Universidade de São Paulo – USP
Escola de Engenharia de São Carlos – EESC
Departamento de Engenharia Elétrica

Trabalho de Conclusão de Curso

Utilização de Computação Paralela para a Resolução de Problemas de Fluxo de Potência Ótimo em Sistemas de Transmissão de Energia Elétrica

Aluno: Gabriel Siqueira Garcia

Orientador: Prof. Dr. Rodrigo Andrade Ramos

São Carlos
2012

GABRIEL SIQUEIRA GARCIA

UTILIZAÇÃO DE COMPUTAÇÃO PARALELA PARA A RESOLUÇÃO DE PROBLEMAS DE FLUXO DE POTÊNCIA ÓTIMO EM SISTEMAS DE TRANSMISSÃO DE ENERGIA ELÉTRICA

Trabalho de Conclusão de Curso apresentado à
Escola de Engenharia de São Carlos, da
Universidade de São Paulo

Curso de Engenharia Elétrica com ênfase em
Sistemas de Energia e Automação

ORIENTADOR: Prof. Dr. Rodrigo Andrade Ramos

São Carlos
2012

Garcia, Gabriel Siqueira
G216u Utilização de computação paralela para a resolução de problemas de fluxo de potência ótimo em sistemas de transmissão de energia elétrica / Gabriel Siqueira Garcia; orientador Rodrigo Andrade Ramos. São Carlos, 2012.

Monografia (Graduação em Engenharia Elétrica com ênfase em Sistemas de Energia e Automação) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2012.

1. Computação Paralela. 2. Fluxo de Potência Ótimo. 3. Análise de Contingências. 4. Programação Orientada a Objetos. 5. Linguagem C++. 6. Sistemas Elétricos de Potência. I. Título.

FOLHA DE APROVAÇÃO

Nome: Gabriel Siqueira Garcia

Título: "Utilização de Computação Paralela para a Resolução de Problemas de Fluxo de Potência Ótimo em Sistemas de Transmissão de Energia Elétrica"

Trabalho de Conclusão de Curso defendido e aprovado em 29 / 11 / 2012,

com NOTA 9,0 (NOVE, ZERO), *pela Comissão Julgadora:*

Prof. Dr. Rodrigo Andrade Ramos (Orientador)
SEL/EESC/USP

Prof. Associado João Bosco Augusto London Júnior
SEL/EESC/USP

M.Sc. Eduardo Werley Silva dos Ângelos
SEL/EESC/USP

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Homero Schiabel

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

API	*Application Programming Interface*
FP	*Fluxo de Potência*
FPO	*Fluxo de Potência Ótimo*
SEP	*Sistemas Elétricos de Potência*
ONS	*Operador Nacional do Sistema Elétrico*
POO	*Programação Orientada a Objetos*

Resumo

Siqueira Garcia, G. **Utilização de Computação Paralela para a Resolução de Problemas de Fluxo de Potência Ótimo em Sistemas de Transmissão de Energia Elétrica**. Trabalho de Conclusão de curso – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2012.

A operação em tempo real de um sistema elétrico de potência (SEP) é composta por uma série de funções de análise e controle da rede. Todas as funções devem ser executas da forma mais rápida possível durante a operação em tempo real. A análise de segurança é uma das funções de rede executadas e é a função que exige maior tempo de processamento para sua execução. O presente trabalho aborda a utilização de computação paralela aplicada à análise de contingências de um sistema elétrico de potência, considerando, para cada caso de contingência, a solução do fluxo de potência ótimo para a condição operacional apresentada. A implementação dos métodos de análise de contingências e fluxo de potência ótimo é realizada utilizando a linguagem C++, em ambiente GNU/Linux, abordando os conceitos de Programação Orientada a Objetos (POO). A aplicação de computação paralela ao programa desenvolvido é realizada através da iterface de programação de aplicativo (do inglês, *Application Program Iterface* – API), openMP. São avaliados os resultados do programa para dois sistemas de potência: sistema sul reduzido brasileiro composto de 45 barras e sul-sudeste-Mato Grosso reduzido composto de 107 barras. A partir dos resultados obtidos foi feita a avaliação do desempenho computacional do programa utilizando a computação paralela em um sistema composto por mais de um núcleo.

Palavras-Chave: Sistemas Elétricos de Potência, Fluxo de Potência Ótimo, Análise de Contingências, Programação Orientada a Objetos, Computação Paralela, Linguagem C++.

Abstract

Siqueira Garcia, G. **Parallel Computing Utilization for Solving Optimal Power Flow Problems on Electrical Transmission Systems**. Trabalho de Conclusão de curso – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2012.

The real time operation of an Electrical Power System (EPS) is composed of a series of analyse e control net functions. All these functions have to be executed as fast as possible at the real time system operation. The security analysis is one of the net functions executed and it is the function that requires more time to be executed. This work adresses the utilization of parallel computing applied to contingency analysis of an electrical power system, considering for each case, the solution of the optimal power flow for the presented operation scenario. The contingency analysis, power flow and optimal power flow methods were implemented using C++ language, on GNU/Linux environment, addressing the concepts of Object Oriented Programming (OOP). The multilanguage API (Application Program Interface), openMP, was used in order to apply parallel computing to the developed program. The program results are evaluated for two electrical power systems: reduced brazilian south electrical system composed of 45 bars and reduced south-southeast-Mato Grosso electrical system composed of 107 bars. The developed program was evaluated in therms of computational performance using parallel computing on a system comprising more than one core.

KeyWords: Electrical Power Systems, Optimal Power Flow, Contingency Analysis, Object Oriented Program, Parallel Computing, C++ Language.

Capítulo 1 Introdução

A análise computacional de um Sistema Elétrico de Potência (SEP) é imprescindível para a operação e controle do mesmo. Na operação em tempo real de um SEP são executadas funções de análise e controle da rede. Todas as funções devem ser executadas de forma rápida e eficiente. Uma das funções de controle executadas é a análise de segurança do SEP. Esta é a função de rede que exige maior tempo computacional para sua execução (BALU *et al.*, 1992).

Durante a análise de segurança, é feito um monitoramento do sistema. Medidas de grandezas físicas do SEP são obtidas em tempo real e passam por um processo de filtragem. Após isso, um processo de estimativa das variáveis de estado do sistema é executado. Com isso, um fluxo de potência *on-line* é executado a fim de se obter o estado de operação do SEP naquele instante (BALU *et al.*, 1992).

A análise do fluxo de potência é capaz de obter todas as tensões complexas nodais do SEP. Esta análise é realizada, usualmente, através de métodos numéricos, como visto na referência (MONTICELLI, 1983), na qual um dos métodos utilizados é o de Newton-Raphson. Durante o processo, a cada iteração, também é necessária a resolução de um sistema linear, geralmente esparso.

Após a verificação do estado de operação do SEP, este pode estar operando em condição segura ou não. Se o sistema está em condição segura, um análise de robustez do sistema é realizada, verificando qual seria novo estado de operação do SEP para uma lista predefinida de possíveis contingências que possam ocorrer no SEP. Desta forma, é feita uma análise de contingências para o SEP em questão, a fim de verificar a operabilidade do SEP na ausência de um ou mais equipamentos.

Após a análise de todas as contingências, pode-se determinar o estado ótimo de operação do SEP, para cada contingência, em termos de tensões complexas nodais, segundo uma função objetivo de minimização. Com relação aos métodos de otimização que podem ser utilizados, pode-se citar um dos métodos mais básicos: o método do Gradiente (HAPP; WIRGAU, 1981). Tal método tem uma modelagem com base nos multiplicadores de Lagrange, tendo como restrição de igualdade as equações do fluxo de potência. Tal modelagem pode ser vista na referência (WOOD; WOLLENBERG, 1996).

Atualmente, existem diversos *softwares* comerciais em uso operacional que realizam análises como as citadas acima. O programa computacional mais utilizado no Brasil é o ANAREDE (Análise de Redes Elétricas), desenvolvido pelo CEPEL (Centro de Pesquisas de Energia Elétrica). Dentre os recursos oferecidos pelo programa podem-se citar: Fluxo de

Potência, Equivalente de Redes, Análise de Contingências, Análise de Sensibilidade de Tensão e Fluxo e Análise de Segurança de Tensão (ANAREDE, 2012).

Face ao exposto, este trabalho tem como objetivo realizar a análise de fluxo de potência ótimo para uma lista de casos de contingência do SEP com base em ferramentas recentes, como a programação orientada a objetos e a programação paralela, utilizando ambiente GNU/Linux.

O ambiente GNU/Linux é essencial para o desenvolvimento de *software*, tanto pela sua versatilidade quanto pelo seu ótimo desempenho funcional. A programação orientada a objetos é um paradigma de programação capaz de criar códigos modulares, com maior manutenabilidade e reusabilidade (RICARTE, 1995). Dentro deste contexto, escolheu-se utilizar a linguagem C++ para o desenvolvimento da presente proposta de projeto de conclusão de curso (DEITEL, 2001).

A análise de contingências, então, será executada processando todos os casos de contingência via o processamento em paralelo (GRAMA, 2003). Assim, o processamento requerido para cada caso de contingência a ser analisado é divido entre os núcleos de um mesmo processador ou mesmo de um grupo de computadores, ligados em forma de *cluster*. Desta forma, será mostrado um considerável aumento no desempenho do processamento, quando comparado ao processamento sequencial.

