delta t, t-2\Delta t$ , por exemplo) para o cálculo do passo  $t+\Delta t$ , e, conseqüentemente, de maior armazenamento em memória. São métodos que devem ser reinicializados na presença de descontinuidades. Nesta categoria, por exemplo, estão os algoritmos da família Adams [8]. Os métodos de integração ainda são subdivididos em **explícitos** e **implícitos**. Os métodos explícitos são aqueles em que para calcular-se o valor de

determinada variável no tempo  $t+\Delta t$ , são necessários valores desta variável em passos anteriores como  $t$ ,  $t-\Delta t$ ,  $t-2\Delta t$  ... , enquanto que, com a utilização de um método implícito necessita-se de valores desta variável nos passos  $t+\Delta t$ ,  $t$ ,  $t-\Delta t$ ,  $t-2\Delta t$  ..... Como exemplos pode-se citar os algoritmos da família Adams-Bashforth que são explícitos, enquanto os da família Adams-Moulton são métodos implícitos, onde destaca-se o método trapezoidal utilizado neste trabalho. Alguns algoritmos utilizam-se dos métodos explícitos e implícitos simultaneamente, ou seja, começam o cálculo de uma determinada variável no passo  $t+\Delta t$  com a utilização de um método explícito, menos complicado e mais rápido, chamada de fase previsor, e concluem o cálculo com um método implícito, mais estável e preciso, chamada de fase corretora. Este tipo de método, que combina os métodos explícitos e implícitos, é chamado previsor-corretor [14].

Os erros agregados à solução obtida ao final de determinado passo de integração, dependem dos erros originados ao longo do processamento do passo e do erro inserido no estabelecimento das condições iniciais deste passo. A estabilidade numérica de determinado método de integração está relacionada com a propagação destes erros nos passos subsequentes. Um método instável é aquele que tende a acumular estes erros [8]. As constantes de tempo envolvidas no SEE têm um importante papel, no sentido de limitarem altos valores de passos de integração, que poderiam vir a originar grandes erros ao longo do processo de simulação . Neste aspecto, métodos de integração estáveis, como o trapezoidal implícito, permitem a utilização de passos maiores, visto que não há propagação de erros ao longo dos passos [8].

## 2.4.2. Esquemas Básicos de Solução

Os dois esquemas básicos de solução do conjunto de equações diferenciais e algébricas que descreve a dinâmica de um SEE são [8]:

- **Esquema Alternado** : as equações diferenciais são transformadas em equações algébricas a diferenças mediante a utilização de um método de integração numérica e cada um dos sistemas, o algebrizado e o originalmente algébrico, é resolvido separadamente, para cada passo de integração, alternando-se as respectivas soluções. Neste esquema de solução, especial atenção deve ser dada no sentido de eliminação do erro de interface, que ocorre quando as variáveis de interface não são devidamente atualizadas ao longo do processo de simulação, sendo uma desvantagem inerente a este esquema.

• **Esquema Simultâneo** : as equações diferenciais são transformadas em equações algébricas a diferenças mediante a utilização de um método de integração numérica, e resolvidas simultaneamente com as equações originalmente algébricas, formando um sistema único de equações, resolvido, geralmente, pelo Método de Newton.

A utilização de métodos de integração explícitos ou implícitos define então quatro esquemas de solução, mostrados na Tabela 2.1 abaixo :

Tabela 2.1 - Esquemas de Solução

MÉTODOS DE INTEGRAÇÃO	ESQUEMAS BÁSICOS	
	ALTERNADO	SIMULTÂNEO
IMPLÍCITO	AI	SI
EXPLÍCITO	AE	SE

Os métodos de integração implícitos têm apresentado melhores resultados [8,15,16] e, geralmente, é utilizado o método trapezoidal implícito, face as suas características de estabilidade e precisão [1].

A seguir é feita uma breve abordagem do Esquema Alternado Implícito ( AI ) e do Esquema Simultâneo Implícito ( SI ).

## 2.5. O Esquema Alternado Implícito ( AI )

As equações (2.1) podem ser reescritas na forma mostrada pelo sistema de equações (2.2), em que as equações algébricas não lineares da rede elétrica e de interface estão explicitadas [8]:

$$\begin{cases} \dot{x} = A.x + B.u \\ I(E,V) = Y.V \\ u = h(E,V) \end{cases} \quad (2.2)$$

onde a matriz  $A$  é quadrada, esparsa e bloco-diagonal e  $B$  é uma matriz retangular esparsa formada por blocos,  $x$  é um vetor de variáveis de estado vinculadas às máquinas e suas malhas de controle,  $u$  é um vetor de variáveis algébricas que aparecem nas equações diferenciais,  $Y$  é a

matriz admitância nodal da rede elétrica, complexa simétrica e esparsa;  $V$  é um vetor de tensões nodais complexas;  $I$  é um vetor de injeções de correntes nodais relativas às cargas e aos estatores das máquinas síncronas e  $E$  é um subvetor de  $x$ , constituído das variáveis de estado necessárias para o cálculo das injeções de correntes dos estatores das máquinas síncronas. Com a aplicação do método trapezoidal implícito às equações diferenciais, o sistema de equações ( 2.2 ), para o intervalo de um passo de integração, pode ser representado conforme as equações ( 2.3 ),( 2.4 ) e ( 2.5 ) onde  $m$  é o número der máquinas e  $n$  o número de barras :

$$\begin{cases} x_1(t) = H_1(x_1(t), x_1(t - \Delta t), u_1(t), u_1(t - \Delta t)) \\ x_2(t) = H_2(x_2(t), x_2(t - \Delta t), u_2(t), u_2(t - \Delta t)) \\ \vdots \\ x_m(t) = H_m(x_m(t), x_m(t - \Delta t), u_m(t), u_m(t - \Delta t)) \end{cases} \quad ( 2.3 )$$

$$\begin{bmatrix} I_1(E_1(t), V_1(t)) \\ I_2(E_2(t), V_2(t)) \\ \vdots \\ I_n(E_n(t), V_n(t)) \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & \dots & Y_{1n} \\ Y_{21} & Y_{22} & \dots & Y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \dots & Y_{nn} \end{bmatrix} \begin{bmatrix} V_1(t) \\ V_2(t) \\ \vdots \\ V_n(t) \end{bmatrix} \quad ( 2.4 )$$

$$\begin{cases} u_1(t) = h_1(E_1(t), V_1(t)) \\ u_2(t) = h_2(E_2(t), V_2(t)) \\ \vdots \\ u_m(t) = h_m(E_m(t), V_m(t)) \end{cases} \quad ( 2.5 )$$

onde :

$x_i = [w_i, \delta_i, E''_{q_i}, E''_{d_i}, E'_{q_i}, E'_{d_i}, Ef_{d_i}]^T$  - vetor de variáveis de estado do  $i$ -ésimo gerador

$E_i = [\delta_i, E''_{q_i}, E''_{d_i}, E'_{q_i}, E'_{d_i}]^T$  - subvetor de variáveis de interface do  $i$ -ésimo gerador

$u_i = [I_{d_i}, I_{q_i}, Pe_i]^T$  - vetor de variáveis das equações de interface do  $i$ -ésimo gerador

Após definir-se :

$H = [H_1, H_2, \dots, H_m]^T$  - vetor de funções não lineares das equações algébricas a diferenças

$x = [x_1, x_2, \dots, x_m]^T$  - vetor de variáveis de estado

$u = [u_1, u_2, \dots, u_m]^T$  - vetor de variáveis das equações de interface

$I = [I_1, I_2, \dots, I_n]^T$  - vetor de injeções de correntes nodais

$V = [V_1, V_2, \dots, V_n]^T$  - vetor de tensões nodais

$h = [h_1, h_2, \dots, h_m]^T$  - vetor de funções não lineares de interface

as equações ( 2.3 ) a ( 2.5 ) podem ser escritas na seguinte forma sintética [1] :

$$\begin{cases} x(t) = H(x(t), x(t - \Delta t), u(t), u(t - \Delta t)) \\ I(E, V) = YV \\ u = h(E, V) \end{cases} \quad (2.6)$$

O algoritmo para o esquema AI, baseado em ( 2.6 ), onde  $k$  é um contador de iterações e  $T$  o tempo de simulação, é apresentado abaixo.

### ALGORITMO AI

Inicialização

Para  $t = 1, 2, \dots, T$

Calcule  $u^*(t)$ , por extrapolação, e faça  $k \leftarrow 0$

Calcule  $x^o(t) = H(x^*(t), u^*(t), x(t - \Delta t), u(t - \Delta t))$ , onde o ‘\*’ indica valor extrapolado

Enquanto  $\|\Delta x^k(t)\|_2^2 > \varepsilon$ , faça :

Resolva  $I(E^k(t), V^k(t)) = YV^{k+1}(t)$

Calcule  $u^{k+1}(t) = h(E^k(t), V^{k+1}(t))$

Calcule  $x^{k+1}(t) = H(x^k(t), u^{k+1}(t), x(t - \Delta t), u(t - \Delta t))$

Calcule  $\Delta x^{k+1}(t) = |x^{k+1}(t) - x^k(t)|$

Faça  $k \leftarrow k + 1$

Fim-Enquanto

Fim-Para

Algumas importantes características do Método Alternado Implícito ( AI ) são comentadas a seguir :

- A utilização de um método de integração implícito torna iterativo o processo de cálculo de  $x(t)$  .
- As injeções de corrente na equação da rede elétrica são, geralmente, funções não lineares de  $V(t)$ , o que implica que estas equações também devam ser resolvidas iterativamente.
- A matriz admitância nodal  $Y$  da rede elétrica é constante entre chaveamentos.
- A extrapolação de  $u(t)$ , realizada no início de cada passo de integração, tem por objetivo a redução do número de iterações, não tendo influência na precisão dos resultados [9].
- O erro de interface, que ocorre quando dentro de um passo de integração o valor de  $x(t)$ , que depende de  $u(t)$ , é calculado várias vezes ( processo iterativo ), mas sempre utilizando-se o mesmo valor inicial de  $u(t)$  extrapolado no início de cada passo, é eliminado com a utilização do algoritmo AI acima, onde  $u(t)$  é atualizado para cada novo cálculo de  $x(t)$ .
- Na presença de descontinuidades, onde não é possível realizar a extrapolação, pode-se utilizar como valores iniciais de  $u(t)$  os do passo anterior, ou seja,  $u(t-\Delta t)$ .
- A eficiência computacional é significativamente melhorada se for realizada uma única iteração no sistema algébrico, ou seja, resolvendo-o como se fosse linear [8]. Este esquema é conhecido como **Esquema Alternado Entrelaçado Implícito ( AEI )**, tendo apresentado desempenho superior ao AI clássico [1,17]. O processo de convergência é controlado pelo processo iterativo diferencial que determina a precisão da solução final [12]. O Esquema Alternado Implícito, neste caso, só é aplicado nos instantes de chaveamentos, por um número limitado de passos de integração, para se evitar grandes erros no cálculo das tensões das barras.

### 2.5.1. Formas de Solução da Rede

As equações da rede elétrica podem ser expressas na forma complexa, conforme a equação ( 2.7 ) :

$$I(E, V) = Y.V \quad (2.7)$$

ou na forma expandida mostrada na equação ( 2.8 ) :

$$I^e(E, V^e) = Y^e V^e \quad (2.8)$$

Neste último caso cada elemento da matriz admitância de barras complexa é transformado em um bloco  $2 \times 2$  com a seguinte forma :

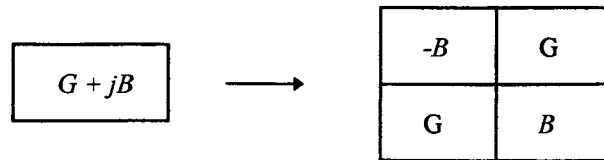


Figura 2.3 - Blocos  $2 \times 2$  da matriz  $Y$  expandida

A susceptância é colocada na diagonal para favorecer a dominância diagonal da matriz e manter a sua simetria, características importantes para a solução de ( 2.8 ).

Os sistemas não lineares ( 2.7 ) ou ( 2.8 ) são em geral, resolvidos por um procedimento iterativo no qual, a cada iteração, resolve-se um sistema linear  $I = Y.V$  [9]. Na solução deste sistema linear pode-se utilizar métodos diretos, baseados na fatoração LU e substituição direta e inversa, ou métodos iterativos, como os do tipo Gradiente Conjugado. A equação da rede na forma expandida ( 2.8 ) ainda pode ser resolvida [8] pelo Método de Newton, reescrevendo-se a equação ( 2.8 ) como :

$$F^e = I^e(E, V^e) - Y^e V^e \quad (2.9)$$

onde  $F^e$  é um vetor de funções não lineares, assumindo o valor nulo na solução exata de ( 2.8 ). Cada iteração da solução de ( 2.9 ) pelo Método de Newton leva a montagem da matriz jacobiana, conforme a equação :

$$F^e = -J^e . \Delta V^e \quad (2.10)$$

sendo a solução de ( 2.10 ) também obtida pela aplicação de métodos diretos ou iterativos.

## 2.6. O Esquema Simultâneo Implícito ( SI )

Neste esquema as equações diferenciais são transformadas em equações algébricas a diferenças, através de um método implícito de integração numérica, e reunidas com as equações



originalmente algébricas, formando um conjunto único de equações algébricas não lineares. Após a aplicação do Método Trapezoidal Implícito no sistema de equações ( 2.1 ) este sistema pode ser apresentado na seguinte forma :

$$\begin{cases} F = x(t) - x(t - \Delta t) - [f( x(t), z(t) ) + f( x(t - \Delta t), z(t - \Delta t) )] \frac{\Delta t}{2} = 0 \\ G = g( x(t), z(t) ) = 0 \end{cases} \quad (2.11)$$

O sistema de equações ( 2.11 ), o qual descreve o comportamento dinâmico do SEE de  $t - \Delta t$  até  $t$ , pode ser reescrito na seguinte forma compacta :

$$H( y(t) ) = 0 \quad (2.12)$$

onde :  $y = [x, z]^T$  e  $H = [F, G]^T$ .

A resolução da equação ( 2.12 ) é comumente realizada pela aplicação do Método de Newton [8], que para a solução de um passo de integração, tem como algoritmo [1] :

#### ALGORITMO DE NEWTON

Fazer  $k = 0$  e especificar  $y^0$

Fazer-Enquanto  $\|H(y^k)\|_2^2 > \varepsilon$

Calcular  $H(y^k)$

$$\text{Resolver } H(y^k) + \left. \frac{\partial H}{\partial y} \right|_{y=y^k} \cdot \Delta y^{k+1} = 0 \quad (2.13)$$

Calcular  $y^{k+1} = y^k + \Delta y^{k+1}$

Fazer  $k \leftarrow k + 1$

Fim-Enquanto

Onde  $y^0$  é a condição inicial pré-definida,  $\varepsilon$  é a tolerância de convergência e  $k$  é o contador de iterações.

A equação matricial ( 2.13 ) é linear e deve ser resolvida várias vezes até a convergência, dentro de um passo de integração, sendo que a cada iteração  $k$  a matriz jacobiana dada por :

$$J = \partial H / \partial y \quad (2.14)$$

deve ser recalculada com o novo valor de  $y^k$ . Esta matriz é real, assimétrica e muito esparsa, características que devem ser exploradas para um melhor desempenho computacional. O maior esforço computacional do algoritmo de Newton e, conseqüentemente, para o esquema SI, está na solução de ( 2.13 ).

Para os métodos que fatoram a matriz jacobiana, para a resolução de ( 2.13 ), é usual manter-se a matriz calculada na primeira iteração de um passo de integração para as iterações subsequentes. Isto visa minimizar o esforço computacional, sendo que, as vezes, tal procedimento é adotado até para vários passos de integração [1]. Esta versão do Método de Newton é conhecida como *Very Dishonest Newton* (VDHN). Neste caso a equação matricial ( 2.13 ) assume a seguinte forma :

$$H(y^k) + \left. \frac{\partial H}{\partial y} \right|_{y=y^0} \cdot \Delta y^{k+1} = 0 \quad (2.15)$$

### 2.6.1. Resolução da Equação Jacobiana

A equação ( 2.15 ) pode ser reescrita na forma :

$$H(y^k) = J(y^k) \cdot \Delta y^{k+1} \quad (2.16)$$

onde  $k$  indica a iteração do Método de Newton e  $J$  é a matriz jacobiana definida por [1] :

$$J = - \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} \quad (2.17)$$

Os elementos da matriz jacobiana de ( 2.17 ), para  $i = 1, 2, \dots, m$ , em que  $m$  é o número de geradores, são :

$$J_1 = \text{diag} [ \partial F_i / \partial x_i ]$$

$$J_2 = [ \partial F_1 / \partial V^e \dots \partial F_m / \partial V^e ]^T \quad (2.18)$$

$$J_3 = [\partial G / \partial x_1 \dots \partial G / \partial x_m]$$

$$J_4 = [\partial G / \partial V^e]$$

Com isto a equação ( 2.16 ) pode ser apresentada como :

$$\begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \\ G \end{bmatrix}^k = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & 0 & \dots & 0 & \frac{\partial F_1}{\partial V^e} \\ 0 & \frac{\partial F_2}{\partial x_2} & & 0 & \frac{\partial F_2}{\partial V^e} \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \frac{\partial F_m}{\partial x_m} & \frac{\partial F_m}{\partial V^e} \\ \frac{\partial G}{\partial x_1} & \frac{\partial G}{\partial x_2} & \dots & \frac{\partial G}{\partial x_m} & \frac{\partial G}{\partial V^e} \end{bmatrix}^k \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_m \\ \Delta V^e \end{bmatrix}^{k+1} \quad (2.19)$$

onde  $F_i = [ F_1, F_2 \dots F_m ]^T$  e  $G$  representam os resíduos para as equações algebrizadas dos geradores e para as equações da rede elétrica, respectivamente. A equação ( 2.19 ) pode ser escrita na seguinte forma compacta :

$$\begin{bmatrix} F \\ G \end{bmatrix}^k = - \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix}^k \begin{bmatrix} \Delta x \\ \Delta V^e \end{bmatrix}^{k+1} \quad (2.20)$$

A matriz jacobiana tem a forma bloco-diagonal com blocos nas bordas ( BBDF - *Block Bordered Diagonal Form* ) . O sistema matricial ( 2.20 ) pode ser resolvido por métodos diretos onde inicialmente é realizada uma eliminação Gaussiana por blocos em ( 2.20 ) para o cálculo de  $\Delta V^e$  :

$$- (J_4 + Y_M) . \Delta V^e = -\bar{J}_4 . \Delta V^e = \bar{G} \quad (2.21)$$

onde  $\bar{G} = G - J_3 . J_1^{-1} . F$  . A expressão  $Y_M = - J_3 . J_1^{-1} . J_2$  representa o efeito dos geradores na matriz admitância nodal.

Numa segunda etapa  $\Delta x$  é encontrado de acordo com :

$$\Delta x = - J_1^{-1} (F + J_2 . \Delta V^e) \quad (2.22)$$

As equações ( 2.21 ) e ( 2.22 ) são, em geral, resolvidas por métodos diretos baseados na fatoração LU e substituição direta e inversa, onde :

- $\bar{J}_4$  possui a mesma dimensão de  $J_4$ , ou seja, de  $Y^e$  .

- a inversão de  $J_1$  não implica em grande esforço computacional. Cada bloco diagonal  $i$  desta matriz possui dimensão igual ao número de equações utilizadas para representar o  $i$  - ésimo gerador e seus controladores. Esta dimensão varia de 2, quando as máquinas síncronas são representadas pelo modelo clássico, a aproximadamente 20 para modelos mais sofisticados.

- no método VDHN somente os elementos de  $\bar{J}_4$  são mantidos constantes por várias iterações ou até passos de integração, as matrizes  $J_1$  e  $J_2$  são recalculadas a cada iteração.

O sistema ( 2.20 ) também pode ser resolvido com a utilização de métodos iterativos para sistemas lineares assimétricos, visto ser a matriz jacobiana não-simétrica. Existem vários métodos do tipo Gradiente Conjugado, por exemplo, que podem ser aplicados neste caso, os quais são apresentados no Capítulo 3 deste trabalho.

## 2.7. A SDSEE em Computadores Paralelos

Na literatura são encontrados vários trabalhos em que a SDSEE é realizada em plataformas computacionais paralelas [1,3,9,10,11,19,21,22,24,25,26,27,39,60,61]. Os algoritmos paralelos encontrados no âmbito destes trabalhos podem ser enquadrados nas seguintes técnicas básicas :

- paralelização no espaço
- paralelização no tempo
- paralelização no espaço e no tempo ( PET )
- relaxação da forma de onda ( RFO ).

cujos aspectos conceituais mais relevantes são apresentados nos próximos subítens.

### 2.7.1. Paralelização no Espaço

Nesta técnica, o conjunto de equações algébricas e algebrizadas, que descrevem a dinâmica de um sistema de energia elétrica em um passo de integração, é dividido entre os diversos processadores da máquina paralela, que resolvem os subconjuntos de equações de forma concorrente, onde todos eles contribuem para a solução do conjunto global de equações. Este é o processo mais natural de paralelização visto que são paralelizados os próprios esquemas básicos de solução do problema da SDSEE. A paralelização no espaço é ilustrada na Figura 2.4, onde uma tarefa  $Q$  é dividida em 4 subtarefas processadas de forma concorrente em 4 processadores.

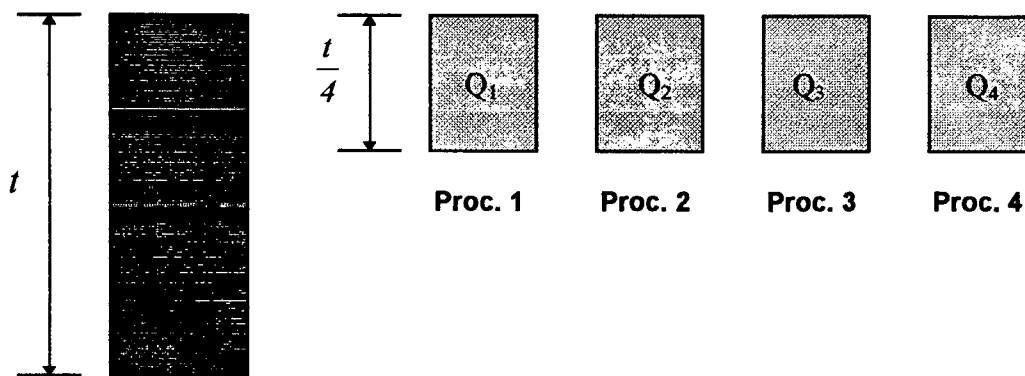


Figura 2.4 - Ilustração da Técnica de Paralelização no Espaço

### 2.7.2. Paralelização no Tempo

Neste caso as equações diferenciais são discretizadas pela aplicação de um método de integração implícito a diversos passos de integração simultaneamente, denominados de janelas de integração. O sistema de equações algébricas a diferenças resultante é resolvido, em muitos casos, pelo Método de Newton e a equação linear jacobiana, associada a toda a janela de integração, é resolvida por um método direto baseado na fatoração LU [1]. Nesta técnica, os vários passos de integração são distribuídos entre os processadores e resolvidos de forma concorrente, onde todos os processadores contribuem para a solução da janela de integração. O primeiro trabalho a aplicar esta técnica de paralelização na SDSEE é de Alvarado [20]. A Figura 2.5, a seguir, mostra, de forma simbólica, a técnica de paralelização no tempo em  $q$  processadores.

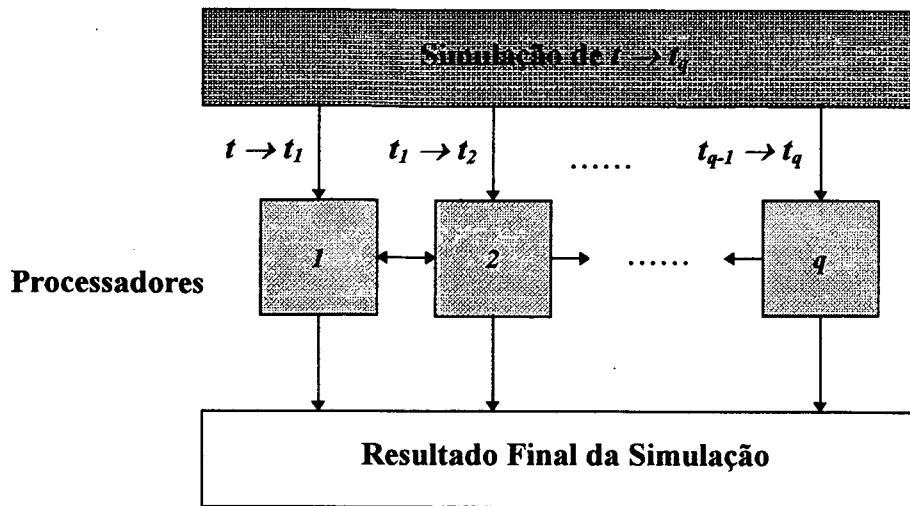


Figura 2.5 - Ilustração da Técnica de Paralelização no Tempo

### 2.7.3. Paralelização Simultânea no Espaço e no Tempo ( PET )

Esta técnica explora simultaneamente o paralelismo existente em cada passo de integração, a paralelização no espaço, e o paralelismo entre vários passos de integração tomados simultaneamente, ou seja, a paralelização no tempo. Em [1] são apresentados vários métodos de solução para a PET, como o Gauss-Jacobi, de substituições sucessivas, e o Gauss-Jacobi-Block-Newton, referenciados em [21,22] respectivamente, e utilizados por LaScala et alli para a resolução do sistema discretizado de equações algébricas não lineares.

### 2.7.4. Relaxação da Forma de Onda ( RFO )

A técnica da relaxação da forma de onda consiste em decompor o sistema de equações diferenciais e algébricas em subsistemas fracamente acoplados entre si, e realizar a solução de cada subsistema independentemente por um número determinado de passos de integração, denominados de janela de integração, obtendo-se uma primeira estimativa da forma de onda. Os resultados são então intercambiados entre os subsistemas, repetindo-se o processo de integração numérica de cada janela até a convergência final das formas de onda. Concluída a solução para uma janela, passa-se à janela seguinte [1]. As principais vantagens da RFO são [1] :

- possibilidade de resolver subsistemas diferentes com passos de integração diferentes.
- evitar a resolução, por métodos diretos, de sistemas algébricos lineares de grande porte.

Cada subsistema é resolvido independentemente dos demais, podendo esta resolução ser efetuada em qualquer ordem ou em paralelo. A efetividade desta técnica, porém, depende fortemente da fase de decomposição. A técnica da RFO tem sido explorada por Crow e Ilic em vários trabalhos [1,23,24,25,26,27].

## 2.8. Comentários

Neste capítulo foram apresentados e discutidos vários aspectos relacionados com a modelagem matemática e esquemas de solução do problema da SDSEE. Foi dado destaque a solução numérica das equações que descrevem a dinâmica destes sistemas, onde foram apresentados os métodos de integração e detalhados os esquemas básicos de solução Alternado Implícito ( AI ) e Simultâneo Implícito ( SI ). Foi apresentado o esquema AEI, que reduz o esforço computacional do AI, sendo este método de fácil implementação computacional, além de fornecer maiores facilidades para a inclusão e tratamento de novos e sofisticados modelos de elementos de um SEE. O fato deste método trabalhar com a matriz de admitância nodal  $Y$  na forma compacta, ou complexa, reduz a área de armazenamento de seus elementos na memória do computador utilizado, quando comparada com a área exigida pela matriz  $Y$  na forma expandida ( real e imaginária ). Além disso, como esta matriz é constante entre chaveamentos na rede elétrica, os seus fatores LU só sofrem alterações nos instantes em que a topologia da rede é alterada. O esquema SI, por utilizar o Método de Newton, cuja convergência quadrática torna o seu número de iterações praticamente independente da dimensão do sistema analisado, é atrativo quanto ao aspecto matemático, porém, a matriz  $Y$  é utilizada na forma expandida e, com referência a matriz jacobiana, esta é modificada a cada iteração, o que altera os seus fatores LU. Uma maneira de reduzir este esforço computacional é manter o jacobiano constante por várias iterações, e até por vários passos de integração. Esta metodologia é conhecida como *Very Dishonest Newton* ( VDHN ). Ao final foram discutidas brevemente as técnicas de paralelização no espaço, paralelização no tempo, paralelização no espaço e no tempo ( PET ) e a relaxação da forma de onda ( RFO ).

## CAPÍTULO 3

# MÉTODOS DO TIPO GRADIENTE CONJUGADO PRÉ-CONDICIONADOS

### 3.1. Introdução

O presente capítulo apresenta os métodos do tipo Gradiente Conjugado Pré-condicionados ( GCP ), assim como várias técnicas de pré-condicionamento, e o estado da arte destes métodos, com destaque especial para a sua aplicação em problemas de sistemas de energia elétrica. Neste trabalho utiliza-se o método Gradiente Conjugado Pré-Condicionado ( GCP ) na solução do sistema algébrico linear de equações associado a rede elétrica, no Esquema Alternado Entrelaçado Implícito.

### 3.2. O Método Gradiente Conjugado

O método Gradiente Conjugado ( GC ) foi introduzido no início dos anos 50 por Hestenes e Stiefel [29]. Este método apresentava uma elegante maneira de tratar a esparsidade da matriz  $A$  de  $A.x = b$ , desde que depende, basicamente, de produtos matriz - vetor ao longo das iterações [30]. Imediatamente após o surgimento do método GC inúmeros estudos demonstraram que a velocidade de convergência do método aumentava muito pela utilização de técnicas de pré-condicionamento no sistema  $A.x = b$  [30].

A solução de um sistema linear do tipo :

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

onde  $A_{n \times n}$  é uma matriz simétrica e positiva definida, é equivalente a solução do seguinte problema de otimização sem restrições :



$$\text{minimizar } f(x) = \frac{1}{2} x^T A x - b^T x \quad (3.2)$$

A função  $f(x)$  tem um mínimo global onde o seu gradiente :

$$\nabla f(x) = Ax - b \quad (3.3)$$

que é a solução de

$$A.x = b \quad (3.4)$$

anula-se.

A forma geral dos métodos iterativos utilizados para minimizar ( 3.2 ) é [28] :

$$x^{k+1} = x^k - \alpha^k p^k \quad , \text{ para } k = 0, 1, \dots \quad (3.5)$$

onde  $p^k$  indica a direção vetorial de minimização e  $\alpha^k$  é um escalar que define a distância de movimento na direção  $p^k$ . A forma de escolha de  $p^k$  e de  $\alpha^k$  define uma grande variedade de métodos [28], incluindo o método Gradiente Conjugado apresentado neste capítulo.

Desse modo o problema de minimizar a função  $f(x)$  é transformado em sucessivos problemas do tipo minimizar  $f(x)$  na direção  $p^k$  de modo que  $\alpha^k$  satisfaça a seguinte equação :

$$f(x^k - \alpha^k \cdot p^k) = \min_{\alpha} f(x^k - \alpha \cdot p^k) \quad (3.6)$$

Para  $x^k$  e  $p^k$  especificados e suprimindo o sobrescrito  $k$ , pode-se escrever a seguinte função em  $\alpha$  :

$$q(\alpha) = f(x - \alpha p) = \frac{1}{2} (x - \alpha p)^T A (x - \alpha p) - b^T (x - \alpha p) \quad (3.7)$$

A função quadrática  $q(\alpha)$  é minimizada quando sua primeira derivada anula-se, ou seja :

$$\frac{d q(\alpha)}{d \alpha} = 0 \quad (3.8)$$

Substituindo-se ( 3.7 ) em ( 3.8 ) e resolvendo-se a equação ( 3.8 ) para  $\alpha$  obtém-se :

$$\alpha = \frac{p^T (Ax - b)}{p^T Ap} \quad (3.9)$$

No Método Gradiente Conjugado as direções de minimização unidirecionais são geradas a partir da direção do vetor gradiente, sucedidas por direções conjugadas de modo que satisfaçam a seguinte equação :

$$(p^i)^T Ap^j = 0 \quad , \text{ para } i \neq j \quad (3.10)$$

A cada iteração, a nova direção  $p$  é obtida por uma combinação linear do sentido negativo do vetor gradiente da iteração corrente com as direções  $p$  conjugadas das iterações anteriores.

A direção do gradiente é obtida da seguinte forma :

Tome a função objetivo

$$f(x) = \frac{1}{2} x^T Ax - b^T x \quad (3.11)$$

e defina  $p^k$  tal que :

$$p^k = -\nabla f(x^k) = -(Ax - b) \quad (3.12)$$

Uma das propriedades básicas dos métodos da direção conjugada é fornecida pelo teorema das direções conjugadas [28] :

“ Se  $A_{n \times n}$  é uma matriz simétrica positiva definida e  $p^0, \dots, p^{n-1}$  são vetores não nulos que satisfazem ( 3.10 ), então para qualquer  $x^0$  as iterações dadas por  $x^{k+1} = x^k - \alpha^k p^k$ , onde  $\alpha^k$  é escolhido de acordo com o princípio da minimização dado por ( 3.9 ), convergem para a solução exata de  $Ax = b$  em não mais que  $n$  passos “ .

Este teorema garante que as iterações de  $x^{k+1}$  convergem e, na ausência de erros de arredondamento, esta convergência é atingida em um número finito de passos, não superior a  $n$ .

Considerando-se  $\varepsilon$  uma constante usada no teste de convergência, o algoritmo do método GC para a resolução de um sistema linear algébrico do tipo (3.1) é dado por:

**ALGORITMO GRADIENTE CONJUGADO**

Escolher uma solução inicial  $x^0$ .

Calcular o sentido negativo do vetor gradiente dado por :

$$r^0 = -(Ax^0 - b) = b - Ax^0, \text{ onde } r \text{ é o vetor de resíduos.}$$

Fazer  $p^0 = r^0$ , calcular  $(r^0, r^0)$  e inicializar o contador de iterações  $k \leftarrow 0$ .

Fazer-Enquanto  $\|r^{k+1}\|_2^2 > \varepsilon$  :

Calcular :

$$\alpha^k = -\frac{[r^k]^T [r^k]}{[p^k]^T [Ap^k]}$$

$$x^{k+1} = x^k - \alpha^k p^k$$

$$r^{k+1} = r^k + \alpha^k Ap^k$$

Calcular a nova direção conjugada  $p^{k+1}$  :

$$\beta^k = \frac{[r^{k+1}]^T [r^{k+1}]}{[r^k]^T [r^k]}$$

$$p^{k+1} = r^{k+1} + \beta^k p^k$$

Incrementar o contador de iterações  $k \leftarrow k+1$

Fim-Enquanto

A fórmula de cálculo de  $r^{k+1}$ , no algoritmo acima, é obtida a partir das seguintes substituições :

$$r^{k+1} = b - Ax^{k+1}$$

$$r^{k+1} = b - A(x^k - \alpha^k p^k)$$

$$r^{k+1} = (b - Ax^k) + \alpha^k Ap^k$$

$$r^{k+1} = r^k + \alpha^k Ap^k$$

( 3.13 )

A dedução detalhada e completa das expressões do algoritmo GC é apresentada em [28].

