



**Simpósio de Métodos
Numéricos em Engenharia**

25 a 27 de outubro, 2017

ARMAZENAMENTO ESPARSO NO PRÉ-CONDICIONAMENTO PARA O MÉTODO ITERATIVO GMRES, APLICADO À SOLUÇÃO DE FLUXO DE POTÊNCIA

Jhonatan de Oliveira Silva
Departamento de Engenharia Elétrica
Universidade Federal de Mato Grosso
Cuiabá, Brasil
jhonatan.oliveira.ene@gmail.com

Carlos Enrique Portugal Poma
Departamento de Engenharia Elétrica
Universidade Federal de Mato Grosso
Cuiabá, Brasil
portugal@ufmt.br

Vanessa de Oliveira Campos
Instituto de Computação
Universidade Federal de Mato Grosso
Cuiabá, Brasil
vanessa@ic.ufmt.br

Resumo - Este trabalho busca integralizar e aproveitar adequadamente os avanços na área de métodos numéricos, usufruindo de suas características e testando-os no problema de fluxo de potência. Propondo novas estratégias para o armazenamento esparsos no pré-condicionamento, com intuito de melhorar significativamente o desempenho do método iterativo. Com o uso destas estratégias, também deverá ser possível solucionar sistemas lineares relativos ao subproblema linear de fluxo de potência que envolve matrizes mal-condicionadas, próximas da singularidade e/ou indefinidas, decorrentes de cenários de condições operacionais adversas, que comumente os solucionadores diretos têm dificuldades de solucionar.

Palavras-chave—artigo; pré-condicionador; fluxo de potência; armazenamento esparsos.

I. INTRODUÇÃO

O sistema brasileiro de energia elétrica passou por grandes transformações desde o projeto para iluminação da estação central da Estrada de Ferro D. Pedro II, em 1879 [1]. Com uma matriz de energia elétrica baseada em sua grande parte por potencial hidráulico e uma vasta dimensão territorial, com as centrais de geração distantes dos principais centros de consumo, tem-se uma alta complexidade em termos da configuração e tamanho do sistema de transmissão de energia elétrica. Contendo uma diversidade de equipamentos com diferentes características e uma variedade de tipo e período de carga e condições adversas em sua operação, aumentam-se cada vez mais as dificuldades em manter o estado seguro das redes de energia elétrica e seu contínuo funcionamento. Consequentemente, a modelagem e solução das equações matemáticas que

representam esse sistema elétrico de energia tornam-se também mais complexas, exigindo ferramentas que possam garantir maior robustez numérica e eficiência.

O objeto de estudo neste trabalho está associado à solução do subproblema linear do fluxo de potência (1), portanto, uma das etapas do método de Newton Raphson [2]. Apresentam-se neste trabalho métodos e estratégias computacionais para a solução de sistemas lineares, abordando as transformações realizadas nestes sistemas, definidas como pré-condicionamento. Essas transformações são realizadas através de métodos numéricos que garantem baixo número de condicionamento da matriz de coeficiente, portanto maior robustez do solucionador.

$$J * \begin{bmatrix} \Delta \theta^i \\ \Delta V^i \end{bmatrix} = \begin{bmatrix} \Delta P^i \\ \Delta Q^i \end{bmatrix} \quad (1)$$

Associando e adaptando modelos matemáticos e métodos numéricos eficientes com intuito de garantir maior robustez e eficiência computacional, utiliza-se neste trabalho o método iterativo Resíduo Mínimo Generalizado (GMRES) [4], combinado às técnicas de pré-condicionamento baseado no método Doolittle [2] com regra de preenchimento e as técnicas de armazenamento esparsos do método de Zollenkopf [3] aplicado à solução de sistemas lineares, sendo as modificações e extensões deste o objeto central de estudo deste trabalho.

II. MÉTODO GMRES

O método iterativo do Resíduo Mínimo Generalizado, conhecido também como GMRES, baseia a busca da aproximação na condição da norma residual mínima, determinando x_k de tal maneira que sua correspondente

norma Euclidiana do resíduo seja mínima sobre o subespaço de Krylov, cujas bases são ortonormalizadas [2].