1.1 Estrutura da Monografia

O texto desta monografia está divido em capítulos de forma a apresentar todo o conteúdo necessário para o entendimento deste trabalho. Na sequência são descritos cada capítulo que compõe o texto:

- **Capítulo 2:** neste capítulo são apresentadas as ferramentas computacionais aplicadas ao desenvolvimento deste trabalho, bem como seus conceitos e maneira utilização.
- **Capítulo 3:** aqui é apresentado o problema do fluxo de potência. A abordagem utilizada, bem como a modelagem matemática e o algoritmo numérico para solução do problema são apresentados neste capítulo. Em seguida é apresentada a modelagem computacional do fluxo de potência implementada no programa desenvolvido neste trabalho. Resultados obtidos para sistemas teste são apresentados para a modelagem desenvolvida.
- **Capítulo 4:** é apresentado o problema do fluxo de potência ótimo. A abordagem utilizada, bem como a modelagem matemática e o algoritmo numérico para solução do problema são apresentados neste capítulo. Em seguida é apresentada a

modelagem computacional do fluxo de potência ótimo implementada no programa desenvolvido neste trabalho. Resultados obtidos para sistemas teste são apresentados para a modelagem desenvolvida.

- **Capítulo 5:** é apresentado o método de análise de contingências utilizado neste trabalho. O algoritmo computacional implementado no programa é apresentado, bem como a utilização de programação paralela. Resultados obtidos para sistemas teste são apresentados para a modelagem desenvolvida.
- **Capítulo 6:** são discutidas as principais conclusões a partir dos resultados obtidos nos testes realizados neste trabalho.

Capítulo 2 Ferramentas Computacionais Aplicadas

Neste capítulo serão apresentados alguns conceitos introdutórios necessários para o melhor entendimento do desenvolvimento computacional realizado neste trabalho. Os conceitos abordados serão: programação orientada a objetos, linguagem de programação C++, diagramas UML (*Unified Modeling Language*) e programação paralela via openMP (*Open Multi-Processing*).

2.1 Programação Orientada a Objetos (POO)

A Programação Orientada a Objetos (POO) é um paradigma de programação que aborda um sistema computacional como um conjunto de objetos. Dentro deste sistema, os objetos se relacionam entre si. Cada objeto possui características e funcionalidades específicas e é através deles que todo o processamento do *software* ocorre (FARINELLI, 2007).

Dentre as principais vantagens relacionadas à orientação a objetos com relação à programação procedural pode-se citar o aumento da reusabilidade e da extensibilidade dos programas, através da utilização dos princípios de herança e polimorfismo. Além disso, a orientação a objetos também engloba outros princípios, não exclusivos desta filosofia, como: abstração e encapsulamento. Tais princípios, unidos aos conceitos de herança e polimorfismo, fazem com que a programação a objetos seja um paradigma muito eficiente para desenvolvimento de *software* (RICARTE, 1995).

Segundo Farinelli (2007, p.4), “a orientação a objetos consiste em considerar os sistemas computacionais como uma coleção de objetos que interagem de maneira organizada”.

Nas subseções seguintes serão descritos, de forma sucinta, os conceitos básicos de orientação à objetos e, em seguida, serão apresentados os princípios deste paradigma.

2.1.1 Conceitos básicos

Um objeto é uma entidade do mundo real que merece representação para o ambiente estudado. Os objetos podem representar entidades concretas ou entidades conceituais. Além disso, cada objeto possui sua identidade, ou seja, objetos com características idênticas são distintos entre si. Desta forma, cada objeto possui uma identificação única que o representa (RICARTE, 1995).

O objeto é uma entidade capaz de reter um estado e possui uma série de operações específicas capazes de manipular este estado. A única maneira de se interagir com um objeto é através das operações que este disponibiliza (FARINELLI, 2007).

Por exemplo, podemos ter um objeto “pessoa”. Podemos enumerar uma série de características presentes em uma pessoa, como: altura, idade, cor do cabelo, etc. Uma pessoa pode também executar várias ações, como: andar, falar, escrever, etc. Além disso, uma pessoa tem uma identificação única, que pode ser nome, ou ainda, seu RG ou CPF.

O conjunto de características de um objeto é representado por meio de atributos. Da mesma forma, o conjunto de operações deste objeto que representa seu comportamento pode ser expresso na forma de métodos. Os atributos são variáveis de um tipo específico que armazenam as características correspondentes de um objeto. O métodos são funções que realizam as ações próprias do objeto (FARINELLI, 2007).

Pode-se então, por exemplo, criar um objeto “João”, o qual possui o conjunto de atributos e métodos vistos na Tabela 2.1.

Tabela 2.1: Objeto João – Atributos e Métodos.

João
Atributos
Altura: 1,87
Idade: 22
Profissão: Engenheiro
Métodos
Falar
Andar
Dormir

Os atributos do objeto "João" são a sua idade, altura e profissão. Seus métodos são falar, andar e dormir. Vale ressaltar que, o objeto “João” só interage com o ambiente ou com outro objeto através de seus métodos e que seu nome o identifica unicamente no contexto em que foi criado.

Uma classe representa um conjunto de objetos que contém características e ações semelhantes. Ela representa a abstração das características do conjunto em termos de atributos e métodos. Um objeto é definido então como uma instância de uma classe. Cada instância tem seus próprios valores para atributos, porém dividem as mesmas características e métodos com outros objetos de mesma classe (RICARTE, 1995).

Por exemplo, poderia existir outro objeto, “José”, o qual possui características diferentes de “João”. Porém, podemos criar a classe “Seres Humanos”, englobando as características e métodos comuns a “João” e “José”.

Uma classe é apenas a generalização de um conjunto de objetos. Classes não existem na realidade, apenas objetos. Objetos de mesma classe possuem a mesma denominação tanto para atributos quanto para métodos. A classe representa então um

gabarito para os objetos da classe e mostra como estes estão estruturados internamente (FARINELLI, 2007).

Quando classes possuem características semelhantes entre si, pode-se agrupar estas características comuns em uma classe superior, chamada superclasse ou classe-pai. Desta forma, as classes que tiveram suas características agrupadas na superclasse são chamadas de subclasses ou classes-filha. A superclasse pode então guardar as características comuns a todas as classes subclasses, bem como os métodos que são comuns entre elas, sem a necessidade que se reescreva cada atributo ou método em cada classe-filha. Este conceito é chamado herança e será melhor detalhado na Seção 0.

2.1.2 Princípios da Orientação a Objetos

Nas subseções seguintes serão apresentados alguns dos principais conceitos da programação orientada a objetos. Os dois primeiros, abstração e encapsulamento, não são exclusivos da orientação a objetos porém são suportados de forma melhor por esta do que por outras metodologias (RICARTE, 1995). Já os conceitos de herança e polimorfismo são exclusivos da orientação a objetos e que unidos com os princípios de abstração e encapsulamento formam uma ferramenta poderosa de programação.

2.1.2.1 Abstração

O conceito de abstração consiste em separar os aspectos relevantes de um problema complexo dos aspectos que não são relevantes para a solução do problema. Foca-se em extrair apenas as características essenciais do problema, ignorando as características “acidentais”. Esta separação ocorre de forma livre, podendo um problema ser subdivido de diversas maneiras diferentes e todas estas convergirem para uma solução de sucesso (RICARTE, 1995).

A aplicação do conceito de abstração à programação orientada a objetos consiste em isolar um objeto da realidade e neste representar apenas as características importantes para a solução do problema. Por exemplo, a característica “Altura” não seria relevante para uma classe pertencente a um sistema de cadastro de pessoas em uma biblioteca. Porém, essa característica teria relevância para um cadastro de pessoas candidatas a entrar em um time de basquete.

2.1.2.2 Encapsulamento

O conceito de encapsulamento também é conhecido como ocultamento de informação e consiste em separar os aspectos externos, acessíveis a outras partes do programa, de aspectos internos do objeto, não acessíveis. Unindo-se o conceito de

abstração a este, pode-se ocultar detalhes de uma estrutura complexa que, se fossem acessíveis, poderiam interferir na correta execução do processo (RICARTE, 1995).

Além disso, o encapsulamento agrega toda informação necessária em uma única unidade. Em POO os atributos e métodos são encapsulados em uma unidade (objeto), diferentemente de outras estruturas de programação. O único modo de se conhecer a informação ou de manipula-la é através de seus métodos.

O objeto funciona então como uma “caixa preta” que disponibiliza toda a informação necessária e toda sua funcionalidade sem que outras partes externas ao objeto conheçam a maneira qual as informações são manipuladas. Isto possibilita que se façam alterações nos atributos ou métodos do objeto sem que outros componentes do sistema sejam afetados por estas alterações. Promove-se então a capacidade de modificação de um objeto, internamente, sem que se precise modificar a forma com a qual os métodos ou atributos são acessados (RICARTE, 1995).

Por exemplo, um site de buscas pode modificar o seu algoritmo de busca sem que os usuários percebam. Traduzindo isto em forma de orientação a objetos, um objeto “Site de busca” pode ter um método “Buscar”. Este método pode ser alterado, sem prejuízo para outro objeto ou sistema que use este objeto, garantindo que os mesmo argumentos de entrada sejam mantidos e o mesmo tipo de resposta seja enviado.

2.1.2.3 Herança

Como visto durante a exemplificação dos conceitos básicos de orientação a objetos, herança é o mecanismo através do qual uma classe pode receber características de uma classe ancestral a esta, chamada superclasse, bem como transmitir suas características a uma nova classe, chamada subclasse. A super classe também pode ser denominada classe base ou classe-pai, bem como a subclasse pode ser denominada classe derivada ou classe-filha.

