

SOLUÇÃO DE EQUAÇÕES ALGÉBRICAS LINEARES POR MÉTODOS BLOCO-ITERATIVOS IMPLEMENTADOS EM PARALELO

A. Bhaya, D. M. Falcão, E. Kaszkurewicz e N. Roqueiro
COPPE/UFRJ - Programa de Engenharia Elétrica
C.P. 68504 - 21945 Rio de Janeiro, RJ

RESUMO

Este artigo descreve a utilização de métodos bloco-iterativos paralelos, síncronos e assíncronos, para a solução de sistemas de equações algébricas lineares de grande porte. Estes métodos são utilizados na resolução do problema de Fluxo de Potência em redes de energia elétrica. O computador utilizado é o ACP (Advanced Computer Program) disponível no CBPF com a seguinte configuração: 02 "crates", 10 processadores (68020 μ P) por "crate" e um MicroVax como hospedeiro.

ABSTRACT

This paper describes synchronous and asynchronous parallel block-iterative methods for the solution of large systems of linear algebraic equations arising in the analysis of power systems. These methods are applied to the load-flow problem of electrical power networks. The computer used is the CBPF's 02 crate ACP (Advanced Computer Program) with ten 68020 μ Ps per crate and a MicroVax as host.

1. INTRODUÇÃO

O recente desenvolvimento de processadores paralelos, do tipo "array" vetoriais com estruturas alternativas de alta velocidade e baixo custo está provocando mudanças significativas nas técnicas e linguagens de programação assim como na estrutura dos algoritmos utilizados. Na computação científica, as várias maneiras de conceber os métodos numéricos e adaptá-los a estas novas máquinas e arquiteturas estão estimulando novas pesquisas teóricas e aplicadas em problemas de convergência, balanceamento de carga, aceleração relativa ("speed-up"), etc. É notável que diversas técnicas abandonadas no passado, devido à falta de recursos computacionais, estão em fase de ressurgimento.

Mais especificamente, para resolver a equação linear clássica $Ax=b$, quando a ordem da matriz $A(n \times n)$ é elevada, os chamados métodos diretos, como eliminação Gaussiana, combinados com técnicas de esparsidade são atualmente considerados os mais eficientes para as máquinas sequenciais. Porém os métodos iterativos tais como Jacobi, Gauss-Seidel, SOR, Gradiente Conjugado, que apresentam convergência lenta em máquinas sequenciais, têm se mostrado bastante eficientes e rápidos para problemas de grande porte, particularmente quando implementados em paralelo [1]. A maioria das implementações dos métodos diretos não tem se mostrado eficientes para

máquinas paralelas, por exigir demasiada comunicação entre os processadores. Dada uma partição adequada (em blocos) da matriz A , os métodos bloco-iterativos freqüentemente apresentam um paralelismo inerente - um caso típico é o clássico método iterativo de Jacobi por blocos - e muitas vezes exigem menos comunicação entre processadores.

Um efeito adverso da comunicação chamada síncrona é que os processadores mais rápidos (aqueles que computam os subsistemas de equações mais esparsas, por exemplo) ficam à espera dos mais lentos, comprometendo desta forma a rapidez de convergência. Uma alternativa é de se implementar o método iterativo com comunicação assíncrona entre os processadores. Os problemas de convergência de métodos iterativos tais como Jacobi, Gauss-Seidel, etc. quando implementados em multiprocessadores distribuídos, com comunicação assíncrona, foram considerados por Chazan & Miranker [2] e posteriormente por Baudet [3].

Na Seção 2 do presente trabalho é apresentado o modelo matemático de métodos iterativos assíncronos proposto por Baudet [3], assim como uma condição suficiente para a convergência destes métodos. Na seção 3 é indicado um procedimento para a aplicação dos métodos referidos ao problema de Fluxo de Potência em redes de energia elétrica. A seção 4 contém uma breve descrição da arquitetura do ACP bem como a lógica de pro-

gramação do hospedeiro ("host") e dos nós. As conclusões são apresentadas na seção 5 e as referências bibliográficas na seção 6.