O maior esforço computacional do algoritmo Gradiente Conjugado está no cálculo do produto matriz-vetor  $A.p^k$  [28], além deste, o algoritmo ainda contém dois produtos internos e três operações do tipo *linked-triad* (vetor + escalar-vetor). Todas as operações citadas acima são vantajosas para a sua implementação em ambiente computacional paralelo ou vetorial, sendo esta uma vantagem dos métodos do tipo GC quando utilizados em algoritmos desenvolvidos para máquinas paralelas. Uma outra vantagem é o menor tempo de processamento do método GC, com relação aos métodos diretos, na solução de sistemas lineares de grande porte [36].

O método GC é utilizado invariavelmente associado a alguma forma de pré-condicionamento da matriz  $A$  de (3.1), o que é apresentado a seguir.

### 3.3. O Método Gradiente Conjugado Pré-condicionado

O algoritmo GC converge em, no máximo,  $n$  iterações, onde  $n$  é a dimensão da matriz  $A$  do sistema (3.1), porém, na maioria das vezes, com um tempo de processamento superior ao obtido pelos métodos diretos [36]. A velocidade de convergência do método GC será acelerada se a matriz  $A$  de (3.1) tiver  $m$  ( $m < n$ ) autovalores distintos, quando então o algoritmo convergirá em, no máximo,  $m$  iterações [1], o que equivale aos autovalores de  $A$  estarem agrupados em clusters [36,46]. Numa situação ideal, se todos autovalores de  $A$  fossem iguais o GC convergiria numa única iteração.

Uma estimativa da razão de convergência do método GC é dada por [28]:

$$\|x^k - \hat{x}\|_2 \leq 2\sqrt{\kappa}\sigma^k \|x^o - \hat{x}\|_2 \quad (3.14)$$

onde:

$x^o \Rightarrow$  solução inicial arbitrada

$\hat{x} \Rightarrow$  solução exata do sistema  $A.x = b$

$x^k \Rightarrow$  estimativa na iteração  $k$

e ainda:

$$\sigma = (\sqrt{\kappa} - 1) / (\sqrt{\kappa} + 1) \quad (3.15)$$

$$\kappa = \text{cond}(A) = \|A\|_2 \|A^{-1}\|_2 = \text{maior } |\lambda(A)| / \text{menor } |\lambda(A)| \quad (3.16)$$

Em ( 3.16 ) a variável  $\kappa$  é o número de condição de  $A$  na norma 2, onde  $A$  é uma matriz positiva definida simétrica com autovalores  $\lambda_{Maior} \geq \dots \geq \lambda_{Menor} > 0$ .

Uma análise da expressão ( 3.15 ) mostra que  $\sigma = 0$  quando  $\kappa = 1$ , e  $\sigma \rightarrow 1$  quando o valor de  $\kappa \rightarrow \infty$ . Quanto maior o valor de  $\kappa$ , mais  $\sigma$  tenderá a unidade e mais lenta será a velocidade de convergência do algoritmo GC.

A ação do pré-condicionador no algoritmo GC deve ser, justamente, reduzir o número de condição  $\kappa$  do sistema  $A.x = b$ , possibilitando que o método GCP aplicado na solução de ( 3.1 ) tenha uma maior velocidade de convergência. Isto é possível mediante uma transformação de congruência em  $A$  da forma [1] :

$$\hat{A} = SAS^T \quad (3.17)$$

em que  $S$  é uma matriz não singular, escolhida adequadamente, de modo que esta transformação de congruência resulte numa clusterização dos autovalores da matriz  $A$ , o que reduz o seu número de condição. O sistema linear equivalente obtido após a transformação de congruência é do tipo:

$$\hat{A}.\hat{x} = \hat{b} \quad (3.18)$$

com  $\hat{x} = S^{-T}.x$  e  $\hat{b} = S.b$ .

A escolha correta da matriz  $S$  é o ponto crucial da metodologia.

O algoritmo GC quando aplicado ao sistema ( 3.18 ), mantendo-se as variáveis originais, resulta [28] :

### ALGORITMO GRADIENTE CONJUGADO PRÉ - CONDICIONADO

Escolher uma solução inicial  $x^0$

Fazer  $r^0 = b - Ax^0$

Resolver  $M.\tilde{r} = r^0$ , fazer  $p^0 = \tilde{r}^0$  e inicializar o contador de iterações  $k \leftarrow 0$

Fazer-Enquanto  $\|r^{k+1}\|_2^2 > \varepsilon$  :

Calcular :

$$\alpha^k = -(\tilde{r}^k, r^k) / (p^k, Ap^k)$$

$$x^{k+1} = x^k - \alpha^k . p^k$$

$$r^{k+1} = r^k + \alpha^k \cdot Ap^k$$

Resolver  $M \cdot \tilde{r}^{k+1} = r^{k+1}$

Calcular a nova direção conjugada  $p^{k+1}$  :

$$\beta^k = (\tilde{r}^{k+1}, r^{k+1}) / (\tilde{r}^k, r^k)$$

$$p^{k+1} = \tilde{r}^{k+1} + \beta^k \cdot p^k$$

Incrementar o contador de iterações  $k \leftarrow k+1$

Fim-Enquanto

conhecido como método Gradiente Conjugado Pré-Condicionado ( GCP ).

A matriz  $M = (S^T S)^{-1}$  é uma matriz positiva definida simétrica que se aproxima da matriz  $A$  de ( 3.1 ). A obtenção do algoritmo GCP é detalhada em [28]. A matriz  $M$  é conhecida como matriz de pré-condicionamento do sistema ( 3.1 ). Se  $M = I$  o algoritmo GCP se reduz ao algoritmo GC .

O sistema auxiliar  $M \cdot \tilde{r} = r$  precisa ser resolvido a cada iteração e este sistema deve ser de fácil solução. Para o pré-condicionamento ser efetivo, a matriz  $M$  deve ser uma boa aproximação de  $A$ , o que leva a uma situação conflitante, visto que quanto mais  $M$  se aproximar de  $A$  mais difícil pode ficar a solução do sistema auxiliar  $M \cdot \tilde{r} = r$  [47].

A matriz pré-condicionada  $\hat{A}$  de ( 3.17 ) deve ter um número de condição inferior ao da matriz  $A$  . Para a determinação do número de condição de  $\hat{A}$  temos que [28] :

$$S^T \hat{A} S^{-T} = S^T (S A S^T) S^{-T} = (S^T S) A = M^{-1} A \quad (3.19)$$

que mostra que a matriz  $M^{-1} A$  é similar a  $\hat{A}$ , com isso o número de condição  $\kappa$  de  $\hat{A}$  pode ser calculado pela seguinte expressão :

$$\kappa(\hat{A}) = \text{maior} \left| \lambda(M^{-1} A) \right| / \text{menor} \left| \lambda(M^{-1} A) \right| \quad (3.20)$$

ou seja, quanto menor o raio espectral de  $M^{-1} A$ , menor é o número de condição de  $\hat{A}$ , e maior a velocidade de convergência do método GCP aplicado na solução de ( 3.18 ).

O método GCP geralmente converge em  $n^{0.5}$  iterações [36] e requer  $sn$  operações por iteração, onde  $s$  é a média de elementos diferentes de zero por linha. Isto totaliza um número de operações de  $sn^{1.5}$ , sendo a necessidade de armazenamento de  $sn$  dados.

No presente trabalho são analisados vários métodos de pré-condicionamento descritos no ítem a seguir.

### 3.4. Técnicas de Pré-Condicionamento

Neste trabalho foram consideradas os seguintes técnicas de pré-condicionamento :

- Pré-condicionamento diagonal
- Pré-condicionamento por fatoração LU incompleta
- Pré-condicionamento bloco - diagonal
- Pré-condicionamento por série truncada de Neumann
- Pré-condicionamento por série truncada de Maclaurin
- Pré-condicionamento polinomial
- Pré-condicionamento de 2 níveis

A seguir é feita uma breve descrição de cada uma das técnicas citadas acima .

#### 3.4.1. Pré-condicionamento Diagonal

A matriz  $M$  é montada com os elementos diagonais da matriz  $A_{n \times n}$  [28], sendo um dos pré-condicionadores de implementação mais fácil . O sistema  $M.\tilde{r} = r$  terá a forma :

$$\begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{ii} \end{bmatrix} \cdot \begin{bmatrix} \tilde{r}_1 \\ \tilde{r}_2 \\ \vdots \\ \tilde{r}_i \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_i \end{bmatrix} \quad (3.21)$$

Existem poucas citações desta técnica de pré-condicionamento na literatura devido a sua baixa efetividade.

### 3.4.2. Pré-condicionamento Por Fatoração LU Incompleta

A fatoração LU da matriz  $A$  de ( 3.1 ) é dada por :

$$A = LU \quad ( 3.22 )$$

onde  $L$  é uma matriz triangular inferior e  $U$  é uma matriz triangular superior, sendo ambas menos esparsas que  $A$  face ao surgimento de enchimentos ( *fill-ins* ) gerados no processo de fatoração. A equação ( 3.22 ) pode ser reescrita como :

$$A = LU + F \quad ( 3.23 )$$

onde agora  $L$  e  $U$  têm elementos não nulos nas mesmas posições dos triângulos inferior e superior, respectivamente, da matriz  $A$ , mas a matriz  $F$  não é nula. O produto  $LU$ , em ( 3.23 ), define uma fatoração incompleta da matriz  $A$  de ( 3.1 ) e é utilizado como matriz  $M$  de pré-condicionamento para o método GCP. Alvarado, em trabalho recente [67], classifica a matriz  $M$  de acordo com o grau de *fill-ins* tolerados nesta matriz, ou seja, quando a matriz  $F$  inclui todos os *fill-ins* gerados na fatoração o grau de *fill-ins* de  $M$  é zero. No caso de serem permitidos alguns *fill-ins* em  $M$ , exceto aqueles gerados na fatoração por outros *fill-ins*, passa-se a ter uma matriz  $M$  com um nível 1 de *fill-ins*, e assim sucessivamente.

### 3.4.3. Pré-condicionamento Bloco-diagonal

A matriz  $M$  é montada em função do número de processadores a ser utilizado. Para  $q$  processadores  $M$  terá  $q$  blocos diagonais ( subsistemas ) quadrados de  $A$ . Na composição destes subsistemas são consideradas as características físicas do sistema de energia elétrica representado por  $A$ , que neste caso é a matriz  $Y$  de admitâncias de barras do SEE, de modo que as interligações entre os subsistemas sejam fracas, o que acelera a velocidade de convergência do método GCP [52].

O sistema auxiliar  $M.\tilde{r} = r$  terá a forma :

$$\begin{bmatrix} A_{11} & & & \\ & A_{22} & & \\ & & \ddots & \\ & & & A_{qq} \end{bmatrix} \cdot \begin{bmatrix} \tilde{r}_1 \\ \tilde{r}_2 \\ \vdots \\ \tilde{r}_q \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_q \end{bmatrix} \quad ( 3.24 )$$



onde cada  $A_{ii}$ , para  $i = 1, 2 \dots q$ , é a matriz de admitância de barras do  $i$ -ésimo subsistema, e  $r_i$  e  $\tilde{r}_i$  são os vetores de resíduos.

Este sistema terá paralelismo perfeito, e em [1] foi resolvido utilizando técnicas de fatoração LU de matrizes esparsas em cada subsistema  $A_{ii} \cdot \tilde{r}_i = r_i$ , para  $i = 1, 2, \dots, q$ , onde cada subsistema foi destinado a um processador.

### 3.4.4. Pré-Condicionamento Por Série Truncada de Neumann

Decompondo-se a matriz  $A$  de ( 3.1 ) em  $A = P - Q$ ,  $P$  não singular, a inversa de  $A$  será dada por :

$$A^{-1} = [ I + P^{-1}Q + (P^{-1}Q)^2 + \dots ]. P^{-1} \quad (3.25)$$

onde o módulo do autovalor de maior módulo de  $P^{-1}Q$  deve ser menor que 1. Definindo-se  $H = P^{-1}Q$  tem-se :

$$A^{-1} = \left[ \sum_{k=0}^{\infty} H^k \right] P^{-1} \quad (3.26)$$

O sistema auxiliar  $M \cdot \tilde{r} = r$  para este caso ficará :

$$\tilde{r} = M^{-1} \cdot r = \left( I + H + H^2 + H^3 + \dots \right) P^{-1} r \quad (3.27)$$

onde o grau de proximidade que  $M$  terá de  $A$  dependerá do número de termos utilizados na Série de Neumann.

### 3.4.5. Pré-Condicionamento Por Série Truncada de Maclaurin

Decompondo-se a matriz  $A$  de ( 3.1 ) em :

$$A = A_{DIAG} + A_{OFF} \quad (3.28)$$

onde  $A_{DIAG}$  é uma matriz diagonal que contém os elementos diagonais de  $A$  e  $A_{OFF}$  é uma matriz com os elementos não diagonais de  $A$ . Substituindo-se ( 3.28 ) em  $A \cdot x = b$  e isolando-se  $x$  obtém-se como expressão resultante :

$$x = (I + A_{DIAG}^{-1} A_{OFF})^{-1} \cdot A_{DIAG}^{-1} \cdot b \quad (3.29)$$

onde  $I$  é uma matriz identidade de dimensão igual a de  $A$ .

O termo  $(I + A_{DIAG}^{-1} A_{OFF})^{-1} \cdot A_{DIAG}^{-1}$  em (3.29) representa a inversa da matriz  $A$ .

Fazendo-se  $P = A_{DIAG}^{-1} \cdot A_{OFF}$  em (3.29) tem-se :

$$x = (I + P)^{-1} \cdot A_{DIAG}^{-1} \cdot b \quad (3.30)$$

Considerando-se a Série de Maclaurin :

$$(I + P)^{-1} = I - P + P^2 - P^3 + \dots - P^{2h-1} + P^{2h} - \dots, \text{ para qualquer } h. \quad (3.31)$$

A matriz  $A$  terá como inversa :

$$A^{-1} = (I - P + P^2 - P^3 + \dots) \cdot A_{DIAG}^{-1} \quad (3.32)$$

onde a proximidade que  $M$  terá de  $A$  dependerá do número de termos utilizados na Série de Maclaurin. O sistema auxiliar  $M \cdot \tilde{r} = r$  ficará com a seguinte forma:

$$\tilde{r} = (I - P + P^2 - P^3 + \dots) A_{DIAG}^{-1} \cdot r \quad (3.33)$$

onde definindo-se  $r_x = A_{DIAG}^{-1} \cdot r$  em (3.33) obtém-se a seguinte expressão :

$$\tilde{r} = r_x - (Pr_x) + P \cdot (Pr_x) - P \cdot (P(Pr_x)) + \dots \quad (3.34)$$

O número de operações matriz-vetor é sempre idêntico ao número de termos em  $P$  utilizados na Série de Maclaurin truncada. Para que a Série de Maclaurin venha a convergir é necessário que o raio espectral de :

$$R = -A_{DIAG}^{-1} \cdot A_{OFF} \quad (3.35)$$

seja menor que um, conforme já enunciado na Série de Neumann [51]. O raio espectral de um sistema é definido pelo módulo do autovalor de maior módulo, ou seja :  $\rho = \text{maior}|\lambda_i|$  para  $i = 1, 2 \dots n$ , sendo  $n$  a dimensão do sistema e  $\rho$  o raio espectral.

### 3.4.6. Pré-Condicionamento Polinomial

A matriz  $A$  de ( 3.1 ) é escalonada e decomposta em  $A = I - C$ , logo a inversa de  $A$  é dada por :

$$A^{-1} = (I - C)^{-1} = I + C + C^2 + C^3 + \dots \quad (3.36)$$

Uma aproximação desta inversa é obtida pela matriz  $M^{-1}$  calculada por um polinômio expresso por [48,49] :

$$M_v^{-1} = \sum_{j=0}^v \gamma_{j,v} \cdot C^j \quad (3.37)$$

onde  $\gamma$  são os coeficientes de cada termo do polinômio e  $v$  é o número de termos utilizados. A grande dificuldade do método está em encontrar os parâmetros  $\gamma$  que acelerem a convergência do algoritmo GCP.

### 3.4.7. Pré-condicionamento de Dois Níveis

Este pré-condicionador foi utilizado por Coutinho et alli [50] na solução de grandes sistemas lineares derivados do estudo de elementos finitos aplicado a Mecânica dos Sólidos. O sistema ( 3.1 ) é decomposto em :

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (3.38)$$

A matriz de pré-condicionamento  $M$  é bloco diagonal do tipo :

$$M = \begin{bmatrix} M_{11} & \\ & M_{22} \end{bmatrix} \quad (3.39)$$

onde  $M_{11}$  pode ser o resultado de uma fatoração de Crout no bloco  $A_{11}$  ou obtida através de uma fatoração incompleta de Cholesky neste mesmo bloco. A matriz  $M_{22}$  contém os elementos diagonais do bloco  $A_{22}$ . A utilização da fatoração de Crout, conforme [50], apresentou resultados melhores.

### 3.5. Métodos Tipo Gradiente Conjugado

Os vetores de resíduos  $r^k$ , gerados ao longo do algoritmo GC, satisfazem uma relação de recorrência de três termos, que, apesar de não ser usada explicitamente no algoritmo, é fundamental para a eficiência do método [31]. Para sistemas lineares assimétricos, geralmente esta propriedade dos vetores de resíduos é perdida, não sendo possível a obtenção da ortogonalidade destes vetores [32] ao longo do processo iterativo. Inúmeros métodos foram propostos para manter a ortogonalidade dos vetores  $r^k$  do método GC nesta situação, os quais são apresentados a seguir, sendo os seus algoritmos detalhados em [31].

O método Gradiente Bi-Conjugado ( BCG ), apresentado em [33] trabalha com uma sequência de vetores de resíduos e foi desenvolvido para trabalhar sobre sistemas lineares indefinidos. Em [34] foram sugeridas várias modificações no método Gradiente Bi-Conjugado, dando origem ao método *Conjugate Gradient Squared* ( CGS ). Devido ao comportamento irregular de convergência apresentado pelo método CGS, em que erros de arredondamento ocasionavam até a perda total da convergência, foi proposto por [31] o método Bi-CGSTAB, de convergência mais rápida e menos oscilatória. A maioria destes métodos alternativos exploram a geração de duas sequências de vetores de resíduos  $r^k$  e  $\hat{r}^k$ , associados às direções  $p^k$  e  $\hat{p}^k$ , respectivamente, que satisfazem a obtenção de mútua ortogonalidade.

Van Der Vorst el alli, em [35], citam que para matrizes  $A$  de ( 3.1 ) simétricas hermitianas a ortogonalidade dos vetores de resíduo é garantida, porém, para casos simétricos não hermitianos a perda da ortogonalidade poderá ocorrer ao longo do processo iterativo. Para matrizes hermitianas uma base ortogonal pode ser criada usando-se uma relação de recorrência com 3 termos, que formarão a base para o método GC convencional. Em [35] é sugerida a criação de um método alternativo COGC ( *Conjugate Orthogonal Gradient Method* ), que mesmo para matrizes não hermitianas, mantém a relação de recorrência de 3 termos para a formação da base ortogonal do subspaço de Krylov. O algoritmo é apresentado em [35].

### 3.6. Aplicação de Métodos do Tipo Gradiente Conjugado a Sistemas de Energia Elétrica

A utilização dos métodos do tipo GC em problemas de SEE é relativamente recente, havendo poucos trabalhos disponíveis na literatura, que abrangem, basicamente, problemas relacionados com fluxo de carga (*Load Flow*), análise de contingências e simulação da dinâmica.

No problema do fluxo de carga a utilização dos métodos GC é citada em [36], onde o método GCP é aplicado em algoritmos de fluxo de carga DC e desacoplado rápido. A resolução do sistema linear jacobiano do fluxo de potência, originado pela aplicação do método de Newton-Raphson, pelo método GCP, é proposta em [44]. Uma modificação no algoritmo padrão que resolve o sistema jacobiano pelo método GCP foi apresentada em [42], reduzindo o tempo de CPU do método em 50-60 %. Alvarado, em [45], apresenta vários estudos com a aplicação do método GCP a sistemas lineares oriundos de matrizes esparsas de sistemas de potência, onde é enfatizado o potencial de paralelização deste método. Semlyen, em [41], apresenta uma metodologia de resolução do problema de fluxo de carga, baseado no Subespaço de Krylov.

Galiana, em [36], apresenta um trabalho onde o método Gradiente Conjugado Pré-condicionado é aplicado na área de análise de contingências.

No problema da simulação da dinâmica de sistemas de energia elétrica vários estudos [1,9,10,11] propuseram a utilização do método GCP na resolução das equações associadas a rede elétrica, quando é empregado o Esquema Alternado Entrelaçado Implícito na resolução do sistema algébrico-diferencial que descreve o comportamento dinâmico do SEE em estudo, ou na solução do sistema linear jacobiano, quando da utilização do Esquema Simultâneo. O processamento paralelo na SDSEE, com o emprego do método GCP, é novamente enfatizado por Decker et alli em [43].

Outros trabalhos utilizam o método GCP em problemas de sistemas de potência. Em [67] é realizado um estudo sobre pré-condicionadores adequados para o processamento paralelo do método GCP. Em [70] os métodos GMRES e GC são comparados na solução de sistemas lineares de sistemas de potência.

### 3.7. Comentários

Este capítulo apresentou os métodos do tipo Gradiente Conjugado Pré-condicionados e várias técnicas de pré-condicionamento. Foi destacada a aplicação destes métodos em problemas de sistemas de energia elétrica. Neste trabalho será utilizado o Esquema Alternado Entrelaçado Implícito na SDSEE onde as equações associadas a rede elétrica serão resolvidas pelo método GCP tradicional, visto que a matriz  $Y$  de admitância de barras é simétrica. Foram apresentadas 7 técnicas de pré-condicionamento para o método GCP, onde tendo-se em vista a adequação dos respectivos algoritmos de pré-condicionamento ao ambiente paralelo e a sua facilidade de implementação computacional, os seguintes aspectos podem ser colocados :

- o pré-condicionadores diagonal e bloco-diagonal são perfeitamente adequados ao ambiente paralelo, onde cada processador pode resolver um subsistema  $M_{ii} \cdot \tilde{r}_i = r_i$ , conforme mostrado nas equações ( 3.21 ) e ( 3.24 ), com paralelismo perfeito. A implementação destes algoritmos é simples. Uma desvantagem da utilização do pré-condicionador bloco-diagonal no método GCP é que a velocidade de convergência do método é influenciada pela decomposição efetuada na rede elétrica.

- o pré-condicionamento por fatoração LU incompleta é bastante efetivo para processamentos sequenciais. Porém, a estrutura do método de fatoração e do próprio algoritmo, dificulta a sua eficiente implementação em ambientes paralelos.

- os pré-condicionadores por série truncada de Neumann e Maclaurin se equivalem quando  $P = A_{DIAG}$  e  $Q = -A_{OFF}$ , no caso da série de Maclaurin. Conforme mostra a equação ( 3.34 ) o grau de paralelismo deste pré-condicionador é grande, face ao acentuado número de operações matriz-vetor do algoritmo. Outra vantagem deste tipo de pré-condicionador é que ele independe da decomposição da rede elétrica em subsistemas, conforme requerido pelo pré-condicionador bloco-diagonal.

- o pré-condicionador polinomial tem como principal dificuldade encontrar os coeficientes  $\gamma$  do polinômio matricial que otimizem, devidamente, a convergência do algoritmo GCP. Já o pré-condicionamento de 2 níveis teria um potencial maior de utilização para o esquema Simultâneo face as características estruturais da matriz jacobiana. Além disso, sua implementação paralela não é muito simples.

Após as considerações acima foram selecionados os pré-condicionadores diagonal, bloco-diagonal e por série truncada de Maclaurin, pela facilidade de implementação computacional e potencial de paralelização dos seus algoritmos, para uma melhor avaliação a ser efetuada no Capítulo 5. Esta avaliação verificará a efetiva atuação de cada um destes pré-condicionadores na redução do número de condição da matriz  $Y$ , e no seu comportamento de convergência, quando da sua utilização para a solução do sistema algébrico linear associado as equações da rede elétrica ( $Y.V=I$ ). A escolha do pré-condicionador bloco-diagonal exige a decomposição da rede elétrica em subsistemas fracamente acoplados entre si. Isto motivou um estudo sobre decomposição de redes elétricas apresentado no capítulo a seguir.

# CAPÍTULO 4

## DECOMPOSIÇÃO DE SISTEMAS DE ENERGIA ELÉTRICA

### 4.1. Introdução

O pré-condicionador bloco-diagonal ( PCB ), apresentado no capítulo anterior, utiliza os blocos diagonais da matriz  $Y$  como matriz  $M$  de pré-condicionamento no método GCP, onde cada bloco representa uma subrede elétrica. Para a aceleração da convergência do método GCP na solução do sistema linear associado a rede elétrica, é conveniente que as diversas subredes sejam fracamente acopladas entre si [52], o que exigiu um estudo sobre decomposição de sistemas de energia elétrica em subsistemas de menor porte, tomando-se os trabalhos de M.Vale [52,54] como referência. Estes estudos originaram o algoritmo **DECOSEE** - Decomposição de Sistemas de Energia Elétrica, o qual foi implementado e utilizados em algumas decomposições deste trabalho. Este algoritmo e uma pequena revisão bibliográfica sobre técnicas de decomposição são apresentados neste capítulo.

### 4.2. Técnicas de Decomposição

A decomposição de grandes sistemas em sistemas de menor porte é uma técnica utilizada em várias áreas do conhecimento. No caso de sistemas de energia elétrica, M.Vale, em [52], faz uma ampla revisão bibliográfica sobre várias técnicas de decomposição, sendo as mais importantes listadas a seguir :

- por meio da teoria de grafos
- programação quadrática indefinida
- programação dinâmica
- heurísticas



- manuais
- decomposição Epsilon

As decomposições heurísticas têm sido as de maior sucesso entre várias classes de problemas de decomposição [52], pois permitem associar a teoria ao conhecimento prévio do sistema a ser decomposto. A decomposições manual, a qual depende totalmente do conhecimento que o usuário tem do SEE a ser decomposto, só é viável para sistemas de pequeno porte. M.Amano et alli, em [53], cita a decomposição Epsilon, que permite particionar grandes sistemas em subsistemas fracamente acoplados.

M. Vale, em [52], propõe a técnica de decomposição baseada no critério do *ranking* nodal. Esta técnica foi aplicada a sistemas de energia elétrica de 45 e 118 barras, tendo gerado várias decomposições diferentes, que foram comparadas com decomposições geradas por outras técnicas, como a decomposição Epsilon, por exemplo. Este comparativo foi realizado resolvendo-se o sistema de equações decomposto pelo método iterativo de Block-Jacobi, onde observou-se o número de iterações requerido para a convergência deste método. O menor número de iterações foi obtido quando o sistema de equações foi decomposto segundo o critério do *ranking* nodal [52]. Esta técnica foi utilizada como referência para o algoritmo DECOSEE apresentado neste capítulo.

### 4.3. A Metodologia do Algoritmo DECOSEE

O algoritmo DECOSEE segue, basicamente, os mesmos passos do algoritmo apresentado por M.Vale em [54], com algumas pequenas modificações conceituais referentes aos critérios utilizados para a formação dos subsistemas finais.

#### 4.3.1. Princípios Básicos

O princípio básico da metodologia proposta em [52] é a montagem dos subsistemas a partir de um número inicial de nós sementes igual ao número de subsistemas desejados. Em geral, este número de subsistemas equivale ao número de processadores que será utilizado na plataforma paralela onde o Método Gradiente Conjugado Pré-condicionado pelo PCB será implementado. Os subsistemas são formados pela agregação dos nós adjacentes a estes nós

sementes segundo determinados critérios. A escolha do nó a ser agregado depende de um *ranking* nodal [54], o qual é baseado em pesos concedidos a cada um dos nós ( barras ) que constituem o SEE a ser decomposto, conforme mostrado no subitem 4.3.2 a seguir. A cada novo passo do processo é agregado, a cada subsistema, o nó com maior *ranking* nodal ligado ao subsistema e que ainda não tenha sido absorvido por outro grupo. O processo termina quando todos os nós já tiverem sido agregados.

### 4.3.2. O *Ranking* Nodal

Os pesos utilizados na formação do *ranking* nodal  $W_i$  são calculados mediante as seguintes expressões propostas em [52,54] :

$$W_i = \sum_{L \in \Omega_i} B_L \frac{B_L}{M} \quad (4.1)$$

$$M = \sum_{L \in \Omega_T} \frac{B_L}{n_L} \quad (4.2)$$

onde :

- $\Omega_i$  = conjunto de todos os ramos conectados ao nó  $i$
- $\Omega_T$  = conjunto de todos os ramos do SEE
- $B_L$  = susceptância do ramo  $L$
- $n_L$  = número de ramos contidos no SEE
- $M$  = média aritmética total

O valor de  $W_i$  representa a magnitude da interconexão elétrica entre um dado nó e seus vizinhos [52,54]. O expoente  $B_L/M$  foi introduzido para diferenciar nós com várias ligações de baixa susceptância daqueles com poucas ligações, porém, de alta susceptância. Esta metodologia, conforme [54], conduz a matrizes com dominância bloco diagonal, o que garante a convergência dos processos iterativos, como, por exemplo, o método Gradiente Conjugado Pré-Condicionado, quando utilizado para a solução da equação da rede elétrica  $I = Y \cdot V$ .

### 4.3.3. A Escolha dos Nós Sementes

O número escolhido de nós sementes determina o número de subsistemas [54] nos quais o SEE será decomposto, sendo que diferentes escolhas de nós sementes conduzem a decomposições, geralmente, bem diferenciadas entre si, sendo todas, porém, adequadas a aplicação de métodos iterativos. As várias decomposições obtidas, ao se trocar os nós sementes, levam a subsistemas com diferentes graus de interconexão elétrica entre si [54] .

A velocidade de convergência do método GCP utilizado na solução da rede elétrica depende muito da decomposição realizada, visto que diferentes decomposições originam matrizes  $Y$  com raios espectrais distintos. Quanto menor o raio espectral da matriz  $Y$  mais rápida será a convergência do método GCP.

Os critérios apontados em [54] para a escolha dos nós sementes são :

- tentativa e erro.
- nós centrais de diferentes grupos de nós fortemente conectados.

O primeiro critério é de difícil aplicação, principalmente para sistemas de energia elétrica de grande porte. A utilização deste critério é facilitada quando o usuário está familiarizado com o SEE a ser decomposto [54]. O critério dos nós centrais tem conduzido a decomposições calculadas de forma mais rápida , além de fornecer subsistemas de dimensão similar [54]. Este segundo critério foi utilizado neste trabalho.

### 4.3.4. A Formação dos Subsistemas

Na metodologia proposta em [52], para a obtenção de  $N$  subsistemas, são escolhidos  $N$  nós sementes. A cada passo do processo é agregado ao nó semente de cada subsistema, ou a um nó já incorporado a ele, o nó de maior valor de  $W_i$  dentre os ligados ao subsistema, e que não tenha ainda sido absorvido por um outro grupo. No caso de um nó estar sendo disputado por 2 subsistemas ele será agregado ao subsistema com o qual estiver conectado com o ramo com maior módulo de admitância (  $|Y_L|$  ). Esta metodologia é ilustrada nas Figuras 4.1 e 4.2, a seguir.

A Figura 4.1 exemplifica a etapa inicial de formação de um subsistema-1, cujo nó semente NS-1 tem um grupo de 4 nós candidatos ( numerados de 1 a 4 ) a participar deste subsistema, e nenhum deles está sendo disputado por dois subsistemas diferentes. Deste grupo, o nó que possuir o maior valor de *ranking* nodal  $W_i$  será agregado. Os nós possuem valores de  $W_i$  tal que  $W_1 > W_2 > W_3 > W_4$ , logo, o nó 1 será agregado ao subsistema-1 em formação.

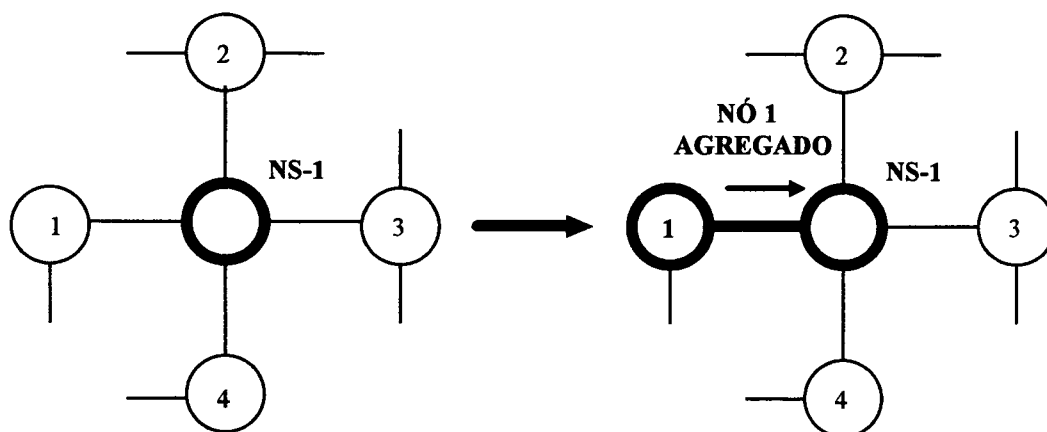


Figura 4.1 - Agregação de nós aos subsistemas - etapa inicial

A Figura 4.2 ilustra o processo de formação do subsistema-1 numa etapa mais avançada, onde todos os 4 nós possíveis de serem agregados ao nó semente NS-1 já foram agregados ao subsistema-1. Os próximos nós candidatos são aqueles conectados aos nós já agregados. Neste caso, há 6 nós candidatos ( numerados de 1 a 6 ) a participar deste subsistema-1 nesta etapa, onde foi suposto que o nó 1 é disputado por dois subsistemas, o subsistema-1 e o subsistema-2, de nó semente NS-2 . O nó a ser agregado no subsistema-1 será o de maior valor de  $W_i$  entre os 6 nós candidatos, e, caso este nó seja o nó 1, ele só será agregado ao subsistema-1 se o módulo da admitância ( $Y_B$ ) do ramo que o une a este subsistema for superior ao módulo da admitância ( $Y_A$ ) do ramo que o une ao subsistema-2 . Os nós possuem valores de  $W_i$  tal que  $W_1 > W_2 > \dots > W_6$  e o módulo de  $Y_A$  é suposto maior que o módulo de  $Y_B$ . De acordo com a metodologia proposta o subsistema-1 formado ao redor do nó semente NS-1 agregará o nó 2.

