

UNIVERSIDADE FEDERAL DE ITAJUBÁ  
INSTITUTO DE SISTEMAS ELÉTRICOS E ENERGIA

Solução do Problema de Fluxo de Potência no Plano Complexo Via  
Métodos de Gradientes Conjugados

André Soares da Silva

Itajubá, Dezembro de 2022

**UNIVERSIDADE FEDERAL DE ITAJUBÁ  
INSTITUTO DE SISTEMAS ELÉTRICOS E ENERGIA**

**André Soares da Silva**

**Solução do Problema de Fluxo de Potência no Plano Complexo Via  
Métodos de Gradientes Conjugados**

Dissertação de mestrado submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Itajubá como parte dos requisitos para obtenção do Título de Mestre em Ciências em Engenharia Elétrica.

**Orientador: Prof. Robson Pires**

**Itajubá, 2022**

## Agradecimentos

Agradeço primeiramente a Deus, por possibilitar a minha vida e a família por ter me dado suporte perene em todos os aspectos. Não é possível fazer um agradecimento justo, qualquer tentativa diminui o valor das pessoas envolvidas, por outro lado, seria injusto não registrar algumas pessoas especiais, que participaram de maneira mais estreita neste trabalho e na minha vida. Agradeço então, ao meu orientador e amigo, Robson Celso Pires, que desde a graduação, vem me orientando com grande paciência, tolerância e resiliência, na minha jornada em busca do conhecimento. Agradeço também ao meu grande amigo de longa data Rafael Álvares Bragança, que sempre me apoiou e ajudou em todos os âmbitos da minha vida pessoal e acadêmica. Agradeço a minha companheira de jornada, Michely Bittencourt dos Santos, que durante todos os anos de nossa união, me incentivou sobretudo nos momentos mais difíceis e desafiadores da minha vida. Agradeço à todas as pessoas que de forma direta ou indireta, ajudaram neste trabalho. Sem a presença e grandes qualidades de espírito dessas pessoas, a iniciativa e continuidade de todos os meus projetos e sonhos seriam prejudicados ou impossibilitados.

# Resumo

A literatura técnica em análise numérica oferta várias alternativas de métodos iterativos para a solução de problemas de otimização não linear aplicados à indústria do setor de energia. No entanto, em se tratando de métodos iterativos aplicados ao fluxo de potência não linear, o método iterativo de Newton-Raphson e suas variações são praticamente unanimidade. A proposta de trabalho desta dissertação de mestrado tem por objetivo investigar o desempenho dos Métodos de gradientes Bi-Conjugados Complexos (CBiCG) não lineares em sua aplicação à problemas de fluxo de potência, em comparação com a formulação clássica, utilizando o método de Newton-Raphson em coordenadas polares no domínio dos números reais. Estes métodos, no entanto, foram implementados no domínio complexo por meio do uso do cálculo de Wirtinger Generalizado e a extensão das séries de Taylor ao domínio complexo. Como principais motivações para essa abordagem tem-se que os métodos de gradiente conjugado são de primeira ordem e não necessitam da fatoração da matriz jacobiana como se faz no tradicional método de Newton-Raphson, reduzindo desta forma o esforço computacional necessário para se obter a solução iterativa, além disso, em vista da complexidade computacional atual, tem-se notado uma tendência que a arquitetura dos processadores incorporem a SIMD (Single Instruction, Multiple Data), que refere-se a um conjunto de operações para manipulação eficiente de uma grande quantidade de dados em paralelo, usando um processador vetorial ou matricial, adequados à álgebra de números complexos.

**Palavras chave:** Gradiente Conjugado, Plano complexo, Fluxo de Potência, Wirtinger, Newton-Raphson.

# Abstract

The technical literature in numerical analysis offers several alternatives of iterative methods for the solution of nonlinear optimization problems applied to the energy industry. However, when it comes to iterative methods applied to nonlinear power flow, the Newton-Raphson iterative method and its variations are practically unanimity. The work proposal of this master's thesis aims to investigate the performance of nonlinear Complex Bi-Conjugate Gradient Methods (CBiCG) in their application to power flow problems, in comparison with the classical formulation, using the Newton- Raphson in polar coordinates in the domain of real numbers. These methods, however, were implemented in the complex domain through the use of the Generalized Wirtinger calculus and the extension of Taylor series to the complex domain. The main motivations for this approach are that the conjugate gradient methods are first order and do not require factoring the Jacobian matrix as is done in the traditional Newton-Raphson method, thus reducing the computational effort required to obtain the solution. In addition, in view of the current computational complexity, there has been a tendency for processor architecture to incorporate SIMD (Single Instruction, Multiple Data), which refers to a set of operations for efficiently handling a large amount of data in parallel, using a vector or matrix processor, suitable for the algebra of complex numbers.

**Keywords:** Gradient Conjugate, Complex Domain, Power Flow, Wirtinger, Newton-Raphson.

# Lista de Figuras

Figura 1.1 - Estrutura Geral dos Sistemas de Potência . . . . .	13
Figura 1.2 - Sistema Elétrico . . . . .	15
Figura 2.1 - Gráfico de contorno de uma função real de variáveis complexas . . . . .	20
Figura 3.1 - Variáveis da barra $k$ . . . . .	26
Figura 3.2 - Convenção da entrada e saída de potência . . . . .	27
Figura 3.3 - Injeção líquida de potência na barra $k$ . . . . .	27
Figura 3.4 - Transformador ideal . . . . .	28
Figura 3.5 - Modelo do Fluxo . . . . .	29
Figura 3.6 - Matriz de admitância de um sistema do SIN de 9899 barras . . . . .	32
Figura 4.1 - Jacobiana complexa de um sistema de 9899 barras. . . . .	46
Figura 4.2 - Jacobiana complexa do sistema europeu de 13659 barras. . . . .	46
Figura 5.1 - Sequência de direções e passos de descida . . . . .	51
Figura 5.2 - Comportamento de zig zag ortogonal do método de Máxima Descida . . . . .	53
Figura 5.3 - Ordenação CAMD do sistema de 9899 barras . . . . .	60
Figura 6.1 - Erros percentuais de magnitude de tensão . . . . .	63
Figura 6.2 - Tempo computacional em segundos . . . . .	64
Figura 6.3 - CBiCG com preconditionador . . . . .	66
Figura 6.4 - CBiCG-ILU preconditionado incluindo sistemas de 145 e 3120 barras . . . . .	67
Figura 6.5 - Tempo computacional em segundos do sistema reduzido . . . . .	68
Figura 6.6 - Tempo computacional em segundos do sistema reduzido . . . . .	69

# Lista de Tabelas

Tabela 1 - Tabela de Iterações CBiCG x MATPOWER . . . . .	65
Tabela 2 - Tabela de Iterações CBiCG-EPF x MATPOWER . . . . .	70

## Lista de Abreviaturas e Siglas

<b>CG</b>	<i>Conjugate Gradient</i>
<b>BiCG</b>	<i>Biconjugate Gradient</i>
<b>CBICG</b>	<i>Complex Biconjugate Gradient</i>
<b>CBICG-ILU</b>	<i>Gradiente Bi-conjugado Complexo com fatoração LU incompleta</i>
<b>CBICG-ILU<sub>Schur</sub></b>	<i>Gradiente Bi-conjugado Complexo aplicado ao sistema reduzido via complemento de Schur</i>
<b>CBICG-EPF</b>	<i>Gradiente Bi-conjugado aplicado ao fluxo de potência exato</i>
<b>CV</b>	<i>Complex-Valued</i>
<b>CV-NRPF</b>	<i>Complex-Valued Newton-Raphson Power Flow</i>
<b>CV-NRP</b>	<i>Complex-Valued Newton-Raphson Power Flow Polar Coordinates</i>
<b>CV-NRR</b>	<i>Complex-Valued Newton-Raphson Power Flow Rectangular Coordinates</i>
<b>CV-PFA</b>	<i>Complex-Valued Power Flow Analysis</i>
<b>FACTS</b>	<i>Flexible AC Transmissions System</i>
<b>AMD</b>	<i>Approximate Minimum Degree</i>
<b>CAMD</b>	<i>Column Approximate Minimum Degree</i>
<b>OLTC</b>	<i>On-Load-Tap-Changer</i>
<b>PSTs</b>	<i>Phase-shifting transformers</i>

# Sumário

<b>1</b>	<b>Introdução</b>	<b>11</b>
1.1	Objetivos . . . . .	12
1.2	Estrutura Geral dos Sistemas de Potência . . . . .	13
1.3	Motivação do Desenvolvimento em Domínio Complexo . . . . .	16
1.4	Apresentação e Descrição da Estrutura do Documento . . . . .	16
<b>2</b>	<b>Fundamentação Matemática</b>	<b>17</b>
2.1	Funções analíticas e não analíticas . . . . .	17
2.2	Cálculos de Wirtinger . . . . .	18
2.3	Cálculo de Wirtinger Generalizado . . . . .	21
2.3.1	Diferenciais Complexas . . . . .	21
2.3.2	Identificação das Diferenciais Complexas . . . . .	23
2.3.3	Matrizes Jacobianas Complexas . . . . .	23
2.4	Complemento de Schur e Operador Swap . . . . .	24
2.4.1	Operador Swap . . . . .	24
2.4.2	Complemento de Schur . . . . .	25
<b>3</b>	<b>Modelo do fluxo de potência</b>	<b>26</b>
3.0.1	Convenção para o sentido das injeções de potência . . . . .	26
3.1	Transformador ideal . . . . .	28
3.2	Equações Básicas do Modelo . . . . .	28
3.2.1	Matriz de Admitância Nodal . . . . .	30
3.3	Tipos de Barras . . . . .	32
3.3.1	Barra <i>Slack</i> ou Barra de Referência . . . . .	32
3.3.2	Barra PQ . . . . .	32
3.3.3	Barra PV . . . . .	33
3.3.4	Barra PQV . . . . .	33
3.4	Fluxo de Potência em domínio Complexo por Ramos . . . . .	35
3.4.1	Equações de <i>mismatches</i> para o <i>PST</i> . . . . .	37
3.4.2	Solução Iterativa - Algoritmo de Newton-Raphson . . . . .	38
<b>4</b>	<b>Formulação Matricial do Fluxo de Potência</b>	<b>39</b>

4.1	Domínio Real . . . . .	39
4.2	Domínio Complexo . . . . .	41
4.2.1	Blocos de Derivadas da Jacobiana Complexa . . . . .	43
4.2.2	Complemento de Schur Aplicado ao Fluxo de Potência . . . . .	47
4.2.3	Formulação Matricial do Fluxo de Potência Exato . . . . .	48
<b>5</b>	<b>Métodos Iterativos de Solução Numérica:</b>	<b>49</b>
5.1	Métodos de Busca Linear Descendente . . . . .	49
5.1.1	Métodos de Busca Linear Descendente de Primeira Ordem . . . . .	51
5.1.2	Condições de Melhoria de Convergência . . . . .	52
5.2	Método de Gradientes Conjugados (CG) . . . . .	53
5.2.1	Direções Mutuamente Conjugadas . . . . .	54
5.2.2	Direções de Fletcher-Reeves . . . . .	54
5.3	Método do Gradiente Bi-Conjugado (BiCG) . . . . .	55
5.3.1	Condicionamento Numérico e Convergência do CBiCG . . . . .	57
5.4	Precondicionadores . . . . .	58
5.5	Efeito da Ordenação . . . . .	59
5.6	Algoritmos Desenvolvidos em MATLAB . . . . .	60
<b>6</b>	<b>Apresentação dos Resultados</b>	<b>61</b>
6.1	Validação do Fluxo em domínio Complexo . . . . .	62
6.1.1	Validação dos Módulos de Tensão e Tempo Computacional . . . . .	62
6.1.2	Número de iterações: CBiCG x MATPOWER . . . . .	65
6.1.3	CBiCG Precondicionado . . . . .	65
6.1.4	CBiCG-ILU Reduzido Via Complemento de Schur . . . . .	68
6.1.5	Fluxo de potência exato CBiCG-EPF . . . . .	69
<b>7</b>	<b>Conclusão</b>	<b>71</b>
<b>8</b>	<b>Trabalhos Futuros</b>	<b>72</b>
	<b>Referências Bibliográficas</b>	<b>73</b>

# 1 Introdução

Sistemas não lineares de equações em domínio complexo são frequentemente encontrados em matemática aplicada como por exemplo, sistemas de energia, processamento de sinais, teoria de controle, redes neurais etc. A solução desses problemas geralmente requer uma aproximação de primeira ou segunda ordem dessas funções não lineares para gerar um novo passo ou direção de busca e alcançar a solução de forma iterativa. No entanto, tais métodos não podem ser aplicados a funções reais de variáveis complexas porque não são necessariamente analíticas, ou seja, não são diferenciáveis e conseqüentemente suas expansões em séries de Taylor não existem. Para contornar esse problema, a função não linear é geralmente redefinida como uma função de suas partes real e imaginária, de modo que os métodos padrões possam ser aplicados. Embora não seja amplamente conhecido, é possível construir uma expansão dessas funções não lineares de variáveis complexas observando que elas podem ser analíticas em seu argumento e seu complexo conjugado. Esta propriedade reside no fato de que se uma função é analítica no espaço gerado por  $\Re\{x\}$  e  $\Im\{x\}$  em  $\mathbb{R}$ , ela também é analítica no espaço gerado por  $x$  e  $x^*$  em  $\mathbb{C}$ . Dada portanto a possibilidade de expansão complexa da série de Taylor e do uso do cálculo de Wirtinger esse trabalho tem como objetivo investigar o desempenho do método Gradiente Bi-Conjugado Complexo (CBiCG) quando forem aplicados ao problema de fluxo de potência e implementados no domínio complexo. As principais motivações dessa abordagem são:

- i) A direção de busca do máximo ou mínimo da função objetivo ou custo é previamente conhecida quando o problema de otimização é desenvolvido no domínio complexo reduzindo, portanto, o esforço computacional.
- ii) O uso dos métodos de gradiente conjugado aplicados ao problema de fluxo de potência não requer a fatoração da matriz Jacobiana o que também se traduz em ganho de desempenho computacional.
- iii) As facilidades de modelagem de dispositivos *Flexible AC Transmissions System* (FACTS) necessários à operacionalização de redes ativas inteligentes (SMART GRID).

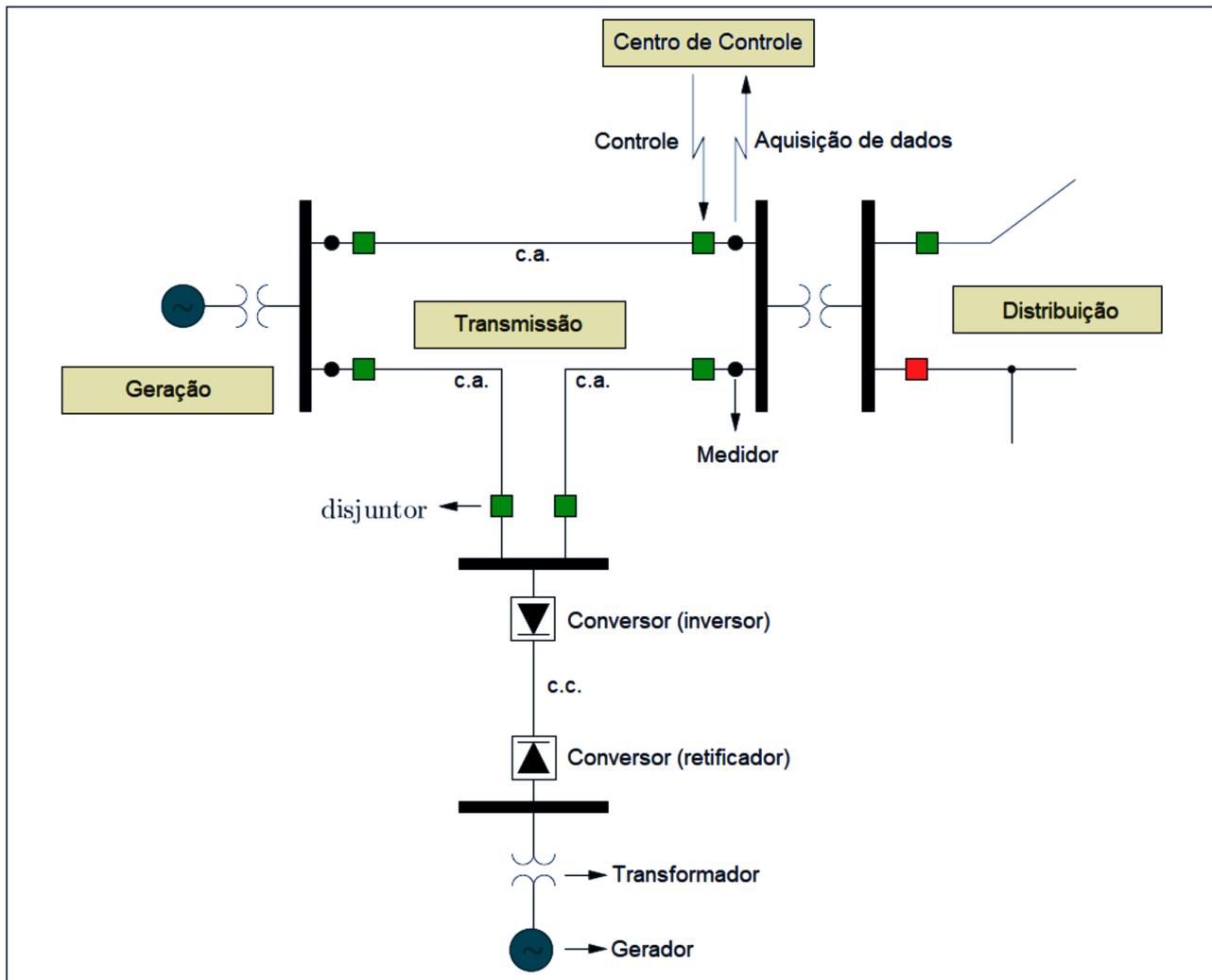
## 1.1 Objetivos

Tipicamente os métodos numéricos são implementados no domínio real porém as aplicações das leis pertinentes ao fluxo de potência e sua modelagem em sistemas AC geram equações e modelos que são naturalmente concebidos em domínio complexo. Portanto, as formulações em domínio real do problema de fluxo de potência fogem do ambiente adequado para modelagem do fasor de tensão e de corrente, que foram introduzidas por Steinmetz [1], [2]. Para contornar essa dificuldade, algoritmos iterativos e não iterativos realizados no domínio complexo foram desenvolvidos recentemente. Devido à crescente inserção de geração renovável o sistema elétrico encaminha-se para um sistema feito de redes híbridas, permeadas por transmissão em corrente alternada, transmissão em corrente contínua e fontes renováveis de geração de energia como eólica e fotovoltaica. Assim sendo, há uma grande e urgente necessidade do desenvolvimento de modelos que permitam análises computacionalmente menos dispendiosas e robustas desse tipo de sistema. Para isso pretende-se, neste trabalho, desenvolver métodos baseados em Gradiente Conjugado em domínio complexo analisando o fluxo de potência sob essa perspectiva e fazendo comparações com o método tradicional de Newton-Raphson em domínio real e complexo em coordenadas polares, deste modo, evidenciando as diferenças e vantagens entre os dois domínios e formulações. Estas análises foram feitas através de resultados obtidos com algoritmos implementados no *software MATLAB* e validados utilizando o *software MATPOWER*, aplicados a sistemas *benchmark* de grande porte disponíveis no *MATPOWER*, como o caso de 13659 barras.

## 1.2 Estrutura Geral dos Sistemas de Potência

Para resolver problemas cada vez mais complexos das redes elétrica, oportuniza-se desenvolver ferramentas matemáticas mais aderentes aos desafios que as tendências evolutivas prenunciam. Em problemas de engenharia, frequentemente tem-se interesse na otimização de parâmetros que recaem em sistemas de equações algébricas com grandes dimensões. No âmbito do setor elétrico, depara-se com sistemas de equações algébricas de muito grande porte naturalmente derivados no plano complexo. Portanto é imprescindível um ferramental capaz de lidar com diferenciação de funções matriciais no domínio complexo. A Fig. 1.1 enseja uma reflexão sobre as tendências presentes nas redes elétricas cada vez mais digitalizadas.