2. MODELOS MATEMÁTICOS DE MÉTODOS BLOCO-ITERATIVOS PARALELOS

O problema da resolução da equação:

$$F(x) = 0 \quad (1)$$

onde $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ é suficientemente "suave", pode ser transformado num problema de determinação dos pontos fixos de um sistema iterativo:

$$x(i+1) = G(x(i), i); \quad x(0) \text{ dado}; \quad i=0, 1, 2, \dots \quad (2)$$

onde $G: \mathbb{R}^n \times \mathbb{N} \rightarrow \mathbb{R}^n$. Achando os pontos x^* que verificam $x^* = G(x^*, i) \forall i$, estaremos determinando aqueles x^* que verificam (1). Em particular, se $F(x) = Ax + b$ (i.e., F linear), então os métodos clássicos iterativos, como por exemplo Jacobi, Gauss-Seidel, SOR, etc., são expressos na forma (2), onde G é linear, [4], [5].

A implementação paralela dos métodos iterativos para a solução de sistemas de grande porte sugere naturalmente a utilização de métodos bloco-iterativos, alocando-se a cada um dos processadores a resolução de um determinado bloco. Estes métodos bloco-iterativos já foram estudados intensivamente no contexto de implementações sequenciais [6] e atualmente estão sendo resuscitados e usados em uma grande variedade de métodos iterativos paralelos, [7], [8], [9].

Considerando-se o caso da comunicação síncrona, uma expressão geral para os métodos bloco-iterativos pode ser dada por:

$$Z_\ell(i+1) = \sum_{j=1}^n H_{\ell j}(Z(i), i) Z_j(i); \quad (3)$$

$$i = 0, 1, 2, \dots$$

$$\ell = 1, 2, \dots, n$$

onde i é o número da iteração (ou "tempo discreto") e, $\forall i$, $Z_j(i) \in \mathbb{R}^{n_j}$ e $H_{\ell j}(Z(i), i) \in \mathbb{R}^{n_\ell \times n_j}$;

$N := \sum_{\ell=1}^n n_\ell$; $Z(i)^T = (z_1(i)^T, \dots, z_n(i)^T)$ e cada uma

das n equações (3) é alocada a um dos n processadores.

Quando a comunicação entre os processadores é assíncrona, o modelo (3) não é adequado e utilizamos o modelo de Baudet [3]. Introduzimos a seguinte notação:

as componentes de um vetor $x \in \mathbb{R}^n$ e um operador $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ são, respectivamente denotadas por x_ℓ , $\ell=1, \dots, n$; e $f_\ell(x)$ ou $f_\ell(x_1, \dots, x_n)$, $\ell=1, \dots, n$. Uma seqüência de vetores em \mathbb{R}^n é denotada por $x(i)$, $i = 0, 1, \dots$.

Definição de iterações assíncronas: Seja $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$. Uma iteração assíncrona correspondendo ao operador F , dada a condição inicial $x(0) \in \mathbb{R}^n$, é uma seqüência $x(i)$, $i=0, 1, \dots$ de vetores em \mathbb{R}^n definidos recursivamente pelas equações:

$$x_\ell(i) = \begin{cases} x_\ell(i-1) & \text{se } \ell \notin J_i \\ f_\ell(x_1(s_1(i)), \dots, x_n(s_n(i))) & \text{se } \ell \in J_i \end{cases} \quad (4)$$

onde $J = \{J_i: i=1, 2, \dots\}$ é uma seqüência de subconjuntos não-vazios de $\{1, \dots, n\}$ e $S = \{s_\ell(i), \dots, s_n(i): i=1, 2, \dots\}$ é uma seqüência de elementos em \mathbb{N}^n . As seqüências J e S estão sujeitas às seguintes condições adicionais, para cada $\ell = 1, \dots, n$:

- (a) $s_\ell(i) \leq i-1$, $i = 1, 2, \dots$
- (b) $s_\ell(i)$, considerada como função de i , tende a infinito.
- (c) $\{i: \ell \in J_i\}$ é um conjunto infinito.

Uma iteração assíncrona correspondendo a F , com condição inicial $x(0)$ e definida por J e S , é denotada por $(F, x(0), J, S)$.

Podemos entender uma iteração assíncrona $(F, x(0), J, S)$ como sendo a seguinte seqüência de computações em um multiprocessador assíncrono:

Seja $\{t_i, i=1, 2, \dots\}$ uma seqüência crescente de instantes de tempo.