O conceito de herança está intimamente ligado à especialização de uma classe com relação a outra, mais primitiva. Desta forma, a classe especialista pode então expandir as características obtidas, incluindo características específicas que a diferenciam da classe mais primitiva (FARINELLI, 2007).

É através da herança que classes podem compartilhar atributos e métodos já existentes de forma a reutilizar o código já escrito. Muito mais que apenas uma economia de código, a herança promove maior integridade do programa, pois qualquer alteração na classe base é transmitida automaticamente às classes derivadas sem que seja necessária a re-implementação das mesmas (FARINELLI, 2007).

Podemos ter uma classe “Pássaro”. Duas possíveis especializações desta classe seriam a classe “Ganso” e a classe “Águia”. Na classe “Pássaro” é possível armazenar as características comuns entre todos os pássaros. Já nas duas subclasses criadas, é possível definir novos atributos e métodos ou mesmo redefinir um método já existente.

2.1.2.4 Polimorfismo

O conceito de polimorfismo consiste em como dois objetos de classes diferentes respondem a uma mesma mensagem ou comando. Para cada objeto, uma mesma mensagem de entrada pode resultar em formas diferentes de execução da mesma. Este conceito está intimamente ligado à como uma superclasse e uma subclasse podem executar de formas diferentes um mesmo método (FARINELLI, 2007).

Como exemplo de polimorfismo, tenhamos as classes “Ganso” e “Águia” da Seção 0. Pode ser definido um método “mover” genérico para todos os pássaros na classe “Pássaro”. Este método pode ser redefinido nas subclasses, especializando-o com relação às características de cada subclasse. Os métodos como gansos e águias se movem são completamente distintos, mesmo tendo semânticamente a mesma designação.

2.2 A linguagem de programação C++

A linguagem C++ originou-se da da linguagem procedural C, porém adaptada à POO. Ela é totalmente compatível com C, ou seja, tudo que pode ser executado em C também pode ser executado em C++.

Nas Seções seguintes serão descritas as exclusividades da linguagem C++ com relação a linguagem C, abordando a definição de classes e a aplicação dos conceitos de POO.

2.2.1 Definição de Classes

As classes são definidas em C++ a partir da estrutura *class*. A utilização desta estrutura é exemplificada abaixo.

```
Class <Nome da Classe> {
Private:
      <código>
Public:
      <código>
}
```

Uma classe é uma estrutura de dados que pode conter tanto variáveis (atributos) quanto funções (métodos). Como pode ser visto, uma classe é formada por duas regiões de dados: uma região privada (*private*) e uma região pública (*public*). A região pública é acessível a todo o código ou ambiente que contém um objeto da classe. Já a região privada só é acessível pela própria classe e tudo que for ali declarado não pode ser acessado por agentes externos nem transmitido para subclasses que possam ser criadas (RICARTE, 2005).

Quando se deseja transmitir as declarações feitas na área privada de uma classe para suas subclasses, utiliza-se um terceiro tipo de região de dados, chamado *protected*. Este tipo de região tem funcionamento similar à região *private*. Tudo que for declarado na região *protected* não pode ser acessado por agentes externos, porém tudo será transmitido para as subclasses desta classe.

As funções pertencentes à classe são chamadas funções membro e tem livre acesso a todas as variáveis da região *private*. Desta forma, o ocultamento da informação é promovido dentro da classe, de maneira que a informação é acessível apenas através das funções que a própria classe fornece (RICARTE, 2005).

2.2.2 Prototipação

Toda função definida em uma classe na linguagem C++ deve ser declarada como um protótipo. O modelo de um protótipo é mostrado abaixo.

```
<Tipo de Retorno> <Nome da Função> (<Tipo de Argumento> <Nome de Argumento>, ...)
```

A prototipação de funções em C++ facilita a detecção de erros com relação a tipos de variáveis na chamada de funções em tempo de compilação. Por exemplo, se uma função possui argumentos do tipo *int* e são passados argumentos do tipo *double* no código que a utiliza, a compilação resultará em erros.

2.2.3 Sobrecarga de Funções

Em C++, uma classe pode ter várias funções com o mesmo nome, desde que cada uma das funções tenha argumentos distintos para sua chamada. Este recurso é chamado de sobrecarga de funções. Através da prototipação o compilador pode reconhecer uma função através de seu nome e seus argumentos de entrada. Quando várias funções de mesmo nome são declaradas, o compilador é capaz de escolher a função correta que está sendo chamada através do número e tipo de argumentos que compõem a mesma.

Por exemplo, seja uma função “área” que calcula a área de uma circunferência a partir de seu raio. O único argumento de entrada dessa função é o valor do raio da circunferência no formato *double*. A função retorna o valor da área em formato *double*. Podemos sobrecarregar essa função, criando outra de mesmo nome para calcular a área de um retângulo a partir de seus lados. Desta forma, a segunda função “área” teria dois argumentos de entrada em formato *double*. Quando essas funções forem chamadas por um programa, se passado apenas um argumento, a função retornará a área de uma circunferência que possui raio no valor fornecido. Se passados dois argumentos, a função retornará a área de um retângulo que possui lados nos valores fornecidos. Este exemplo ilustra a utilização da sobrecarga de forma simplificada. Este recurso foi utilizado durante a programação neste trabalho e será mostrado mais adiante.

2.2.4 Construtores e Destrutores

Construtores e destrutores são funções obrigatórias de toda classe criada em C++. Estas funções especiais tem a função de construir e destruir um objeto quando o mesmo é chamado dentro de um programa computacional. Isto ocorre pois a classe é utilizada em um programa como um tipo de dado. O método como um objeto é criado e as inicializações de suas variáveis podem ser definidas em seu construtor. Analogamente, a forma como as variáveis inicializadas serão destruídas deve ser especificada na função destrutor da classe.

A função construtor sempre tem o mesmo nome da classe e pode ser sobrecarregada. A função destrutor tem o mesmo nome do construtor, porém com o caractere “~” na frente. Um exemplo de protótipos de construtor e destrutor para uma classe chamada “Matriz” podem ser vistos a seguir.

```
Class Matriz {
Private:
      <código>
Public:
      <código>
      Matriz ( )
      ~Matriz ( )
}
```

Se um construtor é declarado sem nenhum argumento ou código, as variáveis são criadas de modo *default*, ou seja, apenas são alocadas sem qualquer atribuição. Porém, o uso de construtores se faz necessário quando se deseja atribuir valores iniciais a variáveis ou ainda quando se deseja alocar dinamicamente uma variável durante a criação de um

objeto. Nesse caso, deve-se passar argumentos ao construtor e definir a maneira como as variáveis serão inicializadas ou alocadas. Quando alocação dinâmica é feita em um construtor, o destrutor deve liberar o espaço em memória alocado inicialmente. Deve-se atentar também para o modo como o objeto é criado quando se aloca dinamicamente variáveis através do construtor.

2.2.5 Herança

O conceito de herança e a criação de classes derivadas é suportado em C++. Para que uma classe seja declarada como derivada de outra classe é necessário que a classe ancestral seja declarada previamente. A partir disso, pode-se criar uma classe derivada utilizando a estrutura exemplificada abaixo.

```
Class <Nome da Classe-Filha> : public <Nome da Classe-Pai> {
Private:
      <código>
Public:
      <código>
}
```

Caso seja necessário que as variáveis (atributos) da classe-pai sejam passadas à classe-filha deve-se utilizar o tipo "*protected*" ao invés de "*private*" na definição da classe-pai. Todas a funções e variáveis declaradas na área pública da classe-pai são passadas à classe-filha. Vale ressaltar que nenhuma das variáveis ou funções passadas são escritas novamente. Quando uma função é chamada na classe-filha e esta veio da definição da classe-pai, o compilador busca a função na classe-pai e não na classe-filha.

A partir da definição de variáveis e funções que devem ser recebidos de seu ancestral, pode-se agora definir novas funções e novas variáveis para a classe-filha, especializando-a com relação à classe-pai.

2.3 Diagramas UML

A linguagem UML (*Unified Modeling Language*) é uma linguagem de representação gráfica de classes e associações para orientação a objetos. A UML não é uma metodologia, pois não define um processo para o modelamento de uma classe, mas sim, oferece um modelo de representação gráfica que pode ser utilizada livremente (FARINELLI, 2007).

A UML pode representar tanto as classes como as associações que podem existir entre elas. O modelo de classe é divido em três partes: nome da classe, atributos da classe e os métodos da mesma. O esquema de representação da classe é mostrado na Figura 2.1.

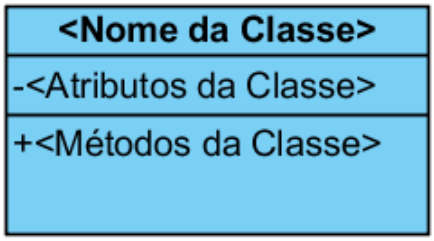

Figura 2.1: Representação gráfica de uma classe em linguagem UML.

A linguagem UML define treze tipos diferentes de diagramas utilizados para modelamento de programas que utilizam POO (OMG, 2012). Os diagramas são divididos em três categorias:

- Diagramas Estruturais: incluem o Diagrama de Classes, Diagrama de Objeto, Diagram de Componentes, Diagrama de Estrutura Composta, Diagrama de Pacote, e Diagrama de Implantação.
- Diagramas Comportamentais: incluem o Diagrama de Casos de Uso, Diagrama de Atividades e Diagrama de Máquina de Estado.
- Diagramas de Interação: incluem o Diagrama de Seqüência, Diagrama de Comunicação, Diagrama de Tempo e Diagrama de Visão Geral de Interação.