Figura 1.1: Estrutura Geral dos Sistemas de Potência



Fonte: Próprio Autor

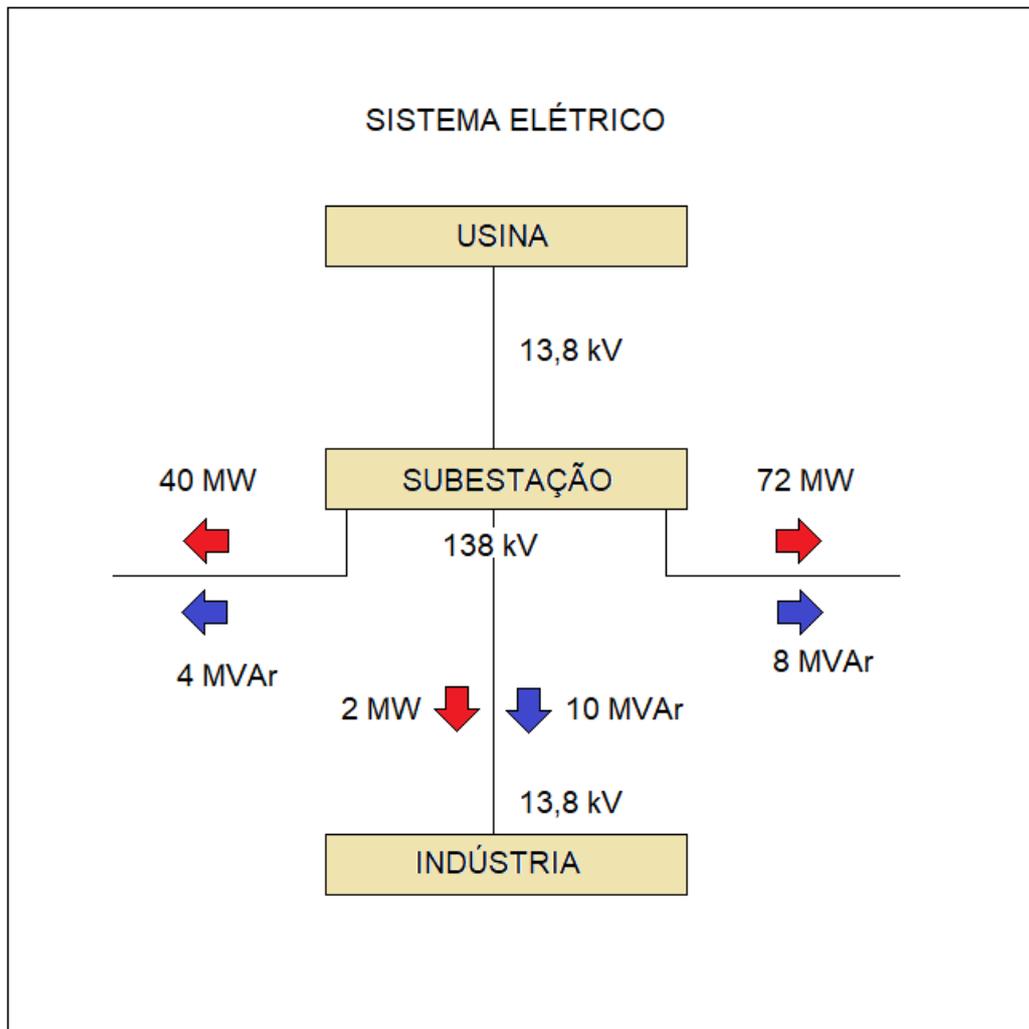
Muito provavelmente, o aplicativo de análise de redes referenciado usualmente como fluxo

de potência ou fluxo de carga, seja a ferramenta mais utilizada no dia a dia das empresas prestadoras de serviços de energia elétrica. Ela permite a obtenção das condições operativas (tensões, fluxos de potência etc.) de uma rede elétrica em função de sua topologia e dos níveis de demanda e geração de energia. A análise do fluxo de carga pode ser feita de forma estática ou dinâmica:

- Modelagem Estática: A rede é representada por um conjunto de equações e inequações algébricas e é analisada em regime permanente.
- Modelagem Dinâmica: A rede é representada por um conjunto de equações e inequações diferenciais e algébricas e é analisada considerando vários aspectos de interesse sobre a dinâmica do sistema.

As aplicações de ambas as metodologias objetivam reproduzir o estado de operação da rede após a ocorrência de contingências. Assim, permite-se detectar as violações de limites de operação, propor ações de controle preventivo, realizar o planejamento da expansão da rede elétrica para atender a demanda futura, entre outros. Associado a isso, propor a inclusão de componentes de redes, monitorar o estado operativo do sistema em tempo real, subsidiar os estudos de estabilidade transitória, formular as ações de controle frente às falhas mais recorrentes de origem interna (curto-circuitos) ou externa, como por exemplo: desligamentos causados por descargas atmosféricas. A Fig. 1.2 remete aos objetivos de uso da ferramenta de análise de fluxos de potência na rede elétrica.

Figura 1.2: Sistema Elétrico



Fonte: Próprio Autor

### 1.3 Motivação do Desenvolvimento em Domínio Complexo

Para resolver problemas cada vez mais complicados é crucial desenvolver ferramentas matemáticas apropriadas. Em problemas de engenharia, frequentemente tem-se interesse na otimização de parâmetros que recaem em sistemas de equações com grandes dimensões. Em se tratando de engenharia elétrica, nos depara-se com sistemas de matrizes a valores de variáveis complexas. Portanto é imprescindível um ferramental capaz de lidar com diferenciação de funções matriciais no domínio complexo. O tema do cálculo diferencial de funções matriciais reais já foi detalhadamente coberto em Magnus e Neudecker (2019) [3], já o desenvolvimento das derivadas matriciais em domínio complexo, pode ser encontrado em [4], uma das principais obras sobre o assunto. A diferenciabilidade complexa que será usada na presente dissertação, se aplica também largamente em áreas como processamento de sinais e comunicações. Portanto, a motivação deste trabalho é fazer uso da diferenciabilidade complexa, no desenvolvimento de implementações do fluxo de potência em domínio complexo, utilizando métodos de gradientes conjugados.

### 1.4 Apresentação e Descrição da Estrutura do Documento

Este trabalho se inicia com toda a fundamentação matemática necessária ao desenvolvimento do fluxo de potência em domínio complexo, onde é explicado a diferenciabilidade complexa e o cálculo de Wirtinger generalizado. Seguido da adoção do modelo de fluxo, bem como a formulação das equações do fluxo de potência e suas derivadas, tanto no domínio real quanto no domínio complexo, da definição dos tipos de barras, e da elucidação da solução iterativa do método de Newton-Raphson tanto para o domínio complexo e real. Logo em seguida se tem uma breve descrição dos algoritmos de gradiente conjugado e sua evolução. Logo após se apresenta os resultados, onde faz-se um comparativo de performance, sob alguns aspectos, dos algoritmos de gradientes conjugados *CBiCG* sem condicionador e com condicionador *CBiCG-ILU*, ambos comparados à formulação clássica do fluxo de potência em domínio real em coordenadas polares. Finalmente chega-se às conclusões e propostas de trabalhos futuros.

## 2 Fundamentação Matemática

Matrizes complexas costumam aparecer nas comunicações digitais, onde os filtros digitais podem conter coeficientes complexos, sistemas de potência, circuitos elétricos, teoria de controle, filtros adaptativos, análise de sensibilidade, sistemas mecânicos, acústicos etc. Geralmente, a representação de sistemas através de matrizes complexas, leva à um menor esforço computacional e de representação, comparada a abordagem via separação das partes real e imaginária, segundo é argumentado em *Brandwood* [5]. No entanto, um cálculo diferencial apropriado em domínio complexo, deve ser levado em conta ao se tratar problemas de otimização. Condições de otimalidade podem ser encontradas no cálculo complexo, através das diferenciais complexas, tratando as variáveis matriciais e suas complexas conjugadas, como variáveis independentes, conforme desenvolvido nesta seção.

### 2.1 Funções analíticas e não analíticas

**Definição 2.1** (Função Analítica). Seja  $D \subseteq \mathbb{C}$  o domínio de definição da função  $f : D \rightarrow \mathbb{C}$ . A função  $f$  é uma **função analítica** ou holomórfica no domínio  $D$  se o limite abaixo existe para todo  $z \in D$ .

$$f'(z) = \lim_{\Delta z \rightarrow 0} \frac{f(z + \Delta z) - f(z)}{\Delta z} \quad (1)$$

Se  $z = x + jy$  para  $x$  e  $y$  reais, então pode-se escrever uma complexa  $f(z)$  como:

$$f(z) = u(x, y) + jv(x, y) \quad (2)$$

Logo, para que a função complexa (2) satisfaça o limite (1) é necessário que:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \quad \frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y} \quad (3)$$

As equações (3) são conhecidas como **equações de Cauchy-Riemann** e podem ser sintetizadas da seguinte maneira:

$$\frac{\partial f}{\partial z^*} = 0 \quad (4)$$

De (4), pode ser visto que a função  $f$  não depende da variável  $z^*$ , caso contrário, ela é dita **não analítica**. Portanto, se uma função depende somente de  $z$ , como nas funções analíticas e não é dependente implícita ou explicitamente de  $z^*$ , a função em geral não pode ser uma função

escalar real. Esta restrição impede a diferenciação das funções escalares reais conforme o limite estabelecido em (1). Duas maneiras [6] são conhecidas para encontrar derivadas de funções escalares reais  $f \in \mathbb{R}$  com respeito à variável matricial complexa  $\mathbf{Z} \in \mathbb{C}^{N \times Q}$  (de uma forma geral). A primeira maneira é reescrever  $f$  como uma função das partes real  $\mathbf{X}$  e imaginária  $\mathbf{Y}$  da variável  $\mathbf{Z}$ , e então encontrar as derivadas da função reescrita, com respeito as variáveis independentes  $\mathbf{X}$  e  $\mathbf{Y}$  separadamente. Note que  $NQ$  variáveis independentes em  $\mathbf{Z}$ , correspondem a  $2NQ$  variáveis independentes reais em  $\mathbf{X}$  e  $\mathbf{Y}$ . A segunda maneira de lidar com esse problema, que é mais elegante e usada neste trabalho, é tratar as diferenciais das variáveis  $\mathbf{Z}$  e  $\mathbf{Z}^*$  como independentes.

## 2.2 Cálculos de Wirtinger

**Definição 2.2** (Derivadas Formais). Seja  $z = x + jy$ , onde  $x$  e  $y \in \mathbb{R}$ , então as derivadas formais com, respeito a  $z$  e  $z^*$  de  $f(z_0)$  em  $z_0 \in \mathbb{C}$  ou **derivadas de Wirtinger** [7], são definidas como:

$$\frac{\partial f(z_0)}{\partial z} = \frac{1}{2} \left( \frac{\partial f(z_0)}{\partial x} - j \frac{\partial f(z_0)}{\partial y} \right) \quad (5)$$

$$\frac{\partial f(z_0)}{\partial z^*} = \frac{1}{2} \left( \frac{\partial f(z_0)}{\partial x} + j \frac{\partial f(z_0)}{\partial y} \right) \quad (6)$$

As derivadas formais acima devem ser interpretadas formalmente porque  $z$  e  $z^*$  não podem variar independentemente [8]. Em Kreutz-Delgado [8], o tópico de cálculos de Wirtinger, também é chamado de  $\mathbb{C}\mathbb{R}$ -Cálculo.

Da definição 2.2, segue que as derivadas de uma função  $f$  com respeito a parte real  $x$  e parte imaginária  $y$ , podem ser expressas como:

$$\frac{\partial f(z_0)}{\partial x} = \frac{\partial f(z_0)}{\partial z} + \frac{\partial f(z_0)}{\partial z^*} \quad (7)$$

$$\frac{\partial f(z_0)}{\partial y} = j \left( \frac{\partial f(z_0)}{\partial z} - \frac{\partial f(z_0)}{\partial z^*} \right) \quad (8)$$

**Exemplo 1** (Derivadas Formais). Usando a definição 2.2, as seguintes derivadas formais são

encontradas:

$$\frac{\partial z}{\partial z} = \frac{1}{2} \left( \frac{\partial}{\partial x} - j \frac{\partial}{\partial y} \right) (x + jy) = \frac{1}{2}(1 + 1) = 1 \quad (9)$$

$$\frac{\partial z^*}{\partial z^*} = \frac{1}{2} \left( \frac{\partial}{\partial x} + j \frac{\partial}{\partial y} \right) (x - jy) = \frac{1}{2}(1 + 1) = 1 \quad (10)$$

$$\frac{\partial z}{\partial z^*} = \frac{1}{2} \left( \frac{\partial}{\partial x} + j \frac{\partial}{\partial y} \right) (x + jy) = \frac{1}{2}(1 - 1) = 0 \quad (11)$$

$$\frac{\partial z^*}{\partial z} = \frac{1}{2} \left( \frac{\partial}{\partial x} - j \frac{\partial}{\partial y} \right) (x - jy) = \frac{1}{2}(1 - 1) = 0 \quad (12)$$

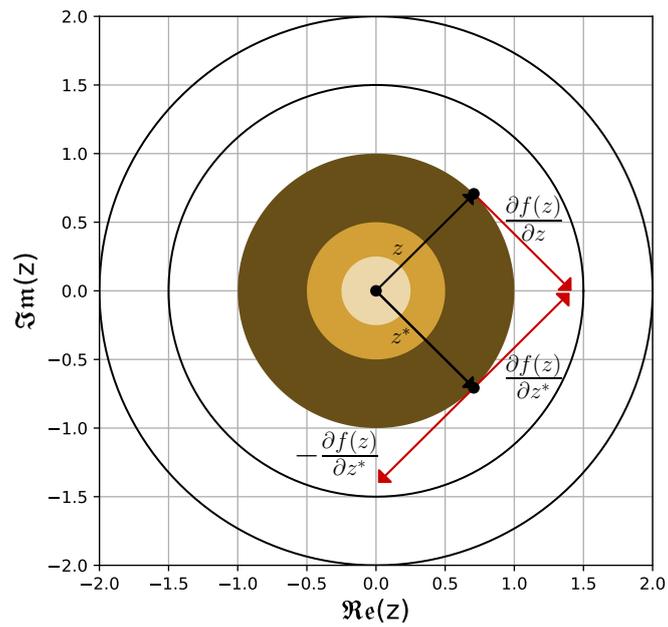
**Exemplo 2.** Seja  $f : \mathbb{C} \times \mathbb{C} \rightarrow \mathbb{R}$  dada por  $f(z, z^*) = zz^*$ . Essa função é diferenciável com respeito a ambas as variáveis  $z$  e  $z^*$  (usando a definição 2.2), e as expressões para as derivadas formais são dadas por:

$$\frac{\partial f(z, z^*)}{\partial z} = z^*, \quad (13)$$

$$\frac{\partial f(z, z^*)}{\partial z^*} = z. \quad (14)$$

Quando as variáveis complexas  $z$  e  $z^*$  são tratadas como variáveis independentes, a função  $f$  é diferenciável nestas duas variáveis. Note que esta mesma função não é diferenciável, usando a expressão tradicional para a derivada de uma função analítica, dada pela definição 2.1. O exemplo 2 também sugere uma interpretação geométrica mostrada na figura 2.1. A taxa máxima de mudança da função objetivo  $f(z, z^*) = |z|^2$ , dada compactamente pelo gradiente  $\frac{\partial}{\partial z^*}$  [8]. Observe que sua direção positiva é referida como um problema de maximização, enquanto a direção oposta diz respeito à minimização da função de custo  $f$ .

Figura 2.1: Gráfico de contorno de uma função real de variáveis complexas



Fonte: Próprio Autor

## 2.3 Cálculo de Wirtinger Generalizado

A teoria desenvolvida para encontrar derivadas de funções escalares reais através do **Cálculo de Wirtinger** pode ser estendida para funções mais gerais dependentes de variáveis matriciais complexas e suas complexas conjugadas, quando todos os elementos dessas matrizes são independentes. Será mostrado nesta seção, com base na principal referência dessa extensão teórica [4], como as diferenciais complexas de uma função podem ser usadas para identificar suas derivadas com respeito a ambas as variáveis matriciais, complexa  $\mathbf{Z}$  e sua complexa conjugada  $\mathbf{Z}^*$ .

### 2.3.1 Diferenciais Complexas

Assim como no caso real, o símbolo  $d$  será usado para denotar a **diferencial complexa**. A diferencial complexa tem a mesma dimensão da matriz em que é aplicada e pode ser encontrada em termos de componentes, isto é,  $(d\mathbf{Z}_{k,l}) = d(\mathbf{Z}_{k,l})$ . Seja  $z = x + jy \in \mathbb{C}$  uma variável escalar complexa, onde  $\text{Re}(z) = x$  e  $\text{Im}(z) = y$ . As seguintes relações entre as partes real e imaginária de  $z$  e seu complexo conjugado  $z^*$ , são válidas:

$$z = x + jy, \quad (15)$$

$$z^* = x - jy, \quad (16)$$

$$x = \frac{z + z^*}{2}, \quad (17)$$

$$y = \frac{z - z^*}{2j}. \quad (18)$$

Das equações (15) e (16), é assegurada a seguinte relação:

$$dz^* = (dz)^*. \quad (19)$$

Seja uma  $f$  uma função complexa escalar  $f : \mathbb{C} \times \mathbb{C} \rightarrow \mathbb{C}$  denotada por  $f(z, z^*)$ . Devido a esta função ser considerada uma função de duas variáveis complexas  $z$  e  $z^*$ , ambas dependentes de  $x$  e  $y$ , ela pode ser considerada uma função das duas variáveis reais  $x$  e  $y$ . Sendo assim a diferencial da função  $f$  pode ser expressa da seguinte forma:

$$df = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy, \quad (20)$$

onde  $\frac{\partial f}{\partial x}$  e  $\frac{\partial f}{\partial y}$  são as derivadas parciais de  $f$  com relação a  $x$  e  $y$ , respectivamente. Inserindo as diferenciais  $dx$  e  $dy$  das equações (17) e (18), na equação (20), encontra-se a seguinte relação:

$$\begin{aligned} df &= \frac{f}{\partial x} \left( \frac{dz + dz^*}{2} \right) + \frac{f}{\partial y} \left( \frac{dz - dz^*}{2j} \right) \\ df &= \frac{1}{2} \left( \frac{\partial f}{\partial x} - j \frac{\partial f}{\partial y} \right) dz + \frac{1}{2} \left( \frac{\partial f}{\partial x} + j \frac{\partial f}{\partial y} \right) dz^*. \end{aligned} \quad (21)$$

Comparando a expressão de (21) com a definição de derivada formal dada em (2.2), chega-se à seguinte expressão:

$$df = \frac{\partial f}{\partial z} dz + \frac{\partial f}{\partial z^*} dz^*. \quad (22)$$

Esta expressão é similar a expressão da diferencial de uma função real 20, ou seja, a estrutura da diferencial se mantém considerando  $f$  como função das variáveis complexas independentes,  $z$  e  $z^*$ .

A análise acima vale também para funções escalares complexas do tipo  $f : \mathbb{C}^{N \times Q} \times \mathbb{C}^{N \times Q} \rightarrow \mathbb{C}$ , denotada por  $f(\mathbf{Z}, \mathbf{Z}^*)$ , que dependem de uma variável matricial complexa  $\mathbf{Z}$  e sua conjugada  $\mathbf{Z}^*$ . De forma análoga às equações (15) - (19), pode-se desenvolver relações pertinentes às matrizes  $\mathbf{Z}$ ,  $\mathbf{Z}^*$  e suas diferenciais:

$$\mathbf{Z} = \mathbf{X} + j\mathbf{Y}, \quad (23)$$

$$\mathbf{Z}^* = \mathbf{X} - j\mathbf{Y}, \quad (24)$$

$$d\text{Re}(\mathbf{Z}) = d\mathbf{X} = \frac{d\mathbf{Z} + d\mathbf{Z}^*}{2}, \quad (25)$$

$$d\text{Im}(\mathbf{Z}^*) = d\mathbf{Y} = \frac{d\mathbf{Z} - d\mathbf{Z}^*}{2j}. \quad (26)$$

onde  $\mathbf{X} = \text{Re}(\mathbf{Z})$  e  $\mathbf{Y} = \text{Im}(\mathbf{Z})$ . A diferencial de  $f$  pode ser expressa em termos das  $NQ$  componentes reais independentes  $x_{k,l}$  e  $y_{k,l}$  de cada matriz real  $\mathbf{X}$  e  $\mathbf{Y}$ :

$$df = \sum_{k=0}^{Q-1} \sum_{l=0}^{N-1} \frac{f}{\partial x_{k,l}} dx_{k,l} + \sum_{k=0}^{Q-1} \sum_{l=0}^{N-1} \frac{f}{\partial y_{k,l}} dy_{k,l}, \quad (27)$$

ou em termos das variáveis complexas  $z_{k,l}$  e  $z_{k,l}^*$ :

$$df = \sum_{k=0}^{Q-1} \sum_{l=0}^{N-1} \frac{f}{\partial z_{k,l}} dz_{k,l} + \sum_{k=0}^{Q-1} \sum_{l=0}^{N-1} \frac{f}{\partial z_{k,l}^*} dz_{k,l}^*, \quad (28)$$

onde as  $\frac{\partial f}{\partial x_{k,l}}$ ,  $\frac{\partial f}{\partial y_{k,l}}$ ,  $\frac{\partial f}{\partial z_{k,l}}$  e  $\frac{\partial f}{\partial z_{k,l}^*}$ , são as derivadas de  $f$  com respeito a  $x_{k,l}$ ,  $y_{k,l}$ ,  $z_{k,l}$  e  $z_{k,l}^*$ , respectivamente. As  $NQ$  derivadas formais  $\frac{\partial f}{\partial z_{k,l}}$  e as  $NQ$  derivadas formais  $\frac{\partial f}{\partial z_{k,l}^*}$  podem ser organizadas em matrizes de diversas formas, dando origem às divergências estruturais das matrizes na literatura do assunto.