Apresenta-se em (2) a equação utilizada para a determinação aproximada de x_k , sendo V uma matriz formada pela expressão (3) que é à base do subespaço de Krylov após sua ortonormalização. Para ortonormalizar os vetores base do subespaço de Krylov $K_k(A, r_0)$, utiliza-se o processo de Arnoldi [4].

$$x_k = x_0 + V \cdot y_k \quad (2)$$

$$[V]_{n \times m} = [V_1^f \ V_2^f \ \dots \ V_k^f] \quad (3)$$

Para ortonormalização do vetor v_{k+1}^i em relação aos outros k vetores $[V_1^f \ V_2^f \ \dots \ V_k^f]$, é utilizada a equação (4), ortonormalizados em iterações anteriores. Para tal processo é aplicado o algoritmo de Arnoldi, conhecido também como método de Gram-Schmidt modificado, onde os vetores $(V_1^f, V_2^f, \dots, V_k^f)$ pertencem ao subespaço original de Krylov $(r_0, A \cdot r_0, A^2 \cdot r_0, \dots, A^k \cdot r_0)$.

$$v_{k+1}^f = \frac{v_{k+1}^i - \sum_{g=1}^k \left[\left[[v_g^f]^T \cdot v_{k+1}^i \right] \cdot v_g^f \right]}{\left\| v_{k+1}^i - \sum_{g=1}^k \left[\left[[v_g^f]^T \cdot v_{k+1}^i \right] \cdot v_g^f \right] \right\|_2} \quad (4)$$

Para satisfazer o teste de convergência, a expressão $\|r_0\|_2 \cdot \hat{e}_1 - H_k \cdot y_k$ deve ser igual à zero, para tanto, os valores dos elementos do vetor y_k devem ser tais que satisfaça a condição, tendendo o resultado da equação a zero (6). O menor valor esperado para a norma do resíduo é próximo ou igual à zero para cada iteração k do GMRES, tornando o problema do tipo mínimo quadrado de pequeno porte (7), que pode ser determinado usando-se as fatorações ortogonais baseadas em rotações no plano ou método de Rotação de Givens [4].

$$\|r_k\|_2 = \|b - A \cdot x_k\|_2 \quad (5)$$

$$\|r_k\|_2 = \|V_{k+1} \cdot [\|r_0\|_2 \cdot \hat{e}_1 - H_k \cdot y_k]\|_2 \quad (6)$$

$$H_k \cdot y_k = \|r_0\|_2 \cdot \hat{e}_1 \quad (7)$$

III. PRÉ-CONDICIONADOR

No princípio o pré-condicionador era utilizado para referenciar as transformações realizadas em um sistema linear com objetivo de diminuir o número de condicionamento da matriz de coeficientes [6], relacionando o menor condicionamento da matriz de coeficientes com a melhora na convergência de um método iterativo [7].

O pré-condicionador P pode ser dividido em duas partes principais, a primeira refere-se ao algoritmo utilizado para montagem de P e, o segundo consiste na regra de preenchimento. Para pré-condicionadores baseados em fatoração incompleta, como o utilizado neste trabalho, normalmente adota-se o algoritmo de fatoração triangular para montar de forma implícita a matriz P , transformando em duas matrizes triangulares L' e U' ($M = L' \cdot U' \approx A$), como apresentado em (8) [8] e [12].

$$P \cdot (L' \cdot U') \cdot P^T \cdot y = P \cdot b, \quad x = P^T \cdot y \quad (8)$$

As matrizes Jacobianas típicas de fluxo de potência são normalmente indefinidas, mal condicionadas e muitas vezes próximas da singularidade, principalmente para sistemas elétricos de maior porte e operando em máximo carregamento [2]. Portanto, torna-se necessário utilizar de estratégias que minimizem ou eliminem as dificuldades relacionadas às características apresentadas, que podem levar a convergência lenta ou falha no processo.

O algoritmo pré-condicionador é dividido em quatro sub-rotinas, o primeiro refere-se ao armazenamento da matriz de coeficiente ou Jacobiana para o estudo de fluxo de potência, subsequente é realizado o reordenamento, que após seu término inicia-se o processo de fatoração triangular Doolittle e, por fim, é efetuada a substituição progressivo-regressiva que obtém o resultado do sistema linear calculado.