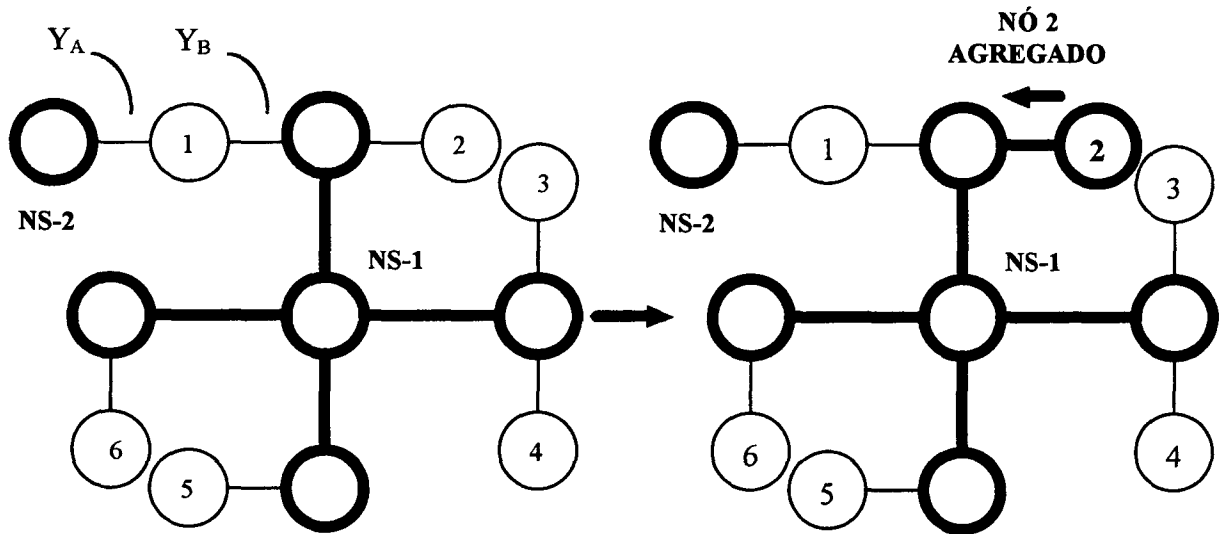


Figura 4.2 - Agregação de nós aos subsistemas - etapa avançada

A metodologia do algoritmo DECOSEE propõe alterações na maneira de montar os subsistemas. Nesta nova metodologia, para a obtenção de  $N$  subsistemas são escolhidos  $M$  nós sementes, sendo  $M > N$ , dando origem, inicialmente, a  $M$  subsistemas parciais, que são formados conforme a metodologia sugerida em [52], ilustrada nas Figuras 4.1 e 4.2. Ao final do processo de montagem dos subsistemas parciais o algoritmo DECOSEE permite que o usuário agregue estes grupos parciais visando a formação dos  $N$  subsistemas finais desejados. Nesta participação do usuário no processo de formação dos subsistemas finais as principais vantagens são :

- menor dependência de uma eficiente escolha dos nós sementes, no início do processo, para a obtenção de subsistemas fracamente acoplados entre si, pois eventuais ligações relativamente fortes entre alguns dos subsistemas parciais podem ser eliminadas pela agregação destes respectivos subsistemas na formação das subredes finais.
- melhor controle da dimensão dos subsistemas por parte do usuário, o que permite a obtenção de subsistemas, sempre que possível, com dimensões similares, o que favorece o balanceamento de carga entre os processadores utilizados na plataforma paralela onde o algoritmo Gradiente Conjugado Pré-condicionado será implementado.

As principais etapas da metodologia do algoritmo DECOSEE são ilustradas na Figura 4.3 e descritas a seguir, onde o traço mais forte da etapa 2, na Figura 4.3, indica uma interligação relativamente forte entre os subsistemas parciais *A* e *C*. O sistema inicial *S* deve ser decomposto em dois subsistemas finais (*SSIS-1* e *SSIS-2*), ou seja,  $N = 2$ .

Etapa 1 - Escolha de  $M$  nós sementes ( $M = 6$  neste exemplo).

Etapa 2 - Montagem dos 6 subsistemas de acordo com a técnica de decomposição pelo critério do ranking nodal, proposta por M. Vale [52,54]. O nó semente *B* foi mau escolhido pois agregou poucos nós ao seu subsistema. Os subsistemas parciais *A* e *C* possuem uma interligação relativamente forte entre si, as demais interligações entre os 6 subsistemas parciais são fracas. Cada subsistema está representado por uma área proporcional ao seu número de nós.

Etapa 3 - Agregação dos 6 subsistemas parciais nos 2 subsistemas finais, onde foram reunidos os subsistemas parciais *A*, *B*, *C* e *D*, para que a interligação forte entre *A* e *C* fosse absorvida pelo subsistema final *SSIS-1*, e os subsistemas parciais *E* e *F*, garantindo-se, também, sistemas com dimensões similares. O subsistema parcial formado pelo nó *B* foi também absorvido. Outra possibilidade de agregação dos subsistemas parciais seria reunir *C*, *D* e *E* em um subsistema final e *A*, *B* e *F* em outro, ainda garantindo-se sistemas com dimensões similares, porém, neste caso, a interligação forte permaneceria entre os dois subsistemas finais.

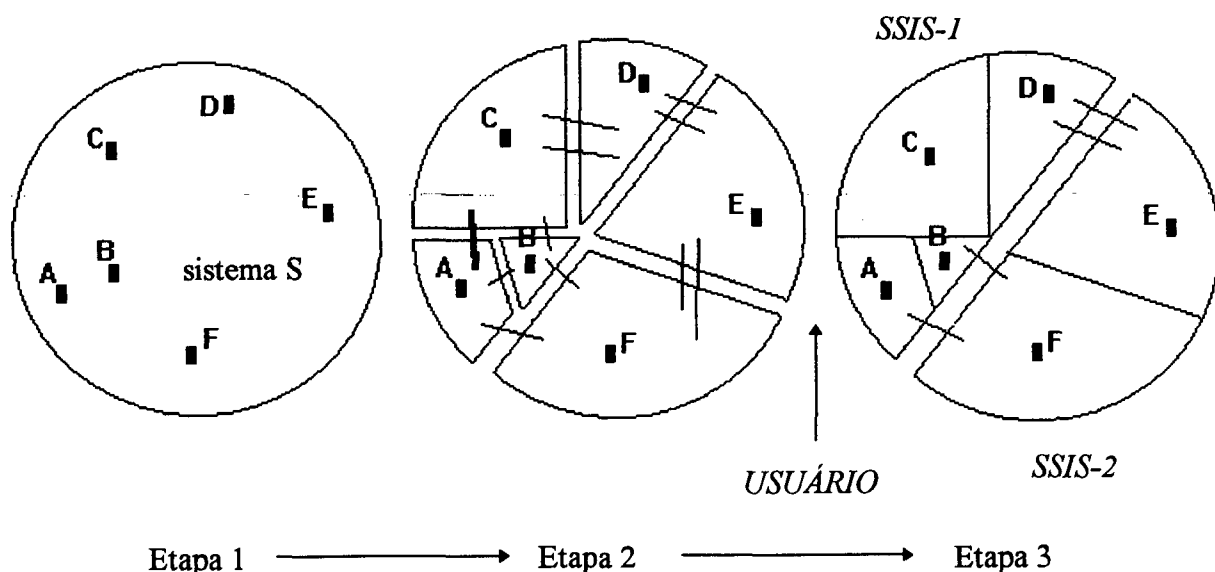


Figura 4.3 - Metodologia do programa DECOSEE.

### 4.3.5. O Algoritmo DECOSEE

O algoritmo DECOSEE é apresentado a seguir, onde é adotada a seguinte simbologia :

SSIS - subsistema	( $\rightarrow$ U ) - saída de dados para o usuário
NADJ - grupo de nós adjacentes	( $\leftarrow$ U ) - entrada de dados pelo usuário
NAGR - grupo de nós agregados	

#### ALGORITMO DECOSEE

Escolha do sistema a ser decomposto e leitura dos dados do sistema (  $\leftarrow$  U )

Cálculo do  $W_i$  de cada nó e emissão do relatório de *ranking* nodal (  $\rightarrow$  U )

Escolha do número  $N$  de subsistemas e dos  $M$  nós sementes (  $\leftarrow$  U )

Fazer-Enquanto o número de nós agregados for menor que o número total de nós

    Formação dos conjuntos NADJ com todos os nós adjacentes a cada subsistema

    Fazer-Enquanto os conjuntos NADJ não estiverem vazios

        Escolha do nó de maior  $W_i$  em cada NADJ.

        Fazer-Enquanto houver nós iguais escolhidos entre os diversos NADJ

            Verificar qual SSIS tem maior  $|Y|$  com o nó comum, este SSIS agregará este nó

            Retirada do nó comum do grupo NADJ dos outros SSIS que o disputavam

            Nova escolha do nó de maior  $W_i$  no NADJ do SSIS que perdeu o nó comum

        Fim-Enquanto.

        Agregar o nó ao respectivo SSIS, o incluindo no grupo NAGR do nó semente.

        Retirar o nó agregado dos grupos NADJ dos outros SSIS.

    Fim-Enquanto

Fim-Enquanto

Emissão dos relatórios de SSIS e barras de fronteira. (  $\rightarrow$  U )

Fazer-Enquanto houver agregação de grupos parciais

    Seleção dos grupos parciais a serem agregados. (  $\leftarrow$  U )

    Formação dos  $N$  subsistemas finais e correção das barras de fronteiras.

Fim-Enquanto

Emissão de relatórios de subsistemas finais e barras de fronteira. (  $\rightarrow$  U )

A investigação e implementação, no algoritmo DECOSEE, de técnicas automatizadas de escolha dos  $M$  nós sementes iniciais e de agregação dos subsistemas parciais ao final do processo, é necessária, principalmente, para os casos de decomposições de sistemas de muito grande porte. M. Vale, em [52], implementou um critério automatizado para a escolha dos  $N$  nós sementes no início do processo de decomposição.

### 4.4. Um Exemplo de Aplicação

O programa que utiliza o algoritmo DECOSEE foi utilizado para decompor os sistemas de energia elétrica de 118, 616 e 1916 barras utilizados neste trabalho. A seguir é apresentado um exemplo de aplicação da metodologia proposta, onde o sistema de 118 barras deve ser decomposto em 2 e 4 subsistemas finais. Inicialmente, este sistema será decomposto em 10 subsistemas parciais, que, ao final, serão agregados para a formação dos 2 e 4 subsistemas finais desejados. A Tabela 4.1 mostra os nós sementes e a distribuição do número de barras dos 10 subsistemas parciais.

Tabela 4.1 - Decomposição do sistema de 118 barras em 10 subsistemas parciais

Subsistema Parcial	Nó Semente	Número de Barras	Subsistema Parcial	Nó Semente	Número de Barras
01	112	05	06	49	22
02	68	09	07	92	20
03	42	04	08	05	13
04	28	07	09	24	10
05	15	16	10	80	12

A Figura 4.4 ilustra o sistema de 118 barras decomposto nestes 10 subsistemas parciais. Todas as interligações obtidas entre estes subsistemas são fracas.

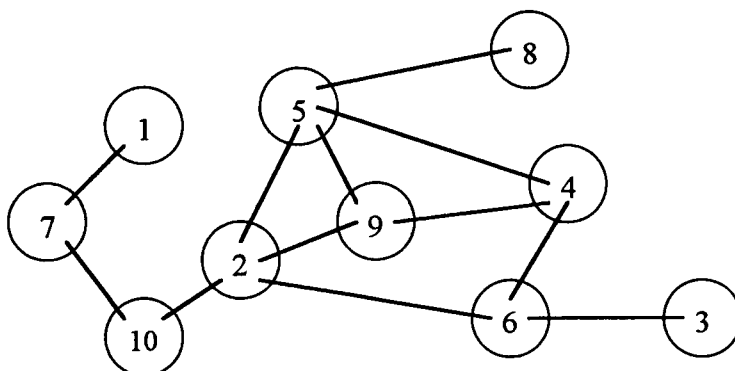


Figura 4.4 - Decomposição do sistema de 118 barras em 10 subsistemas parciais



Para a montagem dos subsistemas finais, após a etapa inicial de decomposição, devem ser observados os seguintes critérios :

- as interligações relativamente fortes entre os subsistemas parciais devem ser absorvidas dentro dos subsistemas finais.
- o número de nós de cada subsistema final deve ser de mesma ordem de grandeza.

A aplicação destes critérios ao sistema de 118 barras, aqui exemplificado, foi facilitada, pois todas as interligações entre os 10 subsistemas finais são fracas, tendo-se, então, que procurar agregar subsistemas parciais que garantam subsistemas finais com dimensões similares. Os subsistemas finais obtidos encontram-se nas Tabelas 4.2 e 4.3 .

Tabela 4.2 - Decomposição do sistema de 118 barras em 2 subsistemas finais

Subsistema Final	Subsistemas parciais agregados	Número de barras	Subsistema Final	Subsistemas parciais agregados	Número de barras
A	1	05	B	3	04
	2	09		4	07
	7	20		5	16
	9	10		6	22
	10	12		8	13
Total	-	56	Total	-	62

Tabela 4.3 - Decomposição do sistema de 118 barras em 4 subsistemas finais

Subsistema Final	Subsistemas parciais agregados	Número de barras	Subsistema Final	Subsistemas parciais agregados	Número de barras
A	1	05	C	5	16
	7	20		8	13
Total	-	25	Total	-	29
B	3	04	D	2	09
	4	07		9	10
	6	22		10	12
Total	-	33	Total	-	31

A Figura 4.5 ilustra os subsistemas finais obtidos após as agregações efetuadas :

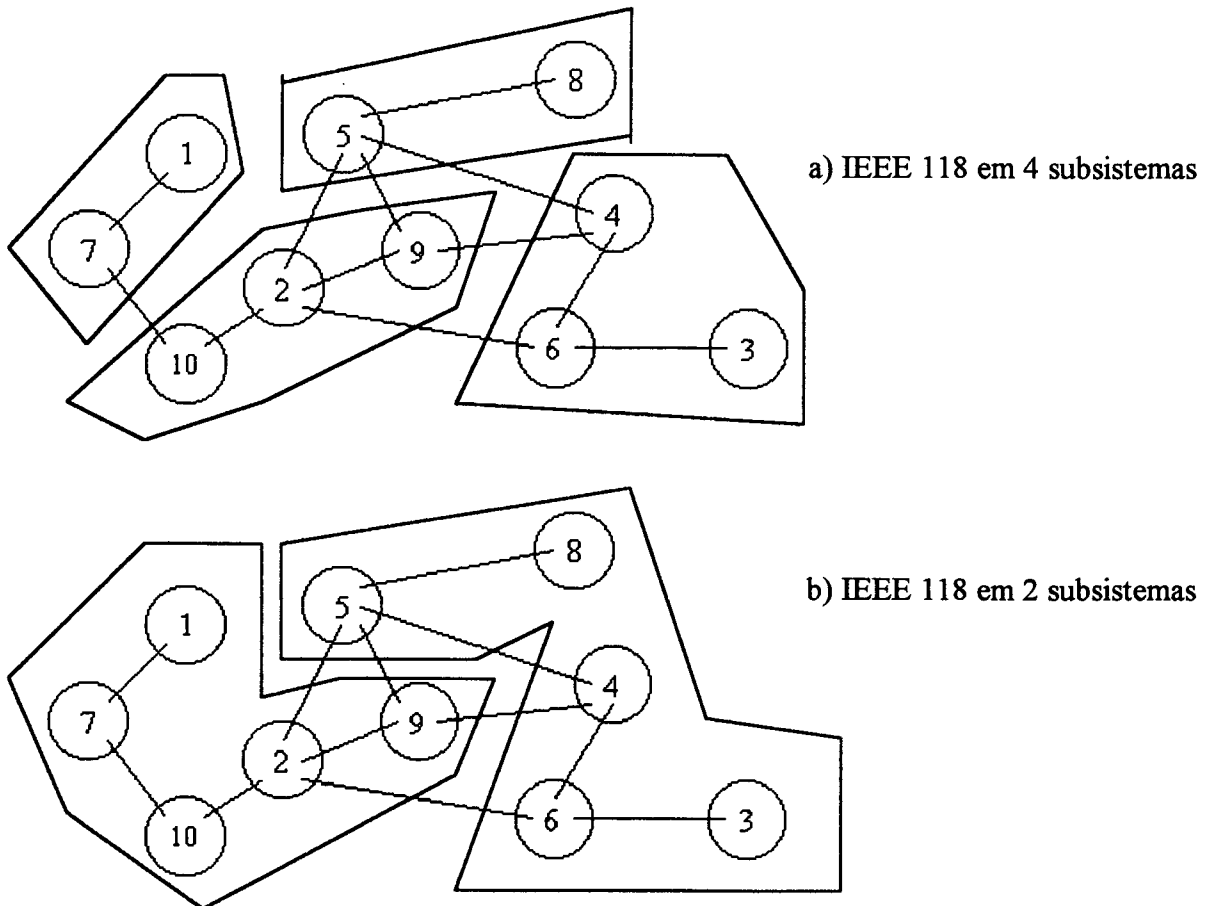


Figura 4.5 - Decomposições do sistema de 118 barras em 2 e 4 subsistemas.

Nos sistemas de 616 e 1916 barras foi adotado o mesmo procedimento, tendo sido obtidas decomposições em até 16 subsistemas finais, com uma fase inicial de 32-40 subsistemas parciais.

Na comparação entre as decomposições utilizadas em [1] para os sistemas de 118 e 616 barras, obtidas pelo critério do *ranking* nodal [52,54], e as decomposições obtidas com a utilização do algoritmo DECOSEE, não foram observadas diferenças acentuadas no número médio de iterações do método GCP, associado ao pré-condicionador bloco-diagonal, na resolução do sistema de equações associadas a rede elétrica.

## 4.5. Comentários

Neste capítulo foi apresentado de forma sucinta uma versão modificada da metodologia de decomposição de SEE em subsistemas menores apresentada em [52]. A modificação consistiu em partir com um número maior de nós sementes do que o número desejado de subsistemas finais, tendo então, ao final, a participação do usuário na agregação dos subsistemas parciais para a formação dos grupos finais. As vantagens principais desta nova metodologia são : a) a menor dependência de uma eficiente escolha dos nós sementes, no início do processo, para a obtenção de subsistemas fracamente acoplados entre si, condição necessária [52] para a aceleração da convergência do método Gradiente Conjugado Pré-condicionado na solução do sistema linear associado a rede elétrica  $I = Y.V$ , e b) o controle da dimensão dos subsistemas por parte do usuário, o que permite a obtenção de subsistemas, sempre que possível, com dimensões similares. O método GCP, associado ao pré-condicionador bloco-diagonal, convergiu, sem problemas, para os sistemas de 118, 616 e 1916 barras, na resolução do sistema de equações associado a rede elétrica decomposta pela metodologia proposta neste capítulo. O trabalho de M.Vale [52,54] foi utilizado como referência principal deste capítulo.

# CAPÍTULO 5

## AVALIAÇÃO DOS MÉTODOS DE PRÉ-CONDICIONAMENTO

### 5.1. Introdução

Os pré-condicionadores diagonal - PCD, bloco-diagonal - PCB e por série truncada de Maclaurin - PCS, selecionados no Capítulo 3 pela facilidade de implementação computacional e potencial de paralelização dos respectivos algoritmos, são avaliados comparativamente neste capítulo quanto aos seguintes aspectos :

- análise de autovalores das matrizes  $Y$  e  $R$  ( do PCS )
- redução no número de condição da matriz  $Y$  do SEE analisado
- comportamento de convergência do GCP

### 5.2. Sistemas de Energia Elétrica Utilizados na Avaliação

Os parâmetros dos sistemas de energia elétrica utilizados para o estudo comparativo entre os pré-condicionadores selecionados são mostrados na Tabela 5.1, onde o número de geradores se refere àqueles representados dinamicamente . A coluna “ modelo “ mostra o modelo utilizado para a representação dos geradores, segundo a classificação de Arrilaga [79] :

Tabela 5.1 - Parâmetros dos SEE utilizados na avaliação dos pré-condicionadores

SEE	Número de Barras	Número de Linhas	Número de Geradores	Modelo
TESTE	09	09	03	III
SSUD 45	45	72	10	IV
IEEE 118	118	179	20	III
SSUD 340	340	684	46	I
SSUD 616	616	995	88	II e III

O sistema de 9 barras foi utilizado para fins didáticos em [75], o sistema IEEE 118 é um sistema padrão de testes do IEEE [76], os sistemas SSUD 616 [1], SSUD 340 e SSUD 45 [76] representam configurações do sistema interligado Sul e Sudeste com algumas subáreas representadas por equivalentes estáticos de rede. Os sistemas de 9, 118 e 616 barras incluem sistemas de excitação. Para o estudo dos pré-condicionadores foram consideradas as seguintes contingências em cada sistema, mostradas na Tabela 5.2, que influenciam na formação das matrizes  $Y$  de cada configuração.

Tabela 5.2 - Contingências consideradas nos SEE

SEE	Barra Curto-Circuitada	Remoção da linha
TESTE	08	7-8
SSUD 45	386	382-386
IEEE 118	65	64-65
SSUD 340	1231	1231-1247
SSUD 616	933	933-1060

### 5.3. Análise de Autovalores

Os autovalores da matriz admitância de barras e da matriz  $R$ , definida no Capítulo 3 pela equação ( 3.35 ) e utilizada pelo pré-condicionador por Série de Maclaurin, são verificados para as três configurações ( pré-falta, sob falta e pós-falta ) de cada sistema de energia elétrica analisado. O raio espectral da matriz  $R$ , caso inferior a 1, garante a convergência da Série de Maclaurin.

#### 5.3.1. Matriz Admitância de Barras

O estudo dos autovalores foi feito com o software **Matlab 4**, o qual trabalha com matrizes na forma esparsa, exceto no cálculo de autovalores onde as utiliza na forma não esparsa, sendo uma de suas desvantagens a limitação de, aproximadamente 4.5 Mbytes para o armazenamento das matrizes, tendo sido utilizado este software até a dimensão de 340 barras somente. Para o cálculo dos autovalores do sistema de 616 barras foi utilizada a Biblioteca **Eispack**.

Os autovalores das matrizes  $Y$  dos sistemas de energia elétrica analisados são mostrados nas Figuras 5.1 a 5.5 a seguir :

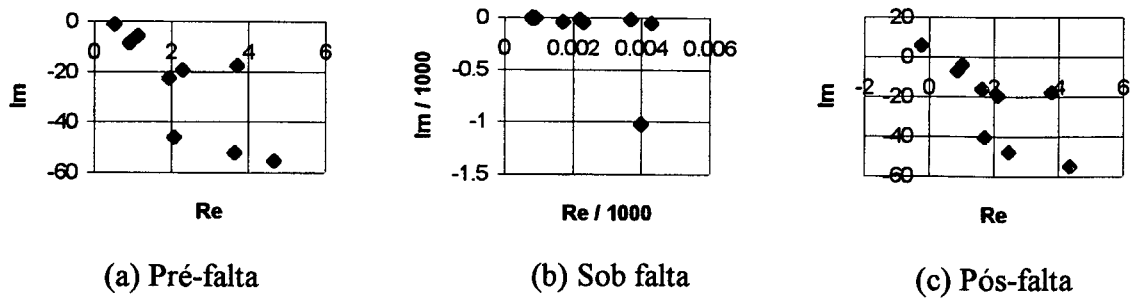


Figura 5.1 - Autovalores das matrizes  $Y$  do sistema de 9 barras

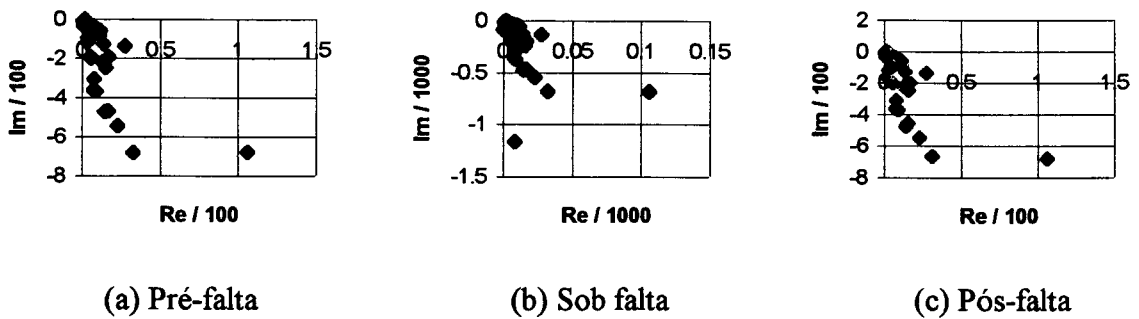


Figura 5.2 - Autovalores das matrizes  $Y$  do sistema de 45 barras

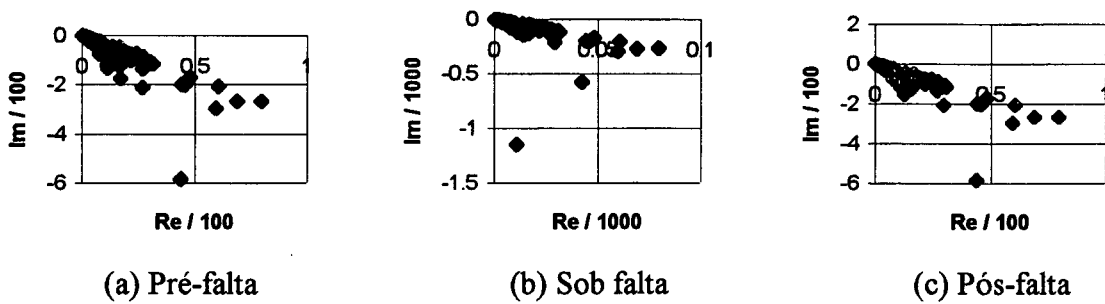


Figura 5.3 - Autovalores das matrizes  $Y$  do sistema de 118 barras

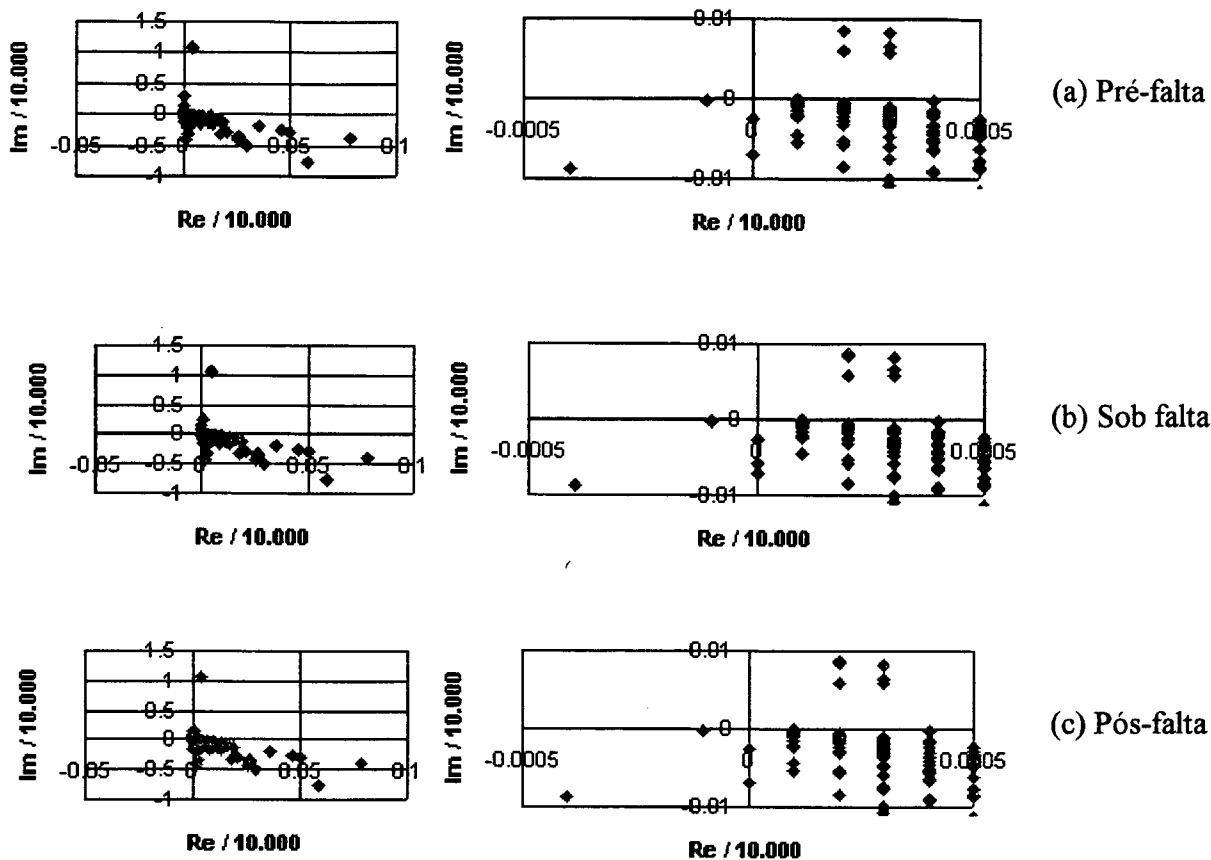


Figura 5.4 - Autovalores das matrizes  $Y$  do sistema de 340 barras com ampliação à direita de cada configuração

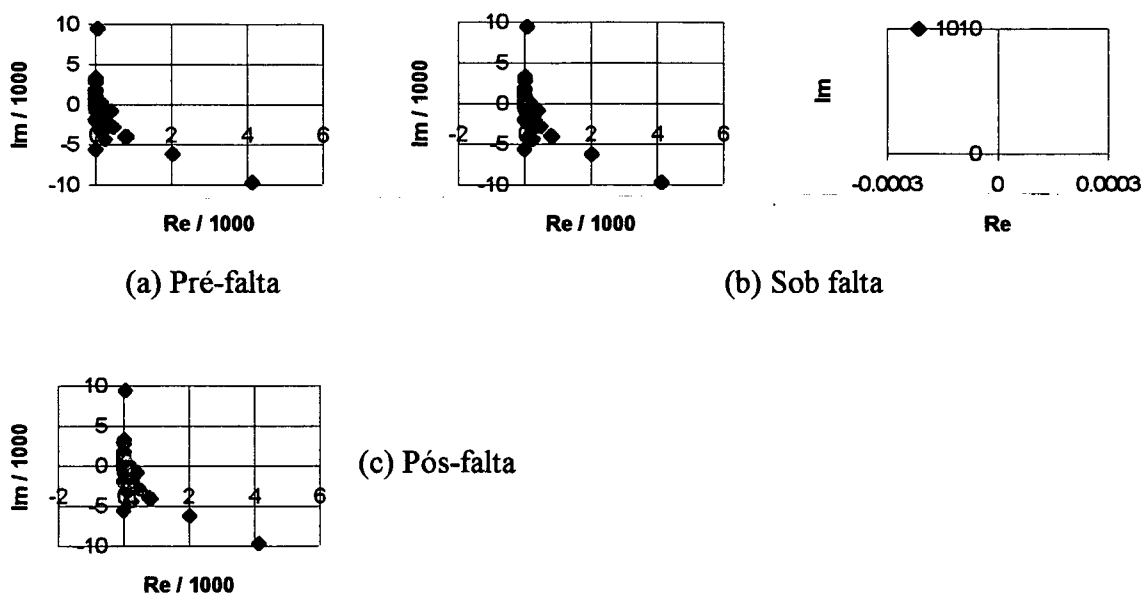


Figura 5.5 - Autovalores das matrizes  $Y$  do sistema de 616 barras, sendo que a configuração sob falta ( b ) está ampliada na sua direita.

Observou-se que as matrizes  $Y$  não são estritamente diagonalmente dominantes e que quase a totalidade dos seus autovalores tem parte real positiva. Constatou-se que nos casos em que a matriz  $Y$  possui autovalores com parte real negativa a sua correspondente matriz  $R$  tem raio espectral superior a 1. Os autovalores das matrizes  $R$  de cada subsistema são mostrados a seguir.

### 5.3.2. Matriz $R$ do Pré-condicionador por Série Truncada de Maclaurin

A matriz admitância de barras  $Y$ , de  $I = Y.V$ , pode ser decomposta em :

$$Y = Y_{DIAG} + Y_{OFF} \tag{5.1}$$

onde  $Y_{DIAG}$  contém somente os elementos diagonais da matriz  $Y$  e  $Y_{OFF}$  contém os elementos de  $Y$  fora da diagonal. A convergência da série de Maclaurin, conforme visto no Capítulo 3, é garantida se o raio espectral  $\rho$  da matriz  $R$  definida por  $R = -Y_{DIAG}^{-1} \cdot Y_{OFF}$  for menor que 1. Os raios espectrais das matrizes  $R$  dos sistemas estudados são analisados pelos autovalores destas matrizes, os quais são mostrados nas Figuras 5.6 a 5.11 a seguir. Os autovalores que estiverem fora do círculo de raio unitário são destacados com um círculo a sua volta.

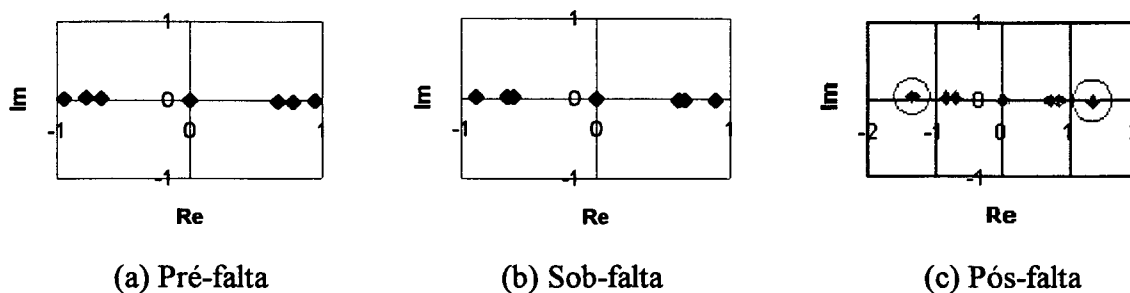


Figura 5.6 -- Autovalores das matrizes  $R$  do sistema de 9 barras

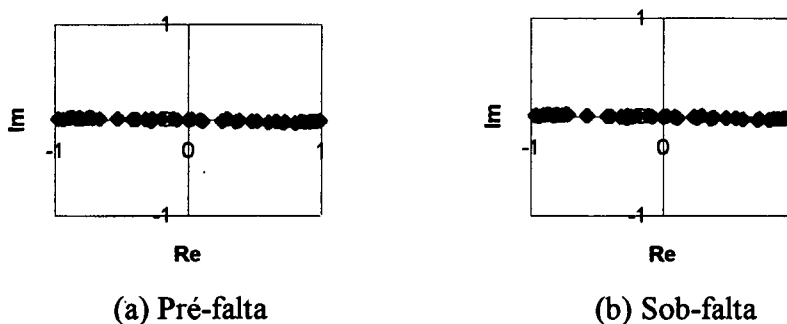


Figura 5.7 - Autovalores das matrizes  $R$  pré-falta e sob-falta do sistema de 45 barras



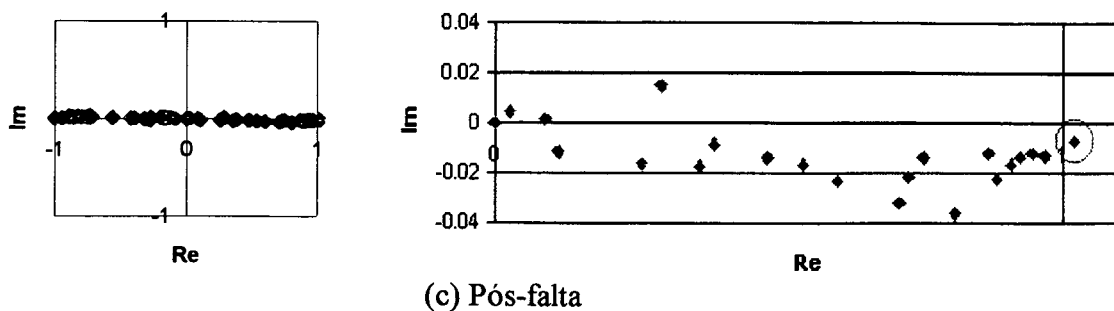


Figura 5.8 - Autovalores da matriz  $R$  pós-falta do sistema de 45 barras onde o semiplano direito dos autovalores foi ampliado a direita.