### 2.3.2 Identificação das Diferenciais Complexas

Sejam duas matrizes complexas  $\mathbf{Z}$  e  $\mathbf{Z}^* \in \mathbb{C}^{N \times Q}$ , onde os elementos de cada matriz são todos independentes. Assume-se que essas duas matrizes podem ser tratadas como variáveis independentes. Então, um procedimento para encontrar as diferenciais complexas de uma função matricial complexa da forma  $f : \mathbb{C}^{N \times Q} \times \mathbb{C}^{N \times Q} \rightarrow \mathbb{C}^{M \times P}$ , denotada por  $f(\mathbf{Z}, \mathbf{Z}^*)$ , é calcular a diferença:

$$\begin{aligned} f(\mathbf{Z} + d\mathbf{Z}, \mathbf{Z}^* + d\mathbf{Z}^*) - f(\mathbf{Z}, \mathbf{Z}^*) &= \\ &= 1^{\text{a}} \text{Ordem}(d\mathbf{Z}, \mathbf{Z}^*) + \text{Ordem Superior}(d\mathbf{Z}, \mathbf{Z}^*), \end{aligned} \quad (29)$$

ou seja, após expandir o lado esquerdo de (29), a soma dos termos que envolvem  $d\mathbf{Z}$  e  $d\mathbf{Z}^*$  de primeira ordem, é a diferencial de  $f$

$$df = 1^{\text{a}} \text{Ordem}(f(\mathbf{Z} + d\mathbf{Z}, \mathbf{Z}^* + d\mathbf{Z}^*) - f(\mathbf{Z}, \mathbf{Z}^*)) \quad (30)$$

**Exemplo 3** (Diferencial de  $\mathbf{AZB}$ ). Seja  $\mathbf{A} \in \mathbb{C}^{M \times N}$ ,  $\mathbf{Z} \in \mathbb{C}^{N \times Q}$ , e  $\mathbf{B} \in \mathbb{C}^{Q \times P}$ , onde  $\mathbf{A}$  e  $\mathbf{B}$  são independentes de  $\mathbf{Z}$  e  $\mathbf{Z}^*$ . Então para se encontrar a diferencial da função matricial complexa  $f(\mathbf{Z}, \mathbf{Z}^*) = \mathbf{AZB}$ , procede-se da seguinte forma:

$$\begin{aligned} f(\mathbf{Z} + d\mathbf{Z}, \mathbf{Z}^* + d\mathbf{Z}^*) - f(\mathbf{Z}, \mathbf{Z}^*) &= \\ &= \mathbf{A}(\mathbf{Z} + d\mathbf{Z})\mathbf{B} - \mathbf{AZB} = \\ &= \mathbf{A}(d\mathbf{Z})\mathbf{B} \Rightarrow \\ df &= \mathbf{A}(d\mathbf{Z})\mathbf{B}. \end{aligned} \quad (31)$$

### 2.3.3 Matrizes Jacobianas Complexas

Para podermos definir a jacobiana complexa de uma forma geral, é necessário definir o operador de vetorização, a seguir:

**Definição 2.3** (Operador Vetorização). Seja  $\mathbf{A} \in \mathbb{C}^{M \times N}$  e denote a  $i$ -ésima coluna de  $\mathbf{A}$  por

$a_i$ , onde  $i \in \{0, 1, \dots, N-1\}$ . Então o operador  $vec(\cdot)$  é definido como o vetor  $MN \times 1$  dado por:

$$vec(\mathbf{A}) = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{N-1} \end{bmatrix}. \quad (32)$$

De posse da definição acima, pode-se determinar de forma mais geral a derivada de uma função matricial complexa com respeito a matrizes complexas e então determinar a jacobiana com respeito as variáveis matriciais complexas.

**Definição 2.4** (Derivada Matricial). Seja  $f : \mathbb{C}^{N \times Q} \times \mathbb{C}^{N \times Q} \rightarrow \mathbb{C}^{M \times P}$ , denotada por  $f(\mathbf{Z}, \mathbf{Z}^*)$ . As derivadas de  $f(\mathbf{Z}, \mathbf{Z}^*) \in \mathbb{C}^{M \times P}$  com respeito as matrizes  $\mathbf{Z} \in \mathbb{C}^{N \times Q}$  e  $\mathbf{Z}^* \in \mathbb{C}^{N \times Q}$ , são denotadas por  $D_{\mathbf{Z}}f$  e  $D_{\mathbf{Z}^*}f$ , respectivamente. Ambas as derivadas, são de dimensão  $MP \times NQ$  e se relacionam na expressão diferencial a seguir:

$$dvec(f) = (D_{\mathbf{Z}}f)dvec(\mathbf{Z}) + (D_{\mathbf{Z}^*}f)dvec(\mathbf{Z}^*). \quad (33)$$

As derivadas  $D_{\mathbf{Z}}f$  e  $D_{\mathbf{Z}^*}f$ , também são chamadas de **Matrizes Jacobianas Complexas** de  $f$  com relação a  $\mathbf{Z}$  e  $\mathbf{Z}^*$ , respectivamente. Da expressão (33) é possível derivar expressões para as diferenciais e jacobianas complexas de funções gerais. Mais propriedades e métodos do cálculo matricial de Wirtinger pode ser encontrado numa das principais referências sobre o assunto, que é o livro *Complex-Valued Matrix Derivatives* [4].

## 2.4 Complemento de Schur e Operador Swap

### 2.4.1 Operador Swap

Frequentemente em problemas de engenharia elétrica, as partições  $A$  e  $D$  em (39), são mal condicionadas. O operador Swap  $S$  vem para o auxílio do condicionamento de matrizes, através da permutação de blocos de linhas ou colunas. O operador swap é definido a seguir.

**Definição 2.5** (Operador Swap). Seja o isomorfismo  $S$  de  $\mathbb{C}$  no espaço dual  $\mathbb{C}^*$ , que obedece

a relação  $S^{-1} = S^T = S$ , define-se  $S$  da maneira seguinte:

$$S \triangleq \begin{bmatrix} 0 & I_n \\ I_n & 0 \end{bmatrix} \quad (34)$$

a relação  $S^{-1} = S^T = S$ , mostra que  $S$  é um operador simétrico e sua própria inversa  $S^2 = I$ . De fato, o operador Swap é uma permutação de blocos de matrizes, o qual permuta linhas ou colunas, dependendo se o operador  $S$  é pré-multiplicado ou pós-multiplicado, respectivamente.

**Exemplo 4** (Operador Swap). Seja  $A$  uma matriz  $2n \times 2n$  dada em blocos e o operador swap  $S = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ , então é possível ver os efeitos da pré-multiplicação e pós-multiplicação de  $S$  por  $A$  e também a multiplicação simultânea pela esquerda e direita de  $A$ :

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad (35)$$

$$SA = \begin{bmatrix} A_{21} & A_{22} \\ A_{11} & A_{12} \end{bmatrix} \quad (36)$$

$$AS = \begin{bmatrix} A_{12} & A_{11} \\ A_{22} & A_{21} \end{bmatrix} \quad (37)$$

$$SAS = \begin{bmatrix} A_{22} & A_{21} \\ A_{12} & A_{11} \end{bmatrix} \quad (38)$$

## 2.4.2 Complemento de Schur

O termo **Complemento de Schur**, foi introduzido pela primeira vez em 1968 pela pesquisadora *Emilie Haynswoth*. O complemento de Schur desempenha um papel fundamental em muitas áreas da matemática e engenharia, sobretudo em problemas que envolvem a solução de sistemas de equações diferenciais. Na análise numérica, por exemplo, estes sistemas aparecem sob a forma de quocientes de determinantes em interpolação polinomial, em frações continuadas e aproximação de Padé. Em Probabilidade e Estatística, o complemento de Schur se aplica em certas inequações matriciais que são úteis nessas áreas, como a inequação de Cramer-Rao, que fornece um limite inferior para a covariância de um estimador não-enviesado, a inequação de Groves-Rothenberg e a inequação multivariada de Cauchy-Schwarz. Existe também, inúmeras aplicações do complemento de Schur em Engenharia de Controle. Uma das principais referências

sobre o assunto se encontra no livro *The Schur Complement and its Applications* [9]. Outras referências importantes de aplicação, podem ser encontradas em [10], [11], [12], [13], e [14].

**Definição 2.6** (Complemento de Schur). Sejam as matrizes complexas,  $A, B, C, D$  de dimensões  $p \times p$ ,  $p \times q$ ,  $q \times p$  e  $q \times q$ , respectivamente. Seja  $M$ , de dimensão  $(p + q) \times (p + q)$ , dada em blocos de matrizes da seguinte forma:

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad (39)$$

O complemento de Schur do bloco  $A$  e  $D$ , na matriz  $M$ , é dado respectivamente por:

$$M/A := D - CA^{-1}B \quad (40)$$

$$M/D := A - BD^{-1}C \quad (41)$$

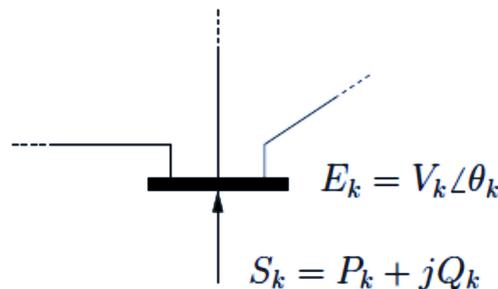
### 3 Modelo do fluxo de potência

Nesta secção apresenta-se a descrição do modelo utilizado no fluxo de potência, seus componentes e convenções.

#### 3.0.1 Convenção para o sentido das injeções de potência

São definidas quatro variáveis à barra  $k$ , correspondentes à tensão e à injeção de potência na barra, conforme figuras 3.1 e 3.2.

Figura 3.1: Variáveis da barra  $k$

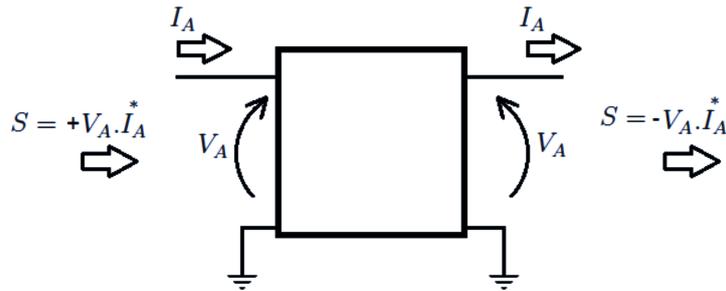


Fonte: Próprio Autor

- $V_k$ : Magnitude de tensão nodal

- $\theta_k$ : Ângulo de tensão nodal
- $P_k$ : Injeção líquida de potência ativa
- $Q_k$ : Injeção líquida de potência reativa

Figura 3.2: Convenção da entrada e saída de potência

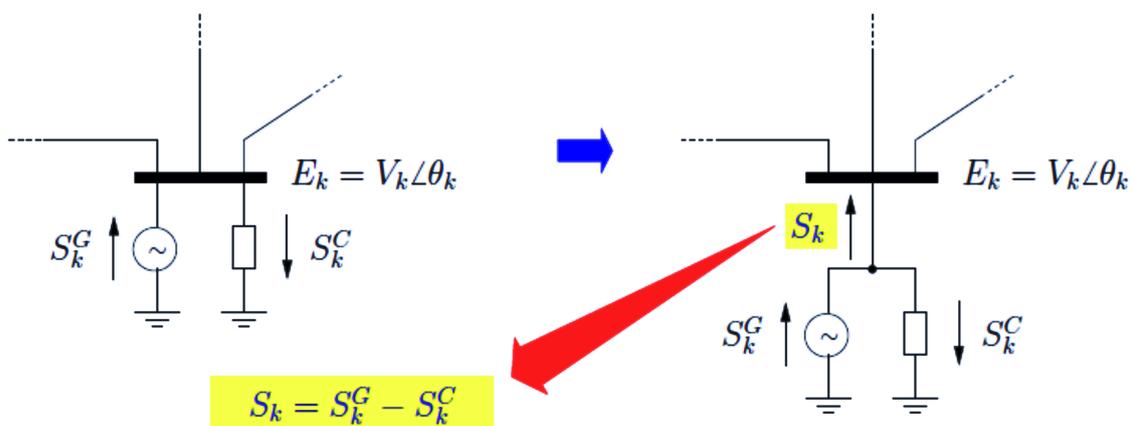


Fonte: Próprio Autor

Convencionam-se a potência entrando em um ramo ou nó, positiva ou negativa, de acordo com a figura 3.2. Conforme o diagrama mostrado pode-se sumarizar a convenção da seguinte forma:

- $P_k > 0$ : A potência ativa esta entrando na barra  $k$ , portanto, a potência é gerada.
- $P_k < 0$ : A potência ativa esta saindo da barra  $k$ , portanto, a potência é consumida.

O balanço de potência na barra  $k$ , considerando elementos de geração e consumo de potência, pode ser visto na figura 3.3.

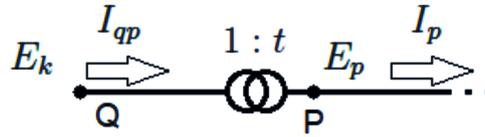
Figura 3.3: Injeção líquida de potência na barra  $k$ 

Fonte: Próprio Autor

### 3.1 Transformador ideal

O modelo esquemático do transformado ideal pode ser visto na figura 3.4:

Figura 3.4: Transformador ideal



Fonte: Próprio Autor

Utilizando a convenção de potência nos pontos P e Q do transformador ideal da figura 3.4 tem-se:

$$S_{qp} + -S_p = 0 \quad (42)$$

### 3.2 Equações Básicas do Modelo

Considerando agora o modelo de fluxo de potência da figura 3.5 tem-se:

$$E_k \cdot I_{km}^* + -E_p I_p^* = 0 \quad (43)$$

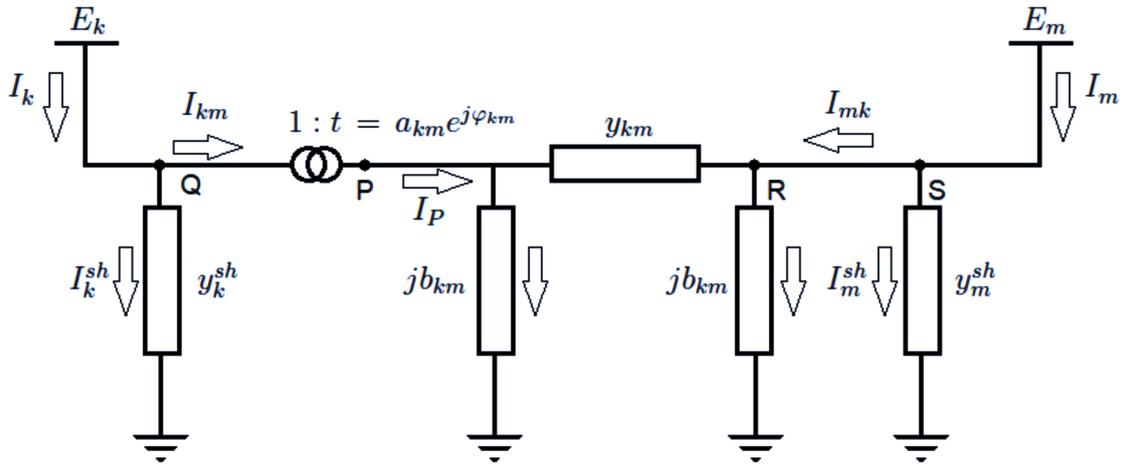
mas da relação entre as tensões no transformador tem-se:

$$\begin{aligned} \frac{E_k}{E_p} &= \frac{1}{t} \\ E_p &= E_k t \end{aligned} \quad (44)$$

portanto, aplicando esse resultado na equação (43), tem-se:

$$\begin{aligned} E_k I_{km}^* + -E_k t I_p^* &= 0 \\ I_p^* &= \frac{I_{km}^*}{t} \\ I_p &= \frac{I_{km}}{t^*} \end{aligned} \quad (45)$$

Figura 3.5: Modelo do Fluxo



Fonte: Próprio Autor

Para a modelagem do fluxo de potência considera-se o modelo  $\pi$  equivalente das linhas de transmissão, bem como um OLTC (on-load tap-changer) com o tap off-nominal, representado por  $t$ , conforme indicado na figura 3.5. A seguir indica-se as variáveis do modelo adotado para um ramo genérico entre as barras  $k$  e  $m$ :

- $E_k$ : Tensão na barra  $k$ .
- $E_m$ : Tensão na barra  $m$ .
- $I_k$ : Corrente saindo da barra  $k$  para a barra  $m$ .
- $I_m$ : Corrente saindo da barra  $m$  para a barra  $k$ .
- $I_k^{sh}$ : Corrente devido à admitância shunt  $y_k^{sh}$  conectada a barra  $k$ .
- $I_m^{sh}$ : Corrente devido à admitância shunt  $y_m^{sh}$  conectada a barra  $m$ .
- $I_{km}$ : Corrente do ramo  $k - m$  saindo da barra  $k$  para a barra  $m$ .
- $I_{mk}$ : Corrente do ramo  $k - m$  saindo da barra  $m$  para a barra  $k$ .
- $t$ : Tap complexo do OLTC, onde  $t = a_{km} e^{j\varphi_{km}}$ .
- $y_{km}$ : Admitância série da linha de transmissão.
- $b_{km}$ : Susceptância da linha de transmissão.

No modelo adotado tem-se que as correntes pela esquerda e direita de  $y_{km}$  são opostas de mesma magnitude e levando em conta as equações (44) e (45) de transformação de tensão e corrente respectivamente, tem-se:

$$\begin{aligned} I_p - jb_{km}E_p &= (E_p - E_m)y_{km} \\ \frac{I_{km}}{t^*} - jb_{km}E_k t &= (E_k t - E_m)y_{km} \end{aligned} \quad (46)$$

mas da lei de Kirchhoff das correntes aplicada ao nó  $Q$ :

$$I_{km} = I_k - y_k^{sh} E_k \quad (47)$$

Levando na equação (46) e organizando os termos, vem:

$$I_k = [tt^*(y_{km} + jb_{km}) + y_k^{sh}]E_k + (-t^*y_{km})E_m$$

de forma análoga chega-se à:

$$I_m = (-ty_{km})E_k + (y_{km} + jb_{km} + y_m^{sh})E_m$$

Portanto, as equações gerais para as correntes nas barras  $k$  e  $m$  são:

$$I_k = [tt^*(y_{km} + jb_{km}) + y_k^{sh}]E_k + (-t^*y_{km})E_m \quad (48)$$

$$I_m = (-ty_{km})E_k + (y_{km} + jb_{km} + y_m^{sh})E_m \quad (49)$$

### 3.2.1 Matriz de Admitância Nodal

As injeções de corrente (48) e (49), podem ser expressas em termos de uma matriz de admitância de ramo  $Y_{br}$ , de dimensão  $2 \times 2$ , cuja as tensões terminais são  $E_k$  e  $E_m$ :

$$\begin{bmatrix} I_k \\ I_m \end{bmatrix} = Y_{br} \begin{bmatrix} E_k \\ E_m \end{bmatrix}, \quad (50)$$

identificando a expressão (50), com as expressões de corrente (48) e (49), tem-se:

$$Y_{br} = \begin{bmatrix} tt^*(y_{km} + jb_{km}) + y_k^{sh} & -t^*y_{km} \\ -ty_{km} & y_{km} + jb_{km} + y_m^{sh} \end{bmatrix}. \quad (51)$$

pode-se ainda desacoplar a matriz de admitâncias shunt  $Y^{sh}$ , que é uma matriz diagonal, cujo os elementos de sua diagonal principal, são as admitâncias shunt conectadas as barras  $E_k$  e  $E_m$ :

$$Y_{br} = \begin{bmatrix} tt^*(y_{km} + jb_{km}) + y_k^{sh} & -t^*y_{km} \\ -ty_{km} & y_{km} + jb_{km} + y_m^{sh} \end{bmatrix} + \begin{bmatrix} y_k^{sh} & 0 \\ 0 & y_m^{sh} \end{bmatrix} \quad (52)$$

A equação da admitância de ramo (51), significa que para cada ramo  $i$ , tem-se 4 elementos da seguinte maneira:

$$Y_{br}^i = \begin{bmatrix} y_{kk}^i & y_{km}^i \\ y_{mk}^i & y_{mm}^i \end{bmatrix} \quad (53)$$

Se  $n_l$  e  $n_b$  são o número de ramos e de barras do sistema, respectivamente, então 4 vetores  $Y_{kk}$ ,  $Y_{km}$ ,  $Y_{mk}$  e  $Y_{mm}$ , de dimensão  $n_l \times 1$  podem ser construídos, de forma que o  $i$ -ésimo elemento de cada vetor, vem do correspondente elemento de  $Y_{br}^i$ . Além disso, as matrizes de conexão  $C_k$  e  $C_m$ , ambas de dimensão  $n_l \times n_b$ , são usadas para a construção da matriz de admitância nodal do sistema  $Y_{bus}$ . Estas matrizes,  $C_k$  e  $C_m$  são definidas de tal forma, que o elemento  $(i, q)$ ésimo de  $C_k$  e o  $(i, p)$ ésimo elemento de  $C_m$  são iguais a 1 para cada ramo  $i$ , se o ramo  $i$  conectar a barra  $q$  à barra  $p$ , todos os outros elementos de  $C_k$  e  $C_m$  são iguais a zero. Define-se ainda  $Y_k$  e  $Y_m$ , ambas de dimensão  $n_l \times n_b$ , de tal forma que:

$$I_k = Y_k V \quad (54)$$

$$I_m = Y_m V, \quad (55)$$

onde  $V$  é o vetor complexo  $n_b \times 1$  de tensões nodais. finalmente, segue que:

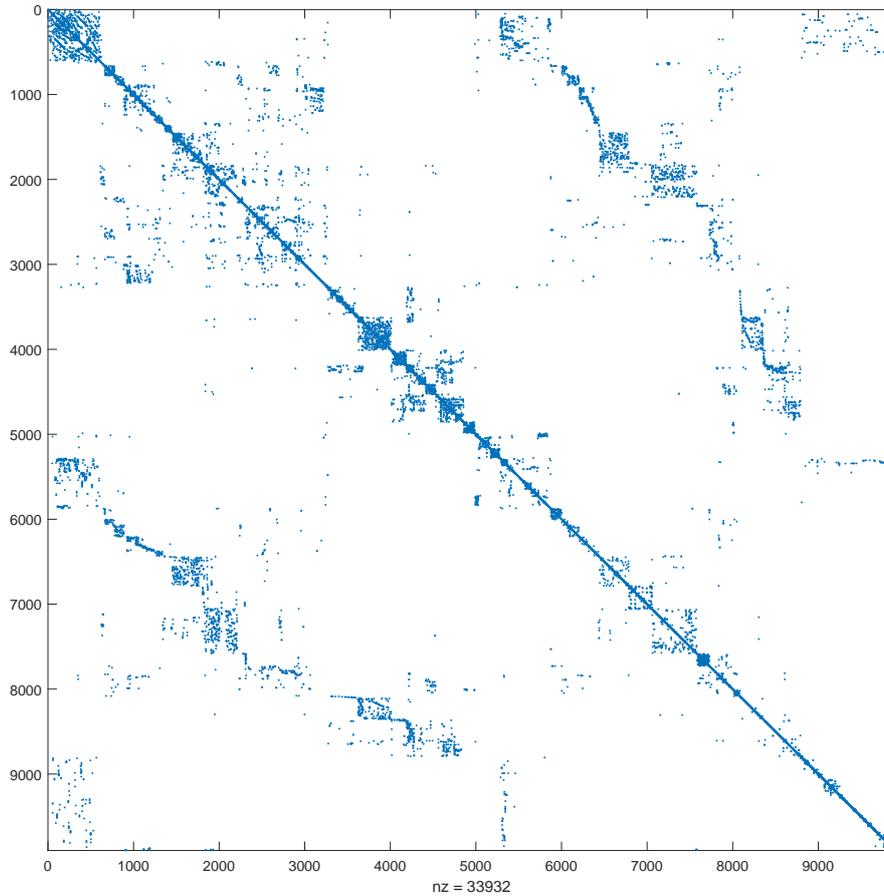
$$Y_k = [Y_{kk}]C_k + [Y_{km}]C_k \quad (56)$$

$$Y_m = [Y_{mk}]C_k + [Y_{mm}]C_m \quad (57)$$

$$Y_{bus} = C_k^T Y_k + C_m^T Y_m + [Y^{sh}]. \quad (58)$$

onde  $[X]$ , significa uma matriz diagonal cuja a diagonal principal é formada pelos elementos do vetor coluna  $X$ . Esta formulação vetorizada da matriz de admitância nodal, pode se encontrada na referência [15]. A seguir um exemplo de matriz de admitância nodal, construída com algoritmos desenvolvidos neste trabalho, a partir de dados de um sistema elétrico real de 9899 barras, extraído do banco de dados do site [www.epe.gov.br](http://www.epe.gov.br):

Figura 3.6: Matriz de admitância de um sistema do SIN de 9899 barras



Fonte: Próprio Autor

### 3.3 Tipos de Barras

#### 3.3.1 Barra *Slack* ou Barra de Referência

A tensão na barra *slack* já é conhecida, uma vez que o valor da magnitude e do ângulo são especificados para a barra de referência do sistema.