Os diagramas mais usualmente utilizados são o Diagrama de Casos de Uso, Diagrama de Classes e Diagrama de Atividades. O Diagrama de Casos de Uso, de uma maneira geral, mostra o sistema computacional na forma de macroatividades. O Diagrama de Classes mostra a composição de cada classe em termos de atributos e métodos e as ligações entre as classes componentes do projeto. O Diagrama de Atividades mostra como se dá a execução das atividades relacionadas a um processo do sistema computacional (MEDEIROS, 2004).

2.3.1 Associações entre Classes

As classes podem ser associadas, a medida que tenham relação umas com as outras. Além da associação simples, também existe a agregação e a composição. Tanto a agregação como a composição indicam uma estrutura todo-parte. Na agregação a estrutura parte pode existir sozinha, mantendo o sentido de sua existência no modelamento. Já na composição, a relação entre as classes é mais forte e a estrutura parte não tem sentido sem a estrutura todo.

Um exemplo de associação simples seria a associação entre as classes “Patrão” e “Empregado”. As classes tem relação uma com a outra, porém não há estrutura todo-parte neste caso, nem generalização de classes. Assim, a ligação entre estas duas classes é dita simples.

Considere uma classe “Casa” e uma classe “Condomínio”. Pode-se dizer que um objeto da classe “Condomínio” é formado por vários objetos da classe “Casa”, indicando uma estrutura todo-parte. Um objeto da classe “Casa” pode existir sem que sua existência perca sentido. O mesmo ocorre para um objeto do tipo “Condomínio”. Ambos podem existir, mantendo o sentido de sua existência no contexto a ser analisado, sendo assim, um exemplo de agregação de classes.

Considere agora a classe “Andar” e a classe “Prédio”. Pode-se dizer que um objeto da classe “Prédio” é formado por vários objetos da classe “Andar”, indicando uma estrutura todo-parte. Porém, neste caso, o objeto “Prédio” não pode existir sem seus objetos da classe “Andar”. Assim, este é um exemplo de composição entre classes.

As representações em linguagem UML para associação simples, agregação e composição são mostradas a seguir.

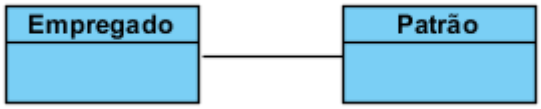

Figura 2.2: Exemplo de ligação simples: Patrão – Empregado.

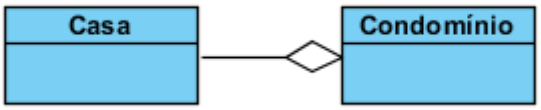

Figura 2.3: Exemplo de agregação: Casa – Condomínio.

Figura 2.4: Exemplo de composição: Andar – Prédio.

2.4 Programação Paralela

A programação paralela é uma forma de computação em que o processamento de um programa é divido em partes e executado concorrentemente, de forma a aumentar o desempenho computacional em termos de tempo de processamento. Com o surgimento de computadores multinúcleo esse tipo de computação passou a ser mais utilizado, dividindo o processamento entre os núcleos do processador.

2.5 OpenMP

O openMP é uma Interface de Programação de Aplicativo (API - *Application Program Interface*) para computação paralela em memória compartilhada formada por um conjunto de diretivas de compilação que pode ser utilizada para programas em C/C++ e Fortran. Utiliza a estrutura *fork/join* (ver Figura 2.5), dividindo o processamento de um programa através do uso de *threads*. Uma *thread* é a menor porção de processamento que pode ser executada por um sistema operacional. Ela funciona como um subsistema que existe dentro de um processo. Se não há processo, a *thread* deixa de existir (LLNL, 2012).

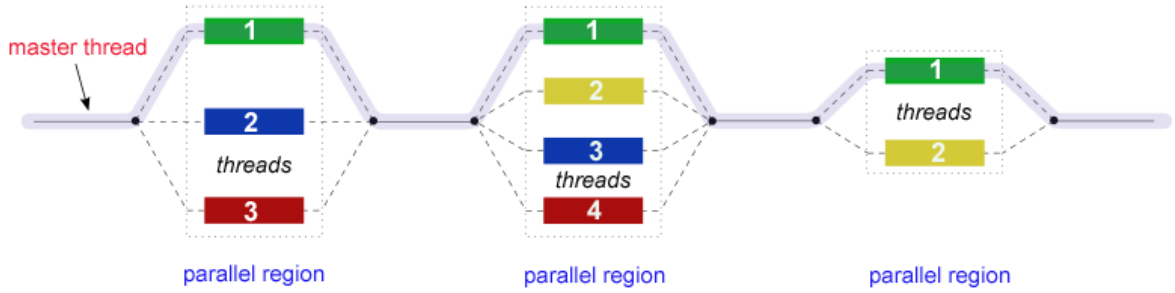

Figura 2.5: Estrutura *fork/join* retirada da referência (LLNL, 2012).

Conforme pode ser visto na Figura 2.5, o programa é executado inicialmente de forma *serial*, por apenas uma *thread*, chamada *master-thread*. A partir de certo momento na execução do código, o programa entra na região paralela. Nesta etapa então, a execução do código se divide (*fork*) e cada parte do processo é divido entre um número arbitrário de *threads*. Cada *thread* executa a parte que lhe cabe do processo segundo especificado pela diretiva de paralelização que iniciou a região paralela. Quando as *threads* terminam a parte paralela do processamento, elas sincronizam seus resultados e deixam apenas a *master-thread* na execução do programa.

A maneira como a divisão do processamento ocorrerá, bem como a quantidade *threads* que serão utilizadas e o momento em que a execução do programa será paralelizada depende das diretivas de programação utilizadas. Algumas diretivas serão apresentadas na seção seguinte, incluindo as que foram utilizadas neste trabalho.

2.5.1 Diretivas de Compilação

As diretivas utilizadas neste trabalho foram as diretivas *parallel* e *for*. Tais diretivas são inseridas diretamente no código em C++ e tem modos distintos de utilização. Todas as diretivas em C++ recebem o termo "*pragma omp*" no início de sua chamada e tem a linha iniciada com o caractere "#". Estas duas diretivas são explicadas a seguir.

2.5.1.1 Diretiva - *Parallel*

A diretiva *parallel* define qual será a região paralela do código e como o compartilhamento de memória entre as *threads* será executado. A estrutura de utilização desta diretiva em C++ é mostrada abaixo.

```
<código sequencial>
#pragma omp parallel <cláusulas> {
      <código executado em paralelo>
}
<código sequencial>
```

As cláusulas inseridas na diretiva definem como será o compartilhamento de memória e o número de *threads* que executarão paralelamente o código. Dentre as principais cláusulas estão: "*private*", "*shared*" e "*num_threads*" (LLNL, 2012).

- Private (<lista de variáveis>)

A cláusula *private* define quais variáveis já declaradas na região serial do código serão redefinidas para cada *thread* na região paralela. Quando uma variável é definida como *private* cada *thread* cria uma cópia da mesma e retém seu valor.

- Shared (<lista de variáveis>)

A cláusula *shared* define quais variáveis já declaradas serão compartilhadas entre todas as *threads* na região paralela. Se nenhuma cláusula for especificada, todas as variáveis são consideradas *shared*. É importante ressaltar que toda variável alocada dinamicamente é necessariamente *shared*.

- Num_threads (<número inteiro ou variável do tipo inteiro>)

Esta cláusula define em quantas *threads* a região paralela será dividida.

2.5.1.2 Diretiva - *For*

A diretiva *for* especifica que o laço *for* declarado diretamente abaixo desta será executado em paralelo pelas *threads* definidas pela diretiva *parallel*. Esta diretiva deve necessariamente ser declarada dentro da região paralela definida pela diretiva *parallel*. A estrutura de utilização desta diretiva em C++ é mostrada abaixo.

```
<código sequencial>
#pragma omp parallel <cláusulas> {
      <código executado em paralelo>
      #pragma omp for <cláusulas>
      for(<termos para execução do for>){
            <código executado em paralelo pelas threads no laço for>
      }
<código executado em paralelo>
}
<código sequencial>
```

Algumas das cláusulas de entrada para a diretiva *for* são: "*schedule*", "*ordered*" e "*nowait*". Dentre as demais cláusulas existentes estas três foram escolhidas, pois são as mais relevantes ao escopo deste trabalho. As cláusulas supracitadas tem a função de definir como as iterações serão distribuídas entre as *threads* (LLNL, 2012).

- Schedule (<tipo de divisão>,<número de divisões>)

A cláusula *schedule* define em quantas partes serão dividas as iterações e o como esta divisão será feita. Os tipos de divisão mais relevantes a este trabalho são: "*static*" e "*dynamic*". Em ambos os modos as iterações são divididas em *n* partes de acordo com o número especificado no segundo argumento da diretiva. No modo *static,* as iterações são divididas estaticamente entre as *threads*. Cada um tem uma porção fixa das iterações para resolver. No modo *dynamic,* as iterações são alocadas dinamicamente para a *thread* que estiver livre no momento (LLNL, 2012).