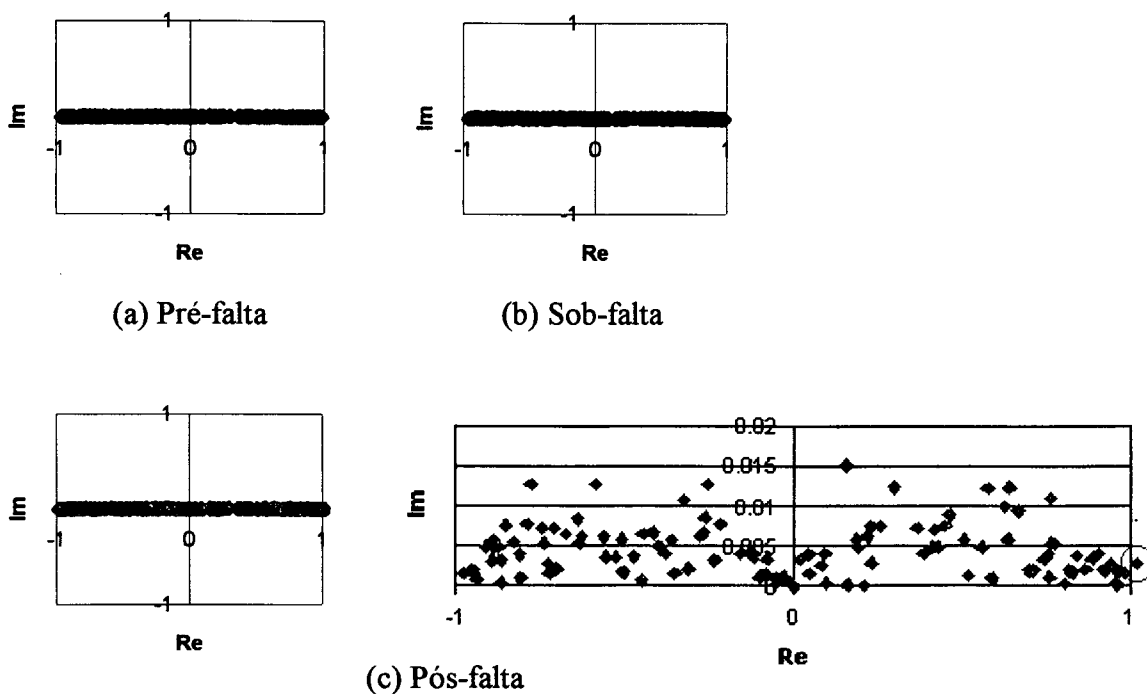


Figura 5.9 - Autovalores das matrizes  $R$  do sistema de 118 barras onde o semiplano direito dos autovalores da configuração pós-falta foi ampliado a direita.

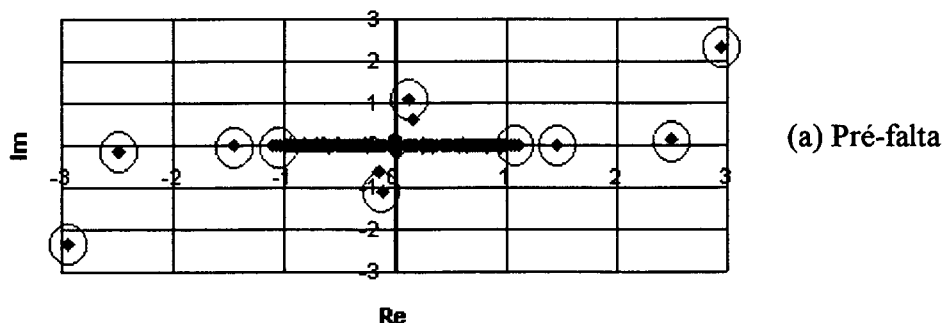


Figura 5.10 - Autovalores da matriz  $R$  pré-falta do sistema de 340 barras.

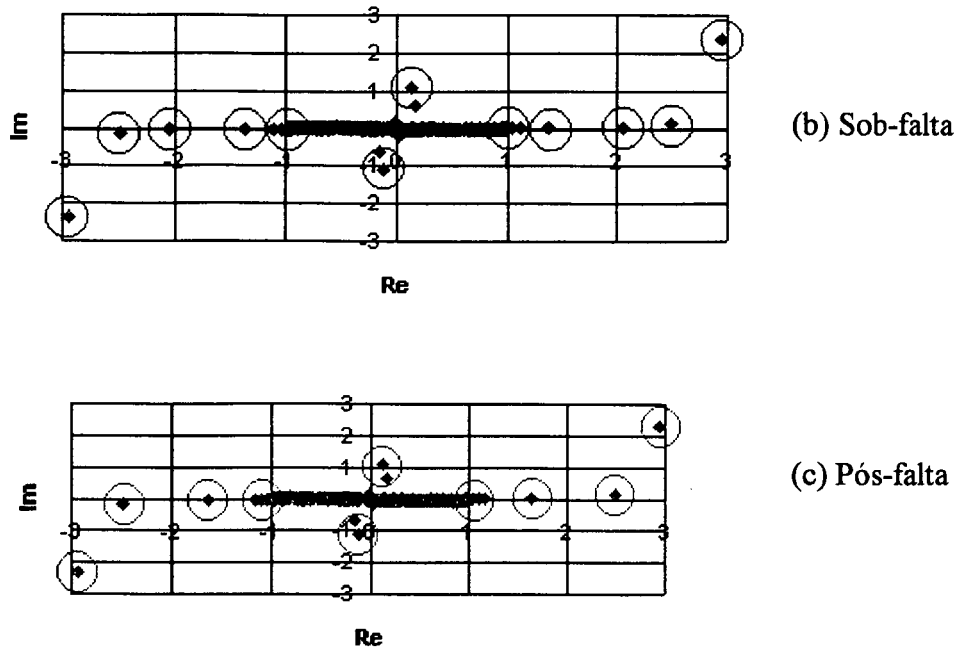


Figura 5.11 - Autovalores das matrizes  $R$  sob-falta e pós-falta do sistema de 340 barras.

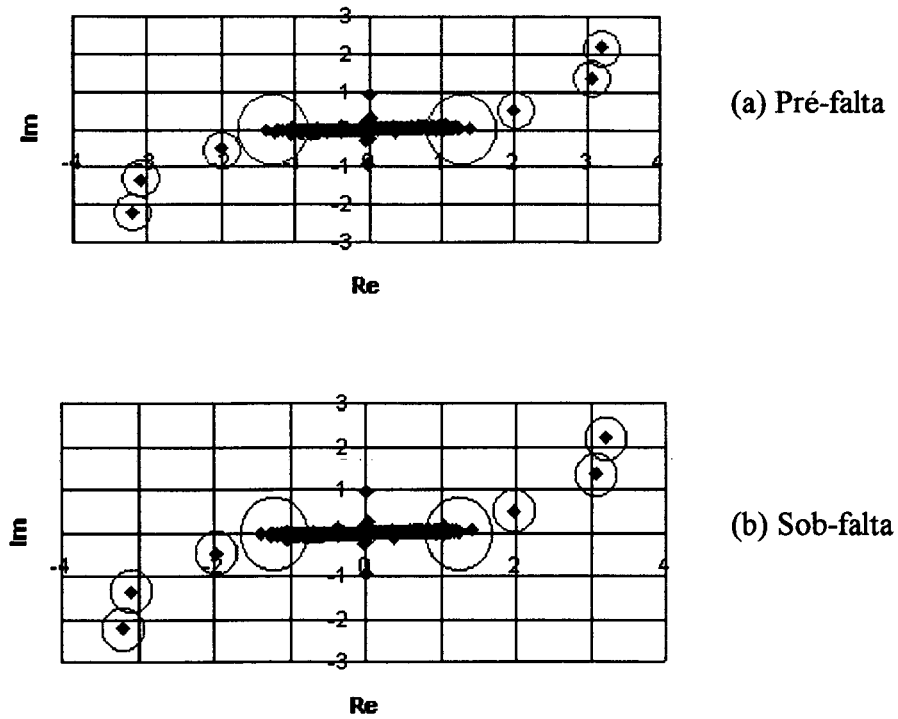


Figura 5.12 - Autovalores das matrizes  $R$  pré-falta e sob-falta do sistema de 616 barras.

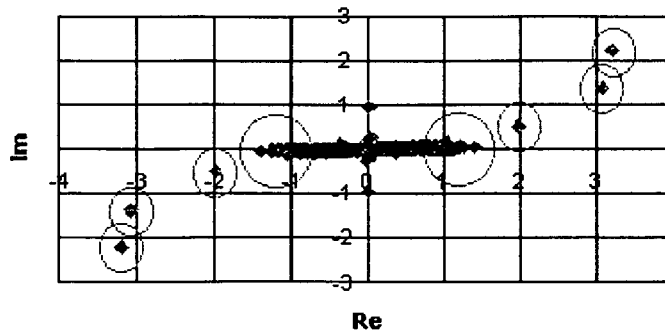


Figura 5.13 - Autovalores da matriz  $R$  pós-falta do sistema de 616 barras

### 5.3.3. Comentários

Observou-se na análise dos autovalores das matrizes  $Y$  e  $R$ , dos sistemas de energia elétrica, que sempre que a matriz  $Y$  contém autovalores no semiplano esquerdo, ou seja, autovalores com parte real negativa, a sua correspondente matriz  $R$  possui raio espectral superior a 1. Também foi observado que em todas as situações em que o raio espectral de  $R$  é superior a 1 a matriz  $Y$  correspondente não é estritamente diagonalmente dominante.

Na análise dos autovalores das matrizes  $R$  constatou-se que todos os sistemas possuem configurações em que o raio espectral de  $R$  é maior que 1, o que pode trazer problemas de convergência a Série de Maclaurin, principalmente, nos sistemas de 340 barras e 616 barras onde o raio espectral é bem maior que 1 e vários autovalores estão fora do círculo de  $\rho = 1$ . O sistema de 9 barras pós-falta também contém um autovalor que confere ao sistema um raio espectral bem superior a 1, onde a Série de Maclaurin também tenderá a ter problemas.

### 5.3.4. O Método da Compensação

Se a matriz  $Y$ , do sistema  $I = Y.V$ , for estritamente diagonalmente dominante, o raio espectral da correspondente matriz  $R$ , definida por  $R = -Y_{DIAG}^{-1} \cdot Y_{OFF}$ , será menor que 1, garantindo-se a convergência da série de Maclaurin [40]. Isto pode ser exemplificado pela matriz  $Y$  pré-falta estritamente diagonalmente dominante do sistema teste de 9 barras, que garantiu um raio espectral, da correspondente matriz  $R$ , menor que a unidade, para esta configuração, conforme mostra a Figura 5.6.(a). Neste mesmo sistema, a matriz  $Y$  pós-falta possui apenas 5 linhas com dominância diagonal, ocasionando um raio espectral maior que a unidade para esta

configuração, conforme mostra a Figura 5.6.(c). Nestas situações, as diagonais da matriz  $Y$ , onde não há dominância diagonal, podem ser reforçadas até que a matriz  $Y$  se torne estritamente diagonalmente dominante. O **Método da Compensação** permite reforçar estas diagonais da matriz  $Y$ , tornando-as diagonalmente dominantes, pela colocação de uma admitância shunt  $y_{comp}$  nas barras do SEE correspondentes a estas diagonais, em paralelo com uma fonte de corrente  $I_{comp}$ , conforme a Figura 5.14, que mostra uma barra fictícia  $i$ , com três ramos longitudinais. A equação deste equivalente Norton aplicado a barra  $i$  da Figura 5.14 é expressa por:

$$I_{comp} = Y_{comp} \cdot V \quad (5.2)$$

onde a notação utilizada foi :

$I_{comp}$  → corrente do equivalente Norton de compensação.

$y_{comp}$  → admitância do equivalente Norton de compensação com o objetivo de reforçar e/ou garantir a dominância diagonal da barra.

$V$  → tensão da barra onde o equivalente Norton de compensação está ligado.

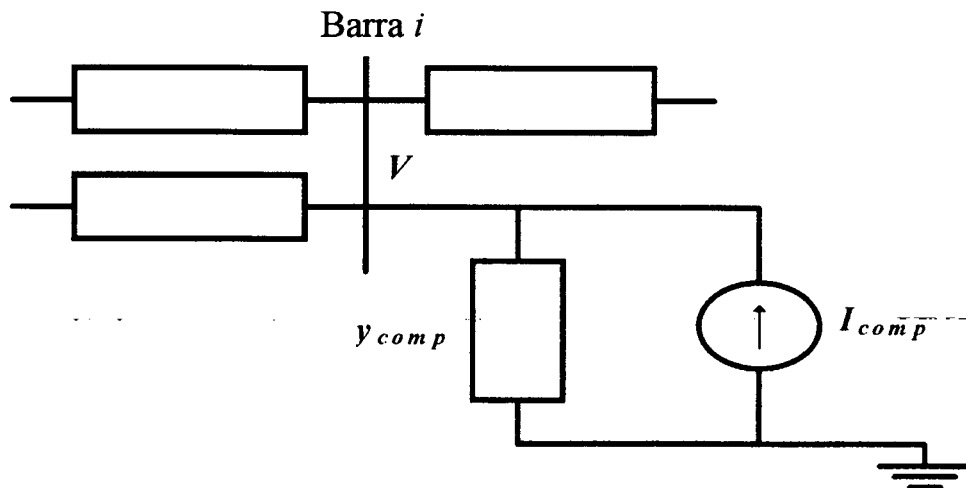


Figura 5.14 - Aplicação do Método da Compensação

A matriz  $Y$  pós-falta do sistema teste de 9 barras, ainda tomado como exemplo, após a aplicação do Método da Compensação ficou estritamente diagonalmente dominante. Foram utilizadas, no equivalente Norton, admitâncias do tipo  $g + jb$ , para  $g = 0$ , com valores

calculados para que o módulo final da admitância diagonal superasse em 10% o módulo da soma inicial dos módulos das admitâncias dos ramos conectados a barra.

Os autovalores das matrizes  $Y$  e  $R$  deste sistema de 9 barras pós-falta compensado são apresentados na Figura 5.15, ilustrando a eficiência do Método da Compensação em reduzir o raio espectral da matriz  $R$ , que antes era superior a unidade, conforme a Figura 5.6.(c). Observou-se que o autovalor da matriz  $Y$  que estava no semiplano esquerdo dos eixos complexos, mostrado na Figura 5.1.(c), foi deslocado para o semiplano direito devido ao reforço efetuado na diagonal da matriz  $Y$ , conforme pode ser constatado na Figura 5.15.(a).

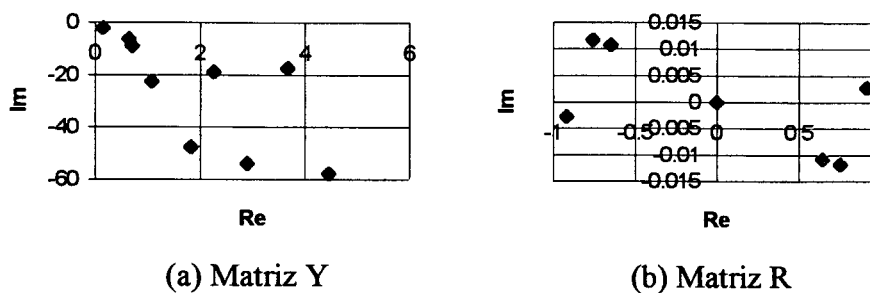


Figura 5.15 - Autovalores do sistema de 9 barras pós-falta compensado.

## 5.4. Análise pelo Número de Condição

Neste subíttem avalia-se o efeito dos 3 pré-condicionadores na redução do número de condição do sistema  $Y.V = I$ , mediante a criação de um novo sistema  $\hat{Y}.\hat{V} = \hat{I}$ , conforme mostrado no Capítulo 3. Para se observar a atuação de cada pré-condicionador é calculado o número de condição da matriz original  $Y$  e da matriz pré-condicionada  $\hat{Y}$  para cada um dos pré-condicionadores selecionados. Os gráficos mostrados nas Figuras 5.16 a 5.20 consideram a seguinte simbologia para os pré-condicionadores:

- sem PC  $\Rightarrow$  sem pré-condicionador
- D  $\Rightarrow$  pré-condicionador diagonal
- BD- $n$   $\Rightarrow$  pré-condicionador bloco-diagonal com  $n$  blocos
- S- $n$   $\Rightarrow$  pré-condicionador Série de Maclaurin com  $n$  termos na série

Os resultados dos números de condição para os sistemas estudados, sem a aplicação do Método da Compensação, são mostrados nos subítens 5.4.1 a 5.4.5 a seguir.

### 5.4.1. Sistema de 9 Barras

A Figura 5.16 mostra os números de condição para o sistema teste de 9 barras. Pode-se notar que o comportamento do PCS no sistema 9 barras pós-falta ocorreu conforme o esperado, ou seja, face ao raio espectral da matriz  $R$  pós-falta ser superior a 1, a Série de Maclaurin não convergiu, o que é mostrado pelo incremento contínuo do número de condição da matriz  $\hat{Y}$  com o aumento do número de termos da série. O uso do PCS com 1 termo da série, porém, teve sempre desempenho superior aos demais.

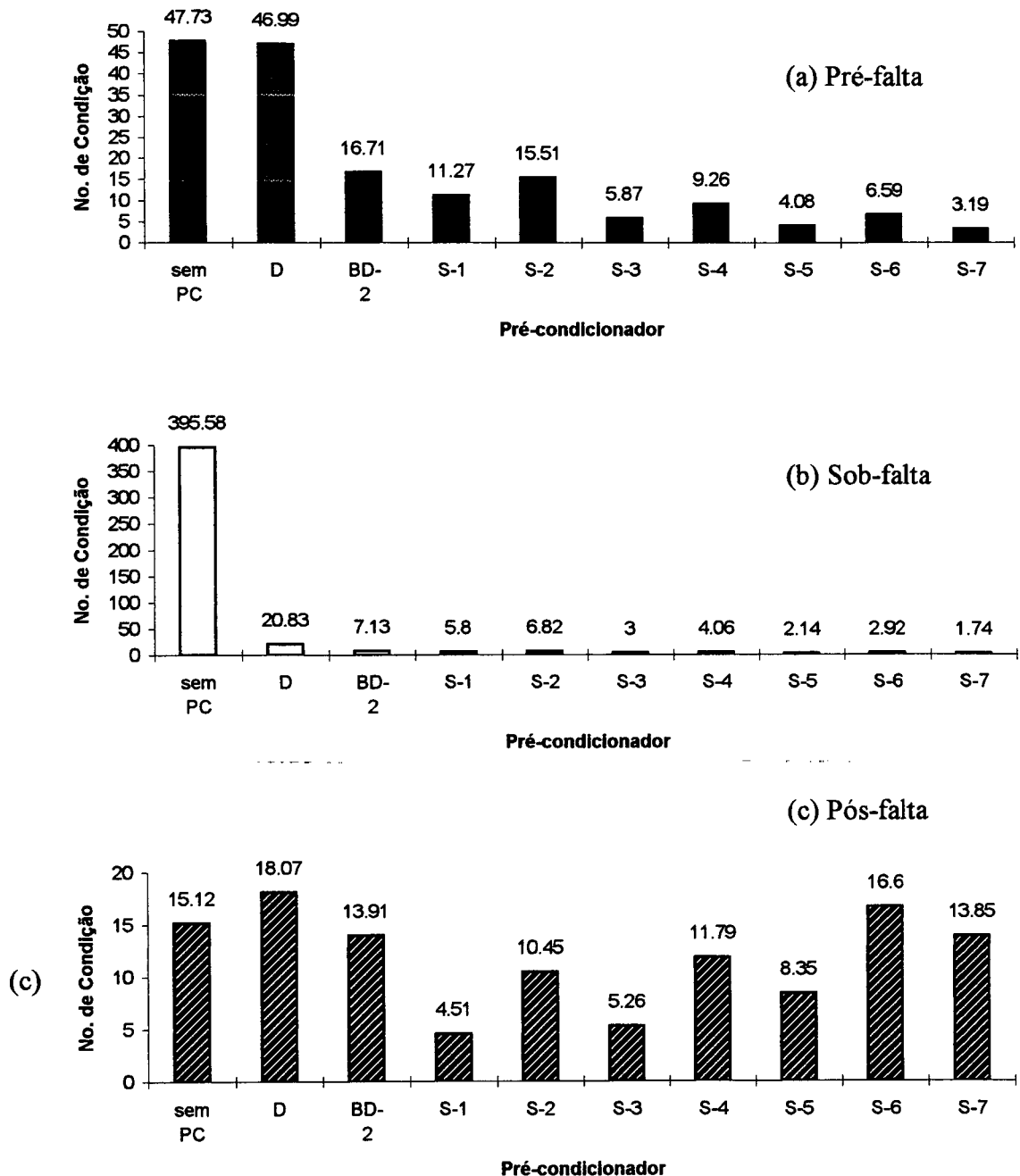


Figura 5.16 - Números de condição para o sistema de 9 barras

A Figura 5.16 mostra que o uso de um número par de termos na série no PCS tem um comportamento diferente do ímpar, peculiaridade discutida em [28]. O PCD não conseguiu reduzir o número de condição da matriz  $Y$  pós-falta.

### 5.4.2. Sistema de 45 Barras

Os números de condição para as 3 configurações do sistema de 45 barras são mostrados na Figura 5.17 a seguir :

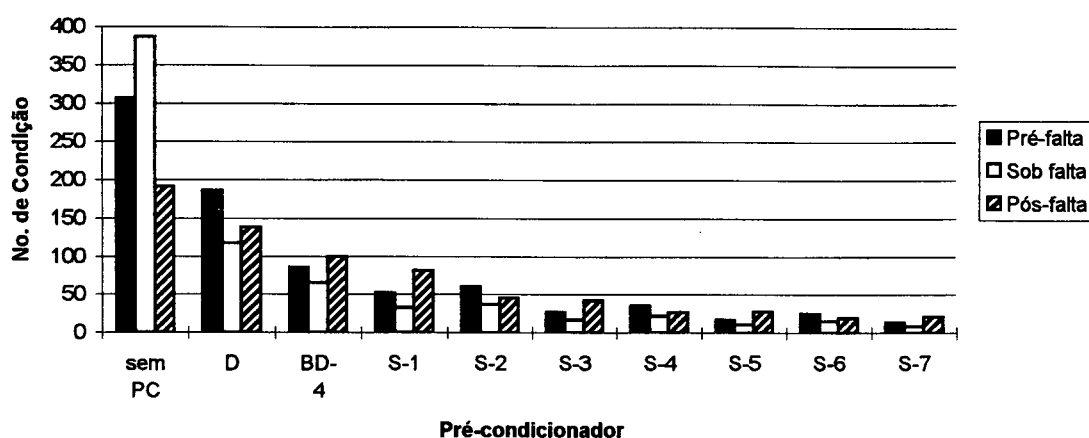


Figura 5.17- Números de condição do sistema de 45 barras

O PCS neste sistema também teve o comportamento esperado, ou seja, apesar de haver um autovalor no sistema pós-falta conferindo um raio espectral ligeiramente maior que 1, a série convergiu normalmente.

### 5.4.3. Sistema de 118 Barras

Os números de condição para as 3 configurações do sistema de 118 barras são mostrados na Figura 5.18. Neste sistema observou-se o mesmo comportamento dos pré-condicionadores que no sistema de 45 barras. O PCB com 4 blocos, quando comparado ao PCD, reduziu menos o número de condição do sistema de 118 barras pós-falta. O PCS com um único termo continua competitivo com o PCB, conforme já visto no sistema teste de 9 barras.

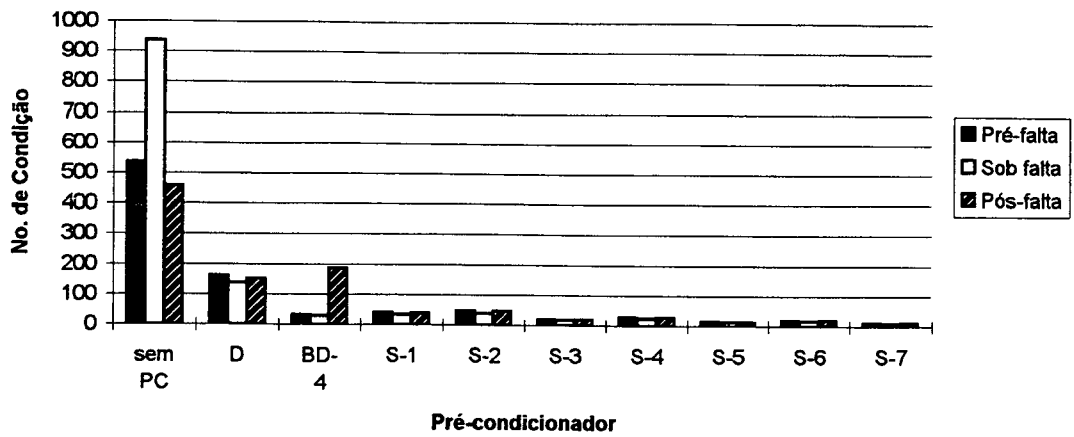


Figura 5.18 - Números de condição do sistema de 118 barras

### 5.4.4. Sistema de 340 Barras

Os números de condição deste sistema são mostrados na Figura 5.19, onde a divergência da série de Maclaurin reflete um comportamento já esperado para o PCS, face a análise feita no raio espectral da matriz  $R$  deste sistema para as três configurações.

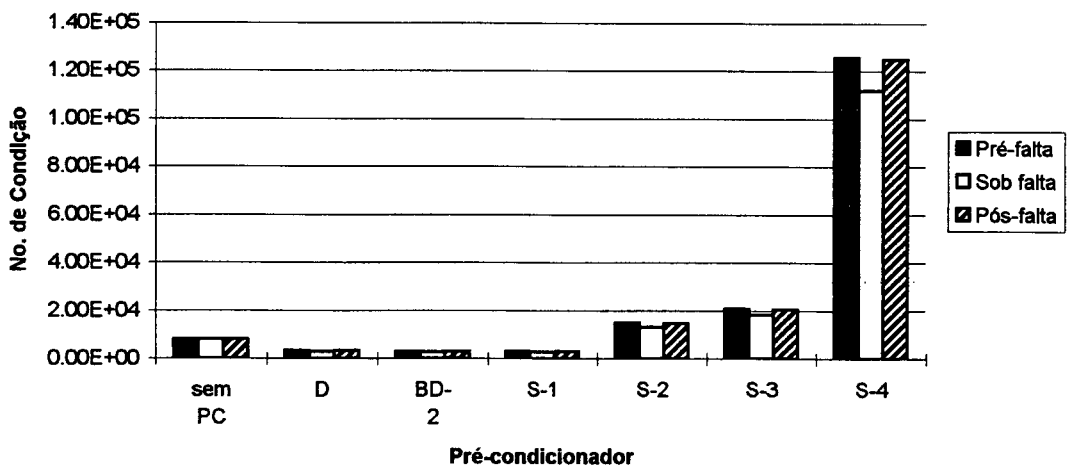


Figura 5.19 - Números de condição do sistema de 340 barras



### 5.4.5. Sistema de 616 Barras

Os números de condição para o sistema de 616 barras não foram analisados, mas a observação do comportamento de convergência da série de Maclaurin nas simulações, quando o número de termos da série era aumentado, mostrou o mesmo comportamento já constatado no sistema de 340 barras, ou seja, a série diverge com o aumento do número de termos. Estes dois sistemas possuem características semelhantes em termos de matriz  $R$ .

### 5.4.6. Sistema de 9 Barras Pós-falta Compensado

O sistema de 9 barras, utilizado como exemplo para a aplicação do Método da Compensação, tem seus números de condição para a configuração pós-falta mostrados na Figura 5.20. A série de Maclaurin apresentou convergência, sendo que neste mesmo sistema não compensado, conforme a Figura 5.16.(c), isto não ocorreu.

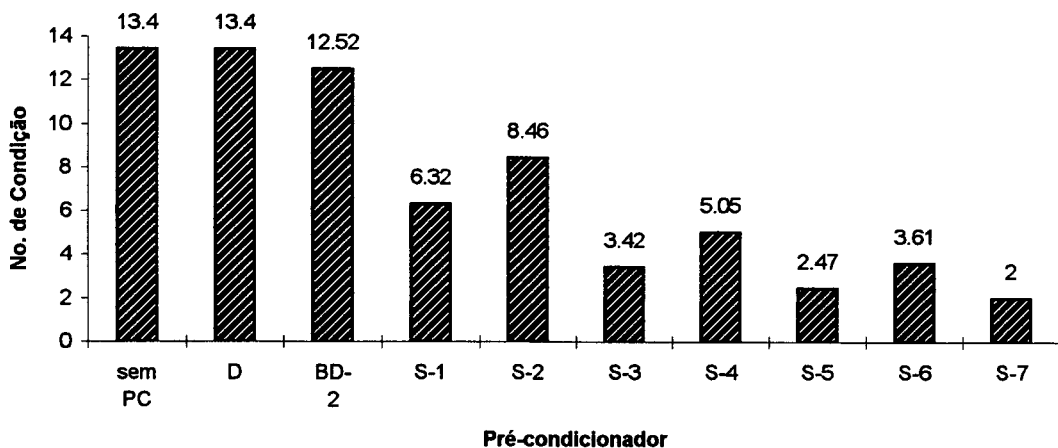


Figura 5.20 - Números de condição do sistema de 9 barras pós-falta compensado

### 5.4.7.COMENTÁRIOS

Da análise da ação dos 3 métodos de pré-condicionamento selecionados, destacam-se os seguintes aspectos principais :

- **PCS** - A maior redução do número de condição da matriz  $Y$  é conseguida pelo PCS. Mesmo nas configurações em que houve a divergência da série de Maclaurin, o PCS com um único termo da série ( S-1, nos gráficos dos números de condição ) reduziu mais o número de condição da matriz

$Y$  que o PCD e PCB. A utilização de mais termos da série de Maclaurin, no PCS, reduz ainda mais este número de condição naqueles casos em que a série é convergente. Corre-se o risco porém, de piorar o número de condição da matriz  $Y$  se a série for divergente, situação a qual só se tem conhecimento com uma prévia análise da matriz  $R$  da configuração do sistema. Esta análise é praticamente inviável de ser feita face as inúmeras combinações possíveis de contingências, como curto-circuitos e aberturas de linhas, por exemplo, que podem ser aplicadas ao sistema, tendo-se uma matriz  $R$  para cada situação. Os casos de sistemas de energia elétrica na configuração curto-circuito e pós-falta em que a série de Maclaurin converge não garantem que, em outras situações de curto-circuito e pós-falta, isso vá ocorrer. O Método da Compensação, que permite reforçar as diagonais da matriz  $Y$ , quando esta não for estritamente diagonalmente dominante, pode ser utilizado, garantindo-se que o raio espectral da correspondente matriz  $R$  será menor que 1 e que a série de Maclaurin irá convergir. Todos estes aspectos mostram que é aconselhável a utilização do PCS para pré-condicionar o GCP quando utiliza-se somente um termo da série de Maclaurin, quando é garantida a redução do número de condição da matriz  $Y$  e se evita a utilização do Método da Compensação.

- **PCD** - Este pré-condicionador mostrou ser o pior dentre os 3 pré-condicionadores analisados, tendo reduzido muito pouco o número de condição da matriz  $Y$ , sendo que no sistema de 9 barras pós-falta chegou a aumentá-lo.

- **PCB** - Este pré-condicionador mostrou um comportamento robusto, reduzindo sempre o número de condição da matriz  $Y$ , tendo como inconveniente que a velocidade de convergência do método GCP associado a este pré-condicionador depende da decomposição do sistema [52].

Na seção seguinte, será avaliado o comportamento de convergência do método-GCP associado aos pré-condicionadores PCS, com 1 termo da série de Maclaurin, e PCB, visto que, a partir deste ponto, o pré-condicionador diagonal ( PCD ) foi abandonado, face a sua baixa efetividade em reduzir o número de condição da matriz  $Y$ .

## 5.5. Análise do Comportamento de Convergência

Nesta seção é analisado o comportamento de convergência do GCP associado aos pré-condicionadores PCS, com 1 termo da série de Maclaurin, e PCB. Para o comparativo entre os

dois pré-condicionadores várias simulações foram feitas com o algoritmo que utiliza o esquema AEI com o método GCP para a solução da rede elétrica. Os parâmetros utilizados foram :

- passo de integração =  $10^{-2}$  segundos
- tolerância para as variáveis de estado =  $10^{-4}$
- tolerância para as tensões das barras ( solução da rede ) =  $10^{-3}$
- tolerância para o GC =  $10^{-2}$
- geradores representados pelo modelo III [79] de máquina síncrona, exceto no sistema de 616 barras onde algumas máquinas são representadas também pelo modelo II [79].
- as máquinas possuem regulador de tensão.

Os tempos de simulação para as configurações pré-falta, sob curto e pós-falta foram, respectivamente, de 0.50, 0.10 e 3.40 segundos para os sistemas de 9 e 118 barras, e de 0.10, 0.02 e 1.00 para o sistema de 616 barras. Foi também utilizado o PCB com um único bloco diagonal para simular um método direto baseado na fatoração LU e substituição direta e inversa. Na Tabela 5.3 abaixo são mostrados os tempos de CPU gastos na solução das equações da rede elétrica e no método GCP, assim como o tempo total de CPU da simulação, obtidos em um computador Pentium 100 Mhz. São também apresentadas as médias de iterações obtidas no esquema AEI, na rede elétrica e no GCP, calculadas pelo número total de iterações realizadas ao longo do período de simulação dividido pelo número de passos de integração.

Tabela 5.3 - Resultados das simulações para avaliação da convergência dos pré-condicionadores

SEE →	9 barras			118 barras			616 barras		
Pré-condicionador →	BD-2	LU	S-1	BD-4	LU	S-1	BD-2	LU	S-1
Tempo Rede Elétrica (s.)	2.00	2.00	2.00	7.00	4.00	5.00	19.0	3.00	366
Tempo GCP (s.)	2.00	1.00	2.00	5.00	3.00	4.00	18.0	2.00	365
<b>Tempo Total (s.)</b>	<b>3.00</b>	<b>2.00</b>	<b>2.00</b>	<b>7.00</b>	<b>5.00</b>	<b>6.00</b>	<b>20.0</b>	<b>4.00</b>	<b>368</b>
Média de Iterações AEI	2.84	2.80	3.23	2.90	2.86	1.98	2.16	2.21	2.18
Média de Iterações na Rede	1.96	1.84	2.47	1.96	1.92	2.17	1.46	1.52	1.48
Média de Iterações no GCP	1.03	1.00	1.05	1.36	1.00	1.34	10.4	1.00	242

### 5.5.1. Comentários

Das simulações realizadas para a avaliação do comportamento de convergência do GCP associado aos PCS e PCB destacam-se os seguintes aspectos principais :

- O GCP convergiu na solução da rede elétrica de todos os sistemas estudados, porém o tempo de processamento da simulação realizada no sistema de 616 barras, utilizando-se o PCS, foi muito superior ao apresentado pela simulação utilizando-se o PCB. O acentuado número de iterações do PCS no SSUD 616 pode ter sido motivado pela perda da ortogonalidade dos vetores de resíduos ao longo do processo iterativo, visto que a matriz  $Y$  é complexa não hermitiana.