#### 3.3.2 Barra PQ

Com a potência ativa e reativa especificadas para o nó PQ, as funções de *mismatches* complexas são expressas por

$$M_k = S_k - (P_{ks} + j Q_{ks}), \quad (59)$$

$$M_k^* = S_k^* - (P_{ks} - j Q_{ks}). \quad (60)$$

Da mesma forma, as funções no domínio real são dadas por:

$$\Delta P_k = P_k^{calc} - P_{ks}, \quad (61)$$

$$\Delta Q_k = Q_k^{calc} - Q_{ks}. \quad (62)$$

Onde  $P_{ks}$  e  $Q_{ks}$ , são as potências ativas e reativas especificadas no nó  $k$ , respectivamente, e  $P_k^{calc}$  e  $Q_k^{calc}$  são as potências calculadas em cada processo iterativo.

### 3.3.3 Barra PV

Como a geração de potência ativa e a magnitude da tensão terminal em uma barra *PV* são ambas especificadas, ou seja,  $P_{ks}$  e  $V_{ks}$ , respectivamente, a soma de  $M_k$  em (59) e  $M_k^*$  em (60) fornece a função residual complexa,  $M_{kg}$ , que está relacionada com a restrição de potência ativa da seguinte forma:

$$\begin{aligned} M_{kg} &= M_k + M_k^*, \\ &= S_k + S_k^* - 2 \times P_{ks}. \end{aligned} \quad (63)$$

A segunda função residual complexa  $E_{kg}$  para um nó gerador  $k$  é formada usando a restrição de magnitude de tensão dada por

$$|E_{kg}| = |V_k|^2 - |V_{ks}|^2, \quad (64)$$

onde  $|V_{ks}|$  é a magnitude de tensão especificada no nó  $k$ .

Assim  $|V_k|^2 = V_k V_k^*$ , então (64) pode ser expressa no domínio complexo como

$$E_{kg} = V_k V_k^* - |V_{ks}|^2. \quad (65)$$

Também para o domínio real, com a geração de potência ativa e a magnitude da tensão terminal em uma barra *PV* especificadas, a equação de restrição no domínio real é dada apenas por (61). Assim,  $|V_k|$ , onde  $k$  é o nó *PV*, permanece igual ao valor especificado.

### 3.3.4 Barra PQV

Este tipo de barramento é referido ao modelo *On-Load-Tap-Changer (OLTC)*, que pode ser um transformador para regulagem de tensão de barramento local ou próximo, ou um trans-

formador de deslocamento de fase para controlar o fluxo de potência ativa transmitida por uma linha [16], ou ambos. Ele também é adequado para modelar um link DC [17], [18], [19], [20]. Como a demanda de potência ativa e reativa é especificada, as funções de *mismatches* complexas, conforme declarado em (59) e (60), são empregadas. No entanto, vale lembrar que a posição do tap do *OLTC* nos permite regular a magnitude da tensão e também a potência ativa transmitida na linha.

Supõe-se que há um defasador puro que controle apenas a potência ativa na linha, isto leva à usar além das equações (59) e (60), a seguinte função de *mismatch*:

$$Ma_{km} = a_{km} a_{km}^* - |a_s|^2, \quad (66)$$

onde  $a_s = 1$ , caracterizando assim um defasador puro, e  $M_a$  é a equação de restrição para o módulo de  $a_{km}$ , que deve permanecer em 1.

Para o domínio real, as funções de *mismatches* são dadas pelas funções (61) e (62), sendo que  $|a_{km}|$ , onde  $k$  é o nó PQV, permanece igual ao valor especificado, no caso,  $a = 1$  para configurar o defasador puro.

Supondo agora que a tensão do barramento  $m$  é regulada, tem-se então, além das funções (59) e (60), a seguinte função de *mismatch*:

$$E_m = V_m V_m^* - |V_{m_s}|^2. \quad (67)$$

Em (67),  $|V_{m_s}|$  é o módulo da tensão especificada no nó  $m$ , ou seja, a tensão nodal regulada. Quando há apenas o controle de tensão pelo tap, ou seja, o transformador é usado apenas para controle de tensão, e não para controle de potência ativa e tensão, é necessário adicionar a seguinte equação de *mismatch*:

$$M_m = a_{km} - a_{km}^*. \quad (68)$$

Essa equação faz com que a parte imaginária do tap seja igual a zero, sendo o tap apenas real, sem defasamento angular, controlando assim apenas o módulo de tensão.

E para o domínio real, as equações de *mismatches* são dadas por (61), (62). Sendo que,  $|V_{m_s}|$  permanece no valor especificado, não entrando no processo iterativo, e o valor de  $a$  se ajusta conforme a necessidade da tensão especificada.

### 3.4 Fluxo de Potência em domínio Complexo por Ramos

Nesta seção são apresentadas as equações de um fluxo de potência, por meio de equações complexas, derivadas diretamente do trabalho de Wirtinger [21], em contraste com a abordagem dada por [22], [23]. A modelagem de fluxo de potência tem base nas equações nodais clássicas apresentadas em [24]. A abordagem do modelo analítico é derivada através das equações gerais do fluxo de potência. A principal razão para esta última opção é o modelo do transformador com posição de tap fora do nominal, incluindo *PSTs* (Phase-shifting transformers) [25], [26].

As equações de fluxo de potência de variáveis complexas que modelam qualquer tipo de ramificação em uma rede elétrica, ou seja, linhas de transmissão e *PSTs* são:

$$S_{km} = V_k \left( \frac{y_{km}^*}{a_{km} a_{km}^*} - j b_{km}^{sh} \right) V_k^* - V_k \frac{y_{km}^*}{a_{km}} V_m^*, \quad (69)$$

$$S_{mk} = V_m \left( y_{km}^* - j b_{km}^{sh} \right) V_m^* - V_m \frac{y_{km}^*}{a_{km}^*} V_k^*. \quad (70)$$

e

$$S_{km}^* = V_k^* \left( \frac{y_{km}}{a_{km}^* a_{km}} + j b_{km}^{sh} \right) V_k - V_k^* \frac{y_{km}}{a_{km}^*} V_m, \quad (71)$$

$$S_{mk}^* = V_m^* \left( y_{km} + j b_{km}^{sh} \right) V_m - V_m^* \frac{y_{km}}{a_{km}} V_k. \quad (72)$$

No conjunto de equações (69 - 72), o modelo geral do transformador de derivação com tap fora do nominal é composto por um transformador ideal com relações  $1 : ae^{j\varphi}$  em série com sua admitância ou impedância [25]. A potência em um determinado nó é definida como

$$S_k = \sum_{m \in \Omega_i}^N S_{km} \quad (73)$$

$$S_k^* = \sum_{m \in \Omega_i}^N S_{km}^* \quad (74)$$

Onde  $\Omega_i$  em (73-74) é o conjunto de barramentos vizinhos conectados ao nó  $k$ — e  $N$  é o número total de barras ligadas ao nó  $k$ —. A seguir, é apresentado o conjunto de derivadas parciais de equações complexas utilizadas no método de Newton-Raphson. Então, aplicando o cálculo de

Wirtinger em relação a (69) tem-se:

$$\left. \frac{\partial S_{km}}{\partial V_k} \right|_{V_k^* = Const} = \left( \frac{y_{km}^*}{a_{km} a_{km}^*} - j b_{km}^{sh} \right) V_k^* - \frac{y_{km}^*}{a_{km}} V_m^*, \quad (75)$$

$$\left. \frac{\partial S_{km}}{\partial V_k^*} \right|_{V_k = Const} = V_k \left( \frac{y_{km}^*}{a_{km} a_{km}^*} - j b_{km}^{sh} \right), \quad (76)$$

$$\left. \frac{\partial S_{km}}{\partial V_m} \right|_{V_m^* = Const} = 0.0, \quad (77)$$

$$\left. \frac{\partial S_{km}}{\partial V_m^*} \right|_{V_m = Const} = -V_k \frac{y_{km}^*}{a_{km}}, \quad (78)$$

$$\left. \frac{\partial S_{km}}{\partial a_{km}} \right|_{a_{km}^* = Const} = -V_k \left( \frac{y_{km}^*}{a_{km}^2 a_{km}^*} \right) V_k^* + V_k \frac{y_{km}^*}{a_{km}^2} V_m^*, \quad (79)$$

$$\left. \frac{\partial S_{km}}{\partial a_{km}^*} \right|_{a_{km} = Const} = -V_k \left( \frac{y_{km}^*}{a_{km} (a_{km}^*)^2} \right) V_k^*. \quad (80)$$

Em relação a (70) tem-se:

$$\left. \frac{\partial S_{mk}}{\partial V_m} \right|_{V_m^* = Const} = (y_{km}^* - j b_{km}^{sh}) V_m^* - \frac{y_{km}^*}{a_{km}^*} V_k^*, \quad (81)$$

$$\left. \frac{\partial S_{mk}}{\partial V_m^*} \right|_{V_m = Const} = V_m (y_{km}^* - j b_{km}^{sh}), \quad (82)$$

$$\left. \frac{\partial S_{mk}}{\partial V_k} \right|_{V_k^* = Const} = 0.0, \quad (83)$$

$$\left. \frac{\partial S_{mk}}{\partial V_k^*} \right|_{V_k = Const} = -V_m \frac{y_{km}^*}{a_{km}^*}, \quad (84)$$

$$\left. \frac{\partial S_{mk}}{\partial a_{km}} \right|_{a_{km}^* = Const} = 0.0, \quad (85)$$

$$\left. \frac{\partial S_{mk}}{\partial a_{km}^*} \right|_{a_{km} = Const} = V_m \frac{y_{km}^*}{(a_{km}^*)^2} V_k^*. \quad (86)$$

Em relação a (71) tem-se:

$$\left. \frac{\partial S_{km}^*}{\partial V_k} \right|_{V_k^* = Const} = V_k^* \left( \frac{y_{km}}{a_{km}^* a_{km}} + j b_{km}^{sh} \right), \quad (87)$$

$$\left. \frac{\partial S_{km}^*}{\partial V_k^*} \right|_{V_k = Const} = \left( \frac{y_{km}}{a_{km}^* a_{km}} + j b_{km}^{sh} \right) V_k - \frac{y_{km}}{a_{km}^*} V_m, \quad (88)$$

$$\left. \frac{\partial S_{km}^*}{\partial V_m} \right|_{V_m^* = Const} = -V_k^* \frac{y_{km}}{a_{km}^*}, \quad (89)$$

$$\left. \frac{\partial S_{km}^*}{\partial V_m^*} \right|_{V_m = Const} = 0.0, \quad (90)$$

$$\left. \frac{\partial S_{km}^*}{\partial a_{km}} \right|_{a_{km}^* = Const} = -V_k^* \left( \frac{y_{km}}{a_{km}^* a_{km}^2} \right) V_k, \quad (91)$$

$$\left. \frac{\partial S_{km}^*}{\partial a_{km}^*} \right|_{a_{km} = Const} = -V_k^* \left( \frac{y_{km}}{(a_{km}^*)^2 a_{km}} \right) V_k + V_k^* \frac{y_{km}}{(a_{km}^*)^2} V_m. \quad (92)$$

E finalmente, em relação a (72) tem-se:

$$\left. \frac{\partial S_{mk}^*}{\partial V_k} \right|_{V_k^* = Const} = -V_m^* \frac{y_{km}}{a_{km}}, \quad (93)$$

$$\left. \frac{\partial S_{mk}^*}{\partial V_k^*} \right|_{V_k = Const} = 0.0, \quad (94)$$

$$\left. \frac{\partial S_{mk}^*}{\partial V_m} \right|_{V_m^* = Const} = V_m^* (y_{km} + j b_{km}^{sh}), \quad (95)$$

$$\left. \frac{\partial S_{mk}^*}{\partial V_m^*} \right|_{V_m = Const} = (y_{km} + j b_{km}^{sh}) V_m - \frac{y_{km}}{a_{km}} V_k, \quad (96)$$

$$\left. \frac{\partial S_{mk}^*}{\partial a_{km}} \right|_{a_{km}^* = Const} = V_m^* \frac{y_{km}}{a_{km}^2} V_k, \quad (97)$$

$$\left. \frac{\partial S_{mk}^*}{\partial a_{km}^*} \right|_{a_{km} = Const} = 0.0. \quad (98)$$

### 3.4.1 Equações de *mismatches* para o *PST*

Além das equações de *mismatches* das barras, é necessário também a adoção de equações de *mismatches* para ajustar o fluxo de potência ativa na linha na qual está instalado o *PST*.

Assim, para o domínio dos números complexos, tem-se que a equação de *mismatch* é dada por:

$$MP_{km} = S_{km} + S_{km}^* - 2 \times P_{km_s}. \quad (99)$$

Sendo  $MP_{km}$  a equação de restrição para a potência ativa transmitida através do *PST*, onde  $P_{km}$  a potência especificada que deve ser transmitida pelo *PST*.

Para o domínio real, é adicionada a equação de fluxo de potência na linha de transmissão onde se deseja controlar o fluxo de potência ativa, que é dada por:

$$P_{km} = \left( \frac{1}{a} V_k \right)^2 g_{km} - \left( \frac{1}{a} V_k \right) V_m g_{km} \cos(\theta_{km} + \varphi_{km}) + \left( \frac{1}{a} V_k \right) V_m b_{km} \sin(\theta_{km} + \varphi_{km}). \quad (100)$$

Então, a equação de *mismatch* no domínio real é dada por

$$\Delta P_{km} = P_{km}^{calc} - P_{km_s}, \quad (101)$$

onde  $P_{km_s}$  é a potência especificada para a linha de transmissão.

### 3.4.2 Solução Iterativa - Algoritmo de Newton-Raphson

Sendo  $\underline{\mathbf{x}}$  o vetor de variáveis de estado, e  $\underline{\mathbf{M}}$  o vetor de *mismatches*, o objetivo do algoritmo é calcular  $\underline{\mathbf{x}}$  de forma que satisfaça

$$\underline{\mathbf{M}}(\underline{\mathbf{x}}) = 0. \quad (102)$$

Segue-se que a linearização de (102) é dada por

$$\underline{\mathbf{M}}(\underline{\mathbf{x}}^{(\nu-1)}) + \mathbf{J}(\underline{\mathbf{x}}^{(\nu-1)}) \Delta \underline{\mathbf{x}}^{(\nu)} = 0, \quad (103)$$

e

$$\underline{\mathbf{x}}^{(\nu)} = \underline{\mathbf{x}}^{(\nu-1)} - [\mathbf{J}^{(\nu-1)}]^{-1} \underline{\mathbf{M}}(\underline{\mathbf{x}}^{(\nu-1)}), \quad (104)$$

ou

$$\Delta \underline{\mathbf{x}}^{(\nu)} = - [\mathbf{J}^{(\nu-1)}]^{-1} \underline{\mathbf{M}}(\underline{\mathbf{x}}^{(\nu-1)}), \quad (105)$$

Onde  $\mathbf{J}$  é a matriz Jacobiana. Então, a equação atualizada é dada por

$$\underline{\mathbf{x}}^{(\nu)} = \underline{\mathbf{x}}^{(\nu-1)} + \Delta \underline{\mathbf{x}}^{(\nu)}. \quad (106)$$

Sendo o critério de convergência assumido dado por

$$\|\Delta \underline{\mathbf{x}}^{(\nu)}\|_{\infty} \leq tol \ (\approx 10^{-4}). \quad (107)$$

Onde  $\|\cdot\|_{\infty}$  é definido como a norma infinita e  $\nu$  é o número de iterações.

Sendo a barra *slack* excluída, tem-se que:

Tanto o vetor de *mismatch* quanto o vetor de variáveis de estado variam de acordo com o sistema a ser modelado. Para o domínio real, umas das formas que o vetor das variáveis de estado pode ser representado é

$$\underline{\mathbf{x}} = [\theta_1, \theta_2, \dots, \theta_{N-1}, |V_1|, |V_2|, \dots, |V_{N-1}|, |a_{km_1}|, \dots, |a_{km_L}|, \varphi_1, \dots, \varphi_J]^T, \quad (108)$$

e o vetor *mismatch* é

$$\underline{M}(\underline{\mathbf{x}}) = [\Delta P_1, \Delta P_2, \dots, \Delta P_{N-1}, \Delta Q_1, \Delta Q_2, \dots, \Delta Q_{N-1}, \Delta P_{km_1}, \dots, \Delta P_{km_L}]^T. \quad (109)$$

Para o domínio complexo, tem-se que o vetor das variáveis de estado é dado por

$$\underline{\mathbf{x}} = \underline{\mathbf{x}}_{\mathbf{c}} = [V_1, V_2, \dots, V_{N-1}, a_{km_1}, \dots, a_{km_L}, V_1^*, V_2^*, \dots, V_{N-1}^*, a_{km_1}^*, \dots, a_{km_L}^*]^T, \quad (110)$$

e o vetor *mismatch* é

$$\underline{M}(\underline{\mathbf{x}}_{\mathbf{c}}) = [M_1, M_2, \dots, M_{N-1}, E_{kg_1}, \dots, E_{kg_P}, M_{km_1}, \dots, M_{km_J}, M_1^*, M_2^*, \dots, M_{N-1}^*]^T. \quad (111)$$

Onde  $N$  é o número de barras.

## 4 Formulação Matricial do Fluxo de Potência

Não foi apresentado neste trabalho a dedução clássica das equações do fluxo de potência no domínio real, devido a uma vasta literatura sobre o estado da arte desta formulação. No entanto uma abordagem utilizando o cálculo matricial, é menos comum e merece ser tratada aqui, uma vez que, os algoritmos computacionais desenvolvidos durante esta dissertação, prezam pela estrutura matricial ou vetorizada. A principal referência utilizada sobre o cálculo matricial diferencial em domínio real, pode ser encontrada em *Matrix Differential Calculus with Applications in Statistics and Econometrics* [3].