- Ordered

A cláusula *ordered* especifica que as iterações deverão ser executadas na mesma ordem que seriam no modo sequencial. Esta cláusula não pode ser utilizada junto com a cláusula *schedule*. Não há argumentos para esta cláusula.

- Nowait

A cláusula nowait pode ser utilizada junto com qualquer umas das outras cláusulas supracitadas. Ela especifica que não é necessário fazer o sincronismo das *threads* ao final da região paralela do código.

Capítulo 3 Fluxo de Potência

O fluxo de potência (FP) caracteriza-se pela determinação do estado da rede e dos fluxos de potência no sistema utilizando-se uma modelagem computacional especificamente criada para este fim. É expresso de forma estática, ou seja, considera que as variações nos parâmetros do sistema são lentas o suficiente para que se possa ignorar os efeitos transitórios. Desta forma, o FP é calculado a cada instante de tempo, *t*, no SEP, considerando seus parâmetros como fixos neste instante (MONTICELLI, 1983).

Entende-se como estado da rede, o conjunto dos módulos e ângulos de todas as tensões em todos os nós, ou barras, do SEP. Através destes parâmetros pode-se determinar os fluxos de potência nas linhas de transmissão do sistema. A modelagem do FP é composta por um conjunto de equações e inequações algébricas não lineares que representam o sistema e seus componentes.

O SEP é formado por basicamente dois tipos de componentes:

- Componentes que se situam entre um nó qualquer e a terra, como os geradores, as cargas, os reatores e os capacitores.
- Componentes que se situam entre dois nós quaisquer, como as linhas de transmissão e os transformadores.

Neste trabalho, os geradores e as cargas não são modelados, mas considerados como injeções fixas (estáticas) de potência ativa e reativa líquida na rede. Os capacitores *shunt* da barras fazem parte da modelagem deste problema, bem como as linhas de transmissão. Os transformadores, tanto em fase como defasadores, não são modelados.

As equações do FP são obtidas através da conservação das potências ativa e reativa nos nós do sistema. Desta forma, segundo a primeira lei de Kirchoff, a soma das injeções líquidas de potência em nó deve ser nula. Através da segunda lei de Kirchoff é possível determinar os fluxos de potência como funções dos módulos e ângulos da tensões nodais (MONTICELLI, 1983).

3.1 Formulação Básica

A formulação básica do problema do FP considera que para cada barra k do sistema podem ser relacionadas 4 variáveis:

- O módulo da tensão na barra, V_k;
- O ângulo da tensão na barra, θ_k;

- A injeção líquida de potência ativa na barra, P_k, composta pela geração subtraída da carga;
- A geração líquida de potência reativa, Q_k.

Para cada barra do SEP, dois destes parâmetros são considerados conhecidos e outros dois são considerados desconhecidos. Desta forma, são definidos três tipos de barras: PQ, PV e Vθ. A barra PQ é a barra de carga, em que são conhecidos apenas as potências ativa e reativa líquidas que são consumidas. A barra PV é a barra de geração, em que apenas a sua potência ativa líquida gerada e o módulo de sua tensão terminal são conhecidas. Por último, a barra Vθ, também chamada de barra *slack*, é a barra de referência do sistema. Esta barra tem função de fornecer a referência angular para as demais tensões no sistema, pois a formulação do fluxo de potência resulta em um sistema indeterminado na variável θ. Além disso, por não ter sua potência especificada, esta barra também fecha o balanço de potências do sistema (MONTICELLI, 1983).

Associando as considerações feitas com relação às leis de Kirchhoff, pode ser definido um conjunto de equações básicas para cada barra k do sistema:

$$P_k = \sum_{m \in \Omega_k} P_{km}(V_k, V_m, \theta_k, \theta_m) \tag{3.1}$$

$$Q_k + {Q_k}^{sh}(V_k) = \sum_{m \in \Omega_k} Q_{km}(V_k, V_m, \theta_k, \theta_m) \tag{3.2}$$

A potência ativa, P_k, injetada na barra é igual a soma das potência ativas que saem através das linhas. Da mesma forma, a soma da potência reativa, Q_k, e da potência reativa *shunt*, ${Q_k}^{sh}$, injetada na barra é igual a soma das potência reativas que saem através das linhas. Ambos os fluxos de potência são equacionados em função dos módulos e ângulos das tensões nas barras (MONTICELLI, 1983).

A potência reativa *shunt*, ${Q_k}^{sh}$, é capacitiva e é utilizada para aumentar a tensão na barra, compensando os reativos predominantemente indutivos do sistema. É modelada como uma carga em derivação da barra para a terra. O conjunto Ω_k é o conjunto de todas as barras adjacentes á barra k (MONTICELLI, 1983).

Para a correta aplicação do balanço dos fluxos de potência são consideradas algumas convenções:

- Nas barras de geração, as potências que entram na barra são positivas e as potência que saem da barra são negativas.
- Nas barras de carga, as potências que entram na barra são negativas e as potência que saem da barra são positivas.

Cada barra do sistema também está sujeita a um conjunto de inequações básicas que representam os limites de tensões nas barras PQ e os limites de potência reativa nas barras PV. Isto é devido ao fato de que esses valores são flutuantes em tais barras e dependem da solução do FP para o sistema. Tal modelagem não será aplicada a este trabalho.

3.2 Modelagem de Linhas

O modelo π básico de uma linha de transmissão é mostrado na Figura 3.1 (MONTICELLI, 1983).

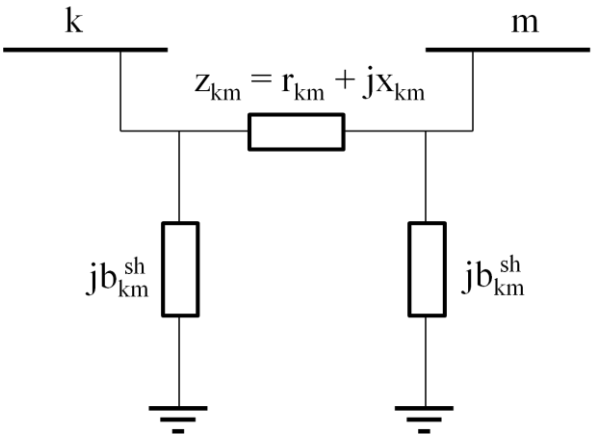

Figura 3.1: Modelo π de uma linha de transmissão.

Como pode ser visto na Figura 3.1, o modelo da linha de transmissão é formado por uma impedância série, z_{km}, e uma reatância *shunt*, ${b_{km}}^{sh}$. A impedância série da linha é composta por uma resistência série, r_{km}, e uma reatância série, x_{km}. Desta forma, a admitância série, y_{km}, pode ser definida na equação (3.3).

$$y_{km} = g_{km} + jb_{km} = \frac{r_{km}}{{r_{km}}^2 + {b_{km}}^2} - j\frac{x_{km}}{{r_{km}}^2 + {b_{km}}^2} \tag{3.3}$$

Pode-se verificar que g_{km} é positivo e b_{km} é negativo. Sendo r_{km} e b_{km} valores reais positivos, isto implica que o modelo da linha de transmissão tem característica indutiva.

Desta forma, a corrente I_{km} pode ser encontrada através da equação (3.4).

$$I_{km} = y_{km}(E_k - E_m) + j{b_{km}}^{sh}E_k \tag{3.4}$$

3.3 Forma matricial

A partir da primeira Lei de Kirchoff, podemos escrever a seguinte expressão para a conservação das correntes em uma barra do SEP:

$$I_k + {I_k}^{sh} = \sum_{m \in \Omega_k} I_{km} \tag{3.5}$$

Considerando a expressão para I_{km} descrita anteriormente, pode-se reescrever a equação (3.5) como mostrado na equação (3.6).

$$I_k = \left({jb_k}^{sh} + \sum_{m \in \Omega_k} ({jb_k}^{sh} + y_{km}) \right) E_k + \sum_{m \in \Omega_k} (-y_{km}) E_m \tag{3.6}$$

A equação (3.6) pode ser escrita então em forma matricial, $I = YE$, em que:

- I – Vetor das correntes injetadas em uma barra k do SEP;
- Y – matriz admitância nodal do SEP;
- E – Vetor das tensões nodais do SEP.

A matriz admitância é formada por dois tipos de elementos: os elementos da diagonal, Y_{kk}, e os elementos de fora da diagonal da matriz, Y_{km}. O equacionamento para cada tipo de elemento da matriz Y é mostrado nas equações (3.7) e (3.8).

$$Y_{km} = -y_{km} \tag{3.7}$$

$$Y_{kk} = {jb_k}^{sh} + \sum_{m \in \Omega_k} ({jb_{km}}^{sh} + y_{km}) \tag{3.8}$$

Sendo $k = 1, \dots, NB$, em que NB é o número de barras do SEP.

A matriz Y é quadrada, de dimensão NB. Esta matriz também é esparsa, ou seja, tem a maioria de seus elementos iguais a zero. Isto acontece pois cada barra do SEP não é ligada a todas as demais barras, e sim a apenas algumas poucas barras adjacentes. Desta forma, os elementos da diagonal, Y_{km}, são nulos tornando a matriz esparsa. Tal esparsidade deve ser tratada durante a modelagem computacional, como será visto na Seção 0.

A partir da definição da matriz admitância Y, a corrente I_k pode ser escrita da seguinte forma:

$$I_k = Y_{kk} E_k + \sum_{m \in \Omega_k} Y_{km} E_m = \sum_{m \in K} Y_{km} E_m \tag{3.9}$$

Em que K é o conjunto de todas as barras adjacentes à barra k, incluindo a própria barra k (MONTICELLI, 1983).