- Observou-se que o PCS é muito sensível ao ponto inicial de cada configuração. Cita-se o caso do sistema de 616 barras que só convergiu usando-se aritmética de precisão dupla nos cálculos efetuados ao longo da simulação. Além disto, a média do número de iterações por solução do GCP, com a utilização do PCS, é reduzida quando é aumentado o número de passos após chaveamentos em que se aplica o método AI a rede elétrica, o que aumenta a precisão das tensões das barras.

A seguir, será realizado um estudo dos resíduos do método GCP com a utilização dos pré-condicionadores PCS e PCB, onde se poderá verificar a possível perda da ortogonalidade dos vetores de resíduos, no caso do PCS, causa provável do elevado número de iterações mostrado na Tabela 5.3 para o sistema SSUD 616.

### 5.5.2. Comportamento dos Resíduos do Método GCP

No método GC, aplicado a matrizes reais positivas definidas, os vetores de resíduos  $r$  satisfazem a seguinte condição de mútua ortogonalidade ( para  $i \neq j$  ) :

$$(r_i, r_j) = 0 \quad ( 5.3 )$$

Por outro lado, em processos iterativos de aritmética finita e para matrizes complexas não hermitianas, como a matriz  $Y$ , poderá haver perda da ortogonalidade citada acima.

Mostra-se na Figura 5.21 o comportamento da ortogonalidade dos vetores de resíduos ao longo de uma das soluções da rede elétrica pelo método GCP, utilizando-se o PCB com 4 blocos diagonais e o PCS com um termo da série de Maclaurin, para o sistema de 118 barras :

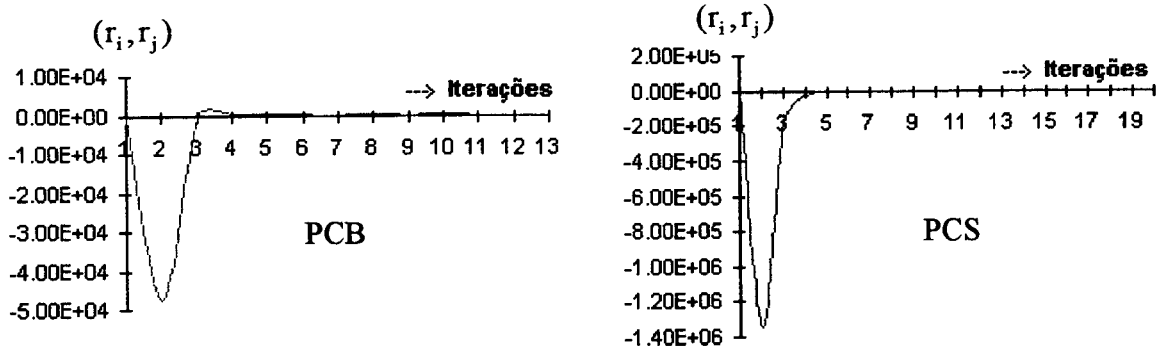


Figura 5.21 - Comportamento da ortogonalidade dos vetores de resíduo no sistema de 118 barras

Fica bastante acentuada a maior amplitude de oscilação da ortogonalidade com o uso do PCS, que inclusive levou mais iterações na solução do sistema de equações da rede elétrica.

Verificou-se também a evolução da norma 2 ao quadrado do resíduo, ao longo de uma das soluções da rede elétrica pelo método GCP, do sistema de 118 barras, mostrada na Figura 5.22, onde constata-se que, com a utilização do PCS, a norma oscila muito mais ao longo das iterações. A Figura também mostra que o GCP associado ao PCS levou mais iterações na solução da rede elétrica do que o GCP associado ao PCB.

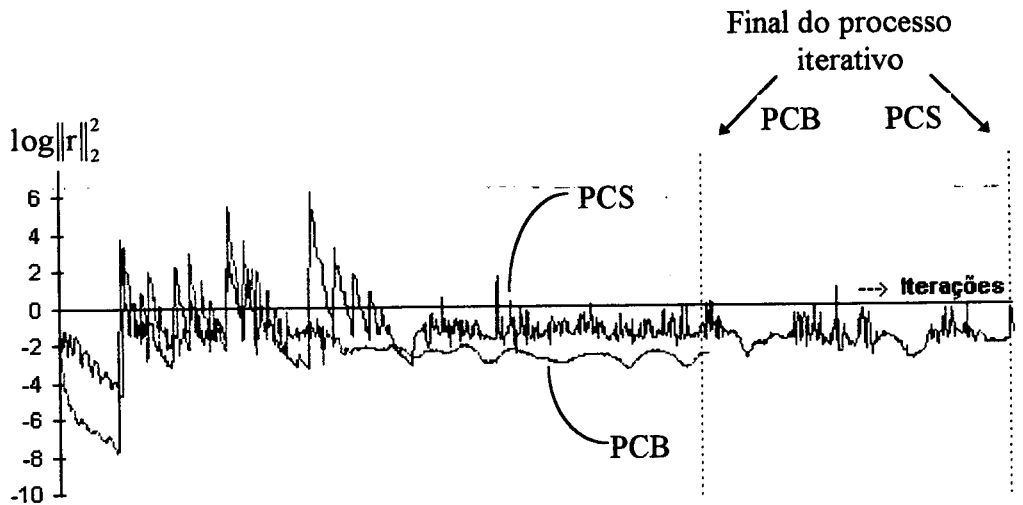


Figura 5.22 - Evolução da norma do resíduo no sistema de 118 barras

Destaca-se mais uma vez a vantagem do PCB sobre o PCS, na análise do comportamento da ortogonalidade dos vetores de resíduos ao longo de uma das soluções da rede elétrica pelo método GCP, utilizando-se o PCB com 2 blocos diagonais e o PCS com um termo da série de Maclaurin, para o sistema SSUD 616, mostrada nas Figuras 5.23 e 5.24. Nestas Figuras pode-se constatar que a ortogonalidade varia muito menos com o PCB do que com o PCS. A oscilação mostrada na Figura 5.23 é muito grande, justificando as mais de 400 iterações gastas pelo GCP associado ao PCS na solução da rede elétrica.

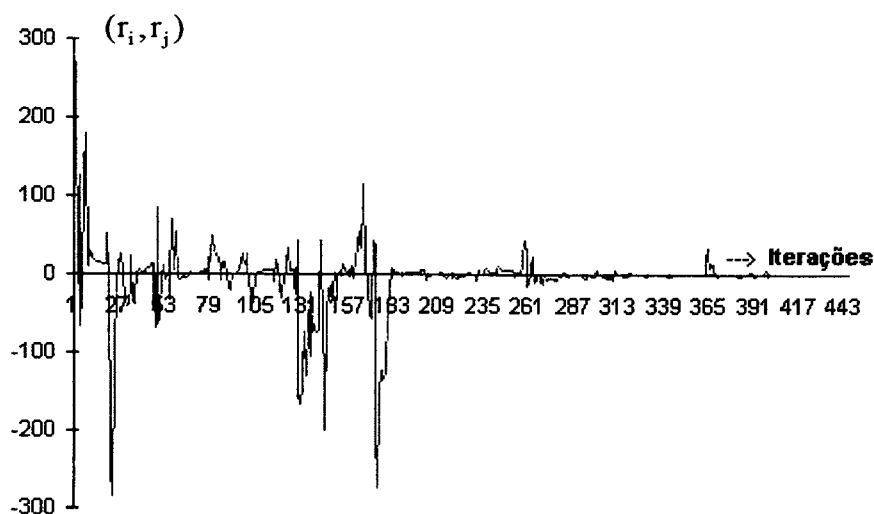


Figura 5.23 - Comportamento da ortogonalidade dos vetores de resíduo no sistema de 616 barras com a utilização do PCS 1 termo

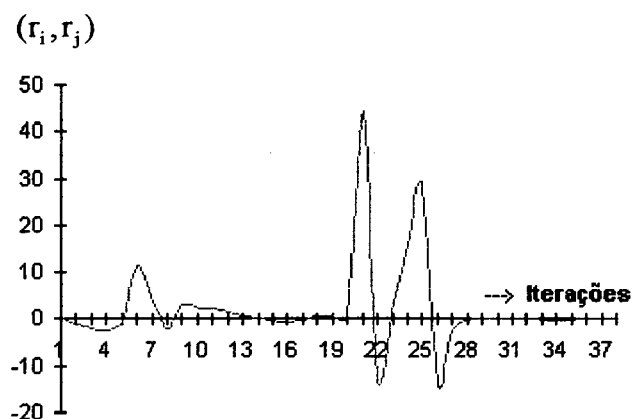


Figura 5.24 - Comportamento da ortogonalidade dos vetores de resíduo no sistema de 616 barras com a utilização do PCB 2 blocos

A análise do comportamento dos resíduos mostrou que a ortogonalidade dos vetores de resíduo não está sendo mantida ao longo das iterações do método GCP na solução da rede elétrica. As variações da ortogonalidade são significativamente mais acentuadas com a utilização do PCS. Conforme já apresentado no Capítulo 3, é citado em [35], que para matrizes simétricas hermitianas a ortogonalidade dos vetores de resíduos é garantida, porém, para casos simétricos não hermitianos a perda de ortogonalidade pode ocorrer ao longo do processo iterativo. A matriz  $Y$  de admitância de barras é complexa, simétrica e não hermitiana. Van der Vorst et alli, em [35] e [31], propõem os métodos COGC e Bi-CGSTAB, respectivamente, citados no Capítulo 3, indicados para matrizes indefinidas. Estes métodos foram implementados e os resultados obtidos são apresentados nos subítens a seguir.

### 5.5.3. O Método COGC

Em [35] é citado que para matrizes complexas hermitianas uma base ortogonal pode ser criada usando-se uma relação de 3 termos de recorrência para os vetores de resíduo, que formarão a base para o método GC convencional. Van der Vorst et alli, em [35], propõe um método alternativo (COGC), onde esta relação de 3 termos é mantida mesmo quando a matriz  $A$  de  $A.x = b$  é complexa não hermitiana. O algoritmo do método COGC, proposto em [35], foi implementado, mas não obteve-se convergência para nenhum dos sistemas estudados.

### 5.5.4. O Método Bi-CGSTAB

Este método proposto por Van der Vorst em [31], e já apresentado no Capítulo 3, foi implementado para verificar o comportamento dos pré-condicionadores PCS e-PCB. Constatou-se uma pequena redução na média do número de iterações deste método na solução da rede elétrica para os sistemas de 9 e 118 barras. O tempo de processamento, na simulação do sistema de 616 barras utilizando-se o PCB, praticamente dobrou, com relação a utilização do GCP convencional. Com o Bi-CGSTAB associado ao PCS não houve convergência.

## 5.6. Comentários Finais

Os pré-condicionadores diagonal, bloco-diagonal e por série truncada de Maclaurin foram comparados neste capítulo quanto a análise dos autovalores das matrizes  $Y$  e  $R$ , a redução no número de condição da matriz  $Y$  e comportamento de convergência do método GCP. Constatou-se que o pré-condicionador diagonal reduz muito pouco o número de condição da matriz  $Y$ .

O pré-condicionador por série truncada de Maclaurin foi o que reduziu mais acentuadamente o número de condição da matriz  $Y$ , mesmo quando utilizou-se somente um termo na série de Maclaurin. A convergência da série de Maclaurin depende de que o raio espectral da matriz  $R$ , associada ao sistema analisado, seja inferior a 1. A grande dificuldade em se utilizar mais de um termo na série de Maclaurin, para uma maior redução no número de condição da matriz  $Y$ , está em garantir que o raio espectral da correspondente matriz  $R$  seja inferior a 1. Salienta-se que a matriz  $R$  modifica-se a cada configuração do sistema ( pré-falta, sob falta e pós-falta ) e a cada chaveamento. O raio espectral da matriz  $R$  será sempre inferior a 1 se a correspondente matriz  $Y$  for estritamente diagonalmente dominante. Pode-se utilizar o Método da Compensação para reforçar eventuais diagonais da matriz  $Y$  que não tenham dominância diagonal. Foi indicada a utilização do PCS com um termo da série de Maclaurin, que além de reduzir o número de condição da matriz  $Y$ , evita o uso do Método da Compensação. Numa outra etapa de avaliação, já com o PCS utilizando somente um termo da série, se observou que o método GCP convergia com este pré-condicionador, na solução da rede elétrica, se fosse utilizada precisão dupla nos processamentos realizados. Constatou-se perdas de ortogonalidade dos vetores de resíduo durante as iterações do método GCP, associado ao pré-condicionador por série truncada de Maclaurin, na solução da rede elétrica.

O pré-condicionador bloco-diagonal mostrou um comportamento muito robusto em todos os sistemas em que foi empregado associado ao método GCP na solução da rede elétrica. A velocidade de convergência do método GCP, neste caso, depende da decomposição efetuada no sistema analisado. Esta decomposição, uma vez realizada, não necessitará ser repetida, a não ser em casos de grandes alterações na topologia da rede. Constatou-se que as perdas de ortogonalidade dos vetores de resíduo durante as iterações do método GCP, associado ao pré-condicionador bloco-diagonal, são significativamente menores do que as perdas de ortogonalidade mostradas com o pré-condicionador por série truncada de Maclaurin.



No caso de sistemas lineares com matrizes complexas não hermitianas, como a matriz  $Y$  de admitância de barras, estas perdas de ortogonalidade podem ocorrer, havendo vários métodos citados na literatura como específicos para estes casos. Dois destes métodos foram implementados, o COGC e Bi-CGSTAB. O método COGC não convergiu em nenhum dos sistemas analisados. O problema da perda de ortogonalidade persistiu com a utilização do Bi-CGSTAB, sendo que este método associado ao pré-condicionador por série truncada de Maclaurin não convergiu para o sistema de 616 barras.

Face aos aspectos acima citados, o pré-condicionador bloco-diagonal será utilizado neste trabalho, associado ao método GCP, na solução das equações algébricas lineares associadas a rede elétrica.

Os estudos realizados neste capítulo sobre pré-condicionadores e ortogonalidade dos vetores de resíduo ao longo das iterações do método GCP mostraram resultados preliminares, sugerindo-se a continuação dos trabalhos nesta linha para a obtenção de conclusões mais definitivas.

# CAPÍTULO 6

## PLATAFORMAS COMPUTACIONAIS PARALELAS

### 6.1. Introdução

Neste capítulo são apresentados alguns conceitos fundamentais em processamento paralelo, as plataformas computacionais IBM/SP1 e SP2 utilizadas nos testes computacionais e os aspectos mais relevantes de outras plataformas computacionais paralelas, usadas em outros trabalhos, para fins de comparação.

### 6.2. Conceitos Fundamentais em Processamento Paralelo

O processamento paralelo é definido por Quinn [2] como uma espécie de processamento de informações que enfatiza a manipulação concorrente de dados pertencentes a um ou mais processos, que busca a solução de um problema simples. Borrowing [59] o define como uma maneira de processar informações na qual dois ou mais processadores, com alguma forma de comunicação entre eles, trabalham juntos para a solução de um problema, enquanto Hwang [59] o coloca como uma forma eficiente de processar informações que enfatiza a exploração de eventos concorrentes no processo de computação.

A utilização do processamento paralelo neste trabalho visa a redução do tempo de computação na simulação da dinâmica de sistemas de energia elétrica, através da exploração de eventos concorrentes nos algoritmos de solução.

A seguir são apresentados vários conceitos importantes e terminologias referentes ao processamento paralelo utilizados neste trabalho.

### 6.2.1. Paralelismo e *Pipelining*

Os dois meios utilizados para conseguir a concorrência citada acima por Quinn [2] e Hwang [59] são o paralelismo e o *pipelining*, onde neste último a concorrência é aumentada pela divisão do volume computacional em um número limitado de passos enquanto o paralelismo utiliza recursos múltiplos para aumentar a concorrência [2].

Um exemplo elucidativo para diferenciar *pipelining* e paralelismo é dado por Quinn em [2]: suponha uma linha de produção de um mecanismo cuja montagem consista de 3 etapas sequenciais *A*, *B* e *C*, cada uma sendo concluída em 1 unidade de tempo. Numa produção sequencial, numa única máquina, primeiro é realizada a etapa *A*, após a *B* e, finalmente, a *C*, sendo a taxa de produção igual a 1 mecanismo para cada 3 unidades de tempo. Já no esquema *pipelined*, uma máquina *A* executa a etapa *A* em 1 unidade de tempo, passa o mecanismo para a máquina *B*, que executa a etapa *B* em mais 1 unidade de tempo, que passa o mecanismo para a última máquina *C* que conclui o mecanismo em mais 1 unidade de tempo, totalizando 3 unidades de tempo para que o mecanismo seja concluído. A diferença, contudo, para a produção sequencial, é que enquanto a máquina *B* processa a etapa *B* do mecanismo, a máquina *A* já começou um novo mecanismo, e assim sucessivamente. Fica claro que quando o *pipeline* estiver cheio, 3 mecanismos serão produzidos a cada 3 unidades de tempo, numa taxa de produção de 1 mecanismo por unidade de tempo. No processamento em paralelo, várias linhas de produção operam uma ao lado da outra, ou seja, 30 linhas farão 30 mecanismos em 3 unidades de tempo, com uma taxa de produção de 10 mecanismos por unidade de tempo. Uma vantagem evidente do paralelismo é o aumento da taxa de produção com o aumento de linhas de produção, já o esquema *pipelined* tem o próprio *pipe* como limite, que uma vez cheio chega a sua taxa limite de produção. O esquema sequencial é claramente limitado.

### 6.2.2. *Speed-up*

*Speed-up* é a razão entre o tempo necessário para o algoritmo sequencial executar determinado processamento pelo tempo utilizado pelo correspondente algoritmo adaptado para uma máquina *pipelined* ou paralela. O conceito de *speed-up* pode ser melhor entendido, utilizando-se o exemplo dos mecanismos mostrado no subitem 6.2.1, onde no esquema sequencial 10 mecanismos serão produzidos em 30 unidades de tempo, enquanto no processo *pipelined*,

supondo o *pipe* cheio, esta produção levaria 10 unidades de tempo, o que fornece um *speed-up* de :

$$\text{Speed-up pipelined} = \frac{\text{tempo sequencial para produzir 30 mecanismos}}{\text{tempo pipelined para produzir 30 mecanismos}} = \frac{30}{10} = 3$$

supondo que estes mesmos 10 mecanismos sejam produzidos em 10 linhas paralelas de produção, gastando 3 unidades de tempo, teremos um *speed-up* de :

$$\text{Speed-up paralelo} = \frac{\text{tempo sequencial para produzir 30 mecanismos}}{\text{tempo paralelo para produzir 30 mecanismos}} = \frac{30}{3} = 10$$

### 6.2.3. A Lei de Amdahl

Em um computador paralelo, contudo, nunca são conseguidas eficiências de 100%, como mostrado no subitem anterior, em que a utilização de 10 linhas de produção reduziu em 10 vezes o tempo de produção. A utilização de 10 processadores em determinado algoritmo, nunca reduzirá em 10 vezes o tempo de execução do algoritmo sequencial equivalente, visto haver saturação do algoritmo em virtude da existência de etapas eminentemente sequenciais, conforme a Lei de Amdahl. Sendo  $f$  a fração de operações que são processadas sequencialmente dentro de determinado algoritmo, onde  $0 \leq f \leq 1$ , a Lei de Amdahl mostra que o máximo *speed-up* a ser conseguido em um computador paralelo com  $q$  processadores é dado por :

$$\text{Speedup máximo} \leq \frac{1}{f + (1-f)/q} \quad (6.1)$$

o que equivale a dizer que se, por exemplo, 10% de um algoritmo deve ser processado de forma sequencial, o máximo *speedup* possível de ser atingido é 10, independentemente de quantos processadores sejam utilizados. Isto pode ser calculado fazendo-se  $q \rightarrow \infty$  na equação (6.1).

A Lei de Amdahl serve como instrumento para identificar se determinado algoritmo é potencialmente promissor em máquinas paralelas.

### 6.2.4. A Lei de Gustafson-Barsis

Em algumas aplicações de processamento paralelo já foram obtidos desempenhos superiores aos fixados pela Lei de Amdahl [81].

A Lei de Amdahl assume que  $(1 - f)$  é independente do número de processadores  $q$ . Gustafson e Barsis entenderam que esta consideração não pode ser generalizada, sendo, portanto, a razão de obter-se maiores ganhos que aqueles fixados pela Lei de Amdahl [82].

A Lei de Gustafson-Barsis apresenta uma nova interpretação da fórmula do *speed-up* apresentada pela expressão ( 6.1 ), tomando como referência o tempo gasto em processamento paralelo e não fazendo qualquer restrição a  $f$ . A nova expressão do *speed-up* é então dada por :

$$S_p = q - (q - 1)f \quad (6.2)$$

A equação ( 6.2 ) mostra que o *speed-up* é limitado por  $q(1 - f)$ , sendo, portanto, dependente do número de processadores. A Lei de Gustafson-Barsis apresenta limites para medidas de desempenho para aplicações em processamento paralelo mais favoráveis, se comparados aos limites impostos pela Lei de Amdahl.

### 6.2.5. As Arquiteturas Computacionais

Flynn, em 1966, propôs uma taxonomia de arquiteturas computacionais baseada nos conceitos de *instruction stream* e *data stream*, sendo o primeiro uma sequência de instruções realizada pelo computador, enquanto o segundo conceito é uma sequência de dados utilizados para executar uma *instruction stream*. Flynn classificou as arquiteturas de acordo com a multiplicidade do *hardware* da máquina em manusear *instruction streams* e *data streams*. As quatro classes propostas por Flynn e utilizadas até hoje são :

- **Single Instruction Stream, Single Data Stream ( SISD )** : categoria em que as máquinas podem decodificar somente uma sequência de instruções simples por unidade de tempo e que abrange, praticamente, todos os computadores seriais. As máquinas de alto desempenho que operam com *pipelining* se enquadram nesta categoria, tendo custo elevado e utilização restrita para as atuais metodologias empregadas na SDSEE [1].

- **Single Instruction Stream, Multiple Data Stream ( SIMD )** : os sistemas tipo *array processors* se enquadram nesta categoria, onde cada processador do arranjo de processadores executa uma mesma sequência de instruções simples, porém com dados diferentes. As máquinas SIMD possuem uma Unidade de Controle, vários elementos de processamento ( EP's ) com memória individual e uma rede de interconexão entre eles. Os EP's devem trabalhar de forma sincronizada [6] e, para tal, é necessário o estabelecimento de um fluxo adequado de dados entre os EP's. São máquinas indicadas para processamento de imagens ( satélites ... ) e para redes neurais, porém, seu fraco poder de comunicação entre os processadores, memória limitada e inflexibilidade de comunicação entre os EP's, as tornam máquinas pouco indicadas para serem utilizadas na SDSEE.

- **Multiple Instruction Stream, Single Data Stream ( MISD )** : nenhum computador se enquadra nesta categoria.

- **Multiple Instruction Stream, Multiple Data Stream ( MIMD )** : nesta categoria estão incluídos a maioria dos sistemas multiprocessados, onde cada processador trabalha de forma independente dos demais, podendo cada um executar um programa diferente, e compartilhando memória, capacidade de I/O e equipamentos periféricos. Os computadores Intel iPSC/860, Intel Paragon e IBM/SP1 e SP2 são exemplos de máquinas que pertencem a esta categoria. Nesta quarta categoria ainda é feita uma subdivisão com respeito a estrutura de organização da memória, ou seja :

- **Memória compartilhada ( Shared Memory )** : todos processadores utilizam uma memória comum que é o meio de comunicação de dados entre os processadores.

- **Memória distribuída ( Distributed Memory )** : cada processador tem a sua própria memória e não há memória comum. A comunicação de dados entre os processadores é feita através de uma rede de conexão por *message passing* , onde cada processador do sistema tem o seu endereço. O tipo de rede de conexão define o subtipo de arquitetura da máquina. A utilização de *message passing* é o tipo de comunicação mais popular entre as máquinas massivamente paralelas ( MPPs ) [6]. O desempenho do sistema é fortemente afetado pelo volume, frequência e velocidade de comunicação de dados entre os processadores, assim como pelo balanceamento de carga computacional entre eles.

- **Sistemas híbridos** : máquinas que trabalham simultaneamente com memórias distribuída e compartilhada. São máquinas de pouca citação na literatura.

### 6.2.6. Latência

A latência de comunicação é definida como o tempo necessário para dois processadores se conectarem e estarem aptos para a troca de dados entre si [6]. A latência de comunicação pode ser classificada em latência de *hardware* e de *software*, definidas a seguir.

- **Latência de *hardware*** : Tempo necessário para dois processadores se conectarem fisicamente, ou seja, para que haja a conexão elétrica entre eles.

- **Latência de *software*** : tempo necessário para a transferência de zero bytes entre os processadores, ou seja, além de incluir a latência de *hardware*, aqui também é considerado o tempo necessário para a máquina acionar os *softwares* e protocolos internos de transferência de dados internodais.

A latência de comunicação de uma máquina paralela é fixa e independe do tamanho da mensagem. Nos algoritmos paralelos deve-se procurar transferir o máximo possível de dados numa única operação de comunicação entre os processadores, procurando-se diluir o tempo da latência de comunicação.

### 6.2.7. Granularidade

A granularidade de uma máquina paralela pode ser definida como a relação entre a sua capacidade de processamento ( CP ), em Mflops, e a sua capacidade de comunicação internodal ( CC ), em MB/s.

$$Granularidade = \frac{Capacidade\ de\ Processamento\ (CP)}{Capacidade\ de\ Comunicação\ (CC)} \quad (6.3)$$

Máquinas paralelas de granularidade grossa são aquelas em que esta relação é alta, enquanto que, nos casos em que a relação é baixa, a máquina é dita ser de granularidade fina. Quanto maior a granularidade de uma máquina paralela maior a dificuldade de se obter altos *speed-ups* nos algoritmos nela processados.

Os algoritmos desenvolvidos para máquinas paralelas também são classificados pela sua granularidade em :

- **Granularidade Grossa** : algoritmos com tarefas grandes que podem ser processadas independentemente em paralelo. Pode-se citar como exemplo a solução de  $n$  fluxos de carga de SEE de grande porte em  $n$  processadores.

- **Granularidade Fina** : algoritmos com tarefas pequenas a serem processadas em paralelo. Pode-se utilizar como exemplo a soma dos elementos de um vetor.

### 6.2.8. Modelos de Programação

Os modelos de programação de uma máquina paralela estão relacionados a maneira como a máquina controla os programas paralelos em andamento, e podem ser subdivididos em :

- **SPMD - *Single Program Multiple Data*** : este modelo tem como característica que não existe controle centralizado e todos os processadores executam o mesmo programa, contribuindo para o processamento a ser realizado. A Figura 6.1 ilustra este tipo de modelo de programação onde  $n$  é o número do processador e  $q$  o número de processadores.

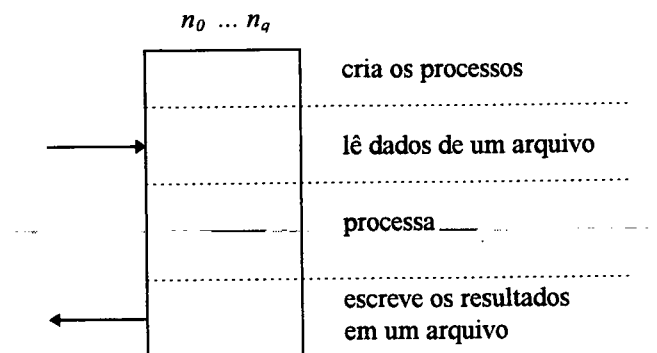


Figura 6.1 - Modelo de Programação SPMD

- **Mestre / escravo** - neste modelo o processo mestre centraliza o controle. Ele inicia os processos escravos, distribue o trabalho, sincroniza a comunicação e executa operações de I/O. São processados programas diferentes no processo mestre e nos processos escravos. O processador mestre pode ou não contribuir para o processamento do problema. Em geral, os



processadores escravos executam todo o processamento. A Figura 6.2 ilustra este tipo de modelo de programação onde  $n$  é o número do processador e  $q$  o número de processadores.

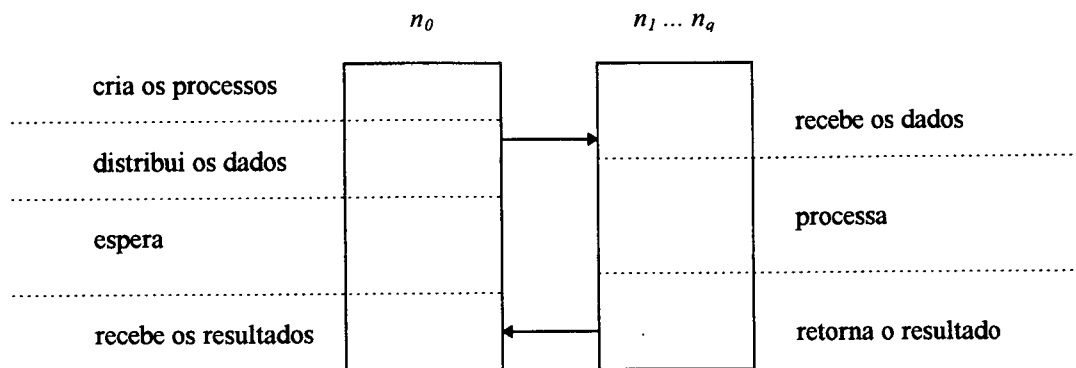


Figura 6.2 - Modelo de Programação Mestre/escravo

### 6.2.9. Transparência

A transparência de um *software* paralelo é a facilidade com que este *software*, escrito para um determinado conjunto de processadores, pode ser reformulado para outro número de processadores [5].

### 6.2.10. Portabilidade

A portabilidade de um *software* indica a facilidade com que este *software* pode ser compilado por diferentes compiladores [5].

### 6.2.11. Escalabilidade

A escalabilidade de uma máquina paralela é a medida da sua capacidade de efetivamente utilizar um número crescente de processadores [71].

## 6.3. Plataformas Utilizadas nas Simulações

As plataformas computacionais paralelas IBM/SP1 do LNCC e IBM/SP2 da UFSC, utilizadas neste trabalho, são apresentadas com maiores detalhes nesta seção.

**IBM/SP1** - O sistema IBM 9076 SP1 do LNCC é um computador paralelo com arquitetura de memória distribuída, suportando programação do tipo SPMD e MIMD. Este sistema possui 16 processadores RISC/6000 370, contendo, cada um, 256 MB de RAM, disco de 2 GB e um desempenho de pico de 125 Mflops. O sistema totaliza 2 Gflops de desempenho de pico. O disco local é utilizado para armazenamento de arquivos temporários e atividades de paginação por parte do Sistema Operacional. A área de dados e programas é suprida por dois *File Servers* ( FS ) RISC/6000 980 com 55 GB de disco no total. O acesso ao sistema paralelo IBM/SP1 é controlado por uma máquina *front end* RISC/6000 570 denominada *view.lncc.br* e, a partir dela, o usuário pode acessar os processadores do supercomputador paralelo. A latência de *hardware* é de 500 nanosegundos e a de *software* de 40 microsegundos . O dispositivo HPS - *High Performance Switch* desta máquina, responsável pela transferência de dados entre os processadores com a utilização de protocolos proprietários da IBM, tem capacidade nominal de 40 MB/s. . A granularidade nominal desta máquina é de 3.12 Mflops/MB/s. . No segundo semestre de 1995 foi feito um *upgrade* de *software* e a máquina atualmente utiliza o *software* do IBM/SP2, resultando em enorme redução da capacidade de comunicação por problemas de adequação do novo *software* a arquitetura do SP1. No primeiro semestre de 1996 está prevista a troca completa do *hardware* de SP1 para SP2 e ampliação para 40 processadores.

**IBM/SP2** - supercomputador paralelo de memória distribuída da Universidade Federal de Santa Catarina - UFSC , com 4 processadores do tipo POWER 2, sendo 2 nós finos ( tipo II *thin 66* ) e 2 nós largos ( tipo III *wide 66* ). Todos processadores têm *clock* de 66.7 Mhz e 4 unidades de ponto flutuante totalizando 266 Mflops de performance de pico e 128 MB de RAM, além de 4.5 GB de disco. A granularidade nominal desta máquina é de 6.65 Mflops/MB/s. Os nós largos diferem dos finos no tamanho da memória *cache*, 4 vezes maior nos nós largos ( 64 KB nos nós finos e 256 KB nos nós largos ) e na capacidade do barramento de dados. A estação de controle deste sistema paralelo é uma RISC/6000 250 com 64 MB de RAM e 2 GB de disco. O sistema totaliza 1 Gigaflop de desempenho de pico. Sua latência de *hardware* é de 500 nanosegundos e a de *software* de aproximadamente 27 microsegundos . O dispositivo HPS desta máquina tem uma capacidade nominal de transferência de dados de 40 MB/s. .

### 6.3.1. Comunicação Internodal no IBM/SP

Em um ambiente computacional paralelo é sempre necessário trocar informações entre os processadores durante a execução de um programa paralelo. Nas plataformas IBM/SP isto é feito, a nível de *hardware*, com a utilização do dispositivo HPS, e a nível de *software*, utilizando-se as subrotinas da biblioteca MPL ( *Message Passing Library* ).

As subrotinas da biblioteca MPL são classificadas em 6 grupos citados a seguir [72]:

**1 • Gerenciamento de Tarefas ( *Task Management Subroutines* )** - este grupo de subrotinas gerencia os programas ( tarefas ) em andamento e fornece parâmetros, tais como o estado corrente das tarefas, mensagens de erro, identificação das tarefas e outros, que permitem que o usuário acompanhe o andamento do seu programa [72].

**2 • Envio de Mensagem Sincronizada ( *Blocking Message Passing Subroutines* )** - este grupo de subrotinas controla o envio de mensagens sincronizadas entre dois processadores em conjunto com o *buffer* de dados, desbloqueando a execução do programa sempre que o *buffer* estiver liberado [72].

**3 • Envio de Mensagem Não Sincronizada ( *Non-Blocking Message Passing Subroutines* )** - este grupo de subrotinas controla o envio de mensagens assíncronas entre dois processadores, desbloqueando a execução do programa mesmo que o *buffer* de dados não esteja liberado [72]. Este tipo de subrotina exige o acompanhamento do *status* das mensagens trocadas entre os processadores.

**4 • Grupos de Tarefas ( *Task Group Subroutines* )** - este grupo de subrotinas organiza as tarefas em grupos de tarefas formando uma “ partição “. Isto permite ao usuário estruturar e controlar os seus programas [72].

**5 • Comunicação Coletiva ( *Collective Communication Subroutines* )** - este grupo de subrotinas controla o envio de mensagens entre todos os processadores e sincroniza as tarefas que estão em andamento [72].

**6 • Utilidades ( *Utility Subroutines* )** - as duas subrotinas deste grupo permitem transformar um *buffer* descontínuo de dados em um *buffer* contínuo, e vice-versa [72].

