

## ALGORITMOS PARA A SIMULAÇÃO DA DINÂMICA DE SISTEMAS DE ENERGIA ELÉTRICA EM COMPUTADORES PARALELOS

I. C. Decker, M.Sc. \*    D. M. Falcão, PhD.    E. Kaszkurewicz, D.Sc.

COPPE-EE/Núcleo de Computação Paralela  
Universidade Federal do Rio de Janeiro  
21945-970 Rio de Janeiro - RJ  
Tel.:(021) 290-7116 Fax:(021) 290-6626  
E-mail: coe10000@ibm.nce.ufrj.br

### Resumo

Neste artigo discute-se a utilização de computadores paralelos na simulação da dinâmica de sistemas de energia elétrica. Inicialmente faz-se uma breve análise da computação científica de alto desempenho procurando destacar as suas características principais, problemas, tendências de evolução e aplicabilidade a problemas de sistemas de energia elétrica, com ênfase na simulação da dinâmica. Posteriormente, apresenta-se um sumário de três algoritmos paralelos de simulação da dinâmica desenvolvidos pelos autores, todos baseados na utilização de métodos iterativos do tipo gradiente conjugado. Apresentam-se também os principais resultados de aplicações desses algoritmos a um sistema com 616 barras, utilizando-se os computadores paralelos NCP I e iPSC/860. Finalmente apresentam-se algumas conclusões destacando-se a significativa redução nos tempos de processamento do problema da simulação da dinâmica de sistemas de energia elétrica e a necessidade de mais investigações nas áreas de "hardware", "software" básico e desenvolvimento de algoritmos.

### Abstract

This paper discusses the use of parallel computers in the simulation of power systems dynamics. Initially, the high performance scientific computation field is briefly analyzed and its main characteristics, problems, evolutionary trends and applicability to power systems problems, with particular emphasis to dynamic simulation, are highlighted. Next, it is presented a summary of three parallel dynamic simulation methods developed by the authors, all based on the Conjugate Gradients approach. The main results of the application of these methods in the simulation of an actual 616 node power system, using the NCP I and iPSC/860 parallel computers, are also presented. Finally, some conclusions regarding the substantial reduction in processing time for the dynamic simulation problem and the need for further research in hardware, software, and algorithm development are presented and discussed.

## 1 O Processamento de Alto Desempenho

Uma das principais aplicações da computação científica é no problema da simulação de sistemas físicos, descritos matematicamente por milhares ou centenas de milhares de equações algébricas, diferenciais ordinárias e algébricas ou diferenciais parciais. O tipo particular das equações depende do sistema físico e da natureza dos fenômenos que se quer descrever. Adicionalmente, os resultados

\*Em doutoramento na COPPE/UFRJ, sob licença da Universidade Federal de Santa Catarina

dessas simulações devem ser apresentados em uma forma suficientemente sintética de modo a facilitar a análise. Esses requisitos demandam elevada capacidade computacional, podendo atingir o nível de Teraflops em certas áreas do conhecimento [1].

A demanda por capacidades de processamento cada vez maiores e as inerentes limitações relativas a velocidade de transmissão de sinais elétricos através de condutores, levaram os fabricantes de computadores a explorarem o paralelismo [2]. O processamento paralelo é definido por Hwang [3], como uma forma eficiente de processar informações que enfatiza a exploração de eventos concorrentes no processo de computação. Neste contexto, as máquinas apresentam uma grande diversidade de arquiteturas computacionais que, para fins de classificação, ainda podem ser organizadas nas três classes definidas por Hwang [4]: “pipeline computers”, “array processors” e sistemas multiprocessadores.

Em “pipeline computers”, também conhecidos como supercomputadores, instruções sucessivas são executadas em um esquema de sobreposição, operando tal como uma linha de montagem, com cada estágio sendo responsável por uma microtarefa antes de enviar o resultado para o próximo estágio. Estas máquinas têm custo elevado e atrativos para o processamento de grandes vetores, com potencialidades moderadas para acelerar as simulações da dinâmica de SEE, usando-se as metodologias tradicionais, devido ao elevado grau da esparsidade associado à falta de uma estrutura padrão das matrizes envolvidas [5].

Os sistemas multiprocessadores constituem-se de diversos processadores que operam independentemente, podendo compartilhar memória, capacidade de entrada/saída e equipamentos periféricos. São em geral classificados em três tipos: memória compartilhada (“shared memory”), memória distribuída (“distributed memory”) e híbridos. Nos sistemas de memória compartilhada, todos os processadores têm acesso a uma memória comum que representa também o meio de comunicação de dados entre os processadores. O desempenho em aplicações específicas é fortemente dependente das características dessa memória e da intensidade da concorrência de acesso dos processadores, exigidas por um dado algoritmo. Nos sistemas de memória distribuída não existe memória comum e a comunicação de dados, entre os processadores, é feita por “message-passing” através de uma rede de interconexão, cuja configuração define o tipo de arquitetura. O volume, frequência e a velocidade dessas transmissões de dados, têm influência significativa no desempenho global. Outro aspecto importante no desempenho de sistemas multiprocessadores é o balanceamento da carga entre os processadores.