A. Armazenamento Esparso

A expansão do sistema elétrico brasileiro e sua interligação elevaram a complexidade e as dificuldades em manter a qualidade do serviço. Dessa forma surge a necessidade de determinar os fluxos por essas redes e algumas outras grandezas elétricas que possibilitam o diagnóstico do estado do sistema analisado. A evolução dos sistemas elétricos, tanto por suas dimensões quanto complexidade, levou há uma necessidade cada vez maior de simuladores computacionais com alta eficiência de processamento de dados. Sendo os sistemas lineares de grande porte e de alta esparsidade, faz-se necessário aplicação de técnicas de armazenamento esparsas.

Desenvolvido com base no método de Zollenkopf [5] e [13], o armazenamento esparsa da matriz Jacobiana é realizado tanto por linha quanto coluna, criando vetores que guardam informações dos elementos diferentes de zero da matriz de coeficientes. Os vetores esparsos possibilitam uma busca rápida aos elementos não nulos da matriz, sendo vantajoso tanto para o armazenamento da matriz de coeficientes como para as matrizes triangulares L e U determinadas no processo de fatoração triangular. Dentre os principais vetores esparsos, destacam-se:

NOZE – quantidade de elementos por linha;

ITAG – índice de coluna dos elementos;

CE – valor dos elementos;

LCOL – posição do primeiro elemento por linha;

LNXT – posição dos elementos, exceto os de LCOL.

Realizado o armazenamento esparsa da matriz de coeficientes, como passo subsequente ao processo de pré-condicionamento, efetua-se o reordenamento da matriz Jacobiana.

B. Reordenamento da Matriz Jacobiana

Com objetivo de diminuir o número de elementos não-nulos da matriz Jacobiana e melhorar a estabilidade numérica durante a fatoração triangular LU, aplica-se a estratégia de reordenamento do tipo Mínimo Grau (Minimum Degree – MD), conhecido como o esquema II de Tinney [9].

O esquema II de Tinney gera o vetor reordenamento NSEQ, sequência numérica que representa a ordem das linhas e/ou colunas que serão permutadas, dando origem as matrizes de permutação P e Q, que uma vez determinadas pode substituir o sistema original por outro equivalente reordenado. Com objetivo de preservar as propriedades da matriz original, aplica-se a permutação simétrica, portanto $Q = P^T$, mantendo estruturalmente simétrica.

O processo de permutação matricial é um tipo de transformação linear, onde se multiplica uma matriz quadrada binária determinada pelo vetor de ordenamento, conhecida como matriz de permutação P, pela matriz de coeficientes. Com objetivo de garantir maior eficiência computacional, não há a realização do produto de matrizes, o processo de permutação é realizado de forma indireta, deslocando linha e coluna ao mesmo tempo.

Algoritmo 1: Algoritmo simplificado de Permutação da Matriz Jacobiana.

```

for i= 1:n
  for j= 1:n
    Passo 1: ii= NSEQ(i)
              jj= NSEQ(j)
    Passo 2: x= LCOL(1,jj)
              y= LCOL(1,jj)+NOZE(1,jj)-1
              while (x ~= (y+1))
    Passo 3: if ii= ITAG2(1,x)
              • Jacobiana Permutada
            end
          end
        end
      end
    end
  
```

Como passo inicial ao processo de permutação da matriz Jacobiana, através do vetor de reordenamento é determinada as novas posições de cada elemento da matriz, armazenado seus respectivos índices de linha e coluna (Passo 1). O segundo passo para permutação é a identificação dos elementos da matriz de coeficientes através dos vetores esparsos. Por fim, como terceiro passo, é armazenada em novos vetores esparsos a matriz Jacobiana permutada.

C. Fatoração Triangular

Neste trabalho utiliza-se o pré-condicionador de fatoração incompleta ILUD(ξ) apresentado em [2] e [10], proveniente de uma versão do algoritmo de eliminação de Gauss conhecido como Doolittle. Uma grande vantagem deste algoritmo é a possibilidade de aplicação da estratégia de preenchimento baseada no erro em ambos os fatores triangulares L e U, de forma eficiente.