As trocas de mensagens entre os processadores das plataformas computacionais IBM/SP podem ser feitas usando-se os protocolos TCP / IP ou os protocolos proprietários da IBM, quando os dados são transferidos pelo dispositivo HPS ( *High Performance Switch* ) que possui uma largura de banda de pico bidirecional de 40 MB/s. [73] . Este dispositivo permite :

- troca simultânea de mensagens entre todos os processadores
- múltiplos caminhos entre os processadores
- que os processadores sejam considerados equidistantes em virtude da baixa latência dos estágios de chaveamento do HPS.

As Subrotinas de Comunicação Coletiva da biblioteca MPL do IBM/SP [71], mais empregadas no presente trabalho, são apresentadas a seguir, com um exemplo explicativo de cada uma.

- ***MP\_scatter*** - esta subrotina distribui mensagens distintas de um processador de origem ( *source* ) para todos os demais processadores que compõem o *pool* de execução, ou seja, faz uma difusão de dados. Este tipo de operação é também conhecida como *one-to-all personalized communication* ou *distribute* . Esta subrotina é indicada para distribuir subvetores a partir de um vetor fonte pelos vários processadores do *pool*, conforme mostra o exemplo abaixo, onde o processador de origem ( *source* ) é o nó 01:

	<u>nó 01</u>	<u>nó 02</u>	<u>nó 03</u>	<u>nó 04</u>
antes da <i>MP_scatter</i>	(1,2,3,4)	-	-	-
depois da <i>MP_scatter</i>	1	2	3	4

- ***MP\_gather*** - é a operação reversa da *MP\_scatter*, ou seja, reúne dados dispersos pelos processadores em um único ( *dest* ), fazendo uma aglutinação de dados. É também chamada de *collect* . Um exemplo desta subrotina é mostrado abaixo ( *dest* = nó 01 ) :

	<u>nó 01</u>	<u>nó 02</u>	<u>nó 03</u>	<u>nó 04</u>
antes da <i>MP_gather</i>	1	2	3	4
depois da <i>MP_gather</i>	(1,2,3,4)	-	-	-

• *MP\_combine* - esta subrotina executa uma operação de redução entre os dados disseminados pelos vários processadores do *pool* e coloca o resultado em todos os processadores. Pode-se citar como exemplos de operações de redução : somar ou multiplicar todos os dados de cada processador ( escalares ou vetoriais ), escolher o maior valor entre eles e outras. O exemplo abaixo supõe uma soma escalar :

	<u>nó 01</u>	<u>nó 02</u>	<u>nó 03</u>	<u>nó 04</u>
antes da <i>MP_combine</i>	1	2	3	1
depois da <i>MP_combine</i>	7	7	7	7

• *MP\_concat* - esta subrotina aglutina os dados de vários processadores em todos os nós, sendo também conhecida como *complete broadcast* ou *expand* , conforme o exemplo :

	<u>nó 01</u>	<u>nó 02</u>	<u>nó 03</u>	<u>nó 04</u>
antes da <i>MP_concat</i>	1	2	3	4
depois da <i>MP_concat</i>	(1,2,3,4)	(1,2,3,4)	(1,2,3,4)	(1,2,3,4)

### 6.3.2. Tipos de Processamento no IBM/SP

Os programas paralelos podem ser processados na plataforma IBM/SP de duas formas distintas [55] :

- processamento interativo
- processamento *batch*
  - ⇒ não exclusivo
  - ⇒ exclusivo

O processamento em modo interativo faz com que o programa paralelo seja processado em conjunto com os outros programas, que, simultaneamente, compartilham os mesmos processadores, visto ser o ambiente paralelo do IBM/SP1 do tipo multitarefa-multiusuário, ou seja, diferentes usuários submetem a processamento diferentes programas, ao mesmo tempo, e em nós comuns. Esta modalidade deve ser utilizada nas fases de montagem e depuração do algoritmo que está sendo desenvolvido, quando vários testes e ajustes são necessários, sem medições de tempo de CPU. Os programas processados nesta modalidade permitem a entrada de dados via console.

O processamento em modo *batch* é controlado pelo programa *Loadlever*, um gerenciador de *jobs* desenvolvido pela IBM. Cada programa *Loadlever* controla um *cluster* de máquinas (*pool*) e cada *Loadlever Pool* possui uma máquina conhecida como *Central Manager* que gerencia as diversas máquinas do *cluster* de acordo com a disponibilidade de cada uma associada aos *jobs* em processamento e processadores disponíveis [55]. O modo *batch*, ao ser acionado, coloca o *job* numa fila de submissão de *jobs*, gerenciada pelo *Central Manager* do *pool* de máquinas do *Loadlever*, quando então o usuário recebe, via console, um código que identifica o seu *job*. Nesta modalidade tem-se a possibilidade de processamento exclusivo, ou seja, no instante em que um determinado programa está sendo processado em um conjunto (*pool*) de processadores, nenhum outro *job* estará disputando estes mesmos nós, garantindo-se, com isto, medições de tempo de CPU confiáveis. Esta modalidade não permite a entrada e saída de dados via console.

#### 6.4. Plataformas Usadas na Comparação de Desempenho

As simulações realizadas neste trabalho nas plataformas IBM/SP foram comparadas com simulações realizadas com os mesmos algoritmos em outras duas plataformas paralelas descritas a seguir.

- **NCP I** - computador desenvolvido na COPPE/UFRJ, possui arquitetura hipercúbica com 8 processadores de 32 bits de tecnologia RISC, do tipo Transputer T800, com memória local de 2 MB. Cada processador possui uma capacidade de pico de 1.5 Mflops, operando a 20 Mhz de *clock*. A capacidade de pico total do sistema atinge 12 Mflops. O sistema de comunicação é do tipo *message passing* com uma capacidade nominal de transferência de dados de 2.5 MB/s. O NCP I é gerenciado por um microcomputador do tipo IBM PC 386. A granularidade nominal desta máquina é de 0.60 Mflops/MB/s.

- **Intel iPSC/860** - computador do tipo multiprocessador com memória distribuída, pertencente ao Laboratório de Computação Paralela da COPPE/UFRJ, de arquitetura hipercúbica, com 8 processadores i860 com tecnologia RISC de 64 bits e 8 MB de RAM. Cada processador possui uma capacidade de pico de 40 Mflops. A capacidade de pico total do sistema atinge 320 Mflops. O sistema de comunicação entre os nós é do tipo *message passing* com uma

capacidade nominal de pico de 2.8 MB/s. Sua latência de *hardware* é de 65 microsegundos. A granularidade nominal desta máquina é de 14.3 Mflops/MB/s.

## 6.5. Plataformas Utilizadas em Outros Trabalhos

A literatura relativa a aplicação de processamento paralelo a problemas de sistemas de energia elétrica registra a utilização de outras máquinas, sendo que as mais utilizadas terão as suas características principais listadas neste subitem.

Uma das máquinas mais utilizadas é o Sequent Symmetry [4,19,67,69,77,78]. Trata-se de uma máquina paralela de memória compartilhada, que suporta programação tipo MIMD com 26 processadores Intel 80386 16 Mhz , CPU de 32 bits, com coprocessador e um acelerador de operações em ponto flutuante ( FPA ) Weitek WTL 3167, onde somente 25 processadores podem ser utilizados em paralelo, ficando um deles encarregado de controlar o processo. A memória comum tem 128 MB e cada processador tem 64 kB de memória *cache*. A capacidade de pico total do sistema é de cerca de 40 Mflops, ou seja, equivalente a de um processador do iPSC/860 e muitas vezes inferior a capacidade das máquinas IBM/SP.

O nCUBE 2, máquina de memória distribuída, com arquitetura hipercúbica, e 128 processadores nCUBE 6400, de 64 bits, cada um com 4 MB de memória RAM ( expandível até 64 Mbytes ) é citada em [69]. A capacidade de processamento deste processador é de 2.5 Mflops e sua capacidade de comunicação é de 2.2 MB/s., o que confere uma capacidade total de pico do sistema de 320 Mflops e uma granularidade nominal de 1.14 Mflops/MB/s.

A máquina paralelas iPSC/2 com 32 processadores, arquitetura hipercúbica, capacidade de processamento de 1.5 Mflops/processador e capacidade de comunicação de 2.8 MB/s. é citada em [19]. A granularidade nominal desta máquina é de 0.53 Mflops/MB/s. e a capacidade total de pico do sistema é de 48 Mflops.

## 6.6. Síntese das Características Principais

A Tabela 6.1 apresenta uma síntese das características principais de várias plataformas paralelas, incluindo as utilizadas neste trabalho e também aquelas frequentemente citadas na literatura . As máquinas estão separadas em três grupos de acordo com a sua capacidade de

processamento. Na última linha da Tabela 6.1 está em destaque a máquina IBM/SP2 com um novo dispositivo HPS, de comercialização prevista para 1996, e que tem uma latência de *hardware* de 5 nanosegundos e 150 MB/s. de capacidade de comunicação internodal [62]. Nesta tabela CP representa capacidade de processamento e CC a capacidade de comunicação.

Observa-se na Tabela 6.1 que as máquinas paralelas, nos últimos 9 anos, tiveram um acentuado aumento na sua capacidade de processamento, não acompanhado, porém, pelo aumento da sua capacidade de comunicação, o que aumentou muito a granularidade das máquinas atuais com relação as mais antigas. Van Ness et alli, em [77], destaca este fato, ao comparar resultados obtidos, para um mesmo algoritmo, no iPSC/1, cuja comunicação entre os processadores era via rede *Ethernet*, e no iPSC/860, cuja comunicação internodal era bem mais rápida, devido a um Módulo de Conexão Direta que acelerava o envio de dados entre os processadores. O acentuado crescimento na capacidade de comunicação do iPSC/1 para o iPSC/860 foi superado pelo maior crescimento da capacidade de processamento do processador i860, quando comparado ao 80286 do iPSC/1. O resultado refletiu-se nos *speed-ups* que foram muito menores no iPSC/860.

Tabela 6.1 - Características de Várias Máquinas Paralelas

Máquina	Ano	CP/nó Mflops	CC MB/s	CP/CC Mflops/MB/s
PP/Telebrás	87	0.5	10	0.05
NCP I	89	1.5	2.5	0.60
Symmetry	87	1.5	-	-
nCUBE 2	90	2.5	2.2	1.14
iPSC/2	87	1.5	2.8	0.53
iPSC/860	90	40 •	2.8	14.3
Intel Delta	91	40 •	65	0.61
Paragon	93	50 •	200	0.25
CM-5	92	128	20	6.40
IBM SP/1	93	125	40	3.12
IBM SP/2	94	266	40	6.65
IBM SP/2 *	96	266	150	1.77

\* a ser lançada em 1996.

• fonte : [72]

Constata-se pela Tabela 6.1 que as máquinas Symmetry, nCUBE 2 e iPSC/2, muito utilizadas em trabalhos apresentados recentemente na literatura [4,19,67,69,77,78], já são



completamente obsoletas quando comparadas com as máquinas atuais, em termos de capacidade de processamento. Estas máquinas possuem baixa granularidade, justificando-se os excelentes *speed-ups* apresentados nos artigos técnicos [4,19,67,78]. Observa-se também na Tabela 6.1 que a máquina NCP I, utilizada em [1], é uma das que possui granularidade mais fina, o que favoreceu a obtenção dos altos índices de *speed-up* apresentados por Decker em [1]. A máquina iPSC/860, de granularidade média alta, também apresentou bons índices de *speed-up* em [1], porém inferiores aos conseguidos com o NCP I.

## 6.7. Comentários

Este capítulo apresentou alguns conceitos fundamentais em processamento paralelo, as plataformas computacionais paralelas IBM/SP utilizadas nas simulações deste trabalho e dados de outras plataformas utilizadas em outros trabalhos, com destaque para o NCP I e iPSC/860, para comparações de desempenho. Um destaque especial foi dado a algumas características de *hardware / software* do IBM/SP como a comunicação internodal, onde foram abordados aspectos sobre o dispositivo HPS, protocolos de comunicação e a biblioteca MPL, os seus grupos de subrotinas e as principais subrotinas de comunicação coletiva. Os diversos tipos de processamento de programas paralelos também foram apresentados, onde foram descritas as modalidades interativa e *batch*. As principais características das plataformas paralelas mais utilizadas na literatura também foram apresentadas. A Tabela 6.1, ao final deste capítulo, mostrou uma síntese das características principais de várias plataformas paralelas, onde foi destacado o novo IBM/SP2, com um HPS de 150 MB/s. de capacidade de comunicação internodal, que será lançado em 1996, reduzindo a granularidade desta plataforma.

Uma avaliação da Tabela 6.1 mostra que as máquinas mais citadas na literatura já estão obsoletas em termos de capacidade de processamento, quando comparadas com as máquinas atuais. O acentuado crescimento da capacidade de processamento das máquinas paralelas nos últimos 9 anos não foi acompanhado, de forma similar, pelo crescimento da capacidade de comunicação das máquinas, conforme mostra a Tabela 6.1. As máquinas paralelas atuais possuem, conseqüentemente, uma granularidade mais alta, que dificulta a obtenção de altos índices de *speed-up* nestas máquinas. A granularidade das máquinas paralelas mais citadas na literatura é fina, o que facilitou a obtenção de altos índices de *speed-up*. A plataforma NCP I, utilizada em [1], onde são citados altos *speed-ups*, possui também granularidade fina.

# CAPÍTULO 7

## IMPLEMENTAÇÕES COMPUTACIONAIS

### 7.1. Introdução

Neste capítulo são apresentados os algoritmos sequenciais e paralelos de resolução das equações que descrevem a dinâmica de sistemas de energia elétrica, utilizados no presente trabalho. Estes algoritmos utilizam o Esquema Alternado Entrelaçado Implícito, cujo algoritmo e equações foram apresentados no Capítulo 2 deste trabalho. Foram feitas implementações de dois algoritmos sequenciais e de um algoritmo paralelo nas plataformas computacionais IBM/SP1 do Laboratório Nacional de Computação Científica ( LNCC ) e IBM/SP2 da Universidade Federal de Santa Catarina ( UFSC ).

### 7.2. Algoritmos Implementados

Os algoritmos implementados e apresentados a seguir são :

- **AEI com GC paralelo** - algoritmo paralelo que utiliza o esquema AEI na resolução do sistema algébrico-diferencial que descreve a dinâmica de um sistema de energia elétrica, e o método iterativo GCP na solução das equações algébricas lineares associadas a rede elétrica.
- **AEI com GC sequencial** - versão sequencial do algoritmo anterior.
- **AEI convencional sequencial** - algoritmo sequencial que utiliza o esquema AEI na resolução do sistema algébrico-diferencial que descreve a dinâmica de um sistema de energia elétrica, e o método direto baseado na fatoração LU e substituição direta e inversa na solução das equações algébricas lineares associadas a rede elétrica.

Destaca-se que os dois primeiros algoritmos foram desenvolvidos e implementados por Decker [1] nos computadores paralelos iPSC/860 e NCP I.

O esquema AEI convencional foi implementado com a finalidade principal de servir de instrumento referencial na comparação de desempenho entre os algoritmos.

### 7.2.1. O Algoritmo Paralelo AEI com GC

O algoritmo paralelo, desenvolvido e utilizado para a SDSEE por Decker em [1], foi utilizado como referência para este trabalho. Este algoritmo utiliza o método Gradiente Conjugado associado a um pré-condicionador bloco-diagonal na resolução das equações algébricas lineares associadas a rede elétrica. Os estudos apresentados no Capítulo 5 deste trabalho mostraram ser este o pré-condicionador mais indicado para aplicação em ambiente paralelo. Além disto, para as situações de alteração na configuração da rede elétrica, apenas os blocos diagonais da matriz  $M$  de pré-condicionamento afetados por estas alterações, precisarão ser redefinidos nos instantes de chaveamento, com os respectivos fatores LU. Os passos principais do algoritmo paralelo AEI com GC são :

- extrapolação das tensões nas barras e variáveis de interface  $u^*(t)$  e  $V^*(t)$ .
- cálculo de  $x^o(t)$  e  $x^{k+1}(t)$  usando as equações algébricas a diferenças oriundas das equações diferenciais.
- cálculo das injeções de corrente  $I(E^k, V^k)$ .
- resolução das equações linearizadas da rede elétrica  $I(E^k, V^k) = Y.V^{k+1}$ .
- cálculo das variáveis de interface  $u^k(t)$ .

Decker, em [1], baseado nestes passos principais, propõe decompô-los em  $q$  tarefas que serão processadas em  $q$  processadores, fazendo ainda as seguintes observações :

- as componentes de  $u(t)$ ,  $E(t)$  e  $I(t)$  são funções somente de variáveis da barra a qual estão associadas, sendo que as componentes de  $u(t)$  e  $V(t)$  são extrapoladas independentemente.
- o acoplamento entre as variáveis de estado de máquinas síncronas diferentes, se dá somente através de variáveis da rede elétrica, conforme a modelagem apresentada no Capítulo 2. Portanto, as equações diferenciais de máquinas diferentes podem ser integradas simultaneamente e independentemente.
- no algoritmo GC, a predominância do esforço computacional está restrita a três tipos de operações básicas : a) somas de vetor por vetor multiplicado por um escalar; b) produtos internos; c) produtos de matriz por vetor. Em princípio, a decomposição desses vetores e matrizes pode ser feita trivialmente.
- os sistemas auxiliares de pré-condicionamento, usando-se a matriz  $M$  bloco diagonal proposta, tem decomposição trivial e a resolução dos subsistemas é completamente desacoplada.

O elevado potencial de paralelização do algoritmo AEI com GC depende de que a matriz de pré-condicionamento bloco-diagonal  $M$ , com os blocos diagonais de  $Y$  ( matriz admitância nodal ), atenda os seguintes requisitos [1] :

- é necessário que os blocos diagonais da matriz  $Y$  sejam fracamente acoplados entre si ( dominância bloco diagonal ) para que  $M$  seja efetiva como pré-condicionador [52]. Esta forma de  $Y$  é conhecida na literatura como NBDF ( *Near Block Diagonal Form* ) e é equivalente a decompor a rede elétrica em subredes fracamente acopladas entre si.

- o número de subredes é fixado a priori em valor igual ao número de processadores que se quer utilizar.

- o número de interligações entre as subredes deve ser o mínimo possível, a fim de minimizar o volume de comunicação de dados entre os processadores na realização dos produtos do tipo matriz por vetor, no algoritmo GC.

- as características das subredes, em termos de dimensão e complexidade, devem ser tais que a solução do sistema auxiliar de pré-condicionamento e os produtos do tipo matriz por vetor, possam ser feitos com um aceitável balanceamento da carga entre os processadores.

O algoritmo paralelo AEI com GC é apresentado a seguir, onde a variável  $i$  refere-se a  $i$ -ésima tarefa (  $i = 1, 2, 3, \dots, q$  ) :

### ALGORITMO AEI COM GC PARALELO

Inicialização

Para  $t = 1, 2, \dots, T$

Para  $i = 1, 2, \dots, q$

Calcule  $u_i^*(t)$  e  $V_i^*(t)$ , por extrapolação

Calcule  $x_i^0(t) = H_i[x_i^*(t), u_i^*(t), x_i(t - \Delta t), u_i(t - \Delta t)]$

Faça :  $k \leftarrow 0$

Enquanto  $\Delta x_i^k(t) \geq \varepsilon$ , para algum  $i$ , faça :

Calcular  $E_i^k(t)$  e  $I_i^k(t)$

Resolver  $I^k(t) = Y.V^{k+1}(t)$  pelo GCP

Calcular  $u_i^{k+1}(t) = h_i(E_i^k(t), V^{k+1}(t))$

Calcular  $x_i^{k+1}(t) = H_i[x_i^k(t), u_i^{k+1}(t), x_i(t - \Delta t), u_i(t - \Delta t)]$

Calcular  $\Delta x_i^{k+1}(t) = \|x_i^{k+1}(t) - x_i^k(t)\|$

Faça :  $k \leftarrow k + 1$

Fim - Enquanto

Fim - Para

Fim - Para

A Figura 7.1 mostra um diagrama de blocos do algoritmo paralelo AEI com GC, obtido de [1], onde são destacadas as comunicações de dados entre os processadores. A etapa de resolução paralela das equações da rede elétrica, representada por um retângulo tracejado, onde tem-se  $I(E^k, V^k) = Y.V^{k+1}$ , é detalhada no diagrama da Figura 7.2, também obtida de [1]. Os outros retângulos tracejados indicam comunicação entre os processadores.

Os oito pontos de comunicação de dados e sincronização entre os processadores, são [1] :

- $I_g$  e  $V_g$  - são, respectivamente, as injeções de correntes e tensões nas barras que contém geradores representados dinamicamente. A dimensão destes vetores independe do número de variáveis de estado usadas para representar as máquinas síncronas e seus controladores.
- $V_f$  - que representa as tensões nas barras de fronteira dos subsistemas da rede elétrica.
- $G$  - testes de convergência global.
- $RR^0$ ,  $RR^{k+1}$  e  $DYD^k$  - escalares resultantes dos subprodutos internos, realizados em cada processador, e cujo somatório para todos os processadores, formará o resultado global dos produtos internos do algoritmo GC.
- $d_f^k$  - subvetor do vetor de direções globais, contendo somente os valores associados às barras de fronteira dos subsistemas da rede elétrica e com dimensão igual a de  $V_f$ .

Os vetores  $I_g$ ,  $V_g$ ,  $V_f$  e  $d_f$ , têm dimensões, em geral, da ordem de 10% da dimensão total do sistema de equações da rede elétrica [1]. Enquanto isso,  $G$ ,  $RR^0$ ,  $RR^k$  e  $DYD^k$  são escalares e o custo para a comunicação dos mesmos independe da dimensão do sistema a ser resolvido.

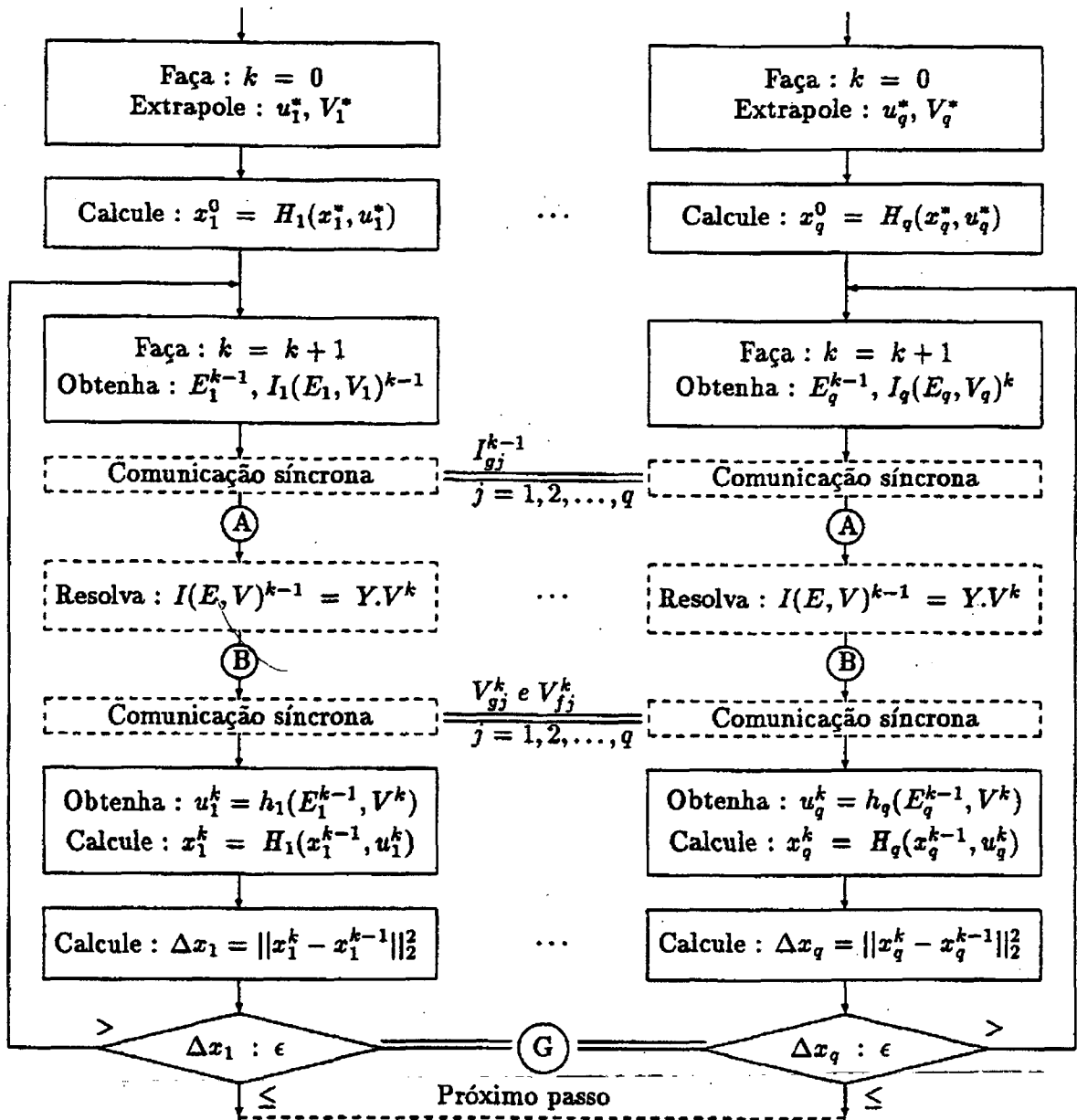


Figura 7.1 - Fluxograma do algoritmo AEI com GC paralelo - 1 passo de integração

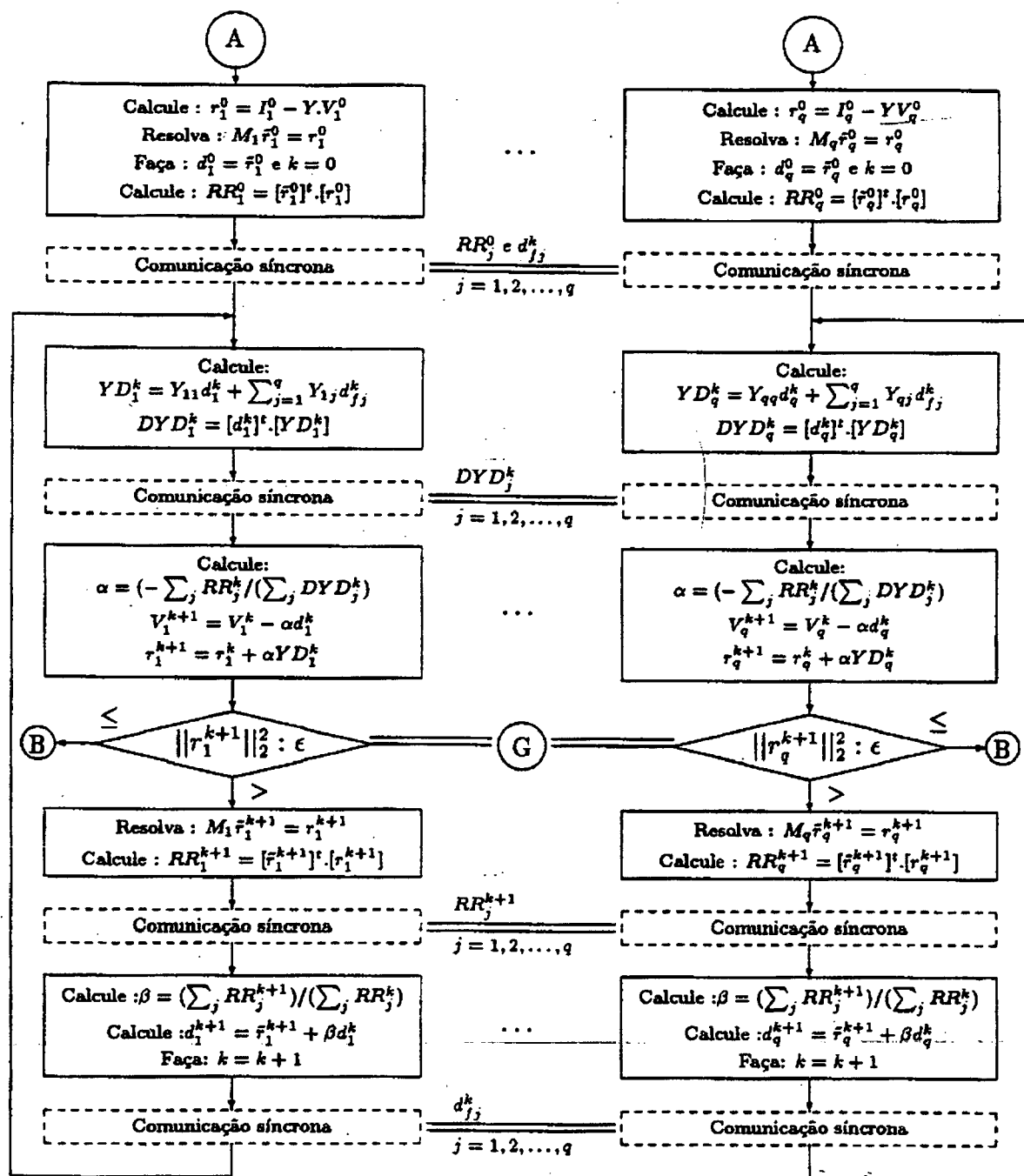


Figura 7.2 - Fluxograma do algoritmo GC para resolver  $I(E^k, V^k) = Y.V^{k+1}$

O algoritmo paralelo AEI com GC, descrito acima, foi adaptado para a implementação na plataforma computacional IBM/SP. As principais modificações feitas são :

- alteração da modelagem das máquinas síncronas para a inclusão dos efeitos sub-transitórios
- implementação de diversos métodos de pré-condicionamento
- implementação dos métodos COGC e BiCGSTAB para a resolução das equações da rede elétrica
- adaptação das formas de trocas de mensagem para a utilização das subrotinas da MPL
- implementação das subrotinas operacionais de gerenciamento de tarefas
- implementação da subrotina de sincronização de tarefas
- substituição das subrotinas de leitura de tempo de CPU pelas correspondentes subrotinas do IBM /SP.

Além destas modificações, o algoritmo AEI com GC teve que sofrer alterações na forma de entrada de dados, originando duas versões deste algoritmo, uma para processamento em modo interativo e outra para processamento em modo *batch*, modalidades de execução já citadas no Capítulo 6.

### 7.2.2. O Algoritmo Sequencial AEI com GC

Este algoritmo, também adaptado de [1], é uma versão sequencial do algoritmo apresentado no subitem anterior, onde foram feitas as mesmas adaptações de modelagem e na entrada de dados que no algoritmo anterior.

### 7.2.3. O Algoritmo Sequencial AEI Convencional

Este algoritmo, também adaptado de [1], sofreu as mesmas adaptações citadas para o algoritmo sequencial AEI com GC, e difere dos dois anteriores por solucionar as equações lineares da rede elétrica com um método direto baseado na fatoração LU e substituição direta e inversa.



### 7.3. Comentários

Neste capítulo foram apresentados os três algoritmos implementados neste trabalho, adaptados daqueles desenvolvido por Decker em [1]. O algoritmo AEI com GC utiliza o método GCP na solução do sistema de equações associado a rede elétrica, onde foi destacado o potencial de paralelização deste algoritmo, tendo sido implementadas as suas versões sequencial e paralela. A velocidade de convergência do método GCP, associado ao pré-condicionador bloco-diagonal, utilizado no algoritmo AEI com GC, é função da decomposição realizada sobre o sistema elétrico em estudo. O algoritmo sequencial AEI convencional também foi apresentado, diferindo dos anteriores por utilizar um método direto baseado na fatoração LU e substituição direta e inversa na solução da rede elétrica.

# CAPÍTULO 08

## AVALIAÇÃO DE DESEMPENHO

### 8.1. Introdução

Neste capítulo são apresentados os resultados obtidos com a aplicação dos algoritmos implementados na simulação da dinâmica de diversos sistemas de energia elétrica de pequeno e muito grande porte. Os processamentos foram realizados nos computadores paralelos IBM/SP1 do LNCC/CNPq e IBM/SP2 do NPD/UFSC, utilizando-se, configurações de 2 e 4 processadores, nos casos de processamento paralelo. Foram realizados também processamentos sequenciais para fins de comparação de desempenho e analisar as perspectivas de paralelização do problema da SDSEE, considerando-se as dimensões do problema e a evolução das máquinas paralelas.

### 8.2. Objetivos

A avaliação de desempenho deste capítulo tem como objetivos principais comparar os algoritmos implementados, comparar as plataformas paralelas utilizadas neste trabalho e em outros trabalhos citados na literatura e estabelecer uma relação entre o perfil de evolução das máquinas paralelas com a perspectiva de obtenção de *speed-ups*.

A nível de algoritmos implementados os principais objetivos a serem alcançados são :

- Comparar o algoritmo AEI convencional sequencial com o algoritmo AEI com GC paralelo, observando-se a viabilidade da aplicação do método GC na SDSEE quando comparado com o tradicional método direto baseado na fatoração LU e substituição direta e inversa, amplamente disseminada nos programas de uso comercial.
- Comparar a evolução dos tempos de CPU dos três algoritmos implementados com o crescimento da dimensão dos sistemas de energia elétrica analisados, onde visa-se observar, principalmente, o comportamento do método direto e do método GC para sistemas de muito grande porte.

- Comparar os dois algoritmos ( sequencial e paralelo ) que utilizam o método GC, onde será observada a paralelização alcançada pelo algoritmo AEI com GC.

Quanto as plataformas paralelas, visa-se comparar aspectos como capacidades efetivas de processamento e de comunicação, *speed-ups* obtidos e granularidade, para a obtenção de um perfil mais detalhado das máquinas paralelas utilizadas que possibilite o estabelecimento de perspectivas de obtenção de *speed-ups*.

### 8.3. Sistemas Elétricos Utilizados nas Simulações

Os parâmetros dos sistemas de energia elétrica utilizados nas simulações são mostrados na Tabela 8.1 a seguir, onde LTs indica Linhas de Transmissão, transformadores e outros elementos série. O número de geradores indica as máquinas síncronas representadas dinamicamente nas simulações efetuadas.

Tabela 8.1 - Parâmetros dos SEE utilizados nas simulações.

Sistema	No. de Geradores	No. de barras	No. de LTs
IEEE 118	20	118	179
SSUD 616	88	616	995
SSUD 2000	198	1916	2788
FIC 4000	396	3832	5579
FIC 8000	792	7664	11166

O sistema IEEE 118 é um sistema padrão de testes do IEEE. O sistema de 616 barras representa uma configuração do sistema interligado das regiões Sul e Sudeste do Brasil, para o ano de 1987, com algumas subáreas representadas por equivalentes estáticos de rede. O sistema de 1916 barras representa uma configuração atual do sistema interligado das regiões Sul e Sudeste do Brasil. Os sistemas FIC 4000 e FIC 8000 são fictícios e foram criados artificialmente mediante vários sistemas SSUD 2000 interligados entre si. O fluxo de carga destes dois últimos sistemas foram obtidos usando o programa NEWDES desenvolvido no Laboratório de Sistemas de Potência da UFSC ( Labspot ). Para os demais SEE foi utilizado o programa ANAREDE do Cepel. As máquinas síncronas foram representadas por modelos de 6a. ordem, exceto para o sistema de 118 barras em que foram usados modelos de 4a. ordem. No caso do sistema SSUD 616 foram utilizadas duas versões : uma onde as máquinas são representadas por modelos de 4a.

ordem, utilizada para comparações com resultados obtidos em [1], e a outra com as máquinas representadas por modelos de 6a. ordem . Os sistemas de excitação são representados pelo modelo IEEE tipo 1. As cargas são modeladas como impedâncias constantes.