A matriz admitância, Y, pode ser dividida em duas partes: G, matriz de condutância nodal, referente a parte real da matriz Y e B, matriz de susceptância nodal, referente a parte imaginária de Y.

A tensão fasorial na m-ésima barra, E_m, pode ser escrita em sua forma polar. Assim, a equação de I_k pode ser reescrita como na equação (3.10).

$$I_k = \sum_{m \in K} (G_{km} + jB_{km})(|E_m|e^{j\theta_m}) \tag{3.10}$$

A potência aparente, S_k, em cada barra, pode ser escrita em função da tensão e a da corrente na barra, como na equação (3.11) a seguir:

$$S_k{}^* = V_k{}^* I_k = |E_k|e^{-j\theta_k} \sum_{m \in K} (G_{km} + jB_{km})(|E_m|e^{j\theta_m}) \tag{3.11}$$

Após algumas passagens algébricas em (3.11) e separando as partes real e imaginária, obtem-se as expressões (3.12) e (3.13) para as potências ativa e reativa em cada barra k do SEP (MONTICELLI, 1983).

$$P_k = V_k \sum_{m \in K} V_m (G_{km} sen\theta_{km} + jB_{km} \cos\theta_{km}) \tag{3.12}$$

$$Q_k = V_k \sum_{m \in K} V_m (G_{km} cos\theta_{km} - jB_{km} \mathrm{sen}\theta_{km}) \tag{3.13}$$

3.4 Algoritmo Básico

O problema do fluxo de potência consiste primeiramente em encontrar o atual estado da rede, ou seja, todos os módulos e ângulos das tensões nas barras do SEP. Após esta etapa as potências em todas as barras podem ser determinadas utilizando-se as expressões (3.12) e (3.13). Desta forma, os fluxos de potência em todas as linhas do SEP podem ser facilmente encontrados a partir do balanço das potências entre as barras.

A partir dos tipos de barras descritos na Seção 0, pode-se formar a Tabela 3.1.

Tabela 3.1: Incógnitas e dados para o problema do fluxo de potência.

Tipo de Barra	P	Q	θ	V
Vθ	Incógnita	Incógnita	Dado	Dado
PV	Dado	Incógnita	Incógnita	Dado
PQ	Dado	Dado	Incógnita	Incógnita

A Tabela 3.1 mostra quais as variáveis de cada barra são dados do problema e quais são incógnitas. Portanto, o algoritmo básico para solução do problema do fluxo de potência pode ser dividido em duas partes:

1. Encontrar os ângulos e módulos das tensões das barras PQ e os ângulos das tensões nas barras PV, determinando assim o estado atual da rede;
2. a partir do estado da rede determinado no passo 1, encontrar as potências ativa e reativa na barra Vθ e a potência reativa na barra PV, utilizando as equações definidas anteriormente para P_k e Q_k.

Cada parte do algoritmo básico será composta de um subsistema de equações a ser resolvido. A definição de cada subsistema será mostrada nas subseções seguintes.

3.4.1 Algoritmo Básico – Parte 1

A primeira parte do algoritmo básico pode ser resolvida utilizando o método de Newton-Raphson para a determinação do estado da rede. Nesta etapa, são dados do sistema:

- As potências P e Q nas barras PQ e as potência P nas barras PV;
- O ângulo e o módulo da tensão na barra Vθ e os módulos das tensões nas barras PV.

Sejam NPQ e NPV o número de barras PQ e o número de barras PV, respectivamente. Desta forma, pode-se definir um sistema de equações, $g(z)$, de dimensão $m = 2NPQ + NPV$, formado a partir dos valores conhecidos de P e Q, apresentado na equação (3.14) (MONTICELLI, 1983).

$$g(z) = \begin{bmatrix} {P_k}^{esp} - V_k \sum_{m \in K} V_m (G_{km} sen\theta_{km} + B_{km} \cos\theta_{km}) \\ {Q_k}^{esp} - V_k \sum_{m \in K} V_m (G_{km} cos\theta_{km} - B_{km} \mathrm{sen}\theta_{km}) \end{bmatrix} \tag{3.14}$$

As potências ${P_k}^{esp}$ serão todas as potências conhecidas do sistema, ou seja, as potências ativas líquidas em cada barra PV e cada barra PQ do SEP. Da mesma forma, as potências ${Q_k}^{esp}$ serão todas as potências conhecidas do sistema, ou seja, as potências reativas de todas as barras PQ do SEP.

O sistema $g(z)$ também pode ser escrito através das equações (3.15) e (3.16):

$$\Delta P = {P_k}^{esp} - {P_k}^{calculado} \tag{3.15}$$

$$\Delta Q = {Q_k}^{esp} - {Q_k}^{calculado} \tag{3.16}$$

Define-se também o vetor, z, de mesma dimensão que $g(z)$, formado pelos valores deconhecidos de V e θ. Desta forma, o vetor z pode ser escrito como:

$$z = \begin{bmatrix} \theta_k \\ V_k \end{bmatrix} \tag{3.17}$$

Em que θ_k é o conjunto de todos os ângulos de tensão desconhecidos do sistema, ou seja, os ângulos de tensão em todas as barras PQ e os ângulos de tensão em todas as barras PV do SEP. Da mesma forma, V_k é o conjunto de todos os módulos de tensão desconhecidos do sistema, ou seja, os módulos de tensão das barras PV do SEP.

Para que o estado da rede seja encontrado, deve-se encontrar o vetor z que satisfaz a seguinte expressão:

$$g(z) = 0 \tag{3.18}$$

A solução deste sistema de equações não lineares pode ser encontrada segundo o método de Newton-Raphson (MONTICELLI, 1983). O método consiste na generalização do método de Newton para um sistema multidimensional e sua aplicação será descrita na seção seguinte.

3.4.1.1 Aplicação do Método de Newton-Raphson

Dado um sistema n-dimensional de equações não lineares, $g(z) = 0$, e um vetor de incognitas z de mesma dimensão, o algoritmo de Newton-Raphson pode ser expresso como:

1. Escolher uma solução inicial $z = z^{(i)} = z^{(0)}$;
2. Calcular $g\left(z^{(i)}\right)$;
3. Verificar se $\left|g\left(z^{(i)}\right)\right| \leq \epsilon$: se a condição for satisfeita o processo convergiu para $z^{(i)}$, senão passa-se para a próxima iteração;
4. Calcular a jacobiana $J\left(z^{(i)}\right)$;
5. Encontrar $\Delta z^{(i)}$ a partir da solução do sistema linear: $-J\left(z^{(i)}\right)\Delta z^{(i)} = g\left(z^{(i)}\right)$;
6. Atualizar $z^{(i+1)} = z^{(i)} + \Delta z^{(i)}$.
7. voltar para passo 2

Ao final do algoritmo encontra-se o vetor z final, o qual representa o estado da rede no momento analisado. Abaixo, serão feitas algumas definições para que a solução do algoritmo possa ser implementada computacionalmente.

O sistema de equações, $g(z)$, pode então ser reescrito como o vetor representado em (3.19).

$$g\left(z^{(i)}\right) = \begin{bmatrix} \Delta P^{(i)} \\ \Delta Q^{(i)} \end{bmatrix} \tag{3.19}$$

O vetor de incógnitas, z, pode ser expresso como:

$$z^{(i)} = \begin{bmatrix} \Delta\theta^{(i)} \\ \Delta V^{(i)} \end{bmatrix} \tag{3.20}$$

A partir dos dois vetores apresentados acima, pode-se escrever a Jacobiana do sistema de equações, $g(z)$, com relação ao vetor z como:

$$J\left(z^{(i)}\right) = \begin{bmatrix} \dfrac{\partial \Delta P}{\partial \theta} & \dfrac{\partial \Delta P}{\partial V} \\ \dfrac{\partial \Delta Q}{\partial \theta} & \dfrac{\partial \Delta Q}{\partial \mathrm{V}} \end{bmatrix}^{(i)} \tag{3.21}$$

Considerando-se as expressões (3.15) e (3.16) para ΔP e ΔQ, e que ${P_k}^{esp}$ e ${Q_k}^{esp}$ são valores constantes, pode-se reescrever a matriz jacobiana da seguinte forma:

$$J(\mathrm{z}) = -\begin{bmatrix} \dfrac{\partial P}{\partial \theta} & \dfrac{\partial P}{\partial V} \\ \dfrac{\partial Q}{\partial \theta} & \dfrac{\partial Q}{\partial \mathrm{V}} \end{bmatrix}^{(i)} \tag{3.22}$$

Desta forma, a matriz jacobiana pode ser dividida em quatro submatrizes, formadas pelas derivadas apresentadas acima. Considerando-se as expressões de cálculo mostradas para P_k e Q_k pode-se escrever cada submatriz como mostrado nos conjuntos de equações abaixo.