O pré-condicionador de fatoração incompleta ILUD(ξ), decompõem a matriz coeficiente em duas matrizes triangulares, L e U, determinados através das equações de Doolittle (10) e (9) respectivamente. As matrizes superior e inferior são mensuradas através da eliminação de Gauss conhecida como Algoritmo Doolittle, que tem como forma generalizada três laços em sequência como em

$$U_{ki} = J_{ki} - \sum_{j=1}^{k-1} [L_{kj} \cdot U_{ji}], \quad i = k, \dots, n \quad (9)$$

$$L_{ik} = \frac{J_{ik} - \sum_{j=1}^{k-1} [L_{ji} \cdot U_{jk}]}{U_{kk}} \quad i = k + 1, \dots, n \quad (10)$$

Algoritmo 2: Algoritmo da Fatoração Triangular LU.

```

for i=1:n
  Passo 1: for j= 1:(NOZEL(i))
            j= ITAGL(z,i)
            h= PCU(j)
  Passo 2: for k= 1:(NOZEU(i))
            x= L*CEU(h)
  Passo 3: L(k)= L(k)-L(h)*U(PU(z,i))
            U(ITAGU(h))= U(ITAGU(h)) - x
            h= (h+1)
            • Atualização ITAGU e NOZEU
          end
        end
  Passo 4: for z= 1:(NOZEU(i))
            j= ITAGL(j,i)
            h= PCL(j)
  Passo 5: for k= 1:NOZEL(j)
            k = ITAGL(hh)
  Passo 6: L(k)= L(k)-L(h)*U(PU(z,i))
            L(k)= L(k)/DEU(i)
            • Atualização ITAGL e NOZEL
          end
        end
  Passo 7: for z= 1:NOZEL(i)
            • NOZEL; IATGL; CEL; LNXTL
          end
  Passo 8: for z= 1:NOZEU(i)
            • NOZEU; IATGU; CEU; LNXTU
          end
        end
  
```

A fatoração triangular com a técnica de esparsidade é iniciada calculando os elementos da matriz triangular superior. Os elementos da matriz Jacobiana são salvos na variável U que durante a fatoração é utilizada para

armazenar os elementos por linha de cada iteração i da própria matriz U . São vinculados ao laço j os elementos diferentes de zero da matriz L (Passo 1), e associam-se os elementos não nulos de U ao laço k (Passo 2).

Definidos os laços através dos vetores esparsos para determinação dos elementos da matriz triangular U , há aplicação da equação de Doolittle (Passo 3). Sendo este um dos processos de maior exigência computacional, a técnica de esparsidade elimina a necessidade de trabalhar com elementos nulos e fazendo buscas diretas aos elementos diferentes de zero.

A determinação da matriz triangular inferior L é realizada por meio da equação de Doolittle, sendo aplicado ao processo a estratégia de vinculação dos laços jk , aos elementos diferentes de zero das matrizes L e U (Passo 4, 5 e 6). Para casos que $J_{ik} = 0$ há a necessidade de atualizar os vetores esparsos ITAGL e NOZEL após o cálculo das variáveis da relativa matriz, em casos que o somatório é igual a zero os vetores esparsos de índice e quantidades de elementos por coluna serão iguais da matriz Jacobiana.

Por fim, para cada iteração i é realizado o armazenamento da linha (U) e coluna (L) em vetores esparsos das matrizes triangulares superior e inferior (Passo 7 e 8) aplicando a regra de preenchimento, expostas em (11) e (12) na devida ordem. Quando o erro associado a variável analisada for menor que a tolerância estipulada ξ , o elemento é descartado.

$$U_{ik} = \begin{cases} 0, & \|R(U_{ik})\| < \xi \\ m_k, & \|R(U_{ik})\| \geq \xi \end{cases}, \quad k = (i, \dots, n) \quad (11)$$

$$L_{ki} = \begin{cases} 0, & \|R(L_{ki})\| < \xi \\ m_k, & \|R(L_{ki})\| \geq \xi \end{cases}, \quad i = (1 + i, \dots, n) \quad (12)$$

D. Substituição Progressiva e Regressiva

Finalizado o processo de determinação das matrizes triangulares pelo método de Doolittle modificado, onde é adotada a técnica de armazenamento esparsa e aplicada a vinculação dos laços do algoritmo aos elementos não nulos das matrizes, executa-se o processo de substituição progressiva e regressiva. Como parte do pré-condicionador, é aplicada a estratégia de transformação do sistema linear em outro equivalente, como em (13), que para determinação da solução executa-se primeiramente o processo de substituição progressiva expresso em (14).

$$A * x = B \rightarrow L * U * x = B \rightarrow L * Z = B \quad (13)$$

$$Z_i = B_i - \sum_{j=1}^{i-1} [L_{ij} * Z_j] \quad (14)$$

Denominado de progressivo em referência a sequência lógica de processamento, a substituição ocorre em dois laços, o primeiro (i) para indicar o elemento de Z que está sendo calculado e j para o somatório. As matrizes

triangulares são armazenadas aplicando-se a técnica de esparsidade, portanto, assim como na fatoração triangular, associa-se ao laço j os elementos diferentes de zero da matriz L , pois o produto dentro do somatório torna-se desnecessário para elementos nulos.