No instante t_i , o multiprocessador M começa a avaliação de $x(i)$; $x(i)$ difere de $x(i-1)$ nas componentes $\{x_\ell: \ell \in J_i\}$ e $\notin J_i$ dos processadores de M (onde $\# J_i$ é o número de elementos de J_i) iniciam o cálculo destas componentes usando valores de componentes calculados em iterações anteriores; a saber: a r -ésima componente do $s_r(i)$ -ésimo iterado, para $r=1, \dots, n$. A escolha das componentes usadas pode ser feita segundo qualquer critério, e, em particular, um critério natural é selecionar os valores mais recentes disponíveis (ULD - use latest data available [8]) tornando assim desnecessário o sincronismo entre processadores. A única restrição, (b), sobre a escolha dos iterados anteriores é de que ela tem que ser atualizada, se bem que a condição (b) permite que o intervalo entre uma atualização e a próxima seja arbitrariamente

grande. A condição (a) exige que apenas componentes de iterados antigos sejam utilizadas na avaliação de um iterado novo. A condição (c) garante que as componentes serão atualizadas.

2.1 - Exemplos e Casos Particulares de Iterações Assíncronas.

Os métodos iterativos clássicos - Jacobi, bloco Jacobi, Gauss-Seidel etc., e alguns mais recentes (e.g. iteração caótica) são casos especiais de (4):

Ex. 1-Jacobi (blocos $l \times l$)

$$J: \{J_i = \{1, 2, \dots, n\} \text{ para } i=1, 2, \dots\}$$

$$S: \{s_\ell(i) = i-1 \text{ para } i=1, 2, \dots, \text{ e } \ell=1, \dots, n\}.$$

Neste caso todos os n processadores são atualizados simultaneamente (iteração paralela com comunicação síncrona)

Ex. 2-Jacobi (blocos $l \times l$)

$$J: \{J_i = \{1 + (i-1 \bmod n)\} \text{ } i=1, 2, \dots\}$$

$$S: \{s_\ell(i) = n \lfloor (i-1)/n \rfloor \text{ } i=1, 2, \dots, n; \ell=1, \dots, n\}$$

onde $\lfloor x \rfloor$ é o maior inteiro menor do que x .

Neste caso, cada componente é atualizada independentemente das outras; n passos deste processo assíncrono correspondem a um passo do processo do Ex. 1. Este é um dos modelos proposto em [2].

2.2 - Condições para Convergência de uma Iteração Assíncrona.

No caso dos algoritmos iterativos lineares invariantes no tempo (para blocos unitários) na sua versão síncrona, a partir da equação (3), podem ser descritos por $Z(i+1) = HZ(i)$; $i=0, 1, 2, \dots$ onde $H=(h_{ij})$ é a matriz constante de iteração. A condição necessária e suficiente de convergência neste caso é dada por $\rho(H) < 1$ onde $\rho(H)$ é o raio espectral da matriz de iteração H , veja a referência [4]. Já no caso assíncrono, as condições obtidas até o presente, por Chazan e Miranker são mais restritivas: o resultado em [2] assegura que a versão assíncrona do método iterativo converge quando $\rho(|H|) < 1$ onde $|H|$ é a matriz obtida a partir de H tomando-se os módulos dos elementos, i.e., $|H| := (|h_{ij}|)$.

Para testar se uma dada matriz de iteração H verifica a condição acima, no caso de sistemas de grande porte, constitui um problema computacional considerável. Portanto condições computacionalmente mais fáceis (mesmo sendo mais restritivas) podem ser úteis. Neste caso, condições como simetria com diagonal dominância estrita, irredutibilidade com diagonal dominância se apresentam como mais convenientes para serem testa-

das no caso das iterações assíncronas [2].

3. APLICAÇÃO AO PROBLEMA DE FLUXO DE POTÊNCIA

O problema de Fluxo de Potência (FP) em redes elétricas consiste na solução da rede em regime permanente para uma dada condição de carga-geração. Este problema, em sua forma básica, ou como componente de problemas mais complexos, é largamente utilizado em estudos de planejamento, operação e controle em tempo real de sistemas elétricos de potência.

3.1 - Formulação do Problema.