$$\begin{aligned} \frac{\partial P_k}{\partial \theta_{\mathrm{m}}} &= V_k V_m (G_{km} sen\theta_{km} - B_{km} \cos\theta_{km}) \\ \frac{\partial P_k}{\partial \theta_{\mathrm{k}}} &= -B_{kk} {V_k}^2 - V_k \sum_{m \in K} V_m (G_{km} sen\theta_{km} - B_{km} \cos\theta_{km}) \end{aligned} \tag{3.23}$$

$$\begin{aligned} \frac{\partial P_k}{\partial \mathrm{V_m}} &= V_k (G_{km} cos\theta_{km} + B_{km} \mathrm{sen}\theta_{km}) \\ \frac{\partial P_k}{\partial \mathrm{V_k}} &= G_{kk} V_k + \sum_{m \in K} V_m (G_{km} cos\theta_{km} + B_{km} \mathrm{sen}\theta_{km}) \end{aligned} \tag{3.24}$$

$$\begin{aligned} \frac{\partial Q_k}{\partial \theta_{\mathrm{m}}} &= -V_k V_m (G_{km} cos\theta_{km} + B_{km} \mathrm{sen}\theta_{km}) \\ \frac{\partial Q_k}{\partial \theta_{\mathrm{k}}} &= -G_{kk} {V_k}^2 + V_k \sum_{m \in K} V_m (G_{km} cos\theta_{km} + B_{km} \mathrm{sen}\theta_{km}) \end{aligned} \tag{3.25}$$

$$\begin{aligned} \frac{\partial Q_k}{\partial \mathrm{V_m}} &= V_k (G_{km} sen\theta_{km} - B_{km} \cos\theta_{km}) \\ \frac{\partial Q_k}{\partial \mathrm{V_k}} &= -B_{kk} V_k + \sum_{m \in K} V_m (G_{km} sen\theta_{km} - B_{km} \cos\theta_{km}) \end{aligned} \tag{3.26}$$

3.4.2 Algoritmo Básico – Parte 2

Nesta etapa do algoritmo o estado da rede já é conhecido, ou seja, todos os módulos e ângulos de tensão de todas as barras do SEP já forma encontrados. Desta forma, deve-se calcular os valores de potência ativa, P, para todas as barras Vθ e os valores de potência reativa, Q, para todas as barras PV e Vθ do SEP. As expressões utilizadas serão as equações (3.12) e (3.13) definidas anteriormente.

3.5 Implementação Computacional – Fluxo de Potência

Nesta Seção será descrita a modelagem computacional implementada para a solução do problema do fluxo de potência, utilizando os conceitos da programação orientada a objetos e os recursos oferecidos pela linguagem C++. Em um primeiro momento serão apresentadas, de forma breve, as classes que fazem parte do *framework* desenvolvido especialmente para desenvolvimento de soluções computacionais para a análise de sistemas elétricos de potência (MANSOUR *et al.*, 2011). Estas classes serão utilizadas na modelagem mas não fazem parte do escopo deste trabalho. Em seguida, serão apresentadas detalhadamente as classes criadas para a implementação do fluxo de potência, bem como seu funcionamento.

3.5.1 Algoritmo computacional

Duas considerações serão feitas para que a implementação do algoritmo apresentado na Seção 0 seja mais prática e eficiente:

- O sistema de equações de dimensão $m = 2NPQ + NPV$ passará a ser representado por um sistema de dimensão $m = 2NB$, em que NB é número total de barras do sistema.
- A solução do sistema linear necessária no algoritmo será realizada através do método de Gauss com critério de Markowitz.

A consideração com relação à dimensão do sistema de equações se deve ao elevado esforço computacional necessário para se montar os vetores e matrizes quando o sistema tem dimensão $m = 2NPQ + NPV$. Neste caso é necessária a varredura de todas as barras do sistema, verificando-se o tipo de cada barra para que se possa tomar decisões sobre a utilização de seus dados para a formação do vetor ou matriz correspondente.

Desta forma, o sistema passa a ter dimensão $m = 2NB$. O vetor z agora terá os valores de ângulo e módulo de tensão de todas as barras do SEP, facilitando a leitura e armazenamento destes valores. O vetor $g(x)$ terá a mesma estrutura, representando os valores de ΔP e ΔQ para todas as barras do sistema. Para que este sistema represente de

forma válida o sistema original, todos os valores de ΔP e ΔQ que não fazem parte do conjunto de equações definido por (3.19) serão nulos, representando que o valor calculado deverá sempre ser igual ao valor esperado. Com isso, a matriz Jacobiana passa a ter dimensão 2NBx2NB. Para que sua implementação seja válida, da mesma forma que no vetor $g(x)$, os componentes da matriz que não fazem parte da matriz original serão nulos e suas diagonais terão valor unitário. A equação (3.27) ilustra o sistema linear a ser resolvido de acordo com as considerações supracitadas e conforme o passo 5 do algoritmo de NewtonRaphson apresentado na Seção 0.

$$-\begin{bmatrix} \frac{\partial P}{\partial \theta} & \frac{\partial P}{\partial V} \\ \frac{\partial Q}{\partial \theta} & \frac{\partial Q}{\partial \mathrm{V}} \end{bmatrix}_{2\mathrm{NBx2NB}} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix}_{2\mathrm{NB}} = \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix}_{2\mathrm{NB}} \tag{3.27}$$

Outro fator relevante no sistema computacional desenvolvido neste trabalho é a exploração da característica esparsa. Conforme foi mencionado nas seções anteriores, as matrizes jacobiana (J) e de admitância notal (Y) apresentam um grau de esparsidade bastante elevado e com o objetivo de diminuir os custos computacionais, neste trabalho utilizou-se uma metodologia desenvolvida em (MANSOUR *et al.*, 2011) para a resolução de sistemas lineares esparsos. Este método baseia-se no método de Gauss-Markowitz, que é um método para solução de sistema lineares que utiliza o critério de pivoteamento de Markowitz para tratar esparsidade.

A partir das considerações apresentadas, o algoritmo computacional desenvolvido para a solução do fluxo de potência é mostrado no fluxograma da Figura 3.2.

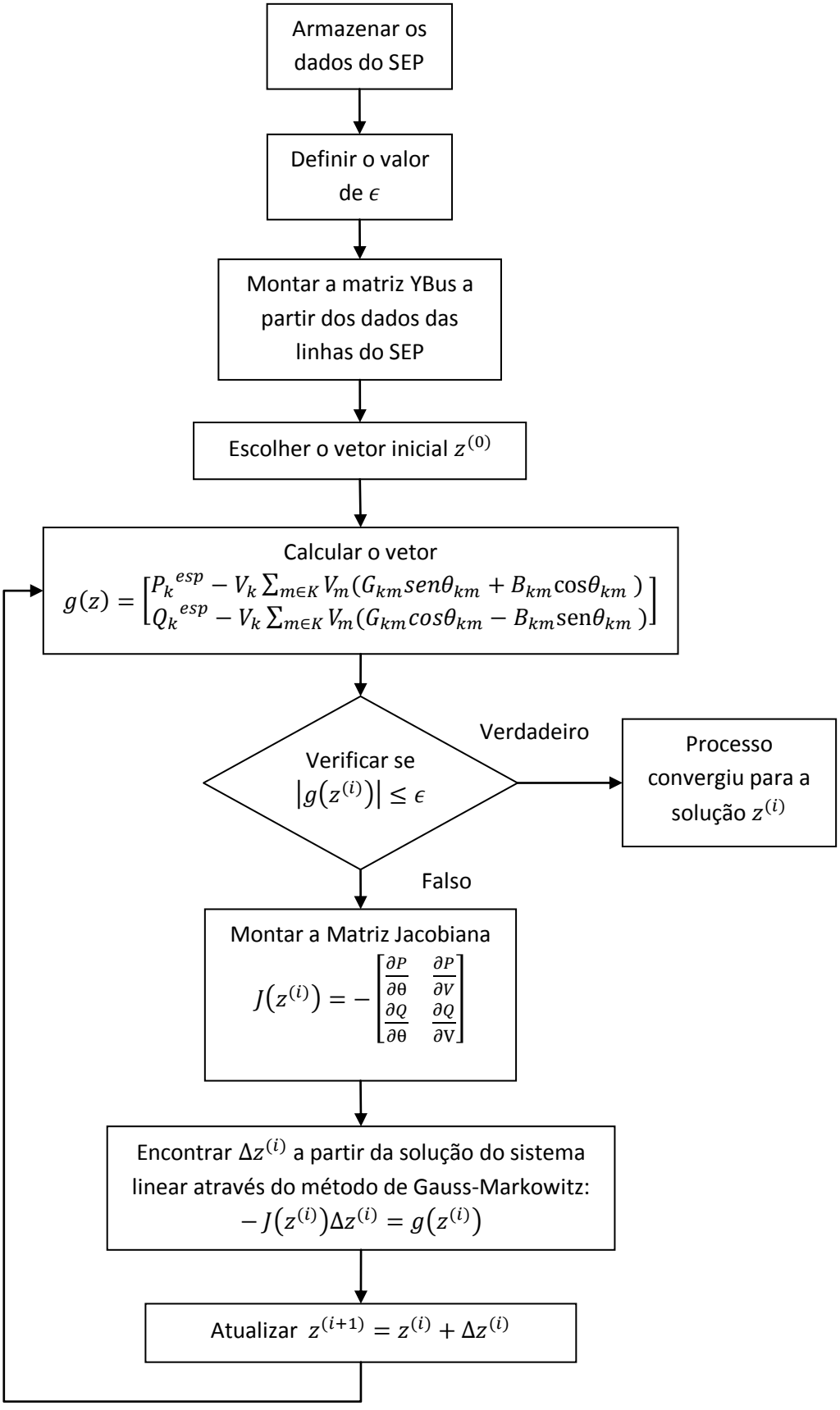

Figura 3.2: Algoritmo computacional implementado para a solução do método de Newton-Raphson.