Atualmente estão comercialmente disponíveis máquinas paralelas constituídas de vários processadores de baixo custo e que tem capacidade de processamento comparável, até mesmo, a um supercomputador [4]. No contexto dos Sistemas de Energia Elétrica, estas máquinas podem ser usadas como estações de trabalho para estudos interativos de planejamento ou ainda incorporadas aos Centros de Supervisão e Controle em tempo real (CSC), como elementos de cálculo de alto desempenho visando, por exemplo, a avaliação e correção da segurança dinâmica. Entretanto, a difusão dessas máquinas entre as empresas de Energia Elétrica ainda depende da superação de alguns aspectos:

- *Novos Algoritmos e Programas* : Para uma utilização eficiente dessas máquinas é necessário adaptar os algoritmos usados em máquinas sequenciais ou, até mesmo, desenvolver novos algoritmos e programas. Isso exige estudos cuidadosos de revalidação dos programas de aplicação, em relação a sua escala de aplicabilidade e precisão dos resultados obtidos;
- *Diversidade de Arquiteturas* : Existe um elevado número de arquiteturas paralelas considerando-se as máquinas comercialmente disponíveis e outras que estão em desenvolvimento, dificultando a elaboração de “software” de uso abrangente. Além disso, a evolução tecnológica na área tem sido extremamente rápida levando os equipamentos a obsolescência poucos anos após

o seu desenvolvimento. Este aspecto faz da tomada de decisão pelo investimento em um determinado equipamento, um ato de elevado risco. Isto pode ser atribuído ao fato de que essa tecnologia é relativamente recente e ainda sem padrões bem estabelecidos;

- *A Carência de "Software"* : A maioria das máquinas ainda têm sistemas de "software" básico bastante primitivos, no sentido de que o programador deve se envolver com detalhes do sistema de "hardware". Isso tem implicações na portabilidade dos aplicativos, aspecto fundamental para a larga aplicação industrial de uma metodologia.

Por outro lado, há dois aspectos que tornam inevitável o interesse de pesquisadores e engenheiros de SEE pela computação paralela [6]: a potencialidade de alcançar grande desempenho computacional a custo relativamente baixo e à necessidade de resposta rápida no caso de aplicações em tempo real.

### 1.1 Aplicabilidade em Análise de Redes Elétricas

A análise de redes elétricas está, em geral, associada a estudos de planejamento da expansão, planejamento da operação, ou a operação em tempo real. No planejamento da expansão do sistema são comuns horizontes de estudos de 20 a 40 anos e, necessariamente, para diversos cenários; demandando enorme esforço computacional pela dimensão dos problemas e pelo elevado número de alternativas a serem avaliadas. Por outro lado, na operação em tempo real, vencer o curto espaço de tempo para a tomada e execução de decisões é um dos principais desafios dos instrumentos de análise de redes elétricas. Especificamente, destaca-se que instrumentos, tais como, *Análise de Segurança* e *Fluxo de Carga Ótimo* [7] têm sido desenvolvidos sem a incorporação dos aspectos dinâmicos dos SEE.

Várias metodologias de análise de redes elétricas tem recebido a atenção de pesquisadores no sentido de sua utilização em computadores paralelos [6]. Alguns exemplos são: Fluxo de Carga [8, 9], Estabilidade Transitória [5, 14, 15, 17, 19, 24, 20, 16], Transitórios Eletromagnéticos [10], a determinação de autovalores e autovetores no âmbito da análise dinâmica para pequenas perturbações [11], etc.

A estabilidade transitória tem apresentado a maior demanda, talvez pela expectativa do desenvolvimento de meios que viabilizem a sua incorporação no conjunto dos instrumentos de monitoração e controle da operação em tempo real. A estabilidade transitória é uma propriedade que os SEE têm de manter o sincronismo, entre as máquinas elétricas, após a ocorrência de grandes perturbações. Tradicionalmente, os estudos de estabilidade transitória têm sido feitos a partir da Simulação da Dinâmica dos Sistemas de Energia Elétrica (SDSEE), que consiste em resolver as Equações Diferenciais e Algébricas (EDAs) que o representam. O diagnóstico é obtido da análise de curvas da evolução no tempo de variáveis eletromecânicas [12]. Esta metodologia permite o uso de uma diversidade grande de modelos de representação dos elementos do sistema, porém, o diagnóstico que se obtém é do tipo pontual, no sentido de que ele é válido para um determinado ponto de operação e sequência de perturbações (contingência).

## 2 Simulação da Dinâmica de SEE

Na área específica da Simulação da Dinâmica de Sistemas de Energia Elétrica (SDSEE), são comuns estudos em sistemas com cerca de 3000 equações diferenciais ordinárias não lineares e cerca de 4000 equações algébricas não lineares [6]. Em uma discussão de Kundur e Rogers (vide referência [13]), citam que, na época (1987), já estavam realizando simulações, para fins de estudos de estabilidade transitória, de SEE constituídos de 1000 máquinas síncronas e 10.000 barras (cerca de 10.000 a 20.000

equações diferenciais ordinárias e 10.000 a 20.000 equações algébricas não lineares, dependendo do modelo utilizado) que demandam diversas horas de CPU em computadores do porte de um VAX 8600 ( capacidade aproximada de 1 Mflop). É importante ainda acrescentar que um estudo de estabilidade transitória requer, em adição, um elevado número de simulações relativas a diferentes configurações e pontos de operação do sistema.

As equações que descrevem a dinâmica de um Sistema de Energia Elétrica constituem um conjunto de equações diferenciais ordinárias não lineares, associadas aos rotores das máquinas síncronas e seus controladores, e um conjunto de equações algébricas não lineares, associadas a rede de transmissão, estatores das máquinas síncronas e cargas representadas por modelos estáticos. Essas equações podem ser expressas na forma seguinte [12]:

$$\dot{x} = f(x, z) \quad (1)$$

$$0 = g(x, z) \quad (2)$$

onde  $f$  são funções vetoriais que definem as equações diferenciais;  $g$  são funções vetoriais que definem as equações algébricas;  $x$  é o vetor das variáveis de estado; e  $z$  é o vetor de variáveis das equações algébricas.