### 4.1 Domínio Real

Considere os vetores de tensões e correntes complexas,  $\mathbf{V} \in \mathbb{C}^n$  e  $\mathbf{I} \in \mathbb{C}^n$ , respectivamente:

$$\mathbf{V} = \begin{bmatrix} v_1 e^{j\theta_1} \\ v_2 e^{j\theta_2} \\ \vdots \\ v_n e^{j\theta_n} \end{bmatrix} \quad (112)$$

e também:

$$I = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} \quad (113)$$

Onde  $v_k$  e  $\theta_k$ , são a  $k$ -ésima magnitude de tensão e o  $k$ -ésimo ângulo de tensão do vetor  $\mathbf{V}$ , respectivamente. Então o vetor de injeção de potência nodal nas barras, fica:

$$\mathbf{S} = [\mathbf{V}]\mathbf{I}^* = [\mathbf{I}^*]\mathbf{V} \quad (114)$$

Onde  $[\mathbf{V}]$  é a matriz diagonal cujos elementos são os elementos de  $\mathbf{V}$ .

$$[\mathbf{V}] = \begin{bmatrix} v_1 e^{j\theta_1} & 0 & \dots & 0 \\ 0 & v_2 e^{j\theta_2} & \dots & 0 \\ \vdots & \vdots & & \\ 0 & 0 & \dots & v_n e^{j\theta_n} \end{bmatrix} \quad (115)$$

De forma genérica,  $\mathbf{V} = v e^{j\theta}$ , então derivando parcialmente  $\mathbf{V}$  e  $\mathbf{I}$  com relação ao módulo  $v$  de tensão, e ao ângulo  $\theta$ , tem-se:

$$\frac{\partial \mathbf{V}}{\partial v} = \left[ \frac{\mathbf{V}}{v} \right] \quad (116)$$

$$\frac{\partial \mathbf{I}}{\partial v} = \mathbf{Y} \left[ \frac{\mathbf{V}}{v} \right] \quad (117)$$

$$\frac{\partial \mathbf{V}}{\partial \theta} = j[\mathbf{V}] \quad (118)$$

$$\frac{\partial \mathbf{I}}{\partial \theta} = j\mathbf{Y}[\mathbf{V}] \quad (119)$$

Onde  $\mathbf{Y}$  é a matriz de admitância nodal do sistema. Derivando agora parcialmente  $\mathbf{S}$  com relação ao módulo  $v$  de tensão, e ao ângulo  $\theta$ , levando em conta as equações anteriores, tem-se:

$$\frac{\partial \mathbf{S}}{\partial v} = [\mathbf{V}]\left(\mathbf{Y} \frac{\mathbf{V}}{v}\right)^* + \mathbf{I}^* \left[ \frac{\mathbf{V}}{v} \right] \quad (120)$$

$$\frac{\partial \mathbf{S}}{\partial \theta} = j[\mathbf{V}](\mathbf{I}^* - \mathbf{Y}^*[\mathbf{V}^*]) \quad (121)$$

Levando em conta as equações 120 e 121, a matriz jacobiana pode ser construída em blocos de matrizes da seguinte forma:

$$J_{11} = \operatorname{Re}\left(\frac{\partial \mathbf{S}}{\partial \theta}\right) \quad (122)$$

$$J_{12} = \operatorname{Re}\left(\frac{\partial \mathbf{S}}{\partial v}\right) \quad (123)$$

$$J_{21} = \operatorname{Im}\left(\frac{\partial \mathbf{S}}{\partial \theta}\right) \quad (124)$$

$$J_{22} = \operatorname{Im}\left(\frac{\partial \mathbf{S}}{\partial v}\right) \quad (125)$$

$$J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \quad (126)$$

Sendo  $n_{PV}$  e  $n_{PQ}$  o número de barras PV e PQ, respectivamente, as dimensões dos blocos na jacobiana (126), se dispõem da seguinte forma:

$$J = \begin{bmatrix} (n_{PQ} + n_{PV}) \times (n_{PQ} + n_{PV}) & (n_{PQ} + n_{PV}) \times (n_{PQ}) \\ (n_{PQ}) \times (n_{PQ} + n_{PV}) & (n_{PQ}) \times (n_{PQ}) \end{bmatrix} \quad (127)$$

Portanto as equações matriciais (120 - 121), são a base para o desenvolvimento de algoritmos vetorizados, usando cálculo matricial. Códigos vetorizados aplicados em fluxo de potência podem ser encontrados em programas como *Matpower* [27].

## 4.2 Domínio Complexo

No desenvolvimento do modelo de fluxo de potência matricial em domínio complexo, precisa-se primeiramente encontrar as expressões matriciais do vetor de mismatches  $M \in \mathbb{C}^n$ , para um sistema de  $n$  barras. Para realizar essa tarefa, encontra-se as equações de mismatches pertinentes a cada tipo de barra, que definirão os blocos matriciais que compõe o vetor de mismatches  $M$ :

- **Barras PQ:**

Nas barras PQ, a potência ativa  $P$  e a potência reativa  $Q$ , são especificadas. Portanto, a potência complexa nodal é especificada  $S_{esp}$ . Ressalta-se que todas as grandezas aqui retratadas, são grandezas vetoriais, por exemplo,  $P_{esp}$  é o vetor de dimensão  $n_{PQ} \times 1$ , onde  $n_{PQ}$  é o número

de barras PQ, nas quais a potência ativa é especificada.

$$\begin{aligned} M_k &= S - S_{esp} \\ M_k &= [V](YV)^* - S_{esp} \end{aligned} \quad (128)$$

▪ **Barras PV:**

Nas barras PV, a potência ativa  $P$  e magnitude de tensão  $|V|$ , são especificadas:

$$\begin{aligned} M_{kg} &= S + S^* - 2P_{esp} \\ M_{kg} &= [V]Y^*V^* + [V^*]YV - 2P_{esp} \end{aligned} \quad (129)$$

A equação de  $M_{kg}$  traduz a especificação da potencia ativa  $P$  nas barras PV. Vale lembrar que  $[V]$  é a matriz diagonal, composta pelos elementos do vetor de tensões  $V$ , vide equação (115). Agora, a equação que traduz a especificação de magnitude de tensão  $|V|$ , é:

$$\begin{aligned} E_{kg} &= |V|^2 - |V|_{esp}^2 \\ E_{kg} &= [V]V^* - |V|_{esp}^2 \end{aligned} \quad (130)$$

Portanto o vetor de mismatches complexo é organizado da seguinte maneira:

$$M = \begin{bmatrix} M_{kg} \\ M_k \\ E_{kg} \\ M_{kg}^* \end{bmatrix} \quad (131)$$

Retomando a expressão (22) das diferenciais complexas, pode-se reorganizá-la em blocos matriciais e usá-la para identificação da matriz jacobiana complexa:

$$dM = \frac{\partial M}{\partial V} dV + \frac{\partial M}{\partial V^*} dV^*, \quad (132)$$

traduzindo em blocos de matrizes:

$$dM = \begin{bmatrix} \frac{\partial M}{\partial V} & \frac{\partial M}{\partial V^*} \end{bmatrix} \begin{bmatrix} dV \\ dV^* \end{bmatrix} \quad (133)$$

relembrando da expressão da diferencial de uma função escalar:

$$dM = JdV_{cv} \quad (134)$$

onde:

$$dV_{cv} = \begin{bmatrix} dV \\ dV^* \end{bmatrix} \quad (135)$$

chama-se  $V_{cv}$  de vetor de estados em coordenadas complexas. Finalmente, comparando os termos de (133) e (134), identifica-se a jacobiana complexa e seus blocos:

$$J = \begin{bmatrix} \frac{\partial M}{\partial V} & \frac{\partial M}{\partial V^*} \end{bmatrix} \quad (136)$$

especificando melhor os termos de  $M$  em (136):

$$J = \begin{bmatrix} \frac{\partial M_{kg}}{\partial V} & \frac{\partial M_{kg}}{\partial V^*} \\ \frac{\partial M_k}{\partial V} & \frac{\partial M_k}{\partial V^*} \\ \frac{\partial E_{kg}}{\partial V} & \frac{\partial E_{kg}}{\partial V^*} \\ \frac{\partial M_{kg}^*}{\partial V} & \frac{\partial M_{kg}^*}{\partial V^*} \end{bmatrix} \quad (137)$$

#### 4.2.1 Blocos de Derivadas da Jacobiana Complexa

Utilizando os conceitos e expressões do cálculo matricial complexo, deduz-se as expressões para as diferenciais que figuram na jacobina complexa (137):

$$\frac{\partial M_{kg}}{\partial V} = \frac{\partial ([V]Y^*V^* + [V^*]YV - 2P_{esp})}{\partial V} \quad (138)$$

$$\frac{\partial M_{kg}}{\partial V} = \frac{\partial ([Y^*V^*]V + [V^*]YV)}{\partial V} \quad (139)$$

$$\frac{\partial M_{kg}}{\partial V} = [Y^*V^*] + [V^*]Y \quad (140)$$

Procedendo de forma análoga para as outras diferenciais, chega-se à:

$$\frac{\partial M_{kg}}{\partial V^*} = [V]Y^* + [YV] \quad (141)$$

$$\frac{\partial M_k}{\partial V} = [Y^*V^*] \quad (142)$$

$$\frac{\partial M_k}{\partial V^*} = [V]Y^* \quad (143)$$

$$\frac{\partial E_{kg}}{\partial V} = [V^*] \quad (144)$$

$$\frac{\partial E_{kg}}{\partial V^*} = [V] \quad (145)$$

$$\frac{\partial M_{kg}^*}{\partial V} = [V^*]Y \quad (146)$$

$$\frac{\partial M_{kg}^*}{\partial V^*} = [YV] \quad (147)$$

para fins de organização, seja:

$$A = [Y^*V^*] \quad (148)$$

$$B = [V]Y^* \quad (149)$$

$$C = [V^*] \quad (150)$$

Então a estrutura simétrica da jacobiana complexa fica evidente:

$$J = \begin{bmatrix} A + B^* & A^* + B \\ A & B \\ C & C^* \\ B^* & A^* \end{bmatrix} \quad (151)$$

seja  $n = n_{PV} + n_{PQ}$ , então:

$$J = \begin{bmatrix} n_{PV} \times n & n_{PV} \times n \\ n_{PQ} \times n & n_{PQ} \times n \\ n_{PV} \times n & n_{PV} \times n \\ n_{PQ} \times n & n_{PQ} \times n \end{bmatrix}_{n \times n} \quad (152)$$

Pode ser visto em (152) as dimensões das matrizes envolvidas e que a jacobiana complexa tem dimensão  $n \times n$ . Para que se possa explorar e reduzir sua estrutura posteriormente, agrupa-se

a matriz jacobiana em quatro blocos matriciais, a saber:

$$J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \quad (153)$$

onde facilmente, se verifica que:

$$J_{11} = \begin{bmatrix} A + B^* \\ A \end{bmatrix} \quad (154)$$

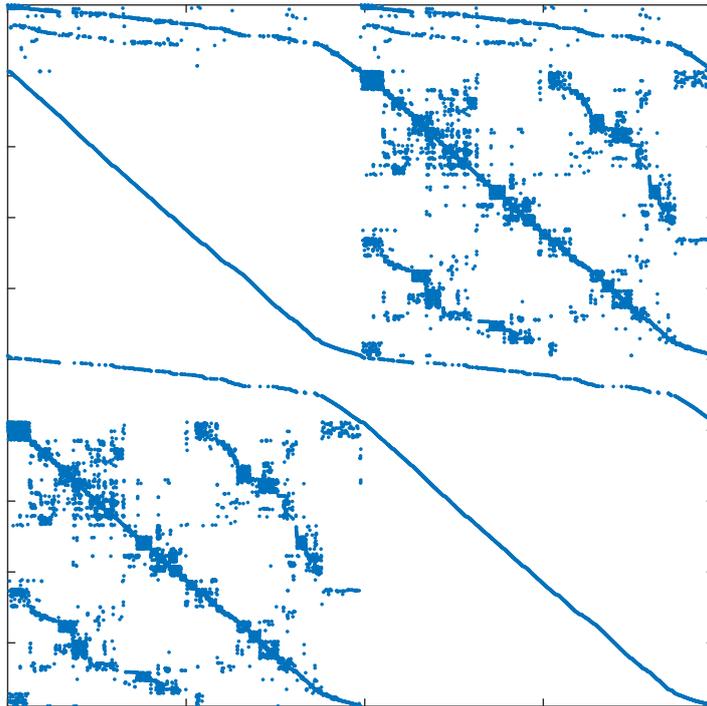
$$J_{12} = \begin{bmatrix} A^* + B \\ B \end{bmatrix} \quad (155)$$

$$J_{21} = \begin{bmatrix} C \\ B^* \end{bmatrix} \quad (156)$$

$$J_{22} = \begin{bmatrix} C^* \\ A^* \end{bmatrix} \quad (157)$$

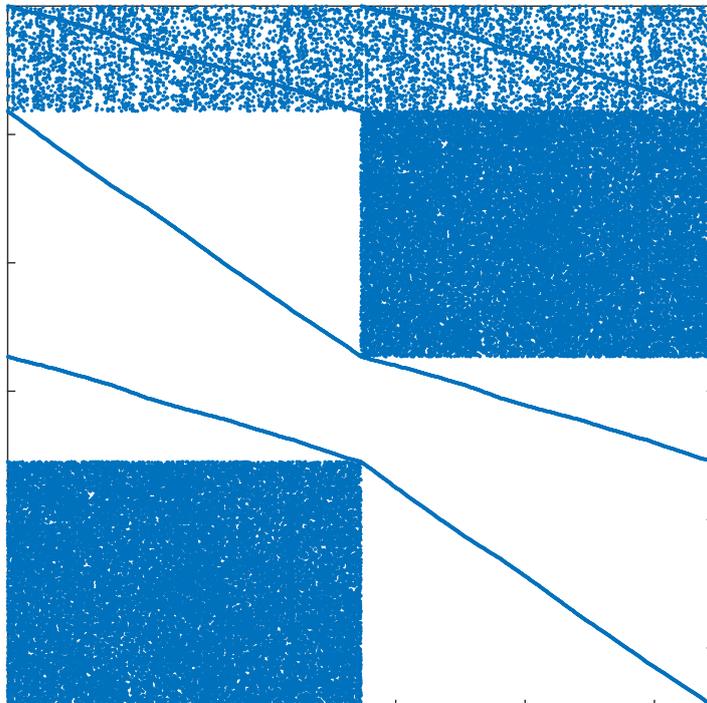
A seguir, dois exemplos de jacobiana complexa, o primeiro é de um sistema real de 9899 barras da região norte, feita a partir da base de dados do site *www.epe.gov.br* [28]. O segundo exemplo, é uma porção de um sistema de transmissão europeu de 13659 barras do projeto PEGASE. Os dados deste sistema foram fornecidos pelo software **MATPOWER**.

Figura 4.1: Jacobiana complexa de um sistema de 9899 barras.



Fonte: Próprio Autor

Figura 4.2: Jacobiana complexa do sistema europeu de 13659 barras.



Fonte: Próprio Autor

### 4.2.2 Complemento de Schur Aplicado ao Fluxo de Potência

O complemento de Schur pode ser uma poderosa ferramenta na resolução de sistemas lineares, pois nos permite eliminar blocos matriciais e assim, reduzir a dimensão do sistema. Primeiramente, retoma-se a formulação do fluxo segundo Wirtinger:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \Delta V \\ \Delta V^* \end{bmatrix} = - \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \quad (158)$$

geralmente, as matrizes A e D são mal condicionadas, portanto usando o operador **swap** pré multiplicando a matriz jacobiana, tem-se:

$$S \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \Delta V \\ \Delta V^* \end{bmatrix} = -S \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \quad (159)$$

$$\begin{bmatrix} C & D \\ A & B \end{bmatrix} \begin{bmatrix} \Delta V \\ \Delta V^* \end{bmatrix} = - \begin{bmatrix} M_2 \\ M_1 \end{bmatrix} \quad (160)$$

as igualdades matriciais acima nos levam em:

$$C\Delta V + D\Delta V^* = -M_2 \quad (161)$$

$$A\Delta V + B\Delta V^* = -M_1 \quad (162)$$

multiplicando a equação (162), pela inversa da matriz B tem-se:

$$B^{-1}A\Delta V + B^{-1}B\Delta V^* = -B^{-1}M_1 \quad (163)$$

$$B^{-1}A\Delta V + \Delta V^* = -B^{-1}M_1 \quad (164)$$

$$\Delta V^* = -B^{-1}A\Delta V - B^{-1}M_1 \quad (165)$$

substituindo  $\Delta V^*$  da equação (165), em (161), vem:

$$C\Delta V - DB^{-1}(A\Delta V + M_1) = -M_2 \quad (166)$$

$$(C - DB^{-1}A)\Delta V = DB^{-1}M_1 - M_2 \quad (167)$$

$$(J/B)\Delta V = DB^{-1}M_1 - M_2 \quad (168)$$

$$J_o\Delta V = M_o \quad (169)$$

A equação (169) nos dá um sistema linear reduzido. Esta redução do sistema vem com um preço, que é a inversão do bloco B. No entanto, este custo não precisa ser tão alto, dada a possibilidade de usar uma fatoração incompleta, como a LU incompleta, para a inversão do bloco em questão. Este procedimento corrobora o uso de redução por complemento de Schur e o uso de métodos iterativos.

### 4.2.3 Formulação Matricial do Fluxo de Potência Exato

O cálculo de Wirtinger matricial pode ser usado para a série de Taylor em ordens superiores. No entanto, as equações envolvidas no fluxo de potência, são equações de natureza quadrática, significando que a expansão da série de Taylor até segunda ordem implica a forma exata das equações do fluxo de potência, conclusão a que chegou Iwamoto em [29]. Nesta dissertação, apresenta-se a formulação do fluxo exato, em domínio complexo, conforme feito em [30]. Entretanto, o desenvolvimento aqui, é feito de forma matricial, explorando as bases do cálculo de Wirtinger generalizado, apresentado em seções anteriores. A vantagem da formulação matricial, é que se tem uma base sólida para desenvolvimentos mais gerais, além de que as fórmulas herdam a natureza vetorizada, que são mais facilmente exploradas por arquiteturas computacionais paralelas. A seguir, recobra-se o vetor de mismatches da formulação complexa, apresentado na equação (131), e faz-se a dedução do fluxo exato para o bloco  $M_k$ , uma vez que, para o restante dos blocos, a demonstração é análoga. Retomando o passo para a identificação das diferenciais de primeira ordem e ordem superiores dado em (29):

$$f(\mathbf{Z} + d\mathbf{Z}, \mathbf{Z}^* + d\mathbf{Z}^*) - f(\mathbf{Z}, \mathbf{Z}^*) = 1^{\mathbf{a}} \text{Ordem}(d\mathbf{Z}, \mathbf{Z}^*) + \text{Ordem Superior}(d\mathbf{Z}, \mathbf{Z}^*)$$

aplicando então ao bloco  $M_k$ :

$$\begin{aligned} M_k(V + dV, V^* + dV^*) - M_k(V, V^*) &= [V + dV]Y^*(V^* + dV^*) - S_{esp} - ([V]Y^*(V^*) - S_{esp}) \\ &= ([V] + [dV])Y^*(V^* + dV^*) - S_{esp} - ([V]Y^*(V^*) - S_{esp}) = \end{aligned} \quad (170)$$

$$= [V]Y^*V^* + [V]Y^*dV^* + [dV]Y^*V^* + [dV]Y^*dV^* - [V]Y^*V^* = \quad (171)$$

$$= [Y^*V^*]dV + [V]Y^*dV^* + [dV]Y^*dV^* \quad (172)$$

com o auxílio das equações (132) - (137) e da expansão da série de Taylor, identifica-se em (172), o que se segue:

$$M_k(V + dV, V^* + dV^*) - M_k(V, V^*) = [Y^*V^*]dV + [V]Y^*dV^* + [dV]Y^*dV^* = \quad (173)$$

$$M_k(V + \Delta V, V^* + \Delta V^*) \approx M_k(V_{cv}) + J\Delta V_{cv} + M_k(\Delta V_{cv}) + S_{esp} = 0 \quad (174)$$

$$M_k(V_{cv}) + J\Delta V_{cv} + M_k(\Delta V_{cv}) + S_{esp} = 0 \quad (175)$$

$$\Delta V_{cv} = -J^{-1} (M_k(V_{cv}) + M_k(\Delta V_{cv}) + S_{esp}) \quad (176)$$

$$\Delta V_{cv} = -J^{-1}M_e \quad (177)$$

onde  $\Delta V_{cv} = [\Delta V \ \Delta V^*]^T$ . A equação (177) nos dá o vetor de correções para o fluxo de potência exato e pode ser alcançada, de modo análogo, para os outros blocos. Note que  $M_e$  é o mismatch exato e retém a não linearidade do problema, ou seja, a matriz Hessiana está contemplada nesta expressão, onde é necessário apenas atualizar o vetor de correções.

## 5 Métodos Iterativos de Solução Numérica:

Na computação científica, a maior parte do tempo computacional é gasto na resolução de sistemas de equações lineares. Esses sistemas podem ser de grandes dimensões, como por exemplo, problemas de dinâmica de fluidos, onde cada equação descreve como o valor de um parâmetro local desconhecido (por exemplo, a velocidade local do fluido), depende de valores (desconhecidos) na vizinhança próxima. Um modelo computacional realista, frequentemente requer muitas equações, como o caso do problema de fluxo de potência em sistemas de grande porte. A ideia por trás dos métodos iterativos é substituir o sistema dado por um sistema aproximado que pode ser mais facilmente resolvido. Ou seja, em vez de  $Ax = b$ , resolve-se o sistema mais simples  $Kx_0 = b$  e toma-se  $x_0$  como uma aproximação para  $x$ . A seguir descreve-se brevemente alguns métodos iterativos, dentro os quais, métodos de gradientes conjugados, usados neste trabalho.

### 5.1 Métodos de Busca Linear Descendente

Nos últimos 40 anos, muitos algoritmos poderosos de busca direta foram desenvolvidos para a minimização irrestrita de funções gerais. Esses algoritmos exigem uma estimativa inicial

para o ponto ótimo, denotado por  $x^0$ . Com o ponto inicial estimado esses algoritmos geram uma sequência de estimativas  $x^0, x^1, x^2, x^3, \dots$ , onde a cada ponto se procura uma direção de descida sob a qual se determina o próximo ponto e assim sucessivamente até que um ou mais critérios de parada sejam atingidos e o ponto de parada  $x^k \cong x^*$  seja considerado o ponto ótimo do processo.