Como ultima sub-rotina para determina a matriz de incógnitas x , temos a substituição regressiva, apresentada em (15). Definida como substituição regressiva em referência ao sentido da sequência de cálculo, o processo ocorre com o primeiro laço variando de $(n - 1)$ até 1, ou seja, uma variação decrescente.

$$L * Z = B \rightarrow Z = U * x \quad (15)$$

$$X_i = \frac{Z_i - \sum_{j=i+1}^n [U_{ij} * X_j]}{U_{ii}} \quad (16)$$

Tem-se em (16) a representação do processo de substituição regressiva, sendo empregados dois laços, o primeiro (i) para indicar o elemento de x que está sendo calculado e j para o somatório. Assim como na substituição progressiva, associa-se ao laço j os elementos diferentes de zero da matriz U .

Finalizado o processo de substituição tem-se o resultado do sistema linear e, o término do pré-condicionador para o método GMRES aplicado ao estudo de fluxo de potência.

IV. SIMULAÇÕES

Devido à complexidade, características do problema estudado e a aplicação de diversos métodos computacionais no programa para solução do fluxo de potência, é utilizado a linguagem interpretada MatLab 7.12.0 R2011a (MATrix LABoratory). O software é baseado em estudos matemáticos de matrizes, sendo uma linguagem de alto nível que possibilita maior facilidade nas implementações computacionais de equações numéricas.

Através do programa Fluxo de Potência desenvolvido neste trabalho é possível alterar o estado do sistema elétrico, com intuito de avaliar a robustez dos métodos computacionais. Para isso, são realizados acréscimos de carga ao sistema simulado, aumentando o nível de carregamento, que por consequência eleva o número de condicionamento da matriz Jacobiana tornando mais difícil a solução do sistema linear [10].

São simulados cinco sistemas elétricos de potência, o primeiro IEEE 30 sobrecarregado, representa uma aproximação simplificada de parte do sistema elétrico de potência americano em dezembro de 1961, possuindo 30 barras, 2 geradores e 4 condensadores síncronos.

Outro sistema simulado é IEEE 118 sobrecarregado, representa uma aproximação simplificada de parte do sistema elétrico de potência americano em dezembro de 1962. Este sistema contém 118 barras, 19 geradores e 177 ramos. Utiliza-se também uma configuração do sistema interligado brasileiro com 2256 barras, 3508 ramos e 200 geradores, e casos de teste desenvolvido pela IEEE - Test

Systems Task Force em 1993, sistemas IEEE 145 e 300 barras.

V. RESULTADOS

Com intuito de avaliar o desempenho do algoritmo desenvolvido neste trabalho, apresentam-se na Tabela I as simulações dos pré-condicionadores ILUD e CROUT [11], uma versão do método de eliminação de Gauss, associados ao solucionador GMRES. Foram utilizados os sistemas IEEE 30, 118, 145 e 300 barras, e o sistema brasileiro JBrasil2256 de 2256 barras.

Mensurado em segundos o tempo de simulação, os resultados dos pré-condicionadores foram divididos em duas partes, tempo de processamento da solução do sistema linear e a duração de todo o processo de solução do Fluxo de Potência. Nota-se na Tabela I uma relativa vantagem do método Crout para a solução do sistema linear do IEEE 30 e 118 barras sobrecarregados, porém havendo uma aproximação no tempo total, que se deve a diminuição da quantidade de iterações do Fluxo de Potência, demonstrando, portanto, maior robustez do algoritmo ILUD para sistemas mal condicionados.

Tabela I: TEMPO DE SIMULAÇÃO DOS PRÉ-CONDICIONADORES.