Tradicionalmente, os algoritmos sequenciais usam um dentre os dois esquemas básicos de resolução das equações (1) e (2): *Alternado* e *Simultâneo*. O esquema *Alternado* consiste basicamente em transformar as equações diferenciais (1) em equações algébricas a diferenças, através de um método de integração, em geral implícito, e resolvê-las alternadamente com as equações originalmente algébricas (2). No esquema *Simultâneo* as equações (1) são discretizadas por um método de integração numérica implícito e reunidas as equações (2), formando um único sistema de equações que é resolvido, para cada passo de integração, por um método tipo Newton. A fim de diminuir o tempo gasto em fatorações LU, tem sido comum manter os mesmos elementos do Jacobiano por várias iterações e até mesmo passos de integração. Essa versão do método é denominada "Very Dishonest Newton" (VDHN) [19].

### 3 Simulação da Dinâmica de SEE em Ambiente Paralelo

A efetiva utilização de computadores paralelos, e em especial na SDSEE, pressupõe um adequado mapeamento do algoritmo de solução entre os processadores. Isso significa decompor o problema em subtarefas que possam ser processadas simultaneamente. Três linhas básicas de investigação são identificadas na literatura:

- **Paralelização no Espaço** : baseia-se na utilização de esquemas tradicionais de solução do problema com a decomposição do espaço de variáveis em subespaços, sendo o processo de paralelização mais natural [5, 14, 15, 16];
- **Relaxação da Forma de Onda (RFO)** : consiste essencialmente em decompor o sistema de EDAs 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 de 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 [17];
- **Paralelização no Espaço e Tempo (PET)** : Visa explorar simultaneamente o paralelismo existente em cada passo de integração (paralelização no espaço) e entre os passos de integração (paralelização no tempo), resolvendo-se vários passos de integração simultaneamente [18, 19].

Os principais apelos da RFO são: a possibilidade de resolver um sistema de grande porte através da solução de subsistemas de pequeno porte; a utilização de passos de integração diferentes em diferentes subsistemas; e o fato de evitar a solução, por métodos diretos, de sistemas algébricos lineares de muito grande porte. Contudo, a efetividade do método depende fortemente da fase de decomposição que representa atualmente um problema crucial desta metodologia. A idéia inicial de se explorar o paralelismo existente entre os passos de integração numérica, em SEE, está descrita em [18] e teve continuidade em [19], com a proposição de um algoritmo denominado de Gauss-Jacobi-Block-Newton (GJBN). Este algoritmo explora o PTE, porém, os resultados de testes mostraram uma rápida saturação da curva de “speed up” com o crescimento do número de processadores [20].

#### 4 Paralelização no Espaço e no Tempo - Algoritmos Desenvolvidos

Neste ítem apresenta-se um sumário de três algoritmos desenvolvidos pelos autores deste trabalho, objetivando viabilizar uma efetiva utilização de computadores paralelos na SDSEE [15, 21, 22]. Nos dois primeiros investiga-se a paralelização no espaço e no terceiro a paralelização no tempo. A principal novidade é a utilização de métodos iterativos do tipo Gradiente Conjugado (GC) na resolução de sistemas lineares simétricos a assimétricos que surgem no âmbito das soluções propostas.

Os métodos do tipo Gradiente Conjugado derivam da família de métodos de solução de sistemas lineares denominada de métodos do subespaço de Krylov ou métodos tipo projetados [23, 2]. O mais popular é o Gradiente Conjugado (CG), utilizável somente em sistemas com a matriz de coeficientes simétrica e definida positiva [23]. Para matrizes assimétricas, têm sido empregados o GMRES (Generalized Minimal Residual) [24], o Gradiente Biconjugado (BCG - Biconjugate Gradient) [9], o CGS (Conjugate Gradient-Squared) [2] e, mais recentemente, o Bi-CGSTAB [25]. Optou-se pela utilização do método GC pré-condicionado, em sistemas simétricos [15, 21], e pelos métodos BCG e Bi-CGSTAB pré-condicionados em sistemas assimétricos [22].

O maior esforço computacional requerido pelos algoritmos associados a esses métodos está no cálculo de produtos internos e multiplicações de matriz por vetor. Essa peculiaridade é muito atrativa para o processamento em computadores paralelos/vetoriais. Por outro lado, há a necessidade de se encontrar uma efetiva matriz de pré-condicionamento  $M$  [23].

Nos algoritmos do tipo GC, o pré-condicionamento é efetuado, em geral, através de modificações no vetor de resíduos  $r = b - Ax$ , a cada iteração, obtidas resolvendo-se um **sistema linear algébrico auxiliar** do tipo  $M\tilde{r} = r$ . Assim, a escolha de  $M$  deve ser uma solução de compromisso tal que  $M$  seja o mais próximo possível de  $A$ , porém, de modo que o sistema auxiliar tenha solução relativamente simples. Sob a ótica dos computadores paralelos/vetoriais, é desejável que  $M$  seja do tipo bloco-diagonal, para que a solução possa ser obtida com paralelismo perfeito, e, na resolução de cada subsistema, utilize-se o máximo possível a capacidade vetorial da máquina.

##### 4.1 Paralelização no Espaço

Foram desenvolvidos dois algoritmos paralelos baseados no esquema *Alternado Entrelaçado Implícito*, denominados AEIGC e AEIGCRD, respectivamente, constituindo-se basicamente do seguinte [15, 21]:

- As equações diferenciais são transformadas em equações algébricas a diferenças, através do método de integração trapezoidal implícito, e resolvidas iterativamente e alternadamente com as equações originalmente algébricas;

- Para cada iteração nas equações diferenciais, a equação que descreve a rede elétrica é resolvida como se fosse linear, a partir da determinação das injeções de correntes para valores das tensões nodais da iteração anterior;
- As equações da rede elétrica são resolvidas por um método híbrido que combina os métodos GC e fatoração LU de matrizes esparsas (algoritmo AEIGC) ou totalmente pelo método Gradiente Conjugado (GC) pré-condicionado (algoritmo AEIGCRD);
- O teste de convergência é feito nas variáveis de estado, calculando-se um vetor de diferenças entre os valores das iterações atual e anterior e comparando-se a sua norma Euclidiana a uma tolerância especificada;

#### 4.1.1 Algoritmo AEIGC

Especificamente, as equações da rede elétrica são ordenadas de modo que a matriz admitância nodal assuma a forma BBDF. Isto é equivalente a decompor a rede elétrica em um conjunto de subredes interligadas somente através de uma única subrede, denominada subrede de interligação ou interconexão.