Básico para esses métodos é a seleção da direção  $u^{i+1}$  em cada iteração que garanta a descida do ponto  $x^i$  até o próximo ponto  $x^{i+1}$  nessa direção, portanto, é necessário que em cada ponto a derivada direcional seja negativa, garantindo que o caminho de busca na função objetivo  $f$  ou custo seja descendente:

$$\left. \frac{df(x^i)}{d\lambda} \right|_{u^{i+1}} = \nabla^T f(x^i) u^{i+1} < 0 \quad (178)$$

A estrutura geral dos métodos de busca linear descendente é dada abaixo:

1. Dado o ponto de partida  $x^0$  e as tolerâncias positivas  $\epsilon_1, \epsilon_2$  e  $\epsilon_3$  faz-se  $i = 1$ .
2. Seleciona-se a direção de descida inicial  $u^i$  tomando uma derivada direcional negativa da função objetivo.
3. Faz-se uma busca linear na direção de  $u^i$ , ou seja:

$$\min_{\lambda} F(\lambda) = \min_{\lambda} f(x^{i-1} + \lambda u^i). \quad (179)$$

4. Faz-se  $x^i = x^{i-1} + \lambda_i u^i$ .
5. Teste de convergência:

$$\text{Se } \|x^i - x^{i-1}\| < \epsilon_1, \text{ ou } \|\nabla f(x^i)\| < \epsilon_2, \text{ ou } |f(x^i) - f(x^{i-1})| < \epsilon_3 \quad (180)$$

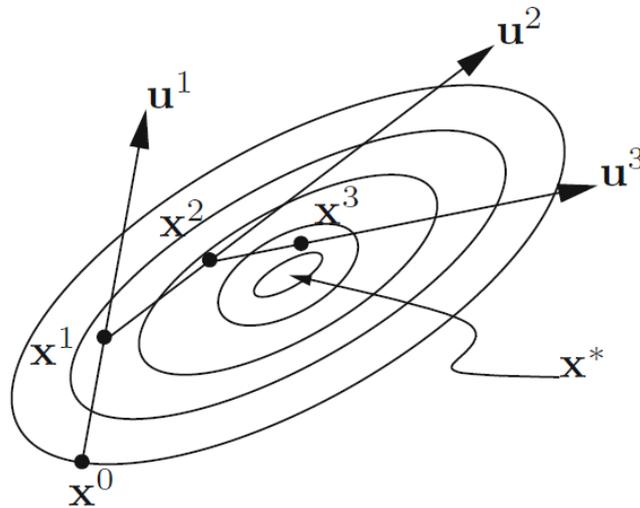
então pare e  $x^* \cong x^i$ , senão deve-se ir ao passo 6.

6. Faz-se  $i = i + 1$  e volta ao passo 2.

A estrutura do método de busca linear descendente descrita pode ser vista na figura 5.1. Os métodos dentro dessa classe diferem de acordo com a maneira como as direções de descida são escolhidas. Outra consideração importante na distinção de tais métodos é a forma como resolvem o sub problema unidimensional da busca linear descrita no passo 3 da estrutura geral do algoritmo. Existem vários algoritmos usados em otimização que resolvem o problema da

busca linear como por exemplo o *Método da Interpolação Quadrática de Powell*, o *Método da Secção de Ouro* e outros.

Figura 5.1: Sequência de direções e passos de descida



Fonte: Próprio Autor

### 5.1.1 Métodos de Busca Linear Descendente de Primeira Ordem

Métodos de busca linear descendente que fazem o uso do vetor gradiente  $\nabla f(x)$  para determinação da direção de busca a cada iteração, são chamados de métodos de primeira ordem pois utilizam as derivadas parciais de primeira ordem da função objetivo  $f(x)$ . O mais simples e mais famoso dessa classe de algoritmos é o *Método da Máxima Descida*, primeiramente proposto por Cauchy em 1847. Seu algoritmo pode ser descrito da seguinte forma:

Dado o ponto de partida  $x^0$  faça para  $i = 1, 2, 3, \dots$  até a convergência:

1. Faz-se  $u^i = \frac{-\nabla f(x^{i-1})}{\|\nabla f(x^{i-1})\|}$
2. Faz-se  $x^i = x^{i-1} + \lambda_i u^i$  onde  $\lambda_i$  é tal que:

$$\min_{\lambda} F(\lambda) = \min_{\lambda} f(x^{i-1} + \lambda u^i) \quad (\text{Busca Linear})$$

Os critérios de parada podem ser os mesmos já citados anteriormente:

1.  $\|x^i - x^{i-1}\| < \epsilon_1$
2.  $\|\nabla f(x^i)\| < \epsilon_2$

$$3. |f(x^i) - f(x^{i-1})| < \epsilon_3.$$

Onde  $\epsilon_1$ ,  $\epsilon_2$  e  $\epsilon_3$  são tolerâncias positivas dadas.

### 5.1.2 Condições de Melhoria de Convergência

Associadas aos custos computacionais dos métodos de busca linear descendentes e em geral dos métodos de primeira ordem, condições de melhoria de performance e economia foram desenvolvidas ao longo do tempo. Essas condições foram desenvolvidas na tentativa de garantir que os passos utilizados nas direções de busca não sejam tão grandes que possam provocar a divergência da sequência de busca nem tão pequenos que tornem o progresso da busca insuficiente. As seguintes quatro condições foram largamente utilizadas com sucesso na atualização dos passos em problemas de otimização irrestrita e por isso merecem citação:

1. Performance:

$$f(x^{i-1} + \lambda_i u^i) \leq f(x^{i-1}) \quad (181)$$

2. Armijo:

$$f(x^{i-1} + \lambda_i u^i) \leq f(x^{i-1}) + c_1 \lambda_i u^{iT} \nabla f(x^{i-1}) \quad (182)$$

3. Curvatura:

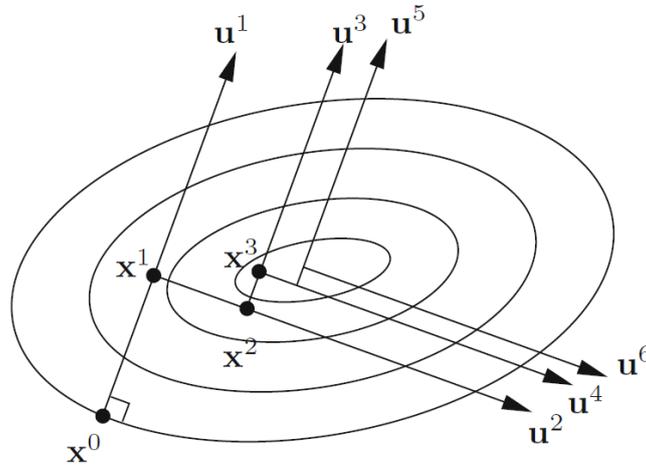
$$c_2 u^{iT} \nabla f(x^{i-1}) \leq u^{iT} \nabla f(x^{i-1} + \lambda_i u^i) \quad (183)$$

4. Curvatura Forte:

$$|u^{iT} \nabla f(x^{i-1} + \lambda_i u^i)| \leq c_3 |u^{iT} \nabla f(x^{i-1})| \quad (184)$$

Onde  $c_1$ ,  $c_2$  e  $c_3$  são parâmetros necessariamente não negativos que controlam o grau sob o qual as condições citadas são reforçadas. A segunda e a terceira condições são conhecidas como *Condições de Wolfe* enquanto que a segunda e quarta são conhecidas como *Condições Fortes de Wolfe*.

Figura 5.2: Comportamento de zig zag ortogonal do método de Máxima Descida



Fonte: Próprio Autor

## 5.2 Método de Gradientes Conjugados (CG)

Apesar de sua descida local ótima o método da Máxima Descida geralmente tem uma pobre performance, sobretudo quando o problema de otimização considerado é mal escalado, ou seja, quando as curvas de nível da função objetivo são muito alongadas. Esse problema no desempenho desse método se dá principalmente porque o algoritmo força a busca direções ortogonais à cada iteração como mostrado na figura 5.2. Embora na teoria o método se prove convergente, na pratica ele pode não convergir de forma eficiente em um número finito de iterações. Dependendo do ponto inicial este métodos pode divergir até mesmo quando aplicado a funções quadráticas positiva definidas. Existe, entretanto, uma classe de métodos de primeira ordem conhecidos como Métodos de Gradientes Conjugados, para os quais pode ser demonstrado que independente da escala do problema, esse método converge em não mais que um predeterminado número  $n$  finito de iterações quando aplicado a funções quadráticas da forma:

$$f(x) = \frac{1}{2}x^T Ax + b^T x + c \quad (185)$$

onde  $c \in \mathbb{R}$ ,  $b$  é um vetor real de dimensão  $n$  e  $A$  é uma matriz  $n \times n$  real e simétrica. Essa poderosa propriedade pode ser estendida à outras funções mais gerais não quadráticas pois muitas funções podem se aproximar localmente por *Séries de Taylor* das condições dadas.

O método de gradientes conjugados começou a ser estudado na década de 50, sendo inicialmente proposto como um método direto para resolver sistemas lineares, não como um método iterativo. Muitos autores citam os trabalhos de Hestenes e Stiefel [31] como referência base ao que foi desenvolvido posteriormente. Segundo [32] o Método de Gradientes Conjugados foi abandonado durante a década de 60, e apenas nos anos 70 foi retomado pela comunidade científica. O método é utilizado principalmente para a resolução de problemas de grande porte, nos quais não é razoável sequer fazer as  $n$  iterações que podem ser demandadas no caso quadrático estritamente convexo.

### 5.2.1 Direções Mutuamente Conjugadas

Dois vetores  $u, v \neq 0$  são ortogonais se o produto escalar  $u^T v = (u, v) = 0$ . O conceito de *conjugação mutua* pode ser definido de maneira semelhante. Dois vetores  $u, v \neq 0$ , são definidos mutuamente conjugados com respeito a matriz simétrica positiva definida  $A$  se  $u^T A v = (u, Av) = 0$ . As direções conjugadas podem ser usadas na seleção das direções de busca dos algoritmos de primeira ordem, evitando o problema das direções ortogonais a cada iteração como ocorre no método da Máxima Descida porém como encontrar tais direções mutuamente conjugadas? Uma maneira seria obter os autovetores da matriz  $A$  porém resolver essa situação por si só representaria um problema computacional da mesma dimensão que o problema original de otimização irrestrita que poderia ser resolvido por qualquer outro método numérico. Para contornar essa situação e obter as direções mutuamente conjugadas o método de *Fletcher-Reeves* foi proposto [33].

### 5.2.2 Direções de Fletcher-Reeves

As *Direções de Fletcher-Reeves*  $u^i$ ,  $i = 1, 2, \dots, n$ , considerando uma matriz  $A$  simétrica positiva definida, podem ser encontradas explicitamente da seguinte forma:

$$u^1 = -\nabla f(x^0) \tag{186}$$

e para  $i = 1, 2, \dots, n - 1$

$$u^{i+1} = -\nabla f(x^i) + \beta_i u^i \tag{187}$$

onde  $x^i = x^{i-1} + \lambda_i u^i$ ,  $\lambda_i$  corresponde ao passo ótimo na interação  $i$  e:

$$\beta_i = \frac{\|\nabla f(x^i)\|^2}{\|\nabla f(x^{i-1})\|^2} \quad (188)$$

A mudança de  $\beta_i$  dá origem à outros métodos de busca como por exemplo as *Direções de Polak-Ribiere* dadas por:

$$\beta_i = \frac{(\nabla f(x^i) - \nabla f(x^{i-1}))^T \nabla f(x^i)}{\|\nabla f(x^{i-1})\|^2} \quad (189)$$

Portanto a estrutura geral do algoritmo de gradiente conjugado de Fletcher-Reeves para funções gerais, pode ser sintetizado nos seguintes passos:

---

**Algoritmo 1:** ALGORITMO CG

---

- 1 Computa-se  $\nabla f(x^0)$  e faz-se  $u^1 = -\nabla f(x^0)$
  - 2 Para  $i = 1, 2, \dots, n$  é feito:
  - 3 Faz-se  $x^i = x^{i-1} + \lambda_i u^i$  onde  $\lambda_i$  é tal que:
  - 4  $\min_{\lambda} F(\lambda) = \min_{\lambda} f(x^{i-1} + \lambda u^i)$  (*Busca Linear*)
  - 5 Computa-se  $\nabla f(x^i)$
  - 6 Se o critério de convergência é satisfeito então pare e  $x^* \cong x^i$ , senão prossiga.
  - 7 Se  $i \leq n - 1$ ,  $u^{i+1} = -\nabla f(x^i) + \beta_i u^i$
  - 8 Faz-se  $x^0 = x^n$  e volte ao passo 1.
- 

### 5.3 Método do Gradiente Bi-Conjugado (BiCG)

O método dos gradientes conjugados (*CG*) descrito anteriormente, não é adequado para solucionar os sistemas lineares de fluxo de carga, já que são aptos apenas para solucionar sistemas simétricos e para que sejam aplicados aos casos não simétricos necessitam da equação normal de gauss ou outras técnicas de simetrização, no entanto, quase todas elas aumentam o número de condicionamento da matriz do sistema. A primeira tentativa para solucionar este impasse é apresentada em [34], com o método conhecido como Gradiente Bi-conjugado (*BiCG*) sendo posteriormente melhorado. O BiCG utiliza uma das formas do algoritmo Lanczos não-simétrico, conhecida como biortogonalização de Lanczos ou simplesmente Bi-Lanczos, para construir as bases ortogonais do Subespaço de Krylov. Em resumo este algoritmo é uma variação do tradicional CG e tem em adicional o cálculo da bi-direção e do bi-resíduo. Como vantagem aos métodos de gradientes tradicionais ele pode ser aplicado à matrizes não simétricas reais sem

alteração do número de condicionamento. A seguir apresenta-se o algoritmo de BiCG:

---

**Algoritmo 2:** ALGORITMO BiCG PRÁTICO

---

- 1 Escolha uma suposição inicial  $x_o$  e faça  $x = x_o$
  - 2  $r_o = (b - Ax_o)$  (resíduo inicial)
  - 3  $\hat{r}_o = r_o^*$  (resíduo conjugado inicial)
  - 4  $p_o = r_o$  (direção de busca inicial)
  - 5  $\hat{p}_o = p_o^*$  (direção de busca inicial conjugada)
  - 6 Sendo  $tol$  a tolerância para o resíduo enquanto  $\|r\| > tol$  faça:
  - 7  $\alpha_i = \frac{\hat{r}_i^T r_i}{\hat{r}_i^T A p_i}$
  - 8  $x_{i+1} = x_i + \alpha_i p_i$
  - 9  $r_{i+1} = r_i - \alpha_i A p_i$
  - 10  $\hat{r}_{i+1} = \hat{r}_i - \alpha_i^* A^H \hat{p}_i$
  - 11  $\beta_i = \frac{\hat{r}_{i+1}^T r_{i+1}}{\hat{r}_i^T r_i}$
  - 12  $p_{i+1} = r_{i+1} + \beta_i p_i$
  - 13  $\hat{r} = \hat{r} + \beta_i^* \hat{r}_i$
- 

Apesar do BiCG se apresentar como uma extensão do CG para matrizes não simétricas ele ainda exige que a matriz seja real. Para que se possa estender os algoritmos ao plano complexo necessita-se de mais uma generalização a matrizes não simétricas e complexas tornando o algoritmo de gradientes conjugados ideal para a aplicação a problemas de fluxo de potência em regime permanente em domínio complexo. Felizmente encontra-se tal generalização em [35] no qual o autor batiza o algoritmo de Comp - BiCG. Por motivos de simplificação o algoritmo é chamado apenas de **CBiCG** - *Complex Biconjugate Gradient*. A dedução desse algoritmo se encontra em [35] e se constitui em uma generalização dos dois métodos anteriores, CG e BiCG. A seguir é descrita a rotina de cálculo do CBiCG:

---

**Algoritmo 3: ALGORITMO CBICG PRÁTICO**


---

- 1 Escolha uma suposição inicial  $x_o$  e faça  $x = x_o$
  - 2  $r = (b - Ax)$  (resíduo inicial)
  - 3  $\hat{r} = r^*$  (resíduo conjugado inicial)
  - 4  $p = r$  (direção de busca inicial)
  - 5  $\hat{p} = p^*$  (direção de busca inicial conjugada)
  - 6 Sendo  $tol$  a tolerância para o resíduo enquanto  $\|r\| > tol$  faça:
  - 7  $\alpha = \frac{\hat{r}^T r}{\hat{r}^T A p}$
  - 8  $x = x + \alpha p$
  - 9  $r = r - \alpha A p$
  - 10  $\hat{r} = \hat{r} - \alpha^* A^H \hat{p}$
  - 11  $\beta = \frac{(-A^H \hat{p})^T r}{\hat{p}^T A p}$
  - 12  $p = r + \beta p$
  - 13  $\hat{p} = \hat{r} + \beta^* \hat{p}$
- 

### 5.3.1 Condicionamento Numérico e Convergência do CBiCG

Em termos gerais, um sistema linear  $Ax = b$  é mal-condicionado se pequenas perturbações na matriz  $A$  ou no vetor  $b$  causam grandes variações na solução. Nesse caso, devido aos erros na representação e operações de pontos flutuantes, não se deve esperar uma solução precisa de um método numérico! Além disso, num sistema linear mal-condicionado, o resíduo pode não revelar a natureza do erro! O condicionamento de uma matriz  $A$  é definido em termos de sua norma e a norma de sua inversa. O número de condição de uma matriz  $A$ , também chamado condicionamento de  $A$ , é definido por  $cond(A) = \|A\| \|A^{-1}\|$ . Este número é de grande importância pois afeta fortemente a convergência da maioria dos métodos iterativos e, em específico, os métodos de gradientes conjugados que apesar de sua versatilidade para operar tanto em ambiente sequencial quanto em paralelo, possuem o grande problema característico de instabilidade numérica perante as diferentes propriedades do sistema linear o que faz com que esses métodos possam apresentar uma convergência lenta ou ainda não convergir à solução exata. Isto acontece em particular se a matriz dos coeficientes é mal condicionada, onde o fato de existir uma grande diferença entre a magnitude dos autovalores origina erros de arredondamento por conta da perda de alguns algarismos significativos no processo da aplicação dos métodos de gradientes conjugados, porém, tanto a eficiência quanto a robustez desses métodos podem ser muito

aprimoradas usando pré-condicionamento. O pré condicionamento é simplesmente transformar o sistema linear original em um equivalente, isto é, com a mesma solução, porém com alguma melhoria nas características matriciais do sistema acarretando uma convergência mais rápida e mais robusta. Citando Saad em 36: “Em geral, a confiabilidade dos métodos iterativos, ao lidar com várias aplicações, depende muito mais da qualidade do pré-condicionador...”. Dessa forma percebe-se que a instabilidade numérica dos métodos de gradientes conjugados, em especial do CBiCG, pode ser contornada ou melhorada, com a utilização de um pré-condicionador conveniente, o que requer estudo dos casos específicos de aplicação.

## 5.4 Precondicionadores

Os métodos de gradientes conjugados fazem parte do subespaço de *Krylov*, eles não são robustos no sentido de que não se pode garantir soluções aproximadas em tempo computacional e armazenamento modestos (modesto com respeito a métodos alternativos de solução). Para alguns métodos (por exemplo, GMRES), é óbvio que eles levam, em aritmética exata, à solução exata em  $n$  iterações máximas, mas isso pode não ser muito prático. Outros métodos estão restritos a classes específicas de problemas (GC, MINRES) ou sofrem de efeitos colaterais desagradáveis, como estagnação ou paradas (Bi-CG, CBiCG, Bi-CGSTAB). Tal baixa convergência depende de maneira muito complicada das propriedades espectrais (distribuição dos auto-valores, campo de valores, condição do auto-sistema, etc.) e estas informações nem sempre estão disponíveis em situações práticas. A alternativa é, então, tentar encontrar algum operador  $K$ , de modo que  $K^{-1}A$  tenha melhores propriedades espectrais (mas ainda desconhecidas). Isto é baseado na observação de que para  $K = A$ , o sistema ideal  $K^{-1}Ax = Ix = K^{-1}b$  e todos os métodos do subespaço forneceriam a solução verdadeira em uma única etapa. A esperança é que para  $K$ , em certo sentido próximo de  $A$ , um método de Krylov selecionado (caso deste trabalho, o CBiCG) e corretamente aplicado para, por exemplo,  $K^{-1}Ax = K^{-1}b$ , precisaria de apenas algumas iterações para gerar uma aproximação suficientemente boa para a solução do sistema dado  $Ax = b$ . Um operador usado para esse fim é chamado de pré-condicionador para a matriz  $A$ .

O problema geral de encontrar um pré-condicionador eficiente é identificar um operador linear  $K$  (o preconditionador) com propriedades tais que:

1.  $K$  é uma boa aproximação de  $A$  em algum sentido.

2. O custo computacional da construção de  $K$  é não proibitivo.
3. O sistema  $Ky = z$  é muito mais fácil de se resolver do que o sistema original.

A pesquisa sobre pré-condicionamento é uma área muito ampla e ativa porém ainda com pouca estrutura. Não existe uma teoria geral sobre a qual é possível se basear com segurança para uma seleção eficiente do pré-condicionador. A principal dificuldade é que o pré-condicionamento se baseia em aproximação e na ausência de informações precisas sobre o comportamento da solução de um dado sistema  $Ax = b$  e das propriedades espectrais de  $A$ , pode ocorrer que a convergência dependa criticamente da informação que é descartada no processo de aproximação. Seleção e construção de um bom pré-condicionador para uma dada classe de problemas é, portanto, na melhor das hipóteses, uma mera suposição.