Sistema Elétrico de Potência	Pré-condicionador			
	ILUD		CROUT	
	Ax=B	Total	Ax=B	Total
IEEE 30	0,0095	0,149	0,0002	0,0051
IEEE 118	0,041	0,5451	0,001	0,0142
IEEE 145	0,0858	0,673	0,0026	0,0188
IEEE 300	0,1067	0,7462	0,0052	0,0372
JBrasil2256	1,0962	2,1972	0,1085	0,2397

^a. Tempo computacional em segundos.

O pré-condicionador ILUD apresenta maior eficiência computacional quanto maior o sistema elétrico de potência simulado, sendo possível observar analisando a comparação entre os tempos de ILUD pelo método de CROUT para cada sistema simulado, como apresentado na Tabela II. Para o sistema IEEE 118, o método ILUD apresenta um tempo de simulação total de 38 vezes o tempo do método CROUT, porém, para o sistema JBrasil2256 essa relação reduz para aproximadamente 9,16.

Tabela II: RELAÇÃO DO TEMPO DE SIMULAÇÃO DOS PRÉ-CONDICIONADORES ILUD e CROUT.

Sistema Elétrico de Potência	ILUD/CROUT	
	Ax=B	Total
IEEE 30	45,2380952	28,7
IEEE 118	40,1960784	38,3603096
IEEE 145	32,6235741	35,6132275
IEEE 300	20,6879845	20,0483611
JBrasil2256	10,1022947	9,16645807

Com intuito de avaliar o desempenho do pré-condicionador desenvolvido neste trabalho em uma

linguagem de programação compilada, foi realizada a conversão para a linguagem Fortran do solucionador do sistema linear, armazenamento esparsa, processo de fatoração triangular e substituição. Realizou-se a simulação do sistema IEEE 300 barras e obteve-se uma melhora significativa, sendo a relação entre ILUD no Fortran pelo Matlab de 8,3 vezes e, redução para aproximadamente 2,5 vezes quando comparado ILUD Fortran com Crout MatLab.

Para a simulação do algoritmo de CROUT foi utilizada a função 'ilu' do MatLab, desta forma, diferente do algoritmo desenvolvido neste trabalho que são interpretados e passam pelo processo de montagem (linguagem de montagem), a função 'ilu' é compilada diretamente. Portanto, as simulações apresentadas neste trabalho para fins comparativos entre CROUT e ILUD não podem ser consideradas igualitárias, mesmo para a comparação ILUD Fortran e Crout MatLab.

Tabela III: PROCESSO DE FATORAÇÃO TRIANGULAR DOOLITTLE E SUBSTITUIÇÃO, COM E SEM TÉCNICA DE ESPARCIDADE.

Sistemas Elétricos de Potência	Com Técnica de Esparcidade		Sem Técnica de Esparcidade	
	Tempo	Mflops	Tempo	Mflops
IEEE 30	0,0122	0,02	0,0152	0,1421
IEEE 300	0,1635	0,0459	3,9674	143,81
JBrasil2256	1,258	0,382	1623	6,12E+04

Apresenta-se na Tabela III o comparativo entre a fatoração triangular e substituição, utilizando a técnica de esparsidade e o algoritmo Doolittle. Observa-se uma melhora significativa no desempenho do processo de fatoração triangular e substituição quando empregada a técnica de esparsidade. Para a simulação do sistema JBrasil2256, temos uma disparidade muito considerável, enquanto o algoritmo esparsa executou em 1,258 segundos, sem a técnica de esparsidade processou em aproximadamente 27 minutos. Destaca-se também, que quanto maior o sistema elétrico de energia simulado, maior a eficiência do algoritmo com esparsidade, isso se deve a característica de o algoritmo não trabalhar com elementos nulos e a relação direta entre o aumento da esparsidade da matriz Jacobiana com o tamanho do sistema.

VI. CONCLUSÕES

A expansão do sistema elétrico brasileiro e sua interligação elevaram a complexidade e as dificuldades em manter a qualidade do serviço. Dessa forma surge a necessidade de determinar os fluxos por essas redes e algumas outras grandezas elétricas que possibilitam o diagnóstico do estado do sistema analisado. A evolução dos sistemas elétricos, tanto por suas dimensões quanto complexidade, levou a uma necessidade cada vez maior de simuladores computacionais com alta eficiência de processamento de dados e alta confiabilidade. Sendo os sistemas lineares de grande porte e esparsos, faz se

necessário que a matriz de coeficientes tenha o menor condicionamento para melhor convergência do método iterativo.