Supondo a existência de  $q$  subredes, interligadas através da subrede de interligação  $s$ , a equação matricial da rede elétrica assume a forma seguinte:

$$\begin{bmatrix} I_1(E_1, V_1) \\ I_2(E_2, V_2) \\ \vdots \\ I_p(E_p, V_p) \\ I_s(E_s, V_s) \end{bmatrix} = \begin{bmatrix} Y_1 & & & & \\ & Y_2 & & & \\ & & \ddots & & \\ & & & Y_q & \\ \bar{Y}_1^T & \bar{Y}_2^T & \dots & \bar{Y}_q^T & Y_s \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_q \\ V_s \end{bmatrix} \quad (3)$$

A equação (3) é resolvida em duas fases: Na primeira, para  $V_s$ , aplicando-se o método de eliminação Gaussiana por blocos e, na segunda, para cada  $V_i$ ,  $i = 1, 2, \dots, q$ , conforme dado a seguir:

**Fase 1:** Resolva

$$\hat{Y}_s V_s = \hat{I}_s \quad (4)$$

onde  $\hat{Y}_s = Y_s - \sum_{i=1}^q \bar{Y}_i^T Y_i^{-1} \bar{Y}_i$  e  $\hat{I}_s = I_s - \sum_{i=1}^q \bar{Y}_i^T Y_i^{-1} I_i$

**Fase 2:** Para  $i = 1, 2, \dots, q$ ; resolva

$$Y_i V_i = I_i - \bar{Y}_i V_s \quad (5)$$

Note que na **Fase 2** devem ser resolvidos  $q$  sistemas lineares completamente desacoplados. Então, usando-se  $q$  processadores, pode-se atribuir um sistema desses para cada processador e resolvê-los pelo método direto baseado na fatoração LU de  $Y_i$ . A matriz  $Y_i$  é esparsa e necessita ser refatorada, ao longo da simulação, somente nos instantes de tempo em que ocorrem alterações na configuração da rede elétrica. A paralelização do processo de resolução da equação (3) é otimizada aplicando-se o método GC para resolver a equação (4), conforme descrito em [15].

Então, aproveitando-se as peculiaridades da matriz admitância nodal na forma BBDF, as equações da dinâmica do sistema são reescritas na seguinte forma [15]:

**UFRGS**  
**INSTITUTO DE INFORMÁTICA**  
**BIBLIOTECA**

3. Cálculo das injeções de correntes  $I(E, V)$ ;
4. Resolução das equações da rede elétrica
5. Cálculo das variáveis de interface ( $u$ ).

Todos esses passos principais possuem um elevado grau de paralelismo, a saber:

- As componentes de  $u$  e  $V$  são extrapoladas independentemente;
- O acoplamento entre as variáveis de estado de máquinas síncronas diferentes, se dá somente através de variáveis da rede elétrica. Portanto, as equações diferenciais de máquinas diferentes podem ser integradas simultaneamente e independentemente;
- As componentes de  $u$ ,  $E$  e  $I$ , são funções somente de variáveis da barra à qual estão associadas;
- No algoritmo AEIGC a solução da rede é feita em duas fases : A **Fase 1** é a que apresenta maiores dificuldades de paralelização, contudo ela é realizada com elevado grau de paralelismo através do método GC. Na **Fase 2** deve-se resolver  $q$  subsistemas algébricos lineares independentes [14, 15];
- No algoritmo AEIGCRD a solução da rede é feita exclusivamente pelo método GC pré-condicionado, cuja predominância do esforço computacional está restrita a três tipos de operações com vetores. Em princípio, a decomposição desses vetores pode ser feita trivialmente. Além disso, 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 [21].

## 4.2 Paralelização no Tempo

Neste item descreve-se sucintamente o algoritmo SIMPT, desenvolvido em [22], que utiliza o paralelismo no tempo na SDSEE . O SIMPT está baseado no esquema *Simultâneo Implícito* e têm as seguintes características gerais:

- As equações diferenciais são discretizadas para vários passos de integração, denominados de janela de integração, e resolvidas juntamente com as equações algébricas desta janela, pelo método de Newton;
- Empregam-se os métodos BCG e Bi-CGSTAB na resolução do sistema linear Jacobiano de grande porte associado a janela de integração;
- A inicialização das variáveis, em cada janela de integração, é feita por série de Taylor;
- A forma de pré-condicionamento utilizada é do tipo fatoração LU incompleta para a matriz Jacobiana.

### 4.2.1 Formulação Matemática

O modelo matemático do sistema, após a discretização para uma janela de integração, é dado pelas seguintes equações [22] :

$$\begin{cases} \hat{F}_t(\hat{x}_t, \hat{x}_{t-1}, \hat{V}_t^e, \hat{V}_{t-1}^e) = 0 \\ \hat{G}_t(\hat{x}_t, \hat{V}_t^e) = 0 \quad t = 1, 2, \dots, p \end{cases} \quad (9)$$

$$\begin{cases} \dot{x}_i = f_i(x_i, u_i) \\ I_i(E_i, V_i) = Y_i V_i - \bar{Y}_i V_s \\ \dot{Y}_s(E, V) = \dot{Y}_s V_s \\ u_i = h_i(E_i, V_i) \\ i = 1, 2, \dots, q \end{cases} \quad (6)$$

onde  $x_i$  é o vetor de variáveis de estado do  $i$ -ésimo subsistema de equações diferenciais;  $f_i$  são funções vetoriais que definem o  $i$ -ésimo subsistema de equações diferenciais;  $u_i$  é o vetor de variáveis do  $i$ -ésimo subsistema de equações de interface;  $h_i$  é o vetor de funções não lineares do  $i$ -ésimo subsistema de equações de interface; e  $q$  é o número de subsistemas.