## 8.4. Parâmetros das Simulações

As simulações realizadas utilizaram os mesmos parâmetros empregados por Decker em [1], para que fosse possível a comparação entre os resultados obtidos nos dois trabalhos, com exceção do *mismatch* para as variáveis diferenciais que foi alterado de  $10^{-5}$  ( muito rígido ) para  $10^{-4}$ . Os parâmetros utilizados são :

- perturbações : curto circuito sólido trifásico eliminado pelo chaveamento de uma linha de transmissão
- tempo de permanência do curto circuito : 0.10 segundos
- período total de simulação : 5.0 segundos, sendo 0.40 segundos de regime permanente, 0.10 segundos de curto circuito e 4.50 segundos de pós-falta
- passo de integração : 0.01 segundos
- tolerância de convergência do Esquema Alternado :
  - para o GC :  $10^{-2}$
  - para a rede elétrica :  $10^{-3}$
  - para as variáveis diferenciais ( AEI ) :  $10^{-4}$
- tipos de processamento :
  - sequencial
  - paralelo ( 2 e 4 processadores )
- níveis de otimização na compilação :
  - sem otimização ( s/o )
  - nível de otimização -O3

## 8.5. Resultados Numéricos

O desempenho dos algoritmos paralelos implementados neste trabalho foi avaliado em termos de número de iterações e tempo de CPU de cada algoritmo, todos utilizando o Esquema Alternado Entrelaçado Implícito. Além disso, foram calculados os *speed-ups* e eficiências do

algoritmo paralelo AEI com GC em relação aos algoritmos sequenciais, grandezas que serão definidas a seguir.

### 8.5.1. O Referencial Adotado

O algoritmo AEI convencional sequencial, que utiliza um método direto baseado na fatoração LU e substituição direta e inversa na resolução das equações da rede elétrica, por ser o esquema mais empregado em programas comerciais de simulação da dinâmica, foi escolhido como referencial para a comparação de desempenho entre os algoritmos implementados. A Tabela 8.2 apresenta o total e a média do número de iterações para o esquema AEI e para a solução da rede elétrica, obtidos com o algoritmo AEI convencional sequencial, considerando-se a simulação de 5 segundos em um único processador do computador IBM/SP1. A média do número de iterações é obtida de acordo com a expressão ( 8.1 ) :

$$\text{Média do número de iterações} = \frac{\text{número total de iterações}}{\text{número de passos de integração}} \quad ( 8.1 )$$

Tabela 8.2 - Número de iterações para o algoritmo AEI convencional sequencial (a)

SEE	Número de Iterações			
	Total		Média	
	AEI	Rede	AEI	Rede
IEEE 118	1010	538	2.02	1.08
SSUD 616	1118	654	2.24	1.31
SSUD 616s	1230	735	2.46	1.47
SSUD 2000	1048	557	2.10	1.11
FIC 4000	1028	539	2.06	1.08
FIC 8000	1040	551	2.08	1.10

Os resultados obtidos por Decker [1] nas máquinas NCP I e iPSC/860, com o algoritmo AEI convencional sequencial e um *mismatch* de  $10^{-5}$  para as variáveis diferenciais do Esquema Alternado Entrelaçado Implícito, são mostrados na Tabela 8.3, onde o número de iterações foi bem maior que os apresentados na Tabela 8.2. As diferenças entre os resultados obtidos nas simulações realizadas neste trabalho e os resultados obtidos por Decker, em termos da evolução

dos vários ângulos dos rotores das máquinas síncronas no tempo, foram desprezíveis, apesar da maior precisão exigida em [1] para as variáveis diferenciais.

Tabela 8.3 - Número de iterações para o algoritmo AEI convencional sequencial (b)

SEE	Número de Iterações			
	Total		Média	
	AEI	Rede	AEI	Rede
IEEE 118	1540	1048	2.96	2.02
SSUD 616	1501	1017	2.89	1.96

A Tabela 8.4 apresenta os tempos de CPU, em segundos, para a solução global e para a resolução das equações associadas a rede elétrica, obtidos nos computadores IBM/SP1 e IBM/SP2, utilizando-se as opções -O3 e sem otimização na compilação. O computador paralelo IBM/SP2 do NPD/UFSC possui 4 processadores, sendo 2 finos e 2 largos. A diferença entre os processadores finos e largos está na memória *cache* e na capacidade do barramento de dados, superiores nos nós largos, conforme citado no Capítulo 6. Os resultados da Tabela 8.4 referentes ao IBM/SP2 foram obtidos em um dos nós finos. No nó largo do IBM/SP2 os tempos de CPU foram, aproximadamente, 10% inferiores com relação aos tempos obtidos no nó fino.

Tabela 8.4 - Tempos de CPU para o algoritmo AEI convencional sequencial (a)

Computador	SEE	TEMPOS DE CPU ( s. )				Tipo de Processamento
		Otimização -O3		Sem Otimização		
		TOTAL	REDE	TOTAL	REDE	
IBM/SP1	IEEE 118	1.05	0.45	2.22	1.06	Batch Exclusivo
	SSUD 616	5.85	2.93	13.36	7.35	
	SSUD 616s	6.45	2.90	15.10	8.45	
	SSUD 2000	18.23	6.97	41.40	19.32	
	FIC 4000	40.20	13.47	96.00	37.57	
	FIC 8000	97.42	28.39	257.68	79.44	
IBM/SP2 Nó Fino	IEEE 118	0.65	0.22	1.16	0.53	Batch
	SSUD 616	3.63	1.53	6.99	3.92	
	SSUD 616s	4.50	2.03	9.03	4.43	
	SSUD 2000	10.75	3.95	22.10	10.28	
	FIC 4000	24.48	7.93	51.77	19.98	
	FIC 8000	62.38	16.88	138.20	40.83	
	FIC 8000	61.49	15.01	140.94	42.97	

A Tabela 8.5 mostra os tempos de CPU ( em segundos ) obtidos por Decker, em [1], nas máquinas paralelas NCP I e iPSC/860, sendo que nesta última máquina foram utilizadas as opções de compilação 0 ( sem qualquer otimização de código ) e 4 ( nível máximo de otimização de código do compilador ), com diferenças desprezíveis entre os tempos de CPU para as duas opções.

Tabela 8.5 - Tempos de CPU para o algoritmo AEI convencional sequencial (b)

Computador	SEE	TEMPOS DE CPU (s.)	
		TOTAL	REDE
NCP I	IEEE 118	109.80	50.40
	SSUD 616	436.00	161.00
iPSC/860	IEEE 118	13.50	5.10
	SSUD 616	62.30	26.60

O melhor desempenho dos computadores IBM/SP1 e SP2 sobre aqueles utilizados por Decker, em [1], fica evidenciado comparando-se os resultados mostrados nas Tabelas 8.4 e 8.5. O IBM/SP2, por exemplo, foi aproximadamente 120 vezes mais rápido que o NCP I para o sistema de 616 barras, e aproximadamente 17 vezes mais rápido que o iPS/860.

### 8.5.2. Eficiência e *Speed-up*

Neste trabalho são utilizadas as mesmas definições de *speed-ups* e eficiências usadas em [1] e rerepresentadas a seguir :

$$S_1 = \frac{\text{tempo de CPU para o algoritmo AEI convencional sequencial}}{\text{tempo de CPU para o algoritmo AEI com GC paralelo}} \quad (8.2)$$

$$S_2 = \frac{\text{tempo de CPU para o algoritmo AEI com GC sequencial}}{\text{tempo de CPU para o algoritmo AEI com GC paralelo}} \quad (8.3)$$

$$E_1 = \frac{S_1}{q} \quad (8.4)$$

$$E_2 = \frac{S_2}{q} \quad (8.5)$$

onde o tempo de CPU está em segundos e  $q$  é o número de processadores utilizados no processamento paralelo. Conforme [1],  $S_1$  e  $E_1$  mostram, respectivamente, o *speed-up* e a eficiência do algoritmo AEI com GC paralelo sobre o algoritmo AEI convencional sequencial, ou seja, estas grandezas refletem o ganho efetivo no emprego de computadores paralelos. De mesma forma,  $S_2$  e  $E_2$  representam uma medida da paralelização alcançada pelo algoritmo AEI que utiliza o método GC.

### 8.5.3. Algoritmos que Utilizam o Método GC

Neste item são apresentados e discutidos os resultados obtidos com a simulação da dinâmica dos sistemas da Tabela 8.1, utilizando-se os algoritmos AEI com GC sequencial e paralelo nos computadores IBM/SP1 e SP2. Conforme mostrado no Capítulo 5, o método GCP será utilizado na solução da equação da rede elétrica, associado a um pré-condicionador bloco-diagonal, o que torna necessária a decomposição da rede elétrica em  $q$  subsistemas, onde  $q$  é o número de processadores.

#### 8.5.3.1. A Decomposição dos Sistemas

A rede elétrica deve ser decomposta em  $q$  subredes tal que a matriz admitância nodal assumira a forma NBDF, em virtude da utilização de um pré-condicionador do tipo fatoração LU incompleta, como o PCB. A decomposição do sistema de equações diferenciais é feita de forma independente da decomposição das equações da rede elétrica, sendo função do número de máquinas síncronas atribuídas a cada processador. A decomposição do sistema de 118 barras em 2 subredes foi feita pelo programa DECOSEE, apresentado no Capítulo 4, e em 4 subredes foi utilizada a mesma decomposição apresentada em [1]. Para o sistema de 616 barras, apesar deste programa ter apresentado várias decomposições similares as utilizadas em [1], foram utilizadas as mesmas de [1] a fim de viabilizar estudos comparativos entre os dois trabalhos. O sistema SSUD 2000 foi totalmente decomposto pelo programa DECOSEE. Os sistemas FIC 4000 e FIC 8000, oriundos da interligação de vários sistemas SSUD 2000, foram decompostos segundo os mesmos parâmetros utilizados para o SSUD 2000. Foram utilizadas apenas decomposições em 2 e 4 subsistemas, em virtude do número máximo de processadores disponíveis para utilização na modalidade batch exclusivo no IBM/SP1, ter sido fixado em 4 processadores. A utilização de um número maior de processadores exige autorização especial que não foi solicitada por razões de



tempo para a finalização do presente trabalho. No IBM/SP2, o número total de processadores é também de quatro. A Tabela 8.6 apresenta o número de máquinas síncronas por processador e o número de barras associadas a cada subrede.

Tabela 8.6 - Parâmetros das decomposições

Decomposição	SEE	Número de Subredes	Número de Barras por Subrede				
			Subrede 01	Subrede 02	Subrede 03	Subrede 04	
Máquinas Síncronas	IEEE 118	02	10	10	-	-	
		04	05	05	05	05	
	SSUD 616	02	44	44	-	-	
		04	22	22	22	22	
	SSUD 2000	02	99	99	-	-	
		04	51	49	49	49	
	FIC 4000	02	198	198	-	-	
		04	99	99	99	99	
	FIC 8000	02	396	396	-	-	
		04	198	198	198	198	
	Barras	IEEE 118	02	58	60	-	-
			04	30	28	29	31
SSUD 616		02	300	316	-	-	
		04	163	137	156	160	
SSUD 2000		02	901	1015	-	-	
		04	492	462	465	497	
FIC 4000		02	1916	1916	-	-	
		04	901	1015	901	1015	
FIC 8000		02	3832	3832	-	-	
		04	1916	1916	1916	1916	

### 8.5.3.2. Resultados das Simulações

Este subitem apresenta os resultados obtidos com processamentos dos algoritmos AEI com GC sequencial e paralelo nas plataformas IBM/SP. Foram utilizados 2 e 4 processadores para as simulações paralelas e para os processamentos sequenciais um único processador. Todos processamentos no IBM/SP1 foram executados na modalidade *batch* exclusivo enquanto no SP2 foi utilizado o modo *batch* não exclusivo. A Tabela 8.7 mostra os resultados obtidos em termos de tempo de CPU, *speed-ups* e eficiências nos processamentos paralelos e sequenciais no IBM/SP1. No IBM/SP2 foi processado somente o algoritmo AEI com GC paralelo e os resultados obtidos são mostrados na Tabela 8.8. Estas Tabelas têm como simbologia : OC indica opção de compilação, SEQ indica sequencial, S/O indica sem otimização e “ Comun. “ indica

comunicação. O tempo de CPU paralelo das Tabelas 8.7 e 8.8 está subdividido em tempo total e tempo de comunicação entre os processadores ( coluna “ Comun. “ ).

Tabela 8.7 - Resultados dos algoritmos AEI com GC no IBM/SP1

SEE	Número de Subredes	OC	ETAPA	TEMPO DE CPU			Speed-Ups e Eficiências			
				SEQ	PARALELO		S <sub>1</sub>	E <sub>1</sub> %	S <sub>2</sub>	E <sub>2</sub> %
					Total	Comun.				
IEEE 118	02	S/O	REDE	2.24	1.98	-	0.54	27.00	1.13	56.50
			TOTAL	3.54	2.81	0.87	0.79	39.50	1.26	63.00
		O3	REDE	0.78	1.38	-	0.33	16.50	0.57	28.50
			TOTAL	1.50	1.87	1.05	0.56	28.00	0.80	40.00
	04	S/O	REDE	2.28	3.78	-	0.28	7.00	0.60	15.00
			TOTAL	3.52	4.91	3.20	0.45	11.25	0.72	18.00
		O3	REDE	0.68	3.50	-	0.13	3.25	0.19	4.75
			TOTAL	1.43	4.23	3.38	0.25	6.25	0.34	8.50
SSUD 616	02	S/O	REDE	30.19	19.35	-	0.38	19.00	1.56	78.00
			TOTAL	36.99	23.46	4.92	0.57	28.50	1.58	79.00
		O3	REDE	9.82	6.85	-	0.43	21.50	1.43	71.50
			TOTAL	13.85	9.14	2.69	0.64	32.00	1.52	76.00
	04	S/O	REDE	37.91	15.50	-	0.47	11.75	2.45	61.25
			TOTAL	44.66	21.24	8.81	0.63	15.75	2.10	52.50
		O3	REDE	9.74	9.16	-	0.32	8.00	1.06	26.50
			TOTAL	14.22	11.06	6.23	0.53	13.25	1.29	32.25
SSUD 2000	02	S/O	REDE	81.85	48.67	-	0.40	20.00	1.68	84.00
			TOTAL	106.84	66.83	10.14	0.62	31.00	1.60	80.00
		O3	REDE	23.21	16.70	-	0.42	21.00	1.39	69.50
			TOTAL	34.65	23.90	6.13	0.76	38.00	1.45	72.50
	04	S/O	REDE	231.28	64.30	-	0.30	7.50	3.60	90.00
			TOTAL	254.23	78.01	14.29	0.53	13.25	3.26	81.50
		O3	REDE	56.80	24.97	-	0.28	7.00	2.27	56.75
			TOTAL	67.62	30.22	13.88	0.60	15.00	2.24	56.00
FIC 4000	02	S/O	REDE	64.35	44.79	-	0.84	42.00	1.44	72.00
			TOTAL	136.20	99.41	3.89	0.97	48.50	1.37	68.50
		O3	REDE	20.00	15.25	-	0.88	44.00	1.31	65.50
			TOTAL	48.65	34.03	2.34	1.18	59.00	1.43	71.50
	04	S/O	REDE	170.56	73.29	-	0.51	12.75	2.33	58.25
			TOTAL	235.27	117.93	28.63	0.81	20.25	1.99	49.75
		O3	REDE	45.55	29.28	-	0.46	11.50	1.56	39.00
			TOTAL	72.40	43.57	18.83	0.92	23.00	1.66	41.50
FIC 8000	02	S/O	REDE	133.50	91.38	-	0.87	43.50	1.46	73.00
			TOTAL	370.27	278.36	5.31	0.93	46.50	1.33	66.50
		O3	REDE	40.86	30.78	-	0.92	46.00	1.33	66.50
			TOTAL	122.08	88.68	3.83	1.10	55.00	1.38	69.00
	04	S/O	REDE	133.19	50.69	-	1.57	39.25	2.63	65.75
			TOTAL	340.58	200.82	8.40	1.28	32.00	1.70	42.50
		O3	REDE	40.70	19.19	-	1.48	37.00	2.12	53.00
			TOTAL	113.94	63.18	6.15	1.54	38.50	1.80	45.00

Tabela 8.8 - Resultados do algoritmo AEI com GC paralelo no IBM/SP2

SEE	Número de Subredes	OC	ETAPA	TEMPO DE CPU PARALELO	
				Total	Comun.
IEEE 118	02	S/O	REDE	1.26	-
			TOTAL	1.78	0.75
		O3	REDE	0.69	-
			TOTAL	1.27	0.64
	04	S/O	REDE	3.38	-
			TOTAL	4.05	3.18
O3		REDE	2.96	-	
		TOTAL	3.58	2.98	
SSUD 616	02	S/O	REDE	11.90	-
			TOTAL	14.49	3.25
		O3	REDE	5.42	-
			TOTAL	7.14	2.39
	04	S/O	REDE	9.44	-
			TOTAL	11.47	4.11
O3		REDE	6.72	-	
		TOTAL	7.97	4.97	
SSUD 2000	02	S/O	REDE	28.81	-
			TOTAL	39.13	5.33
		O3	REDE	12.78	-
			TOTAL	17.67	6.21
	04	S/O	REDE	38.75	-
			TOTAL	46.53	10.38
O3		REDE	19.56	-	
		TOTAL	23.39	10.28	
FIC 4000	02	S/O	REDE	23.96	-
			TOTAL	54.48	1.45
		O3	REDE	9.27	-
			TOTAL	23.24	1.55
	04	S/O	REDE	42.25	-
			TOTAL	66.84	15.30
O3		REDE	18.55	-	
		TOTAL	29.31	10.19	
FIC 8000	02	S/O	REDE	49.61	-
			TOTAL	153.66	1.91
		O3	REDE	20.21	-
			TOTAL	62.62	1.90
	04	S/O	REDE	26.56	-
			TOTAL	110.74	3.31
O3		REDE	11.17	-	
		TOTAL	44.24	2.79	

## 8.6. Avaliação dos Resultados

A utilização da opção de otimização de compilação -O3 resulta em acentuados ganhos nos tempos de CPU dos algoritmos, chegando-se a ganhos de aproximadamente 3 vezes para o sistema FIC 8000. A IBM recomenda que somente após a comparação dos resultados dos algoritmos compilados com a opção -O3 e sem otimização que esta opção de otimização seja utilizada.

A Figura 8.1 mostra a evolução dos tempos de CPU, no IBM/SP1, com o crescimento da dimensão do SEE para os diversos algoritmos testados. No caso dos algoritmos paralelos é apresentado o melhor tempo de CPU obtido entre as duas situações processadas ( em 2 e 4 nós ). No quadro de legendas é usada a simbologia “conv.” para convencional, “seq.” para sequencial e “par.” para paralelo, simbologia utilizada em todos os gráficos deste capítulo.

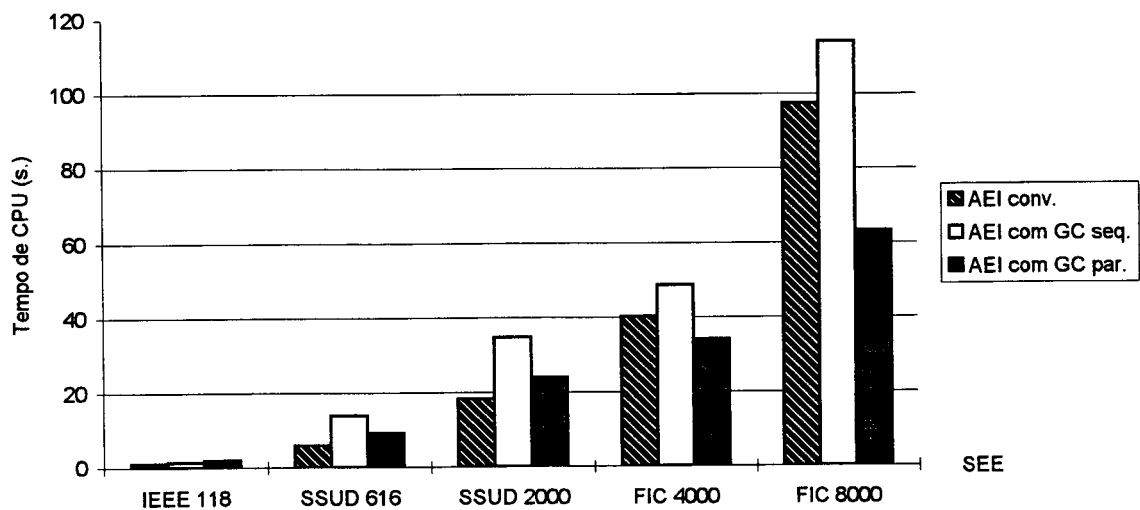


Figura 8.1 - Tempos de CPU dos algoritmos utilizados x Dimensão do SEE

A Figura 8.1 mostra que a partir de SEE com dimensão superior a 2000 barras o algoritmo AEI com GC paralelo é superior ao algoritmo AEI convencional sequencial, apresentando tempos de CPU mais reduzidos. Caso seja desprezada a comunicação o algoritmo paralelo com GC é sempre superior ao convencional sequencial, independentemente da dimensão do SEE. No caso dos dois algoritmos que utilizam o GC, a Figura 8.1 mostra que o paralelo é sempre superior ao sequencial, evidenciando as características de paralelização do algoritmo.

### 8.6.1. A Comunicação entre os Processadores no IBM/SP1 e SP2

A Tabela 8.7 mostra que os tempos de comunicação do algoritmo AEI com GC paralelo estão muito elevados, o que pode ser melhor constatado na Figura 8.2 a seguir, que mostra as parcelas percentuais do tempo de CPU no IBM/SP1 referentes a solução da rede elétrica e a comunicação entre os processadores, com a utilização da opção -O3.

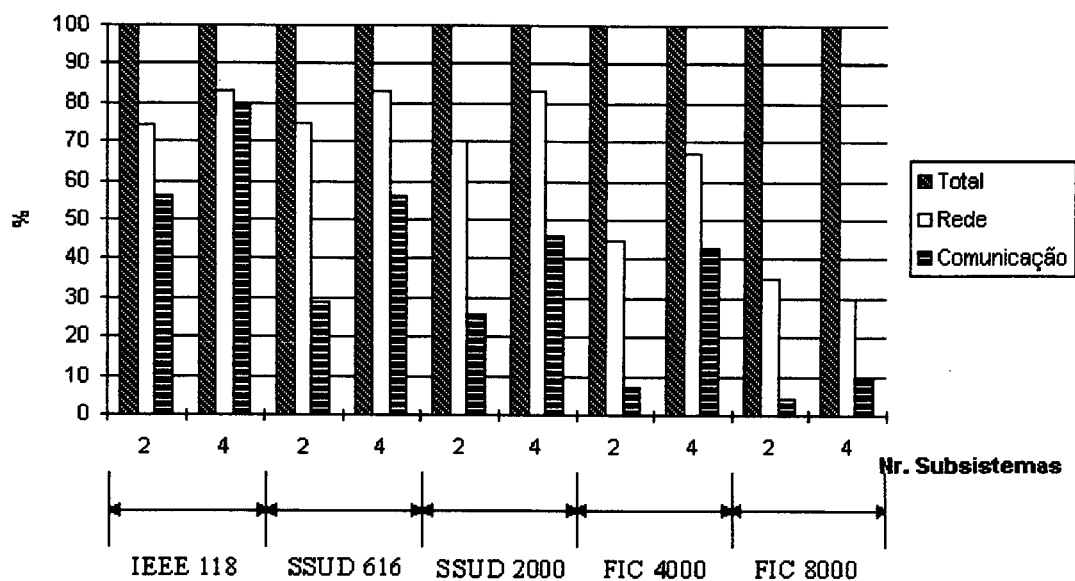


Figura 8.2 - Parcelas percentuais do tempo de CPU no IBM/SP1 .

Os tempos de comunicação entre os processadores no IBM/SP1, conforme mostra a Figura 8.2, estão na faixa de 26-56 % ( para 2 nós ) e 46-80 % ( para 4 nós ) do tempo total de CPU para sistemas até 2000 barras. Nos SEE maiores, como o FIC 4000 e FIC 8000, estas faixas foram reduzidas para 4-7 % ( para 2 nós ) e 10-43 % ( para 4 nós ), mostrando haver “diluição” do tempo de comunicação com o crescimento da dimensão do sistema. Para o IBM/SP2 estas faixas são aproximadamente as mesmas. Decker, em [1], para 2 e 4 processadores do iPSC/860, obteve tempos de comunicação em torno de 5 % e 25 %, respectivamente, do tempo total de CPU, para o sistema SSUD 616. Esperava-se que os percentuais dos tempos de comunicação, com relação ao tempo total de CPU, para as plataformas IBM/SP, fossem inferiores aos percentuais mostrados pelas máquinas utilizadas em [1], em virtude da maior capacidade de comunicação das máquinas IBM/SP, conforme mostra a Tabela 6.1 do Capítulo 6. Os altos

percentuais dos tempos de comunicação, apresentados pelo IBM/SP1, indicam que a taxa esperada de 40 MB/s. de transferência de dados pelo dispositivo HPS desta máquina deve estar muito inferior a este valor. Isto motivou um estudo a respeito da capacidade de comunicação efetiva do HPS do IBM/SP1. Este estudo mostrou :

- O suporte do LNCC/CNPq informou que o resultado da medição da taxa efetiva do dispositivo HPS do IBM/SP1, com *softwares* apropriados, foi de 8 Mbytes/s. A IBM informou que esta baixa taxa se deve a incompatibilidade do HPS do IBM/SP1 ao *software* do IBM/SP2 implementado quando do *upgrade* de *software* realizado no 2o. semestre de 1995.

- O levantamento da curva da capacidade de comunicação do IBM/SP1 e SP/2 mostrou que a real taxa de transferência de dados é da ordem de 4 MB/s. no IBM/SP1 e de 18 MB/s. no IBM/SP2. Os resultados são mostrados na Figura 8.3 e foram obtidos pelo envio de vetores de números reais com dimensões variadas, de um processador para outro, e pela medição do tempo de transferência dos dados. Um vetor de 100 elementos, por exemplo, ao ser enviado 25000 vezes envolve a transferência de  $100 \times 25000 \times 4 / 10^6 = 10$  MB de dados, onde 4 indica os 4 bytes de cada número inteiro. A taxa de transferência de dados em MB/s. foi encontrada mediante a divisão do volume de dados transferidos, em megabytes, pelo tempo gasto na transferência dos dados, em segundos, pelas subrotinas *Mp\_send* ( para enviar ) ou *MP\_recv* ( para receber ). Estas subrotinas pertencem a biblioteca MPL da IBM.

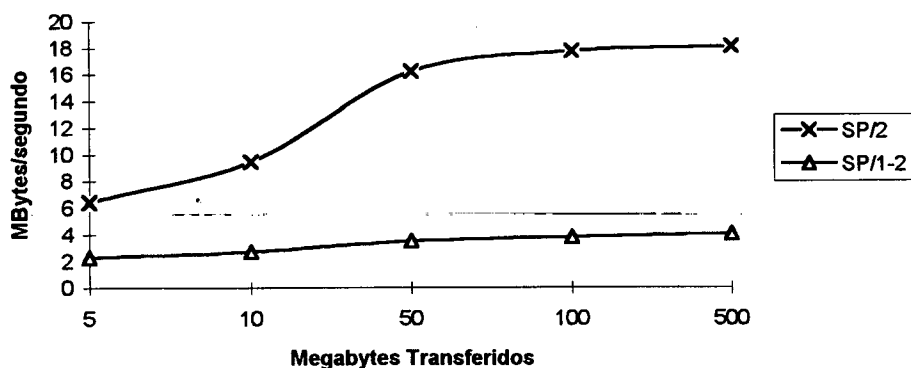


Figura 8.3 - Capacidade de Comunicação do IBM/SP1 e IBM/SP2

Esta redução da capacidade de comunicação do IBM/SP1 afeta a granularidade da máquina que fica aumentada em 10 vezes, passando de 3.125 Mflops/MB/s. para 31.25 Mflops/MB/s., valor muito superior ao apresentado pelas máquinas iPSC/860 e NCP I utilizadas

por Decker em [1], respectivamente de 14.3 e 0.60 Mflops/MB/s. nominais. Isto dificulta a obtenção de bons *speed-ups* nas plataformas IBM/SP.

### 8.6.2. Speed-ups

As Figuras 8.4 e 8.5 mostram os *speed-ups*  $S_1$  e  $S_2$ , respectivamente, obtidos na plataforma IBM/SP1, e nas máquinas paralelas utilizadas por Decker em [1], com a utilização de 4 processadores. A Figura 8.4 apresenta os *speed-ups*  $S_1$ , que mostra a relação entre o tempo de CPU do algoritmo AEI convencional sequencial e o tempo de CPU do algoritmo AEI com GC paralelo, conforme a equação ( 8.2 ).

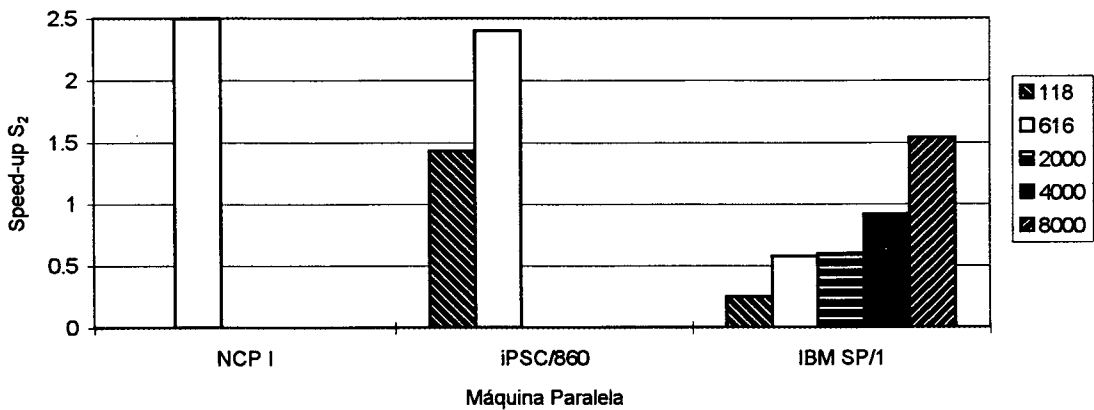


Figura 8.4 - *Speed-ups*  $S_1$  obtidos em várias máquinas paralelas

A Figura 8.5 apresenta os *speed-ups*  $S_2$ , que mostra a relação entre o tempo de CPU do algoritmo AEI com GC sequencial e o tempo de CPU do algoritmo AEI com GC paralelo, conforme a equação ( 8.3 ).

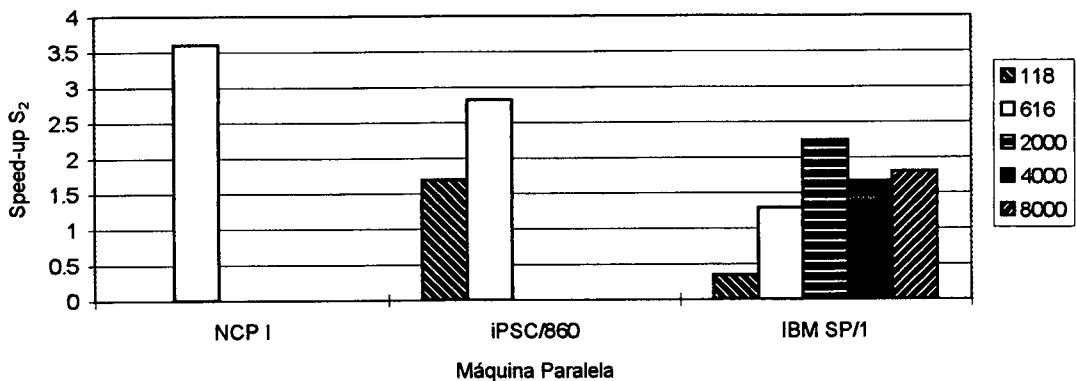


Figura 8.5 - *Speed-ups*  $S_2$  obtidos em várias máquinas paralelas

Na Figura 8.4 ficaram evidenciados os baixos valores de *speed-ups*  $S_1$  obtidos na plataforma IBM/SP1, onde para o SSUD 616, por exemplo, foi obtido um *speed-up*  $S_1$  de 0.53 contra valores próximos a 2.5 no iPSC/860 e NCP I. Os *speed-ups*  $S_2$ , mostrados na Figura 8.5, também apresentaram valores inferiores no IBM/SP1. Estes baixos valores de *speed-ups* são um reflexo da alta granularidade efetiva desta máquina, a qual foi afetada pelo baixo rendimento do seu dispositivo HPS.

### 8.6.3. A Granularidade

Com o objetivo de proporcionar uma melhor visualização da granularidade das quatro máquinas comparadas neste trabalho, efetuou-se uma análise das suas capacidades de processamento e comunicação tendo-se como referência a máquina NCP I utilizada em [1].

A Tabela 8.9 mostra, para as 4 máquinas comparadas neste trabalho, a capacidade de processamento ( CP ), as capacidades de comunicação teórica ( CC ) e efetiva, mostrada na curva da Figura 8.3, e os tempos de CPU, obtidos nestas máquinas, para uma simulação de 5 segundos com o algoritmo AEI convencional sequencial, no sistema SSUD 616.

Tabela 8.9 - Dados para cálculo do desempenho das máquinas relativo ao NCP I

Máquina Paralela	NCP I	iPSC/860	IBM/SP1	IBM/SP2	IBM/SP2 *
CP ( Mflops )	1.5	40	125	266	266
CC ( MB/s. )	2.5	2.8	40	40	150
CC medida ( MB/s. )	-	-	4	18	-
Tempo de CPU ( s. )	436	62.3	5.8	3.6	-

\* a ser lançada em 1996

Partindo-se dos dados da Tabela 8.9 e definindo-se as seguintes variáveis em pu., com relação ao NCP I, para uma máquina  $m$  qualquer ( onde T. indica tempo ) :

$$\text{CP teórica da máquina } m \text{ ( em pu. )} = \text{CP da máquina } m / \text{CP do NCP I} \quad (8.6)$$

$$\text{CP efetiva da máquina } m \text{ ( em pu. )} = \text{T. de CPU da máquina } m / \text{T. de CPU do NCP I} \quad (8.7)$$

$$\text{CC teórica da máquina } m \text{ ( em pu. )} = \text{CC da máquina } m / \text{CC do NCP I} \quad (8.8)$$

$$\text{CC efetiva da máquina } m \text{ ( em pu. )} = \text{CC medida da máquina } m / \text{CC do NCP I} \quad (8.9)$$



pode-se apresentar a Tabela 8.10, com os dados de desempenho das 4 máquinas comparadas neste trabalho, relativos ao NCP I.