Considere as seguintes equações básicas da análise de circuitos em regime permanente senoidal:

$$P_k + j Q_k = V_k I_k^*, \quad k = 1, n \quad (5)$$

$$I_k = \sum_{m \in \Omega_k} Y_{km} V_m, \quad k = 1, n \quad (6)$$

onde

k : número do nó

P_k, Q_k : injeções de potência ativa e reativa

I_k : injeção de corrente

V_k : tensão nodal ($V_k = |V_k| \angle \theta_k$)

Y_{km} : elementos da matriz admitância nodal (Y)

Ω_k : (nós diretamente ligados a k , inclusive k)

De (5) e (6), obtém-se as equações do FP:

$$P_k + j Q_k = V_k \left(\sum_{m \in \Omega_k} Y_{km}^* V_m^* \right), \quad k=1, n \quad (7)$$

No conjunto de equações definido em (7), o termo $P_k + j Q_k$ representa a condição de carga-geração a qual é, em princípio especificada, e os termos V_k ($k=1, n$) são as incógnitas do problema.

Por razões relacionadas com a maneira de se especificar a condição de carga-geração, em alguns nós da rede são especificados P_k e $|V_k|$ ou $|V_k|$ e θ_k ao invés de P_k e Q_k .

3.2 - Métodos de Solução.

A solução do conjunto de equações definido em (7), ou de um conjunto equivalente obtido separando-se as partes reais e imaginárias, pode ser obtido através de processos iterativos que consistem na construção e solução de sistemas de equações lineares a partir de uma condição inicial arbitrária.

Duas classes de métodos são encontradas na literatura [10]:

a. Aproximação da condição de carga-geração por corrente constante

Neste caso, I_k na iteração de ordem i é dado por:

$$I_k(i) = \frac{P_k - j Q_k}{V_k^*(i)} \quad k = 1, n \quad i = 0, 1, 2, \dots \quad (8)$$

Cada iteração do método consiste, então, na solução de

$$I(i) = Y V(i+1) \quad (9)$$

b. Linearização através da fórmula de Taylor

A separação de (7) em suas partes reais e imaginárias, e posterior aproximação linear das funções resultantes em torno do valor das incógnitas na iteração anterior, conduz ao seguinte sistema linear incremental:

$$\begin{bmatrix} \Delta P(i) \\ \Delta Q(i) \end{bmatrix} = \begin{bmatrix} H & N \\ M & L \end{bmatrix} \begin{bmatrix} \Delta \theta(i+1) \\ \Delta |V(i+1)| \end{bmatrix} \quad (10)$$

onde H , N , M e L são submatrizes do Jacobiano da função relacionando P e Q com $|V|$ e θ , calculadas em $V(i)$.

Baseando-se em características particulares das redes elétricas de alta tensão, é possível "desacoplar" o sistema de equações definido em (10) em dois subsistemas

$$\Delta P(i)/|V(i)| = B' \Delta \theta(i+1) \quad (11)$$

$$\Delta Q(i)/|V(i)| = B'' \Delta |V(i)| \quad (12)$$

onde as matrizes B' ou B'' dependem apenas dos parâmetros da rede e, portanto, permanecem constantes durante o processo iterativo. A notação $\Delta P/|V|$ significa a divisão das componentes de ΔP pelas respectivas componentes de $|V|$.

3.3 - Estrutura das Matrizes.

Para as redes elétricas usuais a matriz Y apresenta elevado grau de esparsidade (mais de 95% dos elementos nulos) possuindo elementos não-nulos apenas na diagonal principal e nas posições correspondentes a conexões físicas entre os nós da rede. Esta matriz não possui estrutura facilmente discernível (tridiagonal, bandas, etc.) po-

rém pode, em geral, ser colocada na forma "quase bloco-diagonal" com relativamente poucos elementos não nulos fora dos blocos diagonais.

As submatrizes H , M , N e L na versão linearizada mostrada em (10), assim como as matrizes B' e B'' em (11) e (12), apresentam estruturas similares a Y exceto pela exclusão de algumas linhas/colunas. Essas matrizes em geral apresentam dominância diagonal exceto em algumas poucas situações de interesse prático. A dominância bloco-diagonal é ainda objeto de estudo havendo indicação, nos casos estudados até o momento, que a mesma pode ser obtida na maioria dos esquemas de decomposição de interesse prático.

3.4 - Solução por Métodos Bloco-Iterativos em Paralelo.