#### 4.1.2 Algoritmo AEIGCRD

Os apelos principais para essa proposta são a independência de uma decomposição da rede elétrica para a forma BBDF e a aplicação do método GC em sistemas com dimensões maiores que as propiciadas no algoritmo AIEGC, de modo a permitir a exploração do elevado potencial de vetorização desse método. Por outro lado, os principais desafios são a definição de um efetivo método de pré-condicionamento e o mapeamento do algoritmo em computadores paralelos. Inicialmente, foram utilizadas algumas técnicas de pré-condicionamento encontradas na literatura, tais como: a) pré-condicionamento diagonal [23, 2] e b) séries truncadas de Neumann [23, 2] e MacLaurin [20]. Embora a simplicidade e as potencialidades para o paralelismo e vetorização, essas técnicas não foram efetivas em sistemas de grande porte. Então, optou-se pela proposição de uma nova versão de pré-condicionamento do tipo fatoração LU incompleta, decrita a seguir, que permite a resolução do sistema auxiliar com paralelismo perfeito [21].

Supondo que a rede elétrica constitua-se de  $q$  subredes interligadas, a sua descrição linearizada pode ser representada como segue:

$$\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_q \end{bmatrix}^k = \begin{bmatrix} Y_{11} & Y_{12} & \dots & Y_{1q} \\ Y_{21} & Y_{22} & \dots & Y_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{q1} & Y_{q2} & \dots & Y_{qq} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_q \end{bmatrix}^{k+1} \quad (7)$$

Propõe-se, então, que  $M$  seja constituído somente dos blocos diagonais de  $Y$ . Assim, o sistema auxiliar assume a forma seguinte:

$$\begin{bmatrix} Y_{11} & & & \\ & Y_{22} & & \\ & & \ddots & \\ & & & Y_{qq} \end{bmatrix} \begin{bmatrix} \bar{r}_1 \\ \bar{r}_2 \\ \vdots \\ \bar{r}_q \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_q \end{bmatrix} \quad (8)$$

Note que o sistema auxiliar pode ser resolvido através das técnicas de fatoração LU de matrizes esparsas e com paralelismo perfeito. Entre elas destaca-se o método dos fatores inversos, que apresentou ganhos significativos quando empregam-se técnicas de vetorização [8].

#### 4.1.3 Paralelização

Os algoritmos AEIGC e AEIGCRD, essencialmente, baseiam-se nos seguintes passos principais [14, 15, 21]:

1. Extrapolação das tensões nas barras e variáveis de interface ( $u$  e  $V$ );
2. Cálculos na fase de integração numérica implícita;



nestes  $q$  processadores. O resultado final será o mapeamento das equações da janela de integração em  $q \times p$  processadores, caracterizando-se a chamada paralelização no tempo e no espaço.

## 5 Avaliação de Desempenho em um Ambiente Computacional Paralelo

Neste ítem apresenta-se uma síntese dos resultados da aplicação e avaliação de desempenho dos algoritmos AEIGC, AEIGCRD e SIMPT desenvolvidos. Foram realizadas simulações, sequenciais e paralelas, do comportamento dinâmico de uma configuração do sistema interligado das regiões Sul e Sudeste do Brasil, com 616 barras e 88 geradores representados dinamicamente. Os experimentos consideraram perturbações do tipo curto-circuito trifásico eliminado pelo chaveamento de uma linha de transmissão. As avaliações de desempenho são feitas a partir da definição de figuras de mérito, tais como: tempos de CPU e comunicação, eficiências e “speed ups”.

### 5.1 Computadores Paralelos Utilizados

Os processamentos paralelos reais foram realizados no NCP I, somente para o algoritmo AEIGC, e no Intel iPSC/860, para todos os algoritmos desenvolvidos. Ambos os computadores citados pertencem ao Núcleo de Computação Paralela da COPPE/UFRJ.

O NCP I, desenvolvido na COPPE/UFRJ, possui arquitetura hipercúbica e está, atualmente, com 8 processadores conectados a um microcomputador do tipo IBM PC/386. Cada nó desse computador paralelo possui um processador de 32 bits com tecnologia RISC, do tipo Transputer T800, e memória local de 2Mb. A capacidade de pico de cada processador, operando com clock de 20 MHz, é de aproximadamente 10 Mips e 1,5 Megaflops. O sistema de comunicação é do tipo “message-passing” e tem capacidade de 20 Megabits por segundo. Estes elementos definem uma capacidade de pico de 12 Megaflops e 80 Mips, para a versão atual do NCP I.

O Intel iPSC/860 é do tipo multiprocessador com memória distribuída, topologia hipercúbica e, atualmente, possui 8 nós de processamento que estão conectados a um microcomputador do tipo IBM PC/386. Cada nó contém um microprocessador i860 com tecnologia RISC de 64 bits e 8 Megabytes de memória. O sistema de comunicação entre os nós é do tipo “message-passing” e tem capacidade de pico de 2,8 Megabytes por segundo. Adicionalmente, cada mensagem partindo de um nó toma cerca de 65 microsegundos de tempo de CPU para estabelecer seu caminho. A capacidade de pico de cada processador é de até 80 Megaflops em operações com precisão simples e de até 60 Megaflops em precisão dupla.