Tabela 8.10 - Desempenho das Máquinas Relativo ao NCP I ( em pu. )

Máquina	Capacidade de Processamento		Capacidade de Comunicação	
	Teórica	Efetiva	Teórica	Efetiva
NCP I	1	1	1	1
iPSC/860	27	7	1.12	-
IBM/SP 1	83	75	16	1.6
IBM/SP 2	177	121	16	7.2
IBM/ SP 2 *	177	-	60	-

\* a ser lançada em 1996

A Tabela 8.10 mostra em destaque que a capacidade de processamento efetiva do iPSC/860 atingiu apenas 26 % da capacidade teórica de pico. No IBM/SP1 e SP/2 estes valores ficaram em 90 % e 70 %, respectivamente. Isto justifica ainda mais os altos valores de *speed-ups* obtidos no iPSC/860 em [1], pois a granularidade efetiva desta máquina é acentuadamente menor do que a nominal. Os baixos valores de *speed-ups* obtidos na plataforma IBM/SP1, quando comparados aos obtidos por Decker [1] no iPSC/860 e NCP I, ficam então justificados, visto que:

- a granularidade da máquina IBM/SP1 está aproximadamente 10 vezes mais alta do que a esperada, ou seja, de 3.125 → 31.25 Mflops/MB/s.
- a granularidade efetiva da máquina iPSC/860 é bastante inferior a teórica, em virtude da máquina apresentar uma capacidade de processamento efetiva muito inferior a teórica.
- a granularidade da máquina NCP I é muito fina ( 0.60 Mflops/MB/s ).

#### 8.6.4. A Evolução das Máquinas

As Tabelas 8.9 e 8.10 mostram que as máquinas paralelas atuais tiveram um acentuado crescimento de sua capacidade de processamento, não acompanhado, porém, pelo crescimento da sua capacidade de comunicação. No caso do IBM/SP2, por exemplo, houve um aumento teórico de 177 vezes na capacidade de processamento, com relação ao NCP I, e de apenas 16 vezes ( teórico ) na sua capacidade de comunicação. Este novo perfil das atuais plataformas paralelas

certamente dificulta a obtenção de bons índices de *speed-ups* quando da implementação de algoritmos paralelos. Os altos índices de *speed-ups* apresentados em vários artigos técnicos da literatura [4,19,67,78] foram obtidos em máquinas de granularidade fina e já obsoletas em termos de capacidade de processamento, quando comparadas com as máquinas atuais. O lançamento do novo IBM/SP2, a ser comercializado em 1996, com o novo dispositivo HPS de capacidade de comunicação de 150 MB/s., reduzirá em 3.75 vezes a granularidade desta máquina, trazendo a perspectiva de aumento dos *speed-ups* obtidos nesta máquina.

### 8.6.5. O Paralelismo como Alternativa

A Tabela 8.11 mostra os tempos de CPU para o algoritmo AEI convencional sequencial para os diversos sistemas de energia elétrica analisados neste trabalho, obtidos em um dos nós finos do IBM/SP2. Na coluna Tempo ( pu. ) é apresentada a relação Tempo de CPU / Tempo de dinâmica simulado, onde pode-se observar quantas vezes o tempo de processamento é maior que os cinco segundos de simulação realizados, ou seja, o quanto o tempo de CPU obtido pelo algoritmo AEI convencional sequencial está afastado do tempo real. Note-se que a modelagem usada ainda é relativamente simplificada pois considera somente o sistema de excitação entre os controladores das unidades geradoras.

Tabela 8.11 - Tempos de CPU para o AEI convencional x dimensão dos SEE

SEE	Tempo CPU ( s. )	Tempo ( pu. )
IEEE 118	0.65	0.13
SSUD 616	3.63	0.73
SSUD 2000	10.75	2.15
FIC 4000	24.48	4.90
FIC 8000	62.38	12.48

Na Tabela 8.11 estão salientados os SEE em que não foi possível a obtenção do processamento da simulação da dinâmica em tempo inferior ao tempo real. Esta Tabela mostra que mesmo com a modelagem simplificada utilizada neste trabalho, a simulação convencional, que utiliza um método direto baseado na fatoração LU e substituição direta e inversa, da dinâmica de um SEE de aproximadamente 8000 barras, por exemplo, necessita de um tempo de CPU aproximadamente 12.5 vezes maior que o período da dinâmica simulado. Esta relação tende a se agravar com a sofisticação da modelagem e o aumento da dimensão do sistema. Isto mostra que

para sistemas de energia elétrica de porte médio e elevado o desenvolvimento de algoritmos paralelos para a simulação da dinâmica de SEE ainda continua sendo um tema a ser investigado. Notadamente a utilização de máquinas de grande capacidade de processamento por nó e granularidade relativamente fina, como o Intel Paragon e o novo IBM/SP2, contribuirão para a obtenção de *speed-ups* razoáveis.

## 8.7. Comentários

Este capítulo apresentou os resultados obtidos nas plataformas IBM/SP com a aplicação dos algoritmos implementados na simulação da dinâmica de vários sistemas de energia elétrica. A partir destes resultados foi possível avaliar o desempenho dos algoritmos e das máquinas paralelas IBM/SP1 e SP/2, utilizadas neste trabalho, e de outras plataformas paralelas citadas na literatura. Pode-se também avaliar o estágio de evolução das máquinas paralelas atuais e projetar perspectivas quanto à utilização destas máquinas.

Observou-se que a utilização da opção de otimização de compilação -O3 nas plataformas IBM/SP resulta em acentuados ganhos nos tempos de CPU dos algoritmos. Contudo esta opção deve ser usada com cuidado em relação a exatidão dos resultados.

Quanto aos algoritmos implementados constatou-se que o algoritmo AEI com GC paralelo apresentou menores tempos de CPU que o algoritmo AEI convencional sequencial para sistemas com dimensão superior a 1916 barras. Ao desconsiderar-se o tempo de comunicação entre os processadores isto ocorreu para sistemas de qualquer dimensão. A aplicação do método GCP na SDSEE, proposta por Decker em [1], foi estendida a sistemas de muito grande porte, onde a solução da equação da rede elétrica foi obtida sem problemas.

Em termos de *speed-ups*, constatou-se que os valores obtidos no IBM/SP1 são muito pequenos, quando comparados aos obtidos por Decker [1] no iPSC/860 e NCP I. Isto se deve a grande diferença de granularidade do IBM/SP1 e a destas máquinas. Uma estimativa da granularidade efetiva do iPSC/860 mostrou que ela é acentuadamente menor que a teórica, o que facilitou a obtenção dos altos *speed-ups*. Já o NCP I tem uma granularidade muito fina. A plataforma IBM/SP1, face a problemas apresentados pelo seu dispositivo HPS, está com uma granularidade efetiva aproximadamente 10 vezes maior que a teórica, o que inviabilizou a obtenção de bons *speed-ups* nesta máquina. As máquinas paralelas mais citadas na literatura ( Sequent Symmetry, nCUBE 2 ... ) são de granularidade fina. Consequentemente, os *speed-ups*

apontados nos artigos técnicos são altos. Observa-se, porém, que estas máquinas já estão totalmente obsoletas quanto a capacidade de processamento, quando comparadas com as máquinas paralelas atuais.

Quanto a evolução das máquinas paralelas, observou-se que as máquinas paralelas atuais tiveram um acentuado crescimento de sua capacidade de processamento, não acompanhado, porém, pelo crescimento da sua capacidade de comunicação. Ou seja, a granularidade das máquinas atuais é maior, dificultando a obtenção de altos índices de *speed-ups* nestas máquinas.

Constatou-se neste capítulo que a utilização do algoritmo AEI convencional sequencial em sistemas de energia elétrica de muito grande porte resultou em tempos de CPU muito superiores ao período de tempo simulado. Observa-se então que, dificilmente, o algoritmo AEI convencional sequencial, amplamente utilizado nos programas comerciais de simulação da dinâmica, conseguirá obter tempos de processamento inferiores ao tempo de dinâmica simulado ( real ) em SEE de médio e grande porte. Aponta-se como alternativa viável para este problema a utilização de algoritmos paralelos de simulação da dinâmica de SEE tais como o algoritmo paralelo AEI com GC utilizado neste trabalho.

# CAPÍTULO 9

## CONCLUSÕES

O objetivo deste trabalho foi realizar um estudo da simulação da dinâmica de sistemas de energia elétrica nas plataformas computacionais paralelas IBM/SP, abordando a paralelização do algoritmo de solução, a aplicação de métodos tipo GC, técnicas de pré-condicionamento, a decomposição de SEE e a avaliação em SEE de grande porte. Os algoritmos que exploram a paralelização no espaço, desenvolvidos por Decker em [1], serviram como referencial para este trabalho, e após algumas modificações foram implementados na plataforma computacional de alto desempenho IBM/SP1 do Laboratório Nacional de Computação Científica - LNCC e IBM/SP2 da Universidade Federal de Santa Catarina - UFSC . Esta implementação dos algoritmos no sistema paralelo IBM/SP , é uma das contribuições deste trabalho.

Os algoritmos desenvolvidos em [1] tinham como principal novidade a utilização de métodos iterativos do tipo Gradiente Conjugado Pré-Condicionado, empregados na solução das equações da rede elétrica, quando da aplicação do Esquema Alternado Entrelaçado Implícito, ou na solução do sistema linear jacobiano, quando da utilização do Esquema Simultâneo . Um levantamento do estado da arte dos vários métodos do tipo Gradiente Conjugado, com especial destaque a sua aplicação a problemas de sistemas de energia elétrica, foi apresentado no Capítulo 3 deste trabalho, tornando-se mais uma contribuição do mesmo. Foram apresentados vários métodos da família GC para casos de sistemas lineares simétricos e assimétricos. Como este trabalho utilizou o Esquema AEI, foi indicada a utilização do método GCP tradicional, para sistemas simétricos, por suas facilidades de implementação e características de convergência.

O pré-condicionador bloco-diagonal adotado em [1], composto por blocos diagonais retirados da matriz de admitância de barras  $Y$  do sistema estudado, exige que este sistema seja decomposto em subredes fracamente acopladas entre si [52] para que a convergência do processo iterativo do GC seja facilitada. Cada subrede, em geral, é atribuída a um processador, garantindo paralelismo perfeito na solução do sistema auxiliar de pré-condicionamento do método GCP. Visando eliminar a dependência de decomposição da rede em sistemas menores, foi realizado neste trabalho um estudo de diversos pré-condicionadores, com a intenção de substituir o PCB,

utilizado em [1], por um outro de características paralelizáveis e rápida convergência. No Capítulo 3, foram selecionados para uma segunda etapa de avaliação, pelo potencial de paralelização de seus algoritmos e facilidade de implementação computacional, o próprio pré-condicionador bloco-diagonal (PCB), o pré-condicionador por série truncada de Maclaurin (PCS) e o pré-condicionador diagonal (PCD). Nesta segunda etapa de avaliação, realizada no Capítulo 5, o PCD foi abandonado pela sua baixa efetividade na redução do número de condição da matriz  $Y$ . A análise do comportamento de convergência do PCB e do PCS mostrou acentuadas perdas de ortogonalidade dos vetores de resíduos, ao longo do processo iterativo, no caso do pré-condicionador por série truncada de Maclaurin, que também mostrou ter a sua convergência dependente dos autovalores do sistema estudado. Devido a perda da ortogonalidade dos vetores de resíduos, mostrada pelo PCS, e de forma mais atenuada pelo PCB foram implementados os algoritmos dos métodos Bi-CGSTAB e COGC, citados na literatura como específicos para problemas deste tipo, porém, o problema persistiu e alguns sistemas não convergiram. O pré-condicionador bloco-diagonal, já utilizado em [1], mostrou ser o melhor dos pré-condicionadores estudados, e foi indicado para pré-condicionar o método GCP neste trabalho.

Os estudos realizados neste trabalho sobre pré-condicionadores e ortogonalidade dos vetores de resíduo ao longo das iterações do método GCP mostraram resultados preliminares, sugerindo-se a continuação dos trabalhos nesta linha para a obtenção de conclusões mais definitivas. Este estudo dos diversos tipos de pré-condicionadores, apresentado nos Capítulos 3 e 5, foi outra contribuição deste trabalho.

A escolha do PCB como um dos pré-condicionadores indicados para a segunda etapa de avaliação citada acima, motivou um estudo sobre técnicas de decomposição de SEE em sistemas de menor porte. Foi desenvolvido e implementado o algoritmo DECOSEE, que baseado no critério do *ranking* nodal proposto em [52,54], realiza a decomposição de sistemas de energia elétrica. A metodologia deste algoritmo propõe a escolha de um número inicial de nós sementes superior ao número desejado de subredes. Estes subsistemas parciais, obtidos após o processo proposto em [52,54], são agregados numa etapa final, com participação do usuário, para a formação do número desejado de subredes finais. Esta etapa de agregação tem como vantagens a menor dependência de uma eficiente escolha dos nós sementes, no início do processo, para a obtenção de subsistemas fracamente acoplados entre si e o controle da dimensão dos subsistemas, o que permite a obtenção de subsistemas, sempre que possível, com dimensões similares. O desenvolvimento de critérios automáticos de escolha dos nós sementes, já adotado em [52,54], e

de agregação dos subsistemas, é necessário, principalmente no caso de sistemas de muito grande porte. O algoritmo DECOSEE é uma outra contribuição do presente trabalho.

Foram utilizados três algoritmos de SDSEE neste trabalho, onde todos utilizam o Esquema EAI. São eles :

- **AEI com GC paralelo** - onde as equações da rede elétrica são resolvidas por um método do tipo GC pré-condicionado pelo PCB. A rede elétrica é decomposta em subredes.
- **AEI com GC sequencial** - versão sequencial do algoritmo anterior.
- **AEI convencional** - programa sequencial onde as equações da rede elétrica são solucionadas por um método direto baseado na fatoração LU e substituição direta e inversa. Não há decomposição da rede em subredes.

Estes algoritmos foram implementados nas plataformas paralelas IBM/SP, tendo sido realizadas simulações de sistemas de energia elétrica de até 7664 barras ( FIC 8000 ). Estes SEE de grande porte foram montados artificialmente a partir da união de vários sistemas de 1916 barras ( SSUD 2000) . Esta montagem foi mais uma contribuição deste trabalho, assim como a aplicação do algoritmo AEI a estes SEE de grande porte. Nas simulações realizadas com os algoritmos AEI com GC sequencial e paralelo ficou mais uma vez constatada a viabilidade da utilização do método GC na SDSEE.

Os *speed-ups* obtidos na plataforma IBM/SP1 do LNCC foram baixos, motivado pelo baixo desempenho apresentado pelo seu dispositivo HPS, o que elevou a granularidade desta máquina em aproximadamente 10 vezes quando comparada com os parâmetros nominais. A enorme redução da capacidade de comunicação do HPS do IBM/SP1 se deve a problemas de adequação do novo *software* do IBM/SP2, instalado no *upgrade*-feito no segundo semestre de 1995, à arquitetura do SP1. Os altos *speed-ups* obtidos por Decker [1] nos computadores iPSC/860 e NCP I se devem ao fato de que o NCP I possui uma granularidade muito fina, enquanto o iPSC/860 opera com uma capacidade de processamento efetiva muito inferior a teórica, reduzindo, com isso, a sua granularidade.

Foi obtida uma acentuada redução nos tempos de processamento das simulações realizadas com a utilização das plataformas IBM/SP, comparativamente aos tempos obtidos por Decker [1] nos computadores iPSC/860 e NCP I. Além disso foram realizadas simulações em tempos inferiores ao tempo real para sistemas com até 616 barras.

As máquinas paralelas atuais tiveram um acentuado crescimento de sua capacidade de processamento, não acompanhado, porém, pelo crescimento da sua capacidade de comunicação. No caso do IBM/SP2, máquina paralela lançada em 1994, por exemplo, houve um aumento teórico de 177 vezes na sua capacidade de processamento, com relação ao NCP I, máquina paralela de 1989, e de apenas 16 vezes ( teórico ) na sua capacidade de comunicação. Isso dificulta a obtenção de bons índices de *speed-ups* para problemas de granularidade relativamente fina como a SDSEE.

Na literatura pertinente a SDSEE em computadores paralelos, tem sido encontrados valores relativamente altos de *speed-ups*, porém, obtidos em máquinas de granularidade fina e com uma capacidade de processamento por nó muito inferior a apresentada pelas plataformas IBM/SP. Este perfil da evolução das máquinas paralelas associado a perspectiva de obtenção de *speed-ups* é outra contribuição do presente trabalho.

Este trabalho mostrou que, mesmo com a modelagem simplificada utilizada, a simulação convencional da dinâmica, baseada na fatoração LU e substituição direta e inversa, de um SEE de aproximadamente 8000 barras, por exemplo, necessita de um tempo de CPU quase 13 vezes maior que o próprio período de tempo simulado. Uma alternativa viável, para a obtenção de tempos de processamento inferiores ao tempo de dinâmica simulado nestas situações, é a paralelização e desenvolvimento de novos algoritmos de simulação da dinâmica de SEE. Pode-se citar também o desenvolvimento de novas técnicas de paralelização e a utilização de máquinas paralelas de alto desempenho e granularidade relativamente fina, como é o caso do IBM/SP2 com um novo HPS a ser comercializada ainda no presente ano.

Os estudos efetuados no IBM/SP2 foram comprometidos pelo desbalanceamento de carga entre os processadores, face as suas diferentes arquiteturas. Os dados obtidos nesta máquina foram válidos para efeitos comparativos com o IBM/SP1, sendo esta comparação ( SP1 x SP2 ) outra contribuição do presente trabalho.

## 9.1. Sugestões para Trabalhos Futuros

As principais sugestões para trabalhos futuros são :

- melhorar a modelagem utilizada, principalmente nas malhas de controle das máquinas síncronas, com a inclusão de reguladores de velocidade, controle de frequência e outros.
- explorar a paralelização no tempo e no espaço ( PET ).



- implementar os algoritmos desenvolvidos por Decker em [1] que utilizam o Esquema Simultâneo - SI ( paralelização no tempo ) no IBM/SP1 do LNCC ( futuro IBM/SP2 ).
- investigar outros métodos do tipo GC
- investigar métodos de decomposição de SEE.
- quanto a pré-condicionadores :
  - ⇒ estudar as causas da perda de ortogonalidade dos vetores de resíduo quando da aplicação do método GCP em sistemas lineares simétricos complexos.
  - ⇒ desenvolver melhores pré-condicionadores para o método GCP
  - ⇒ estudar a utilização da fatoração LU incompleta nas submatrizes do PCB

## BIBLIOGRAFIA

- [1] DECKER, I.C. , *Algoritmos para a Simulação da Dinâmica de Sistemas de Energia Elétrica em Computadores Paralelos* , Tese de Doutorado, Universidade Federal do Rio de Janeiro UFRJ, 1993.
- [2] QUINN, M.J., *Designing Efficient Algorithms for Parallel Computers* , McGraw-Hill International Editions, 2nd. Edition, 1988.
- [3] SPONG, M. I. et alli, “ Transient Stability Simulations by Waveform Relaxation Method “ , *IEEE Transactions on Power Systems*, Vol. PWRS-2, PP. 943-952, November 1987.
- [4] WU, J.Q. and BOSE, A., “ Parallel Solution of Power System Problems “ , ICSPT-94, Beijing, China.
- [5] IEEE Committee Report, “ Parallel Processing in Power Systems Computation “ , *IEEE Transactions on Power Systems*, Vol. 7, PP. 629-638, May 1992.
- [6] ZORPETTE, G., “ The Power of Parallelism “ , *IEEE Spectrum*, PP. 28-33, September, 1992.
- [7] FALCÃO, D.M., “ Parallel and Distributed Processing in Power System Simulation and Control “ , IV SEBOPE, Foz do Iguaçu, Brazil, PP. 01-08, May, 1994.
- [8] STTOT, B., “ Power System Dynamic Response Calculations “ , *Proceedings of the IEEE*, Vol. 67, PP. 219-241, February, 1979.
- [9] DECKER, I.C., FALCÃO, D.M. and KASZKUREWICZ, E. , “ An Efficient Parallel Method for Transient Stability Analysis “ , *Proceedings of the 10th Power Systems Computation Conference*, Graz, Austria, PP. 509-516, August 1990.
- [10] DECKER, I.C., FALCÃO, D.M. and KASZKUREWICZ, E., “ Parallel Implementation of a Power System Dynamic Simulation Methodology Using the Conjugate Gradient Method “ , *IEEE Transactions on Power Systems*, Vol. 7, No. 1, PP. 458-465, February 1992.

- [11] DECKER, I.C., FALCÃO, D.M. and KASZKUREWICZ, E., “ Conjugate Gradient Methods for Power System Dynamic Simulation in Parallel Computers “ , IEEE PES *Summer Meeting*, 1994.
- [12] PADILHA, A. , *Cálculo da Estabilidade Transitória em Sistemas de Energia Elétrica Utilizando o Esquema Simultâneo Implícito* , Tese de Mestrado, Unicamp, 1986.
- [13] SILVA, A. S. e COSTA, A.S. , *Aspectos Dinâmicos do Controle de Sistemas de Energia Elétrica*, Apostila, Universidade Federal de Santa Catarina - UFSC, 1994.
- [14] DECKER, I.C. , *Estabilidade I*, Apostila , Universidade Federal de Santa Catarina - UFSC, 1994.
- [15] DOMMEL, H.W. and SATO, N., “Fast Transiente Stability Solutions“, IEEE *Transactions on Power Apparatus and Systems*, Vol. PAS-91, PP. 1643-1650, July/August, 1972.
- [16] ARNOLD, C.P. , *Solutions of the Multi-Machine Power System Stability Problem* , Tese de Doutorado, Universidade de Manchester, 1976.
- [17] FRANÇA, A.L.M., “ Simulação da Dinâmica de Curto Prazo em Sistemas de Energia Elétrica : Comportamento do Esquema Alternado Implícito “ , V CBA / I CLA, Campina Grande, Brasil, 1984.
- [18] COSTA, V.F., *Análise de Estabilidade Transitória em Sistemas de Energia Elétrica* , Tese de Mestrado, FEC / Unicamp, São Paulo, 1981.
- [19] CHAI, J.S. and BOSE, A. , “ Bottlenecks in Parallel Algorithms for Power System Stability Analysis “ , IEEE *Transactions on Power Systems*, Vol. 8, No. 1, PP. 09-15, February 1993.
- [20] ALVARADO, F.L. , “ Parallel Solution of Transient Problems by Trapezoidal Integration “ , IEEE *Transactions on Power Apparatus and Systems*, Vol. PAS-98, PP. 1080-1090, May/June, 1979.
- [21] LaSCALA, M. et alli , “ A Highly Parallel Method for Transient Stability Analysis “ , IEEE *Transactions on Power Systems*, Vol. 5, PP. 1439-1446, November 1990.

- [22] LaSCALA, M. et alli , “ A Gauss-Jacobi-Block-Newton Method for Parallel Transient Stability Analysis “, *IEEE Transactions on Power Systems*, Vol. 5, PP. 1168-1177, November, 1990.
- [23] ILIC-SPONG, M. et alli , “ Transient Stability Simulation by Waveform Relaxation Method”, *IEEE Transactions on Power Systems*, Vol. PWRS-2, PP. 943-952, November 1987.
- [24] CROW, M. L. and ILIC, M. , “ The Parallel Implementation of the Waveform Relaxation Method for Transient Stability Simulations “, *IEEE Transactions on Power Systems*, Vol. 5, PP. 922-931, August 1990.
- [25] CROW, M. L. , *Waveform Relaxation Methods for the Simulation of Systems of Differential/Algebraic Equations with Applications to Electric Power Systems* , PhD Thesis, University of Illinois at Urbam-Cahampaign, 1990.
- [26] CROW, M. L. and ILIC, M. , “ Parallel Algorithms for Large Scale Power System Dynamic Simulation “, in *Proceedings of 1988 American Control Conference*, PP. 1795-1798, 1988.
- [27] CROW, M. L. and ILIC, M. , “ The Parallel Implementation of the Waveform Relaxation Method for the Simulation of Structure-Preserved Power Systems “, in Paper IEEE CH2868-8/90, PP. 1285-1288, 1990.
- [28] ORTEGA, J.M. , *Introduction to Parallel and Vector Solution of Linear Systems* , New York, Plenum Press, 1988
- [29] HESTENES, M.R. and STIEFEL E., “ Methods of Conjugate Gradients for Solving Linear Systems “, *J. Res. Natn. Bur. Standards* 49, PP. 409-436, 1952.
- [30] COUTINHO, A. G. et alli , “ Comparison of Lanczos and Conjugate Gradients for the Element-by-Element Solution of Finite Element Equations on the IBM 3090 Vector Computer “, *Computers and Structures*, Vol. 39, No. 1/2, PP. 47-55, 1991.

- [31] VAN der VORST, H. A. , “ BI-CGSTAB - A Fast and Smoothly Converging Variant of BI-CG for the Solution of Nonsymmetric Linear Systems “, *SIAM J. Sci. Stat. Comput.*, Vol. 13, No. 2, PP. 631-644, 1992.
- [32] DONGARRA, J. J. et alli , “ Solving Linear Systems on Vector and Shared Memory Computers “, *SIAM - Society for Industrial and Applied Mathematics*, 1991.
- [33] FLETCHER, R. , “ Conjugate Gradient Methods for Indefinite Systems “ , *Lecture Notes Math.* 506, Springer-Verlag, Berlin, PP. 73-89, 1976.
- [34] SONNEVELD, P. , “ A Fast Lanczos-Type Solver for Nonsymmetric Linear Systems “ , *SIAM J. Sci. Statist. Comput.*, Vol. 10, PP. 36-52, 1989.
- [35] VAN der VORST, H. A. and MELISSEN, J.B.M. , “ A Petrov-Galerkin Type Method for Solving  $Ax = b$ , Where A is Symmetric Complex “, *IEEE Transactions on Magnetics*, Vol. 26, No. 2, PP. 706-708, 1990.
- [36] GALIANA, F. D. et alli , “ On the Application of a Preconditioned Conjugate Gradient Algorithm to Power Network Analysis “, *IEEE Transactionson Power Systems*, Vol. 9, No. 2, PP. 629-636, May, 1994.
- [37] CHEN, C. T. , *Linear System Theory and Design* , HRW Series in Electrical and Computer Engineering, 1984.
- [38] ALVARADO, F.L. et alli , “ Sparse Matrix Inverse Factors “, *IEEE Trans. on Power Systems*, Vol. 5, No. 2, PP. 466-473, 1990
- [39] CHAI, J.S. et alli , “ Parallel Newton Type Methods for Power Systems Stability Using Local and Shared Memory Multiprocessors “, *IEEE Trans. on Power Systems*, Vol. 6, PP. 1539-1545, Nov. 1991
- [40] NOBLE, B. and DANIEL, J.W. , *Álgebra Linear Aplicada* , Prentice/Hall do Brasil, 1977
- [41] SEMLYEN, A. , “ Fundamental Concepts of a Krylov Subspace Power Flow Methodology “, *IEEE/PES Summer Meeting*, Portland, Or., July, 1995.

- [42] HENNENBERG, G. et alli , “ An Accelerated Newton-Raphson Method Associated With the ICCG Algorithm “, *IEEE Trans. om Magnetics*, Vol. 26, No. 2, March, 1990.
- [43] DECKER, I.C., FALCÃO, D.M. and KASZKUREWICZ, E., “ Conjugate Gradient Methods for Power System Dynamic Simulation on Parallel Computers “ , *IEEE/PES Summer Meeting*, Portland, OR., July, 1995.
- [44] DAG, H. and ALVARADO, F. , “ Direct Methods Versus GMRES and PCG for Power Flow Problems “ , *Proceedings of NAPS*, PP. 274-278, October 5-9, Washington, DC, 1993.
- [45] ALVARADO, F. et alli , “ Block-Bordered Diagonalization and Parallel Iterative Solvers “, *Proceedings of Colorado Mountain Conference on Iterative Methods*, April 5-9, Breckeridge, CG, 1994.
- [46] JENNINGS, A. , “ Influence of the Eingevalue Spectrum on the Convergence Rate of the Conjugate Gradient Method “, *J. Inst. Maths. Applics.*, Vol. 20, PP. 61-72, 1977.
- [47] DECKER, I. C. , *Técnicas Computacionais Aplicadas a Sistemas de Potência*, Apostila, Universidade Federal de Santa Catarina - UFSC, 1994.
- [48] MICCHELLI, C. et alli , “ Polynomial Preconditioners for Conjugate Gradient Calculations “, *SIAM J. Sci. Stat. Comput.*, Vol. 20, No. 2, PP. 362-376, April, 1983.
- [49] TYLAVSKY, D. and GOLALAKRISHNAN, B. , “ Precedence Relationship Performance of an Indirect Matrix Solver “, *IEEE Transactions on Power Systems*, Vol. 5, No. 2, PP. 412-418, May, 1990.
- [50] COUTINHO, A.G. et alli , “Two Level Preconditioners for Hierarchical Finite Elements Equations and their Applications in Solid Mechanics “, *Comunicação Interna*, COPPE/UFRJ, 1991
- [51] UNIVERSIDADE FEDERAL DE ITAJUBÁ , *Métodos Computacionais em Sistemas Elétricos*, Apostila, CESE/1985
- [52] VALE, M.H. , *Decomposição de Redes Elétricas para Processamento Paralelo* , Tese de Doutorado, Universidade Federal do Rio de Janeiro - UFRJ, Junho, 1995.

- [53] AMANO, M. et alli , “ An Improved Block-Parallel Newton Method Via Epsilon Decompositions for Load-Flow Calculations “, *IEEE/ PES Summer Meeting* , July, 1995.
- [54] VALE, M.H. , “ Electrical Power Network Decomposition for Parallel Computations “, *Proceedings of the IEEE Symposium on Circuits and Systems*, San Diego, CA, PP. 2761-2764, May, 1992.
- [55] INTERNATIONAL BURROUGHS MACHINES-IBM, *Manual IBM Loadlever : User's Guide*, 1993.
- [56] IEEE Commitee Report , “ Computer Representation of Excitation Systems “, *IEEE Transactions on Power Apparatus ans Systems*, Vol. PAS-87, No. 6, PP. 1460-1464, June, 1968.
- [57] IEEE Commitee Report , “Excitation System Models for Power System Stability Studies“, *IEEE Transactions on Power Apparatus ans Systems*, Vol. PAS-100, No. 2, PP. 494-509, February, 1981.
- [58] DOMMEL, H.W. and SATO, N. , “ Fast Transient Stability Solutions “, *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-91, PP.1643-1650, July/August, 1972.
- [59] HWANG, K. and BRIGGS, F., *Computer Archicteture and Parallel Processing* , McGraw Hill, New York, 1984.
- [60] PADILHA, A. and MORELATO, A. , “ A W Matrix Methodology for Solving Sparse Network Equations on Multiprocessors Computers “, *IEEE Transactions on Power Systems*, Vol.7, PP. 1023-1030, August, 1992.
- [61] TAOKA, H. et alli , “ Real-time Digital Simulator for Power System Analysis on a Hypercube Computer “, *IEEE Transactions on Power Systems*, Vol. 7, PP. 1-10, Feb., 1992.
- [62] TYLAVSKY, D. ,“ Parallel-vector Processing Synergy and Systems Dynamic Simulations ”, *Proceedings of the EPRI/NSF Workshop on Application of Advanced Mathematics to Power Systems*, Redwood City, CA, Sep. 1991.

- [63] AYKANAT, C. et alli , “ Iterative Algorithms for Solution for Large Sparse Systems of Linear Equations on Hypercubes “, *IEEE Trans. on Computers*, Vol. 37, PP. 1554-1568, December, 1988.
- [64] ANGELERI, F. et alli , “ Studies of Finite Element Procedures - An Evaluation of Preconditioned Iterative Solvers “, *Computers and Structures*, Vol. 32, No. 3/4, PP. 671-677, 1989.
- [65] MANSUR, W.J. et alli ,” Solution of BEM Systems of Equations Via Iterative Techniques “, *International Journal of Numerical Methods in Engineering*, 1992.
- [66] SHAKIB, F. et alli , “ A Multi-element Group Preconditioned GMRES Algorithm for Nonsymmetric Systems Arising in Finite Element Analysis “, *Computer Methods in Applied Mechanics and Engineering*, Vol. 75, PP. 415-456, North-Holland, 1989.
- [67] DAG, H. and ALVARADO, F.L. , “ Computation-free Preconditioners for the Parallel Solution of Power System Problems “, Presented at the *IEEE/PES Winter Meeting*, January 21-25, Baltimore, MD, 1996
- [68] ALVARADO, F. and DAG, H. , “ Partitioned Incomplete Inverse Preconditioners “, Submitted to *Parallel Computing*, June 1994.
- [69] LaScala, M. et alli , “ Comparison of Algorithm for Transient Stability Simulations on Shared and Distributed Memory Multiprocessors “, Presented at the *IEEE/PES Winter Meeting*, January 21-25, Baltimore, MD, 1996.
- [70] PAI, M.A. and KULKARNI, A.Y. , “ A Simulation Tool for Transient Stability Analysis Suitable for Parallel Computers “, Presented at the *4th IEEE Conference on Control Applications*, Albany, NY, 1995.
- [71] GUPTA, A. and KUMAR, V. , “ Performance and Scalability of Preconditioned Conjugate Gradient Methods on Parallel Computers “, *IEEE Trans. on Parallel and Distributed Systems*, Vol. 6, No. 5, May, 1995.
- [72] INTERNATIONAL BURROUGHS MACHINES-IBM, *IBM AIX Parallel Environment Parallel Programming Reference* , Release 1.0, September, 1993.



- [73] KUZELA, J.M. , *IBM POWERparallel System - SP2 Performance Measurements* , Power Parallel Systems, September, 1994.
- [74] INTERNATIONAL BURROUGHS MACHINES - IBM, *SP2 High Performance Switch Overview* , [www.mhpcc.edu/training/workshop/html/ibmhws/switch.html](http://www.mhpcc.edu/training/workshop/html/ibmhws/switch.html).
- [75] ANDERSON, P.M. and FOUAD, A.A. , *Power System Control and Stability*, The Iowa State University Press, Ames, Iowa, USA, Vol. 1, 1977.
- [76] DECKER, I.C. , *Análise de Estabilidade Transitória em Sistemas de Potência usando o Conceito de Superfície Limite de Energia Potencial*, Tese de Mestrado, Universidade Federal de Santa Catarina - UFSC, 1984.
- [77] LIN, S.L. and VAN NESS, J.E. , “ Parallel Solution of Sparse Algebraic Equations “, *IEEE Trans. on Power Systems*, Vol. 9, No. 2, May, 1994.
- [78] WU, J.Q. et alli, “ Parallel Implementation of Power System Transient Stability Analysis “, *IEEE/PES Summer Meeting*, San Francisco, CA, July, 1994
- [79] ARRILAGA, J. and ARNOLD, C.P. , *Computer Modelling Electrical Power Systems*, John Wiley & Sons, 1983
- [80] CANTO, A.F. e PEREIRA, J.L., “ Solução de Equações de Redes Elétricas Utilizando Processamento Sequencial e Paralelo “, *Anais do 10o. Congresso Brasileiro de Automática (CBA)*, 19-23 setembro, PP. 281-286, Rio de Janeiro, 1994
- [81] LEWIS, T.G. and EL-REWINI, H., *Introduction to Parallel Computers*, Prentice-Hall International Editions, Englewood Cliffs, New Jersey, 1992.
- [82] CAMPAGNOLO, J.M. , *Aplicação de Processamento Paralelo na Avaliação da Estabilidade a Pequenas Perturbações : Algoritmos para o Cálculo Parcial de Autovalores e Autovetores*, Tese de Doutorado, Universidade Federal do Rio de Janeiro, 1994.