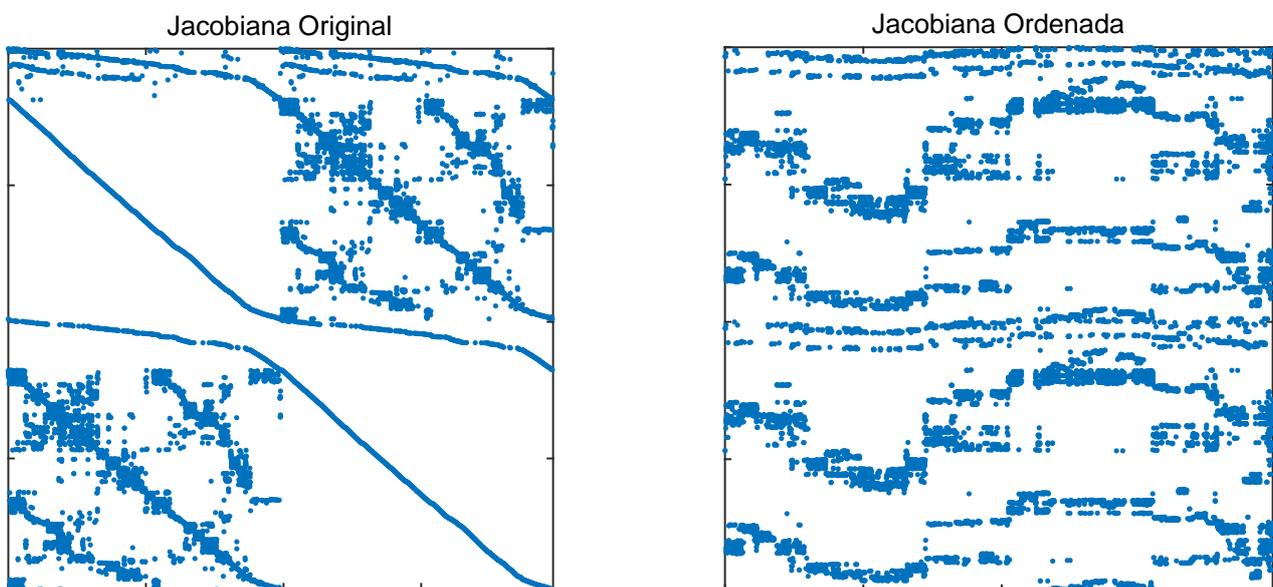
Existe uma grande liberdade na definição e construção de pré-condicionadores para os métodos do subespaço Krylov e essa é uma das razões pelas quais esses métodos são tão populares e tão bem sucedidos. Observe que em todos os métodos de Krylov (CG, BiCG, CBiCG e etc), não é necessário conhecer os elementos individuais de  $A$  e nem modificar suas partes, necessita-se apenas de uma sub-rotina que gera, para um dado vetor  $y$  de entrada, um vetor  $z$  de saída tal que matematicamente pode ser descrito como  $z = Ay$  e esse é mais um atrativo dos algoritmos de gradientes conjugado.

## 5.5 Efeito da Ordenação

É sabido que a ordenação das variáveis pode ter um efeito significativo na convergência dos métodos de gradientes conjugados como pode ser visto, por exemplo, no trabalho 37 ou [38] no qual os autores investigam os efeitos de várias estratégias de ordenação e particionamento no desempenho do algoritmo CG paralelo pré-condicionado. Já existe uma considerável literatura sobre os efeitos da ordenação em problemas de diferenças finitas, sobretudo quando se aplica métodos de gradientes conjugados em sua forma paralelizada pois uma ordenação específica pode melhorar a performance da computação paralela ao ordenar convenientemente as variáveis além de possibilitar a redução da quantidade de fill-in's e manter a estrutura de esparsidade. Em muitos casos, bons resultados foram obtidos com ordenações baseadas em sistemas reduzidos como a ordenação diagonal, espiral, red-black e outras. A título de exemplo da profunda alteração causada pela ordenação, na estrutura matricial, aplica-se a ordenação *Column Approximate Minimum Degree* (CAMD), à matriz jacobiana computada em flat-start,

de um sistema de 9899 barras, extraído da base de dados da Empresa de Pesquisa Energética (EPE), em seu site *www.epe.gov.br* [28]. Este sistema trata-se da configuração da rede elétrica representada nos dados de estudos da transmissão, que são atualizados a partir dos resultados dos estudos de expansão regionais (Relatórios R1), dos empreendimentos de transmissão já licitados e daqueles já indicados para licitação, das projeções de mercado e do plano de geração e dos requisitos de intercâmbios entre subsistemas. A jacobiana foi calculada utilizando-se de algoritmos desenvolvidos nesta dissertação.

Figura 5.3: Ordenação CAMD do sistema de 9899 barras



## 5.6 Algoritmos Desenvolvidos em MATLAB

Nesta seção descreve-se brevemente os algoritmos de fluxo de potência desenvolvidos em domínio real e complexo, utilizando-se do software *MATLAB*, os quais foram desenvolvidos e implementados ao longo deste trabalho de mestrado.

1. Um arquivo de extensão *.txt* formatado nos moldes do IEEE (Common Data Format), contendo os dados do problema de fluxo de potência a ser estudado.
2. Um arquivo de extensão *.pwf* o mesmo formato usado para processar casos no programa *ANAREDE*.
3. Um arquivo de extensão *.mat* o mesmo formato usado para processar casos no programa *MATPOWER*.

Ainda sobre o Newton Raphson tradicional implementado, ele se constitui no estado da arte desse método mas vale ressaltar que a montagem da matriz de admitâncias e do jacobiano foi feita de forma otimizada utilizando formas vetorizadas conforme 39. Tais formas de implementação figuram também nas versões mais recentes do software *MATPOWER* e podem ser vistas com detalhes no manual desse programa [15]. Os demais algoritmos foram feitos utilizando-se dos cálculos já explicados nas seções anteriores como o método de Newton Raphson no domínio complexo, o CG - 8, BiCG - 13 e o CBiCG.

Apesar de alguns métodos terem sido apresentados, para as análises e comparações dos resultados obtidos será utilizado apenas o método CBiCG em comparação ao método de Newton Raphson tradicional em domínio real utilizando coordenadas polares *RV-NRP*. Para análise do desempenho, utiliza-se o algoritmo CBiCG no domínio real da seguinte forma:

---

**Algoritmo 4:** METODOLOGIA DO CBiCG

---

- 1 Escolha uma suposição inicial  $x_o$  e faça  $x = x_o$
  - 2  $J_o = J(x_o)$  (Jacobiano em  $x_o$ )
  - 3 Sendo  $tolM$  a tolerância para o Mismatch enquanto  $\|M\| > tolM$  faça:
  - 4  $dx = cbicg(J_o, -M(x), x_{cbi_o}, bitol)$
  - 5  $x = x + dx$
- 

Nota-se nesse algoritmo que  $J_o$  é constante, a jacobiana é calculada apenas uma vez no ponto inicial  $x_o$  e o algoritmo CBiCG entra em ação resolvendo o sistema linear  $J_o dx^i = -M(x^i)$  para cada iteração  $i$  e  $bitol$  é a tolerância especificada para a resolução desse sistema através do CBiCG. Então para cada iteração  $i$  tem-se sub iterações da rotina do CBiCG na resolução de  $J_o dx^i = -M(x^i)$  até a convergência global do algoritmo em  $i = n$  quando a tolerância para o Mismatch ou resíduo  $\|M\| < tolM$  é alcançada. Note que o algoritmo global pode divergir se para um número máximo  $N$  de iterações  $i$  não for satisfeita a condição  $\|M\| < tolM$  ou se dentro da rotina do CBiCG não for alcançada a convergência na resolução do sistema linear para cada iteração  $i$ .

## 6 Apresentação dos Resultados

Neste capítulo apresenta-se os resultados das simulações utilizando os algoritmos desenvolvidos em domínio complexo (*CBiCG* e *CBiCG - ILU*) e em domínio real *RV - NRP*, todos em

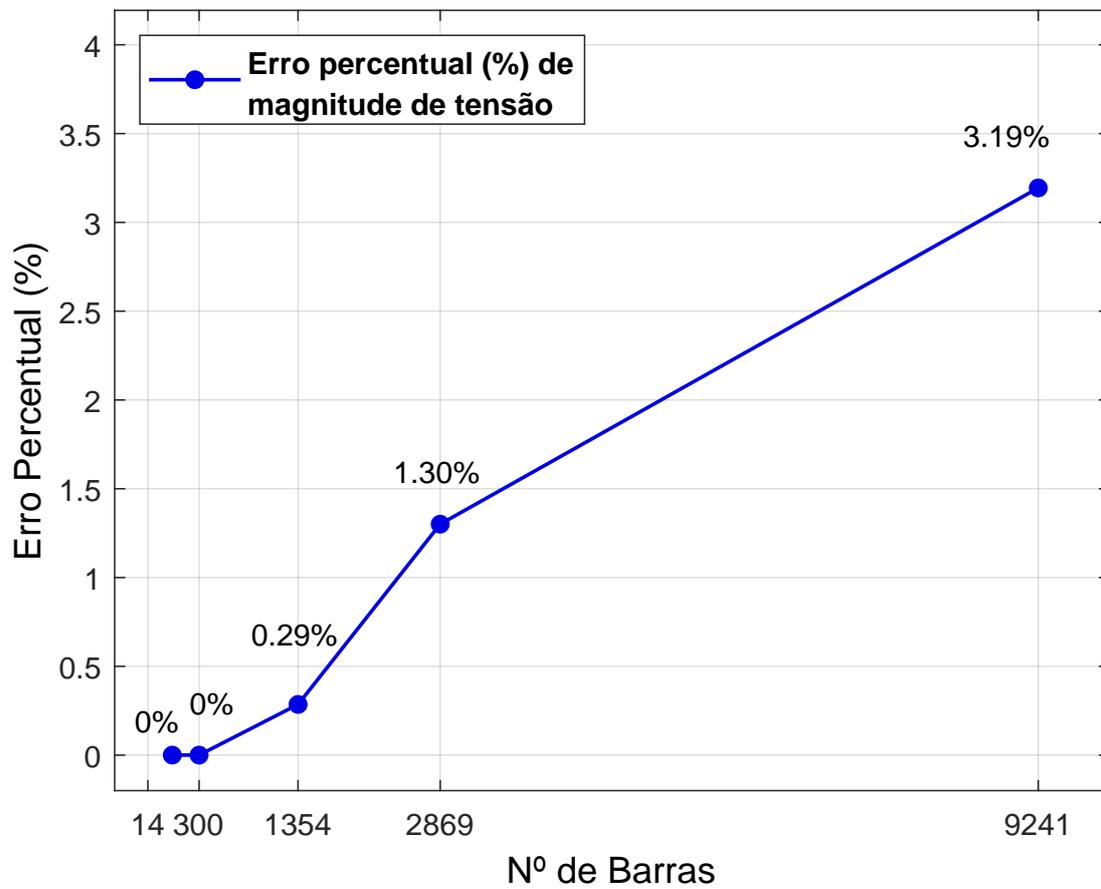
coordenadas polares. Foram feitos comparativos de performance entre as duas formulações do fluxo de potência, para os sistemas fornecidos pelo software *MATPOWER* de 14, 300, 1354, 2869 e 9241 barras com características de condicionamento distintas. As comparações foram feitas levando-se em conta alguns aspectos, como o uso de ordenação por colunas no fluxo complexo *CBiCG-AMD*, preconditionadores através do algoritmo *CBiCG-ILU*, usando fatoração LU incompleta, aplicação do complemento de schur para a redução do sistema *CBiCG-ILU<sub>schur</sub>* e, finalmente, o fluxo de potência exato, *CBiCG-EPF*. Todos os algoritmos envolvidos foram feitos utilizando-se técnicas de esparsidade e foram desenvolvidos no software *Matlab*. Os testes numéricos foram executados usando-se um Intel Core i7 modelo OptiPlex3090 com 16GB de RAM e sistema operacional windows 11, 64-bits.

## 6.1 Validação do Fluxo em domínio Complexo

### 6.1.1 Validação dos Módulos de Tensão e Tempo Computacional

Para fins de validação do fluxo de potência complexo através do algoritmos de gradientes bi-conjugados *CBiCG* e *CBiCG-ILU*, utiliza-se o software de simulação de código aberto *MATPOWER*. A seguir a comparação dos módulos de tensão, dados como resultado da convergência do algoritmo *CBiCG*, com o fluxo em domínio real *RV-NRP* em coordenadas polares, dado pelo software *MATPOWER*. A comparação foi feita levando-se em conta o erro percentual entre os módulos de tensão, para os sistemas de barras de 14, 300, 1354, 2869 e 9241. É possível consultar o manual do *MATPOWER*, para maiores detalhes sobre os sistemas de barras utilizados, vide referência [15].

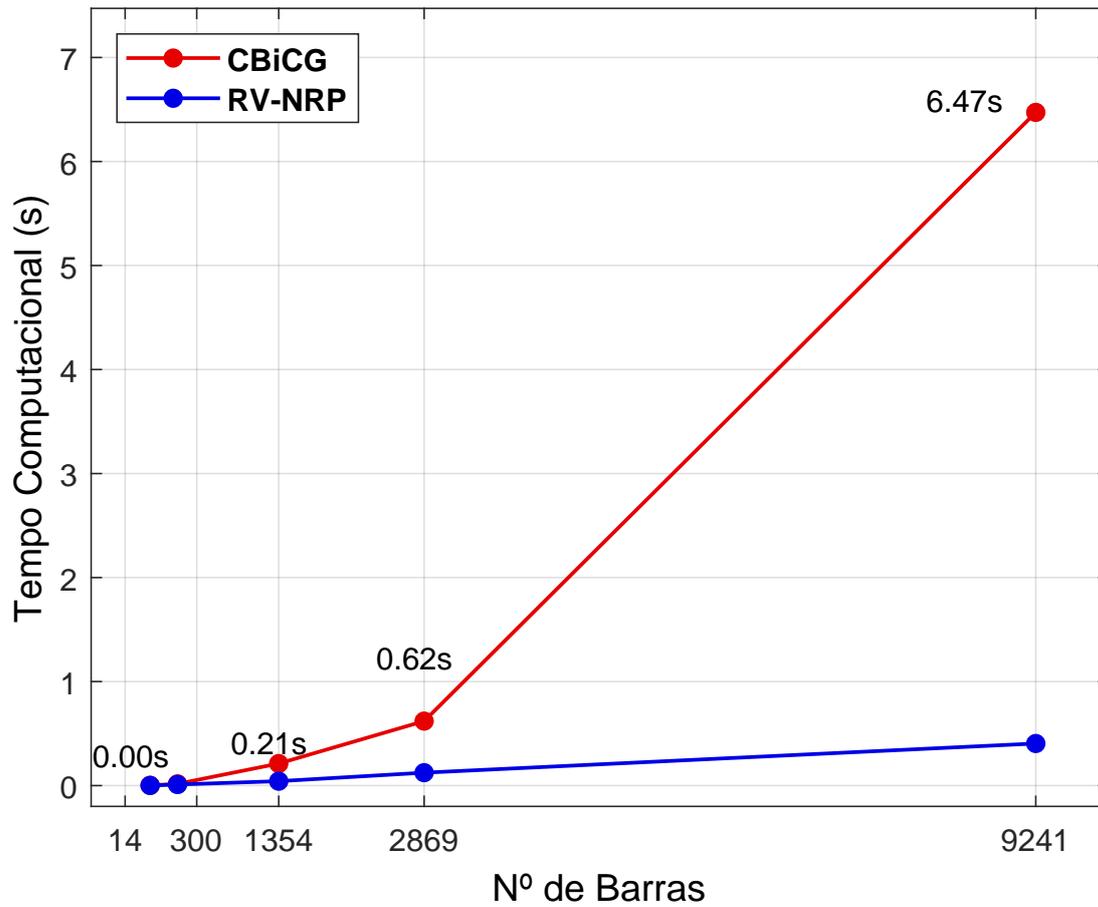
Figura 6.1: Erros percentuais de magnitude de tensão



Fonte: Próprio Autor

O maior erro percentual cometido foi de 4% e se deu no maior caso testado, de 9241 barras. Vale ressaltar que nesta primeira simulação, não foi usada nenhuma técnica de melhoria de condicionamento numérico da matriz jacobiana, o que foi feito nas simulações posteriores. Em seguida, compara-se o tempo computacional, em segundos, desta primeira simulação.

Figura 6.2: Tempo computacional em segundos



Fonte: Próprio Autor

É perceptível, o aumento expressivo do tempo computacional como função da dimensão do sistema, para o algoritmo *CBiCG*. Essa mesma relação é mais discreta e linear para o algoritmo *RV-NRP*, em domínio real. Apesar de o tempo computacional ser sensível a dimensão do sistema, a diferença entre o número de iterações se manteve aproximadamente nula, como se pode ver na tabela 1.

### 6.1.2 Número de iterações: CBiCG x MATPOWER

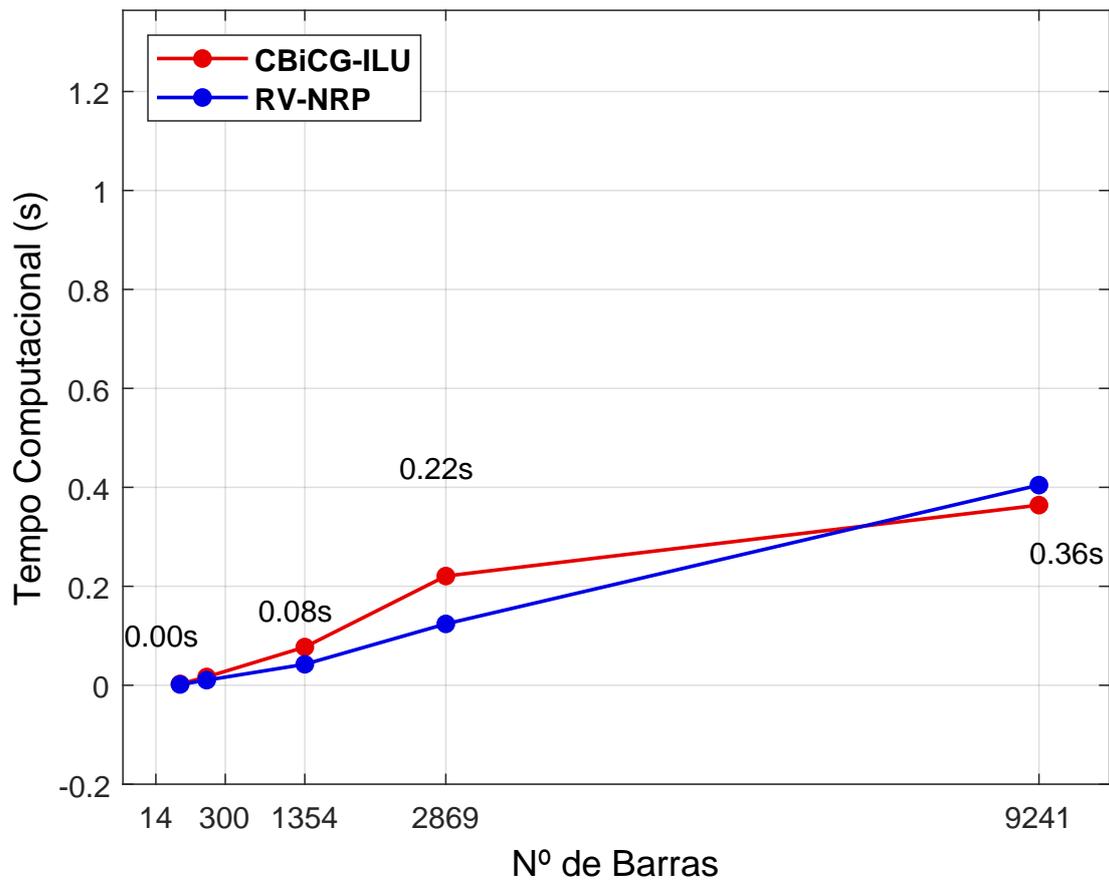
Tabela 1: Tabela de Iterações CBiCG x MATPOWER

Barras	Iterações CBiCG	Iterações RV-NRP
5	2	2
9	3	3
14	3	3
18	3	3
22	2	2
30	3	2
39	4	3
57	3	3
69	3	2
85	3	3
118	4	3
141	2	2
145	7	4
300	5	4
1354	5	4
2869	6	6
3120	5	5
9241	7	6

### 6.1.3 CBiCG Precondicionado

O tempo computacional divergiu de forma mais acentuada que as interações, isso se deu, porque a aplicação do algoritmo *CBiCG* ao fluxo de potência, utilizando Newton-Raphson, requer dois laços, um externo e outro interno onde ocorre efetivamente a resolução do sistema linear  $J(x)dx = -M(x)$ . Este laço interno é mais suscetível ao número de condicionamento da matriz jacobina, o que pode causar um queda de performance no número de iterações do laço externo. Na simulação seguinte, técnicas de ordenação e preconditionamento, promovem uma melhoria significativa dos tempos computacionais para o algoritmo em domínio complexo. No caso a seguir utiliza-se, como preconditionador, a fatoração LU incompleta.

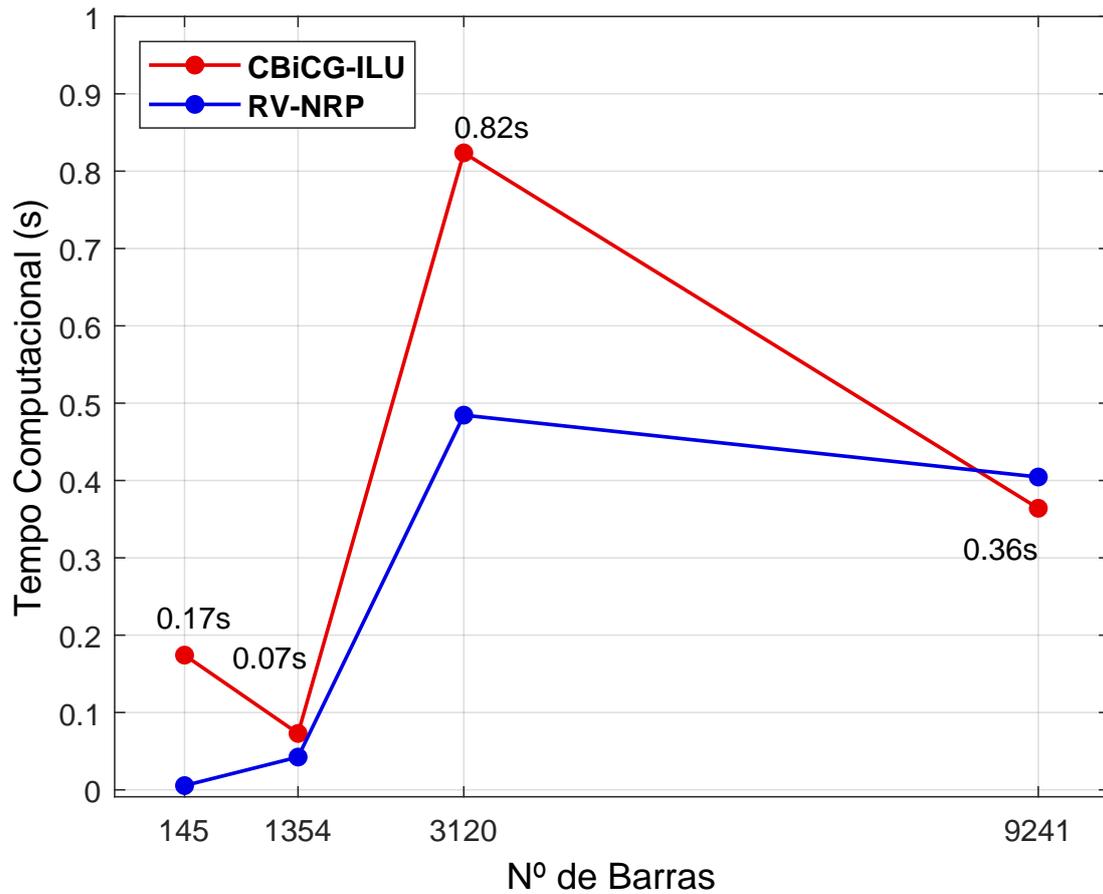
Figura 6.3: CBiCG com preconditionador



Fonte: Próprio Autor

Na figura 6.3 pode-se perceber os efeitos do preconditionador. Nesta simulação o efeito de performance computacional foi mais acentuado no maior sistema, de 9241, onde se obteve uma melhoria no tempo computacional de quase 20 vezes. No entanto esta melhoria está sujeita a morfologia do sistema, o que quer dizer, que o preconditionador utilizado para um sistema, pode não promover a melhoria nos demais sistemas. A seguir, apresenta-se o caso onde sistemas menores de 145 barra e 3120 barras tiveram suas performances prejudicasm utilizando-se o mesmo preconditionar que promoveu a melhoria do sistema de 9241 barras.