### 5.2 Programas Computacionais

As implementações computacionais, sequenciais e paralelas, dos algoritmos desenvolvidos estão organizadas em uma estrutura de sistema computacional denominado PPSDS (“Parallel Power System Dynamic Simulation”). Este sistema, desenvolvido pelos autores deste trabalho, contempla os esquemas de solução *Alternado Entrelaçado Implícito* e *Simultâneo Implícito*, e os algoritmos AEIGC, AEIGCRD e SIMPT. Neste contexto encontram-se diversas opções de modelos dos elementos de SEE, diversos métodos de solução de sistemas lineares simétricos e assimétricos e diversos métodos de pré-condicionamento.

### 5.3 Eficiências e “Speed ups”

Definem-se dois conjuntos de “speed ups” e eficiências: um para o esquema alternado, algoritmos AEIGC e AEIGCRD, e outro para o esquema simultâneo, algoritmo SIMPT.

### 5.3.1 Para o esquema Alternado

$$S_{a1} = \frac{t_{CPU} \text{ para o algoritmo AEIC (sequencial)}}{t_{CPU} \text{ para o algoritmo AEI com GC (paralelo)}} , \quad E_{a1} = S_{a1}/q$$

$$S_{a2} = \frac{t_{CPU} \text{ para o algoritmo AEI com GC (sequencial)}}{t_{CPU} \text{ para o algoritmo AEI com GC (paralelo)}} , \quad E_{a2} = S_{a2}/q$$

onde AEIC refere-se ao esquema Alternado Entrelaçado Implícito Convencional [12, 15],  $t_{CPU}$  representa o tempo de CPU em segundos e  $q$  é o número de processadores utilizados em paralelo. O “speed up”  $S_{a2}$  e a eficiência  $E_{a2}$  representam uma medida da paralelização alcançada pelo algoritmo que utiliza o método GC (AEIGCRD ou AEIGC). Por outro lado,  $S_{a1}$  e  $E_{a1}$  representam, respectivamente, o “speed up” e a eficiência alcançados pelo algoritmo AEIGCRD ou AEIGC paralelos, comparativamente ao algoritmo AEIC sequencial. Consequentemente, eles refletem o ganho efetivo no emprego de computadores paralelos.

### 5.3.2 Para o esquema Simultâneo

$$S_{s1} = \frac{t_{CPU} \text{ para } p \text{ passos (serial)}}{t_{CPU} \text{ para } p \text{ passos (paralelo)}} , \quad E_{s1} = S_{s1}/p$$

$$S_{s2} = \frac{t_{CPU} \text{ para } 1 \text{ passos (serial)}}{t_{CPU} \text{ para } p \text{ passos (paralelo)}} , \quad E_{s2} = S_{s2}/p$$

onde  $t_{CPU}$  representa o tempo de CPU em segundos e  $p$  é o número de passos de integração tomados simultaneamente, assim como também o número de processadores nos processamentos paralelos. O “speed up”  $S_{s1}$  e a eficiência  $E_{s1}$ , representam uma medida da paralelização alcançada com o algoritmo SIMPT, pois consideram, como referência, o processamento sequencial com o mesmo número de passos de integração dos processamentos paralelos. Por outro lado, o “speed up”  $S_{s2}$  e a eficiência  $E_{s2}$ , representam, respectivamente, o “speed up” e a eficiência alcançados pelo algoritmo SIMPT paralelo, em relação ao esquema Simultâneo Implícito para 1 (um) passo de integração.

## 5.4 Resultados Computacionais

A tabela 1 mostra os tempos de CPU, “speed ups” e eficiências obtidos com os algoritmos paralelos AEIGC e AEIGCRD, ou seja, com a paralelização no espaço do esquema *Alternado Entrelaçado Implícito*. Também nesta tabela e para fins de comparação, encontram-se os tempos obtidos em processamentos sequenciais do algoritmo AEIC em diversas máquinas. Note que todos os valores de “speed ups” e eficiências, associados a um determinado computador, foram obtidos em relação a processamentos sequenciais realizados em um único processador do mesmo computador.

Na figura 1 mostram-se tempos de CPU e comunicação ( $c$ ) em processamentos, paralelos (*Par.*) e sequenciais (*Seq.*), realizados com o algoritmo SIMPT no computador iPSC/860. Os retângulos tracejados referem-se a utilização do método Gradiente Biconjugado pré-condicionado (BCG) e os em linha cheia a utilização do método Bi-CGSTAB pré-condicionado, na resolução da equação matricial linear Jacobiana.  $p$  é o número de passos de integração tomados simultaneamente e também o número de processadores nos processamentos paralelos.

## 5.5 Comentários

### Paralelização no Espaço

Table 1: Tempos de CPU, "Speed ups" e Eficiências - paralelização no espaço

| Algoritmo | Computador | q | CPU (s) | "Speed up" e Eficiência                              |                   |                 |                   |
|-----------|------------|---|---------|--|-------------------|-----------------|-------------------|
|           |            |   |         | S <sub>a1</sub>                                      | E <sub>a1</sub> % | S <sub>a2</sub> | E <sub>a2</sub> % |
| AEIGC     | NCP I      | 2 | 334     | 1,30   | 65                | 2,00            | 100               |
|           |            | 4 | 175     | 2,50   | 62                | 3,60            | 90                |
|           |            | 8 | 112     | 3,90   | 48                | 5,60            | 70                |
| AEIGC     | iPSC/860   | 2 | 38,7    | 1,61   | 80                | 1,95            | 98                |
|           |            | 4 | 26,7    | 2,33   | 58                | 2,85            | 71                |
|           |            | 8 | 23,9    | 2,60   | 33                | 3,18            | 40                |
| AEIGCRD   | iPSC/860   | 2 | 37,6    | 1,66   | 83                | 1,90            | 95                |
|           |            | 4 | 26,0    | 2,40   | 60                | 2,82            | 71                |
|           |            | 8 | 23,5    | 2,65   | 33                | 3,44            | 43                |
| AEIC      | NCP I      | 1 | 436     | q : Num. de processadores<br>CPU : Tempos de CPU (s) |                   |                 |                   |
|           | IBM 4381   | 1 | 194     |  |                   |                 |                   |
|           | IBM 3090   | 1 | 20      |  |                   |                 |                   |
|           | iPSC/860   | 1 | 62,3    |  |                   |                 |                   |