Os sistemas de equações lineares definido em (9), (10), (11) e (12) podem ser resolvidos por qualquer um dos métodos descritos neste artigo. Como esses sistemas representam iterações na solução de um problema não-linear, o vetor de termos independentes (função das incógnitas), pode ser atualizado simultaneamente com o processo iterativo de solução do sistema linear, minimizando desta maneira a troca de informações entre processadores. Desta maneira cada processador resolveria o problema de FP para uma região da rede considerando a influência das demais regiões de forma estática. Após a convergência desses subproblemas, haveria a troca de informações e a influência recíproca das redes seria, então, atualizada.

4. O ACP - ARQUITETURAS E PROGRAMAÇÃO

4.1 - Breve Descrição do ACP [12].

O ACP (Advanced Computer Program) é um multiprocessador desenvolvido no Fermilab orientado fundamentalmente para solucionar problemas na área de física de altas energias que são de difícil tratamento em processadores seqüenciais.

Em particular a reconstrução de eventos a partir de dados obtidos experimentalmente foi a motivação principal para o desenvolvimento do computador. Dado que a reconstrução de eventos é um problema com estrutura inerentemente paralela, na qual os dados de cada evento podem ser processados independentemente, no software de apoio deste computador não se deu ênfase à intercomunicação entre processadores.

O ACP tem arquitetura de tipo Árvore com um hospedeiro (normalmente MicroVax) na raiz e microprocessadores denominados nós, conectados aos

ramos. A partir do hospedeiro a comunicação com os nós é feita através de um "bus" de alta velocidade denominado Branch Bus. Este "bus" está conectado com 16 "crates" VME (barramento assíncrono de 32 bits) através de um módulo de Interface do Branch Bus ao VME (BVI), o qual atua como "master" do VME onde se encontram os nós.

Os microprocessadores utilizados nos nós são Motorola 68020 ou AT&T 32100, com seus respectivos processadores numéricos e 2 Mbytes de memória; a programação é feita com a linguagem FORTRAN 77.

4.2 - Lógica de Programação.

Dado que a resolução de sistemas lineares do tipo $Ax=b$ não pertence à classe de problemas usualmente tratados no ACP, está-se desenvolvendo uma programação específica que leva em consideração a necessidade de transmissão de dados e informações entre os diferentes nós.

Visando o problema da intercomunicação entre os nós, em uma primeira etapa utiliza-se o hospedeiro como gerenciador da informação recebida e enviada aos nós e como analisador dos resultados obtidos em cada um dos nós.

Particionando-se em blocos o sistema $Ax=b$ na forma abaixo:

$$\begin{array}{l}
 \text{Proc.1} \rightarrow \\
 \text{Proc.2} \rightarrow \\
 \vdots \\
 \text{Proc.n} \rightarrow
 \end{array}
 \begin{array}{c}
 \left[\begin{array}{cccc}
 A_{11} & A_{12} & A_{13} & \dots & A_{1n} \\
 \dots & \dots & \dots & \dots & \dots \\
 A_{21} & A_{22} & A_{23} & \dots & A_{2n} \\
 \dots & \dots & \dots & \dots & \dots \\
 \vdots & \vdots & \vdots & \vdots & \vdots \\
 A_{n1} & A_{n2} & \dots & \dots & A_{nn}
 \end{array} \right]
 \begin{array}{c}
 \left[\begin{array}{c}
 x_1 \\
 \vdots \\
 x_2 \\
 \vdots \\
 \vdots \\
 x_n
 \end{array} \right]
 =
 \begin{array}{c}
 \left[\begin{array}{c}
 b_1 \\
 \vdots \\
 b_2 \\
 \vdots \\
 \vdots \\
 b_n
 \end{array} \right]
 \end{array}
 \end{array}$$

utiliza-se um método iterativo do tipo Jacobi por blocos: envia-se um "bloco linha" a cada um dos processadores e obtém-se uma parte da solução. A "solução" é enviada, via hospedeiro, aos outros nós que novamente atualizam os respectivos "subvetores solução".

No esquema abaixo apresenta-se sucintamente as operações realizadas pelo hospedeiro e pelos nós.