Este artigo propõe a aplicação do método de armazenamento esparsos de Zollenkopf ao pré-condicionador ILUD, associando e adaptando os métodos de forma a garantir maior eficiência computacional e robustez. A eficiência computacional do método ILUD tende aumentar na medida em que os sistemas elétricos de energia simulados aumentam, característica de grande relevância para o estudo de fluxo de potência, devido aos aspectos dos sistemas de energia elétrica. Essa melhora também se justifica pelas mudanças nas características e propriedades espectrais da matriz Jacobiana, considerada uma das principais dificuldades pertinentes ao problema estudado, que podem dificultar o processo de solução.

Considerado atualmente uma referência na fatoração incompleta, o algoritmo de CROUT derivado do método de eliminação de Gauss, apresentado em [11], apresenta um tempo de processamento superior para todos os sistemas simulados, entretanto cabem algumas observações. O pré-condicionador ILUD consegue reduzir o número de interações do solucionador, demonstrando maior robustez do ILUD. Comprova-se também, uma melhora significativa no desempenho computacional do pré-condicionador desenvolvido neste trabalho, conforme o sistema simulado aumenta, havendo uma aproximação ao tempo do método de Crout e sugerindo que para sistemas maiores podem se igualar ou até mesmo superar.

AGRADECIMENTOS

Agradecemos à Fundação de Amparo à Pesquisa do Estado de Mato Grosso (FAPEMAT) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo suporte financeiro, sem o qual não seria possível a realização desta pesquisa.

REFERÊNCIAS

- [1] Cabral, Ligia M. M.; Cachapuz, Paulo B. B.; Lamarão, Sergio T. N. Panorama do Setor de Energia Elétrica no Brasil. 1 ed. Rio de Janeiro: Centro da Memória da Eletricidade no Brasil, 1988.
- [2] POMA, Carlos Enrique Portugal. Um Solucionador Iterativo Para Sistemas-Lineares: Aplicação no Problema do Fluxo de Carga. Rio de Janeiro, 2010. Dissertação (Doutorado em Engenharia Elétrica) – PUC-Rio, 2010.
- [3] ZOLLENKOPF, K. Bi-Factorisation Basic Computational Algorithm and Programming Techniques, a paper from Large Sparse Sets of Linear Equations, edited by J. K. Reid, Academic Press, pp. 75-96, 1971.
- [4] SAAD, Y.; SCHULTZ, M. H. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. pp. SIAM J. Sci. Stat. Comput. 7, p. 856, 1986.
- [5] ZOLLENKOPF, K. Bi-Factorisation Basic Computational Algorithm and Programming Techniques, a paper from Large Sparse Sets of Linear Equations, edited by J. K. Reid, Academic Press, pp. 75-96, 1971.
- [6] TURING, A. M. Rounding-off errors in matrix processes. Q. J. Mech. Appl. Math. 1, p. 287, 1948.

- [7] EVANS, D. J. The use of pre-conditioning in iterative methods for solving linear systems with symmetric positive definite matrices. J. Inst. Math. Its Appl. 4, p. 295, 1968.
- [8] CHEN, K. Matrix Preconditioning Techniques and Applications. Cambridge Monographs on Applied and Computational Mathematics, 2005.
- [9] TINNEY, W. Direct solutions of sparse network equations by optimally ordered triangular factorization. Proceedings of the IEEE, Volume: 55, Issue: 11, pp. 1801- 1809, 1967.
- [10] POMA, Carlos E. Portugal, PRADA, Ricardo B., PESSANHA, José Eduardo O., Error-Based ILU Preconditioner for the Solution of Linear Equations. IEEE Transactions On Power Systems, Volume: 32, Nº. 1, pp. 326 – 332, 2017.
- [11] LI, Na; SAAD, Yousef, Crout Versions Of ILU Factorization With Pivoting For Sparse Symmetric Matrices, Electronic Transactions on Numerical Analysis – Kent State University, Volume 20, pp. 75-85, 2005.
- [12] SAAD, Y. Iterative Methods for Sparse Linear Systems, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, 2003.
- [13] CASTRO, Carlos A. C Jr. Análise Matricial de Sistemas de Energia Elétrica. Departamento de Sistemas e Energia – Faculdade de Engenharia Elétrica e Computação/UNICAMP, 2010.