- Observa-se na tabela 1 que o processamento paralelo reduziu significativamente os tempos das simulações dinâmicas realizadas. Além disso obteve-se tempos comparáveis aos de mainframes que são muitas vezes mais caros. Ilustra-se este fato destacando-se dois resultados: 1) Usando-se 4 Transputers, o tempo de CPU foi inferior ao de um IBM 4381. Observa-se que atualmente é possível adquirir placas com 4 transputers T800, acopláveis a um microcomputador pessoal, a um custo inferior a dez mil dolares; 2) os tempos de CPU no iPSC/860 são comparáveis aos de um IBM 3090.
- O algoritmo AEIGCRD foi levemente superior, mesmo tendo tempos de comunicação maiores e sendo mais sensível a decomposição que o AEIGC;
- Na configuração de 8 processadores, os valores de "speed up" e eficiência, para o NCP I, são significativamente superiores. Isto é devido a razão capacidade de processamento pela capacidade de comunicação, muito superior no iPSC/860.
- Uma parcela significativa da comunicação de dados entre os processadores independe da dimensão do sistema, indicando que os índices de desempenho devem ser ainda melhores em sistemas de maior dimensão. Além disso, a literatura fora da área de sistema de energia elétrica registra, em estudos realizados com processamentos sequencias, maior competitividade dos métodos tipo GC, em relação ao métodos diretos, para sistemas com dimensões muito superiores as utilizadas nos teste realizados neste trabalho.

#### Paralelização no Tempo

A figura 1 ilustra com clareza os principais méritos e problemas da paralelização no tempo tal como foi proposta. Eles são:

- O algoritmo SIMPT, em sua versão sequencial, apresenta um significativo "slowdown" com o crescimento do número de passos de integração tomados simultaneamente. Há evidências que esse "slowdown" está diretamente associado ao método de pré-condicionamento utilizado e que maiores investigações neste tema poderão reduzi-lo consideravelmente;

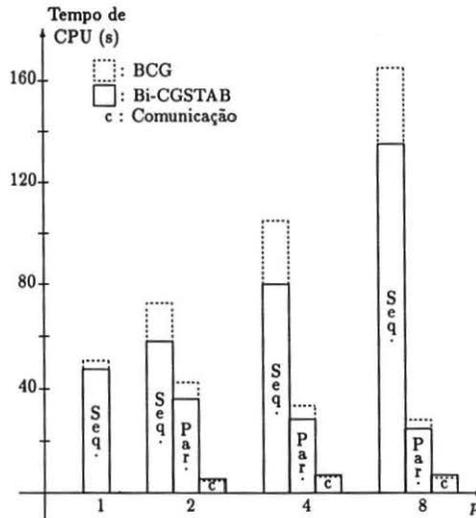


Figure 1: Tempos de CPU e Comunicação para o algoritmo SIMPT sequencial e paralelo

- Os “speed ups” são elevados na comparação direta com a versão sequencial para o mesmo número de passos tomados simultaneamente (“speed up”  $S_{s1}$ ), mostrando a efetividade da paralelização proposta;
- A paralelização no tempo tal como foi proposta apresenta ganhos significativos, mesmo com o “slowdown” sequencial do algoritmo (“speed up”  $S_{s2}$ );
- O método Bi-CGSTAB mostrou-se superior ao BCG na resolução do sistema linear Jacobiano;
- Os tempos de comunicação permaneceram praticamente constantes com o crescimento do número de processadores.

## 6 Conclusões

Neste artigo discutiu-se a aplicabilidade de computadores paralelos de alto desempenho na resolução de problemas de análise de redes elétricas. Em mais detalhes, foi apresentado um sumário de três algoritmos paralelos desenvolvidos pelos autores para a Simulação da Dinâmica de Sistemas de Energia Elétrica. As principais conclusões são:

- O interesse dos profissionais de Sistemas de Energia Elétrica (SEE) pela computação paralela é inevitável em razão da dimensão dos problemas envolvidos, associados a necessidade de alta velocidade de cálculo, principalmente nos Centros de Supervisão e Controle da operação em tempo real (CSC); e da potencialidade de se dispor de estações de trabalho de grande capacidade computacional a custo relativamente baixo. Contudo, trata-se de uma tecnologia relativamente recente e que apresenta, na diversidade de arquiteturas computacionais, uma variável adicional aos problemas de SEE. Maiores investigações são necessárias a nível de “hardware”, “software” e no desenvolvimento de algoritmos, no sentido de transformar

a elevada capacidade computacional e as alternativas de arquiteturas oferecidas por estas máquinas, em ganhos de tempo e qualidade na realização dos estudos.

- No que se refere aos três algoritmos desenvolvidos, a principal conclusão é a efetiva redução obtida nos tempos de CPU de simulações da dinâmica de um sistema com 616 barras e 88 geradores, em ambientes paralelos reais. Mostrou-se também a efetividade da utilização de métodos iterativos do tipo Gradiente Conjugado, no âmbito das metodologias de resolução das equações que descrevem a dinâmica de SEE. Por outro lado, os trabalhos de pesquisa devem continuar, principalmente na definição de métodos de pré-condicionamento mais efetivos que os utilizados.

## Agradecimentos

Ao CEPEL pelo apoio financeiro na manutenção do Laboratório de Computação Paralela da COPPE/UFRJ. O primeiro autor também agradece a Universidade Federal de Santa Catarina e a CAPES/PICD pelo suporte financeiro.