————— Hospedeiro —————

```

Entrada de dados
Partição do sistema em blocos
Envio de dados comuns a todos os nós
Envio de "flag" 0 (opção: cálculo da matriz de
iterações)

LOOP

Envio dos "blocos linha" correspondentes a cada
nó

ENDLOOP

Envio de "flag" 1 (opção: início do cálculo ite
rativo)

LOOP

Análise do status do nó (done)
Leitura do subvetor correspondente à "solução"
obtida no nó
Envio da "solução" atualizada para o nó
Início da nova atualização
Análise de erro

ENDLOOP

Armazenamento de resultados

STOP END
  
```

————— Nós —————

```

Análise de "flag"
If (flag = 0) THEN
Cálculo de  $A_{ll}^{-1} A_{lj}$  e  $A_{ll}^{-1} b_l$ 
GO TO ITER
ITER: Cálculo do subvetor
 $x_l := \sum_{j \neq l}^n A_{ll}^{-1} A_{lj} x_j + A_{ll}^{-1} b_l$ 
STATUS DONE
  
```

Em uma segunda etapa estuda-se a opção de utilizar a comunicação direta entre os nós, a qual é possível se os mesmos estiverem no mesmo "crate". O objetivo é diminuir o tempo gasto em comunicações com o hospedeiro.

5. CONCLUSÕES

Este artigo apresenta métodos de solução de sistemas de equações algébricas lineares, usando uma técnica iterativa assíncrona em paralelo, aplicável ao problema de fluxo de potência utilizando a arquitetura paralela do multiprocessador ACP. Esta arquitetura é explorada de maneira a obter a melhor implementação assíncrona de uma versão modificada do método Bloco de Jacobi. Atualmente a equação linearizada da formulação

desacoplada do fluxo de potência ativa (11) está sendo resolvida por este método. O sistema exemplo utilizado é o sistema padrão de 118 barras do IEEE [11]. A implementação de outros métodos de solução do fluxo de potência, como aqueles descritos pelas equações (8)-(10), está sendo considerada, assim como a utilização de um sistema teste com dimensão mais elevada (1000 barras). Resultados preliminares disponíveis atualmente, os quais serão apresentados no Simpósio, são promissores e indicam a existência de compromisso entre a aceleração relativa ("speed up") devida ao número de processadores e o "overhead" em comunicação imposto pela arquitetura do ACP. Assim, é provável que para uma dimensão fixa do problema (dimensão da matriz) exista um número ótimo de processadores e uma partição ótima correspondente.

Agradecimentos: Os autores agradecem ao Prof. Alberto Santoro e ao pessoal do Laboratório de Física Experimental de Altas Energias do Centro Brasileiro de Pesquisas Físicas, pela cooperação e pelo apoio na utilização do ACP.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] G. A. Lyzenga, A. Raefsky & B. H. Hager, "Finite Elements and the Method of Conjugate-Gradients on a Concurrent Processor", Solving Problems on Concurrent Processors, Volume III: Scientific and Engineering Applications, Prentice-Hall, Inc., (1988).
- [2] D. Chazan & W. Miranker, Chaotic Relaxation, Linear Algebra and Appl. 2 (1969), 199-222.
- [3] G. M. Baudet, Asynchronous Iterative Methods for Multiprocessors, J. ACM, 25 (2), April 1978, 226-244.
- [4] R. Varga, Matrix Iterative Analysis, Prentice-Hall, Inc. Englewood Cliffs, N.J., 1962.
- [5] G. H. Golub & C. F. Van Loan, Matrix Computations, Johns Hopkins University Press, Baltimore, 1983.
- [6] D. M. Young, Iterative Solutions of Large Linear Systems, Academic Press, New York, 1971.
- [7] D. Heller, A Survey of Parallel Algorithms in Numerical Linear Algebra, SIAM Review 20(4), Oct. 1978, 740-777.
- [8] EPRI Report EL-3317, Distributed Processing Algorithms and Assignments, Nov. 1983.
- [9] Third SIAM Conference on Parallel Processing for Scientific Computing, Los Angeles, CA, Dec. 1-4, 1987.
- [10] B. Stott, Review of Load-Flow Calculation Methods, "Proc. of the IEEE", 62 (Jul. 1974), 916-929.
- [11] Wallach, Y., Calculations and Programs for Power System Networks, Prentice-Hall, 1986.
- [12] I. Gaines, et al., The ACP Multiprocessor System at Fermilab, Computer Physics Communications 45 (1987), 323-329.