Figura 6.4: CBiCG-ILU preconditionado incluindo sistemas de 145 e 3120 barras



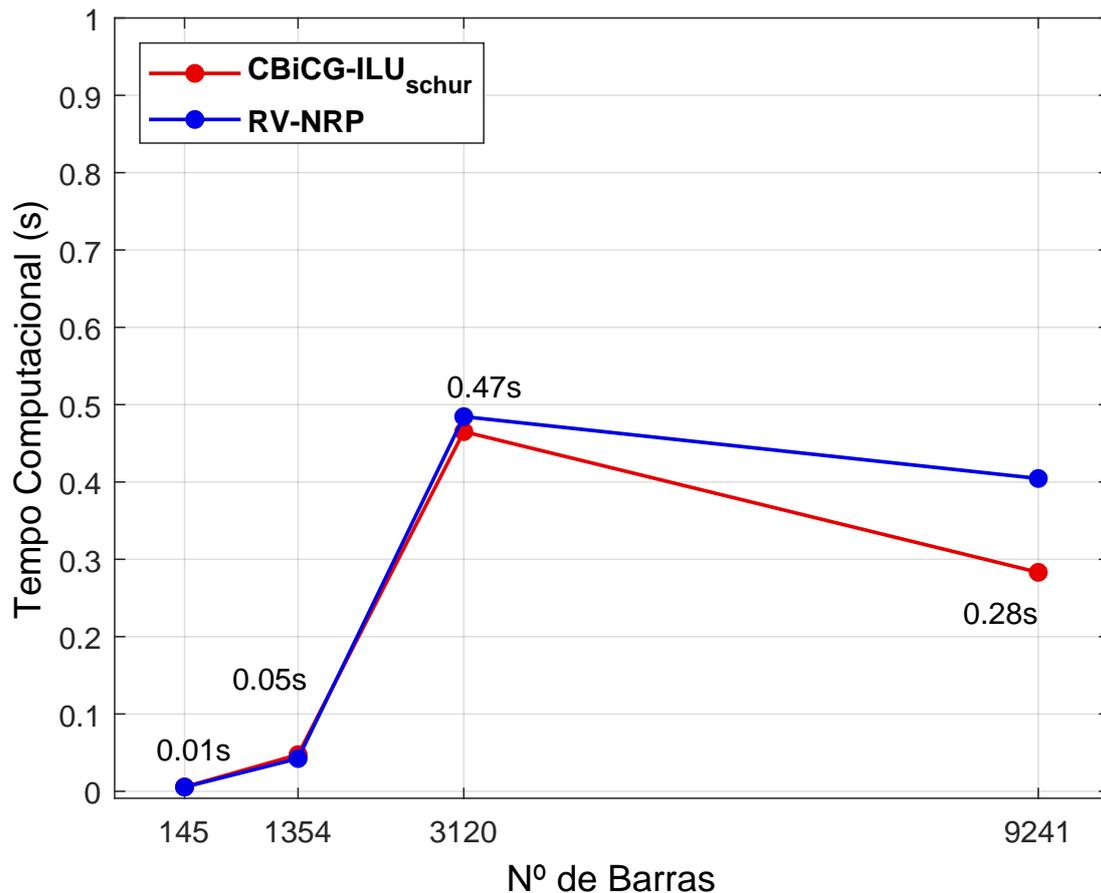
Fonte: Próprio Autor

ou seja, para que o preconditionador possa melhorar a performance computacional, é necessário que se conheça a topologia da rede. Isto, porque dependendo da topologia do sistema, alguns blocos que compõem a matriz jacobiana complexa, podem ser tornar mal condicionados, levando à uma piora no desenvolvimento do algoritmo. Devido a este fator, deve-se através de um processo heurístico, encontrar um preconditionador capaz de promover melhorias no sistema, além disso pode-se aplicar técnicas de ordenação [38] ou usar o operador SWAP para a permutação dos blocos matriciais mal condicionados. Outra forma de melhoria, é a redução da dimensão do sistema via Complemento de Schur 2.6, aplicado a jacobiana 153, que se transforma na jacobiana reduzida  $J_o = J_{11} - J_{12}J_{22}^{-1}J_{21}$ , redução esta que figura nas referências [40] e numa obra mais recente [41].

### 6.1.4 CBiCG-ILU Reduzido Via Complemento de Schur

Conforme dito anteriormente, é possível fazer a redução sistemas considerado, via complemento de Schur, exposto na subseção 4.2.2. Nesta simulação, mantém-se as premissas das simulações anteriores quanto à inicialização.

Figura 6.5: Tempo computacional em segundos do sistema reduzido



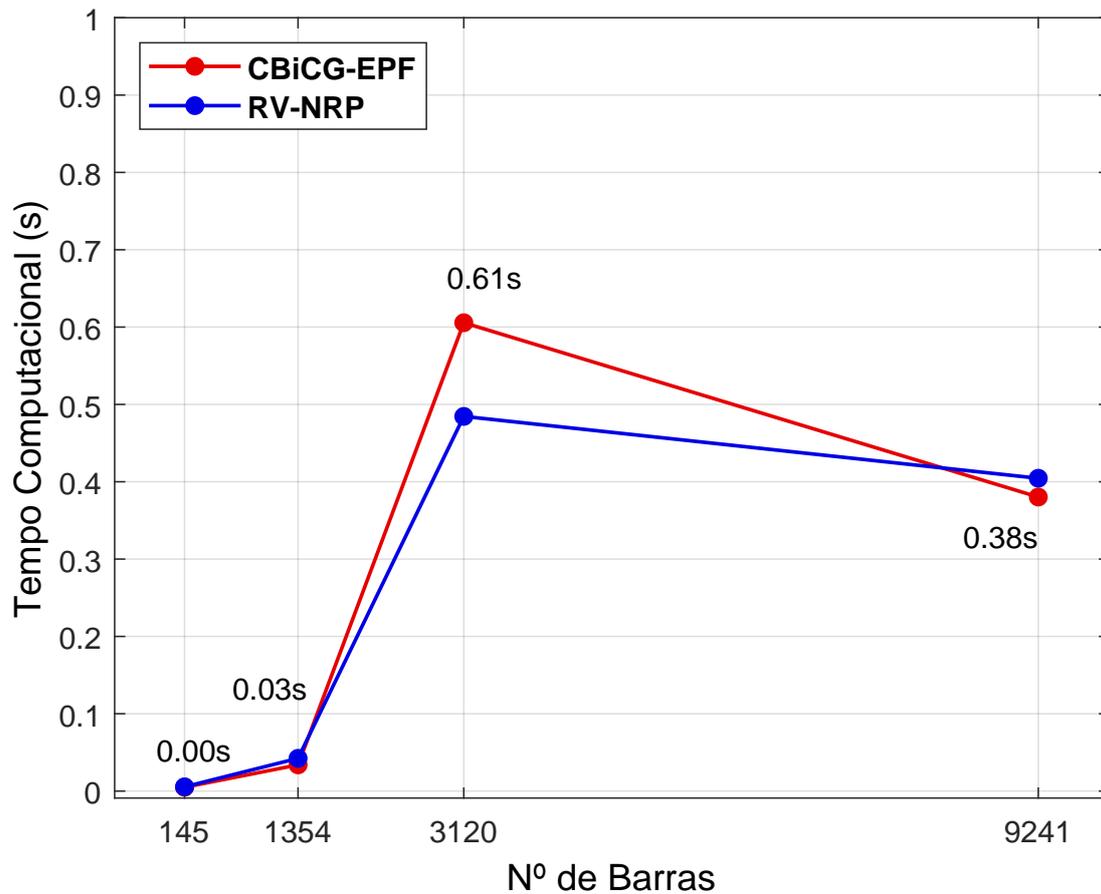
Fonte: Próprio Autor

Os tempos computacionais se mantiveram inalterados para os casos menores, no entanto, houve um ganho computacional para os dois maiores casos de 3120 e 9241 barras. Vale ressaltar aqui, que durante a redução dos sistemas, houve a alteração de blocos matriciais mal condicionados via operador swap. Este fato pode justificar a melhoria de performance do sistema de 3120 barras, que devido ao seu mal condicionamento, acarretou um aumento de tempo computacional visto na figura 6.4. O número de iterações se manteve inalterado e pode ser visto na tabela 1.

### 6.1.5 Fluxo de potência exato CBiCG-EPF

Nesta simulação incorpora-se o efeito da matriz Hessiana, através da formulação matricial de Wirtinger já exposta neste trabalho na subseção 4.2.3. É possível obter mais informações através dos trabalhos de [30] e [29].

Figura 6.6: Tempo computacional em segundos do sistema reduzido



Fonte: Próprio Autor

Nesta simulação, os tempos computacionais se mantiveram semelhantes à simulação representada na figura 6.4, com ligeira redução do tempo computacional para o caso de 3120 barras, provavelmente porque o condicionamento das matrizes envolvidas era o mesmo. A grande diferença se deu no número de iterações até a convergência, fato esperado, uma vez que a incorporação do efeito da matriz Hessiana, traz maior precisão nas direções de convergência e vetor de correção. A seguir apresenta-se a tabela contendo o número de iterações para todos os casos testados:

Tabela 2: Tabela de Iterações CBiCG-EPF x MATPOWER

Barras	Iterações CBiCG	Iterações RV-NRP
5	2	2
9	2	3
14	2	3
18	2	3
22	2	2
30	2	2
39	3	3
57	3	3
69	2	2
85	2	3
118	3	3
141	2	2
145	-	4
300	4	4
1354	3	4
2869	4	6
3120	5	5
9241	5	6

O número de iterações em todos os casos se manteve abaixo do número de iterações fornecido pelo software **MATPOWER**, com exceção do caso de 145 barras onde o algoritmo **CBiCG-EPF** divergiu. No entanto, a performance geral em praticamente todos os casos, foi melhor quanto ao número de iterações.

## 7 Conclusão

A estruturação e modelagem do problema de fluxo de potência nos domínios real e complexo permite a visualização da complexidade de cada caso. A implementação se mostrou muito mais simples e poderosa no domínio complexo uma vez que utilizando o cálculo de Wirtinger generalizado foi possível desenvolver todos os algoritmos de forma direta e natural além de que as estruturas de aritmética de números complexos do software utilizado estão otimizadas para tal fim, bem como é uma tendência, a arquitetura de computadores **SIMD** - *Single Instruction Multiple Data* [42], adequado ao paralelismo e tratamento de conjuntos regulares de dados, como matrizes e vetores, ou seja, aritmética de números complexos. A separação de variáveis na implementação dos algoritmos em domínio real se mostra complicada, visto o grande número de equações e de difícil organização e manutenção. Em se tratando do plano complexo a fundamentação teórica dada pelo cálculo de Wirtinger, permite que se aborde problemas de otimização mais gerais e novas possibilidades, vide [43]. Sobre os métodos dos gradientes conjugados, se constituem em métodos muito simples de serem implementados e rápidos, utilizados por sua facilidade de vetorização, rapidez e aplicação em computação paralela ([44], [45] e [46]). Esses métodos se mostram robustos e eficientes na presença de técnicas e condicionadores específicos.

## 8 Trabalhos Futuros

São grandes as possibilidades de pesquisa dentro deste campo de atuação. Devido ao cálculo matricial generalizado de Wirtinger, existem inúmeros aspectos dos algoritmos iterativos e da formulação complexa, que podem ser explorados, uma vez que, a matriz jacobiana em domínio complexo, apresenta grande simetria e estrutura de blocos que podem ser exploradas via processamento paralelo, usando linguagens de programação mais eficientes do que a usada neste trabalho, como por exemplo C++, em conjunto com outras técnicas que podem fazer o condicionamento por blocos matriciais. Além disso, um tratamento generalizado e sistematizado pelo cálculo de Wirtinger nos permite incorporar modelos de fluxo de potência mais complexos, como os modelos de **FACTS** em geral.

## Referências

- [1] C. P. Steinmetz, “Complex quantities and their use in electrical engineering,” *Proceedings of the International Electrical Congress*, pp. 33–74, 1893.
- [2] J. Glover, M. Sarma, and T. Overbye, *Power System Analysis and Design*. Cengage Learning, 2011. [Online]. Available: <https://books.google.com.br/books?id=uQcJAAAAQBAJ>
- [3] J. R. Magnus and H. Neudecker, *Matrix differential calculus with applications in statistics and econometrics*. John Wiley & Sons, 2019.
- [4] A. Hjørungnes, *Complex-valued matrix derivatives: with applications in signal processing and communications*. Cambridge University Press, 2011.
- [5] D. H. Brandwood, “A complex gradient operator and its application in adaptive array theory,” *Communications, Radar and Signal Processing, IEE Proceedings F*, vol. 130, no. 1, pp. 11–16, February 1983.
- [6] M. H. Hayes, *Statistical digital signal processing and modeling*. John Wiley & Sons, 1996.
- [7] W. Wirtinger, “Über den weierstrass’ schen vorbereitungssatz.” 1927.
- [8] K. Kreutz-Delgado, “The complex gradient operator and the CR-calculus,” *ArXIV e-print, arXIV:0906.4835v1 [math.OA]*, pp. 1–74, June 25, 2009.
- [9] F. Zhang, *The Schur complement and its applications*. Springer Science & Business Media, 2006, vol. 4.
- [10] I. Gohberg *et al.*, *I. Schur methods in operator theory and signal processing*. Springer, 1986.
- [11] W. Hackbusch, B. N. Khoromskij, and R. Kriemann, “Direct schur complement method by domain decomposition based on h-matrix approximation,” *Computing and visualization in science*, vol. 8, no. 3, pp. 179–188, 2005.
- [12] Y. Saad and M. Sosonkina, “Distributed schur complement techniques for general sparse linear systems,” *SIAM Journal on Scientific Computing*, vol. 21, no. 4, pp. 1337–1356, 1999.

- [13] I. N. Imam and L. M. Lamont, “An algorithm using the schur complement in inverting large matrices,” in *Proceedings. IEEE Energy and Information Technologies in the Southeast*. IEEE, 1989, pp. 421–426.
- [14] A. Mahmood, D. Lynch, and L. Philipp, “A fast banded matrix inversion using connectivity of schur’s complements,” in *IEEE International Conference on Systems Engineering*, 1991, pp. 303–306.
- [15] R. Zimmerman and C. Murillo-Sanchez, “Matpower (version 7.1)(2020).”
- [16] J. Verboomen, D. V. Hertem, P. H. Schavemaker, W. L. Kling, and R. Belmans, “Phase shifting transformers: principles and applications,” in *2005 International Conference on Future Power Systems*, Nov 2005, pp. 6 pp.–6.
- [17] E. Acha and L. M. Castro, “A generalized frame of reference for the incorporation of multi-terminal vsc-hvdc systems in power flow solutions,” *Electric Power Systems Research*, vol. 136, no. Supplement C, pp. 415 – 424, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378779616300566>
- [18] E. Acha and B. Kazemtabrizi, “A new statcom model for power flows using the newton-raphson method,” *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2455–2465, Aug 2013.
- [19] A. de la Villa Jaen, E. Acha, and A. G. Exposito, “Voltage source converter modeling for power system state estimation: Statcom and vsc-hvdc,” *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1552–1559, Nov 2008.
- [20] X.-P. Zhang, “Multiterminal voltage-sourced converter-based **HVDC** models for power flow analysis,” *IEEE Transactions on Power Systems*, vol. 19, no. 4, pp. 1877–1884, Nov 2004.
- [21] W. Wirtinger, “Zur Formalen Theorie der Functionen Von Mehr Complexen Veranderlichen,” *Math. Ann.*, vol. 97, pp. 357–375, 1927.
- [22] H. L. Nguyen, “Newton-raphson method in complex form [power flow analysis],” in *Transmission and Distribution Conference, 1996. Proceedings., 1996 IEEE*, Sep 1996, pp. 591–595.

- [23] —, “Newton-raphson method in complex form [power system load flow analysis],” *Power Systems, IEEE Transactions on*, vol. 12, no. 3, pp. 1355–1359, Aug 1997.
- [24] T. T. Nguyen and C. Vu, “Complex-variable newton-raphson load-flow analysis with FACTS devices,” in *Transmission and Distribution Conference and Exhibition, 2005/2006 IEEE PES*, IEEE, Ed. <http://dx.doi.org/10.1109/TDC.2006.1668480>: IEEE, 2006, pp. 183 – 190.
- [25] L. Barboza, H. Zürn, and R. Salgado, “Load tap change transformer: A modeling reminder,” *IEEE Power Engineering Review*, pp. 51–52, 2001.
- [26] R. Pires, “Unbalanced phase-to-phase voltage compensators applied to radial distribution feeders,” *Power Delivery, IEEE Transactions on*, vol. 19, no. 2, pp. 806–812, April 2004.
- [27] R. D. Zimmerman, C. E. Murillo-Sánchez, and D. Gan, “Matpower: A matlab power system simulation package,” *Manual, Power Systems Engineering Research Center, Ithaca NY*, vol. 1, 1997.
- [28] epe, “Empresa de pesquisa energética,” [www.epe.gov.br](http://www.epe.gov.br), 2022.
- [29] S. Iwamoto and Y. Tamura, “A fast load flow method retaining nonlinearity,” *IEEE Transactions on Power Apparatus and Systems*, no. 5, pp. 1586–1599, 1978.
- [30] R. Pires, G. Chagas, and L. Mili, “Enhanced power flow solution in complex plane,” *International Journal of Electrical Power & Energy Systems*, vol. 135, p. 107501, 2022.
- [31] M. R. Hestenes and E. Stiefel, “Methods of conjugate gradients for solving,” *Journal of research of the National Bureau of Standards*, vol. 49, no. 6, p. 409, 1952.
- [32] J. R. Shewchuk *et al.*, “An introduction to the conjugate gradient method without the agonizing pain,” 1994.
- [33] R. Fletcher and C. M. Reeves, “Function minimization by conjugate gradients,” *The computer journal*, vol. 7, no. 2, pp. 149–154, 1964.
- [34] C. Lanczos, “Solution of systems of linear equations by,” *Journal of research of the National Bureau of Standards*, vol. 49, no. 1, p. 33, 1952.

- [35] D. A. Jacobs, “A generalization of the conjugate-gradient method to solve complex systems,” *IMA journal of numerical analysis*, vol. 6, no. 4, pp. 447–452, 1986.
- [36] Y. Saad, *Iterative methods for sparse linear systems*. siam, 2003, vol. 82.
- [37] L. Oliker, X. Li, P. Husbands, and R. Biswas, “Effects of ordering strategies and programming paradigms on sparse matrix computations,” *Siam Review*, vol. 44, no. 3, pp. 373–393, 2002.
- [38] I. S. Duff and G. A. Meurant, “The effect of ordering on preconditioned conjugate gradients,” *BIT Numerical Mathematics*, vol. 29, no. 4, pp. 635–657, 1989.
- [39] F. Milano, *Power system modelling and scripting*. Springer Science & Business Media, 2010.
- [40] G. Viviani, “Complex vector load flow,” *Electric machines and power systems*, vol. 10, no. 5-6, pp. 443–452, 1985.
- [41] G. Granelli and M. Montagna, “Matlab implementation of the complex power flow,” *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, 2013.
- [42] C. Vilacha, J. Moreira, E. Miguez, and A. F. Otero, “Massive jacobi power flow based on simd-processor,” in *2011 10th International Conference on Environment and Electrical Engineering*. IEEE, 2011, pp. 1–4.
- [43] I. Džafić, R. A. Jabr, and T. Hrnjić, “High performance distribution network power flow using wirtinger calculus,” *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3311–3319, 2018.
- [44] R. Helfenstein and J. Koko, “Parallel preconditioned conjugate gradient algorithm on gpu,” *Journal of Computational and Applied Mathematics*, vol. 236, no. 15, pp. 3584–3590, 2012.
- [45] M. Ament, G. Knittel, D. Weiskopf, and W. Strasser, “A parallel preconditioned conjugate gradient solver for the poisson problem on a multi-gpu platform,” in *2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing*. IEEE, 2010, pp. 583–592.

- [46] H. C. Elman and E. Agrón, “Ordering techniques for the preconditioned conjugate gradient method on parallel computers,” *Computer Physics Communications*, vol. 53, no. 1-3, pp. 253–269, 1989.