## References

- [1] G. Meurant, "The evolution of scientific computing on parallel computers," August 1992. I Escola de Computação Científica de Alto Desempenho, LNCC/CNPq, Rio de Janeiro, Brasil.
- [2] J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. Van der Vorst, *Solving Linear Systems on Vector and Shared Memory Computers*. SIAM - Society for Industrial and Applied Mathematics, 1991.
- [3] K. Hwang and F. Briggs, *Computer Architecture and Parallel Processing*. New York: McGraw-Hill Book Company, 1984.
- [4] K. Hwang, "Advanced parallel processing with supercomputer architectures," *Proceedings of the IEEE*, vol. 75, pp. 1348-1379, Oct 1987.
- [5] F. M. Brasch, J. E. Van Ness, and S. C. Kang, "Simulation of a multiprocessor network for power system problems," *IEEE Trans Power Apparatus and Systems*, vol. PAS-101, no. 2, p. , 1982.
- [6] IEEE Committee Report, "Parallel processing in power systems computation," *IEEE Trans. on Power Systems*, vol. 7, pp. 629-638, May 1992.
- [7] M. Rodrigues, O. R. Saavedra, and A. Monticelli, "Modelo de programação assíncrona para a solução concorrente do fluxo de potência Ótimo com restrições de segurança," in *9o CBA - Congresso Brasileiro de Automática*, (UFES - Vitoria/ES), pp. paper WSP2-3, Setembro 14-18 1992.
- [8] A. Gomes and R. Betancourt, "Implementation of the fast decoupled load flow on a vector computer," *IEEE Trans. on power systems*, vol. 5, no. 3, pp. 977-983, 1990.
- [9] C. L. T. Borges, *Investigação do desempenho de métodos de solução do fluxo de potência para processamento paralelo e vetorial*. Master's thesis, COPPE/UFRJ, Rio de Janeiro, RJ, 1991.
- [10] D. M. Falcão, E. Kaszkurewicz, and H. L. S. Almeida, "Application of parallel processing techniques to the simulation of power system electromagnetic transients," in *IEEE/PES Winter Meeting*, pp. paper WM 287-3 PWRs, 1992.

- [11] J. M. Campagnolo, N. Martins, J. C. R. Pereira, L. T. G. Lima, H. J. C. P. Pinto, and D. M. Falcão, "Fast small-signal stability assessment using parallel processing," in *IEEE/PES Summer Meeting*, July 1993.
- [12] B. Stott, "Power system dynamic response calculations," *Proceedings of the IEEE*, vol. 67, pp. 219-241, Feb 1979.
- [13] M. Ilić-Spong, M. L. Crow, and M. A. Pai, "Transient stability simulation by waveform relaxation method," *IEEE Trans. on Power Systems*, vol. PWRS-2, pp. 943-952, Nov 1987.
- [14] I. C. Decker, D. M. Falcão, and E. Kaszkurewicz, "An efficient parallel method for transient stability analysis," in *Proc. of 10<sup>th</sup> PSCC, Power Systems Computation Conference*, (Graz, Austria), pp. 509-516, Aug. 19-24 1990.
- [15] I. C. Decker, D. M. Falcão, and E. Kaszkurewicz, "Parallel implementation of power system dynamic simulation methodology using the conjugate gradient method," *IEEE Trans. on Power Systems*, vol. 7, pp. 458-465, Feb 1992.
- [16] H. Taoka, I. Iyoda, H. Noguchi, N. Sato, and T. Nakazawa, "Real-time digital simulator for power system analysis on a hypercube computer," *IEEE Trans. on Power Systems*, vol. 7, pp. 1-10, Feb 1992.
- [17] M. Crow and M. Ilić, "The parallel implementation of the waveform relaxation method for transient stability simulations," *IEEE Trans. on Power Systems*, vol. 5, pp. 922-931, Aug. 1990.
- [18] F. L. Alvarado, "Parallel solution of transient problems by trapezoidal integration," *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-98, pp. 1080-1090, May/June 1979.
- [19] M. LaScala, M. Brucoli, F. Torelli, and M. Trovato, "A gauss-jacobi-block-newton method for parallel transient stability analysis," *IEEE Trans. on Power Systems*, vol. 5, pp. 1168-1177, Nov 1990.
- [20] J. S. Chai and A. Bose, "Bottlenecks in parallel algorithms for power systems stability analysis," in *IEEE/PES 1992 Winter Meeting*, (New York), pp. 92 WM 285-7 PWRS, Jan 26 - 30 1992.
- [21] I. C. Decker, D. M. Falcão, and E. Kaszkurewicz, "Esquema alternado entrelaçado implícito paralelo - aplicação do método gradiente conjugado pré-condicionado as equações da rede elétrica," Relatório Interno, Programa de Engenharia Elétrica, COPPE/UFRJ, Ilha do Fundão, Bloco H, Rio de Janeiro -RJ., 1993.
- [22] I. C. Decker, D. M. Falcão, and E. Kaszkurewicz, "Esquema simultâneo implícito paralelo no tempo - aplicação dos métodos gradiente biconjugado e bi-cgstab," Relatório Interno, Programa de Engenharia Elétrica, COPPE/UFRJ, Ilha do Fundão, Bloco H, Rio de Janeiro -RJ. 1993.
- [23] J. M. Ortega, *Introduction to parallel and vector solution of linear systems*. New York: Plenum Press, 1988.
- [24] M. A. Pai, P. W. Sauer, and A. Y. Kulkarni, "Conjugate gradiente approach to parallel processing in dynamic simulation of power systems," in *1992 ACC/TA10*, pp. 1644-1647, 1992.
- [25] H. A. Van der Vorst, "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.