3.5.2 Framework para desenvolvimento de soluções computacionais para análise de SEP (MANSOUR *et al.*, 2011)

Este *framework* é formado por um conjunto de bibliotecas desenvolvidas para a análise de um SEP de grande porte. Cada biblioteca é formada por um conjunto de classes que unidas formam a estrutura funcional da biblioteca. O framework inclui as seguintes bibliotecas:

- EPSD (*Electrical Power System Data*): estrutura de armazenamento de dados de um sistema elétrico.
- LSSOLVER (*Linear System Solver*): biblioteca com funções para resolução de sistemas lineares esparsos utilizando método de eliminação de Gauss com critério de Markowitz.
- DBModels (*Data Base Models*): biblioteca com funções de conversão de arquivos ANAREDE para XML (eXtensible Markup Language) e de XML para a estrutura EPSD.

Neste trabalho serão utilizadas classes pertencentes a cada uma dessas bibliotecas. As classes inerentes a este trabalho são: “XMLReader”, “EPSD”, “SpMatrix” e “GaussMarkowitz”. Cada classe será apresentada de forma breve nas subseções seguintes.

3.5.2.1 Classe XMLReader

A classe XMLReader faz parte da biblioteca DBModels e tem a função de ler arquivos de entrada em formato XML, contendo as informações referentes ao SEP, e armazena-los em uma estrutura EPSD. A utilização desta classe neste trabalho é restrita apenas a leitura de dados e o seu respectivo armazenamento em um objeto do tipo EPSD, descrito na subseção seguinte.

3.5.2.2 Classe EPSD

A classe EPSD é a principal classe da biblioteca EPSD e utiliza várias classes pertencentes a essa biblioteca para armazenar os dados elétricos de um SEP. Através dessa classe pode-se obter informações sobre todas as barras e linhas do SEP.

As barras possuem dados essenciais, como: injeções de potência ativa e reativa, módulo e ângulo de tensão e o seu respectivo tipo. Todas as barras do SEP são armazenadas em vetor de objetos do tipo barra e suas informações podem ser acessadas a partir de métodos da classe EPSD. Da mesma forma, os valores de uma linha de transmissão, como resistência e reatância série e reatância *shunt* podem ser obtidos facilmente através dos métodos desta classe.

3.5.2.3 Classe SpMatrix

A classe SpMatrix faz parte da biblioteca LSSolver e é responsável por armazenar uma matriz esparsa de forma eficiente em termos de uso de memória. A classe armazena apenas os elementos não nulos da matriz esparsa, não alocando memória do sistema computacional para armazenar os elementos nulos, os quais formam a maior parte da matriz. Desta forma, reduz-se consideravelmente a utilização de memória quando um SEP de grande porte é considerado.

3.5.2.4 Classe GaussMarkowitz

A classe GaussMarkowitz é a principal classe da biblioteca LSSolver. Esta classe resolve um sistema linear esparso de matriz quadrada com dimensões NxN utilizando o método de Gauss com critério de Markowitz. Para este fim são utilizadas outras classes pertencentes à biblioteca LSSolver, incluindo a classe SpMatrix.

O critério de Markowitz determina, de forma ótima, o melhor pivô para o método de Gauss considerando a esparsidade do sistema. Desta forma, reduz-se de forma considerável o processamento computacional necessário para se resolver o sistema linear.

Esta classe é utilizada neste trabalho para resolver o sistema linear esparso gerado a cada iteração do método de Newton-Raphson. A classe recebe como entrada uma matriz esparsa alocada em formato *double*. Esta matriz é convertida automaticamente pela classe para um objeto do tipo SpMatrix. Em seguida, o método de Gauss com critério de Markowitz é executado. Como saída, obtém-se o vetor independente, solução do sistema linear, em formato *double*.

3.6 Classes – Fluxo de Potência

Nesta seção serão descritas em detalhes as classes desenvolvidas neste projeto. Serão apresentados os Diagramas de Classe desenvolvidos para cada uma delas, bem como seus atributos e métodos. As classes desenvolvidas são: “YBus” e “NewtonRaphson”.

3.6.1 Classe YBus

A classe YBus representa a matriz admitância nodal, Y, apresentada durante o equacionamento do fluxo de potência. Esta classe armazena as matrizes de condutância nodal, G, e de susceptância nodal, B, em forma de atributos e calcula cada um de seus elementos através de seus métodos. De acordo com o Diagrama de Classe mostrado na Figura 3.3 pode-se compreender melhor a estrutura da classe YBus.

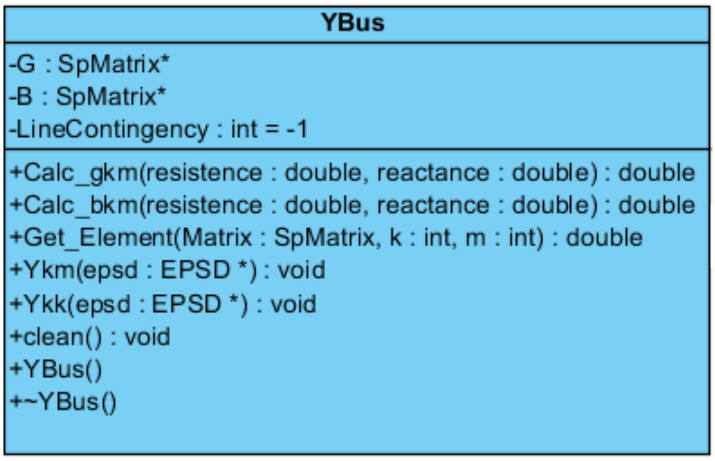

Figura 3.3: Diagrama de Classe da classe YBus.

3.6.1.1 Atributos

Os principais atributos da classe YBus são as matrizes ***G*** e ***B*** que representam as matrizes de condutância e susceptância nodal, respectivamente. Podemos verificar na estrutura do objeto que ***G*** e ***B*** são dois ponteiros da classe SpMatrix apresentada na Seção 0.

O programa desenvolvido deve ser capaz de receber os dados de entrada para um SEP de grande porte. Não devem haver restrições com relação ao número de barras ou linhas que formam o SEP. Como as dimensões das matrizes ***G*** e ***B*** dependem do número de linhas do SEP, o programa deverá ser capaz de alocar espaço em memória de forma variável, de acordo com o tamanho do SEP analisado. Desta forma, sendo ***G*** e ***B*** ponteiros, isto permite a alocação dinâmica de memória pelo programa, em tempo de execução, podendo atender aos objetivos supracitados.

A alocação dinâmica das matrizes ***G*** e ***B*** ocorre durante a criação de um objeto desta classe quando o construtor da classe é chamado. Desta forma, o construtor deve receber o tamanho da matriz a ser alocada, como será mostrado mais adiante na seção de métodos.

Além disso, o atributo ***LineContingency*** é responsável por armazenar qual linha do SEP está fora de operação. Esta variável é utilizada durante a análise de contingências. O valor *default* para o atributo é igual a -1 e indica que nenhuma linha está fora de operação.

Em resumo, os atributos da classe podem ser descritos como mostrado abaixo:

- ***SpMatrix * G***: ponteiro de objeto da classe SPMatrix responsável por armazenar os dados da matriz de condutância nodal.

- ***SpMatrix * B:*** ponteiro de objeto da classe SPMatrix responsável por armazenar os dados da matriz de susceptância nodal.
- ***int LineContingency:*** inteiro responsável por indicar qual linha do SEP está fora de operação para o processo de análise de contingências.

3.6.1.2 Métodos

Os métodos da classe YBus são responsáveis por calcular cada elemento das matrizes ***G*** e ***B***, a partir das equações (3.7) e (3.8) apresentadas na Seção 0. A descrição detalhada de cada método será feita a seguir.

- **double calc_gkm(double r, double x)**

O método **Calc_gkm** é responsável por calcular o parâmetro g_{km} que compõe a admitância série, y_{km}, do modelo de linha de transmissão. Através da equação (3.3) o método recebe como argumentos os valores reais de r_{km} e x_{km} e retorna o valor de g_{km} em formato *double*.

- **double calc_bkm(double r, double x)**

Da mesma forma, o método **Calc_bkm** é responsável por calcular o parâmetro b_{km} que compõe a admitância série, y_{km}, do modelo de linha de transmissão. Através da equação (3.3) o método recebe como argumentos os valores reais de r_{km} e x_{km} e retorna o valor de b_{km} em formato *double*.

- **void Ykm(EPSD* epsd)**

O método **Ykm** é responsável por calcular e alocar os elementos de fora da diagonal da matriz Y. Como pode ser visto na equação (3.7), o elemento Y_{km} modelado é a própria admitância série do modelo de linha de transmissão para a linha situada entre as barra k e m. Desta forma, o método calcula o valor de y_{km} para cada linha de transmissão do SEP e aloca o valor calculado na posição (k,m) da matriz YBus.

O objeto ***epsd*** da classe EPSD, passado por referência a este método, armazena um vetor com os dados de todas as linhas do SEP. Através dos métodos oferecidos pela classe EPSD é possível fazer a varredura de todas as linhas do sistema e obter os valores de resistência, reatância série e reatância *shunt*, bem como os índices das barras as quais a linha está conectada (k e m). Através dos métodos **Calc_gkm** e **Calc_bkm** é possível obter o valor de y_{km} e através dos índices k e m obtidos pode-se alocar o valor calculado na posição correta da matriz. O algoritmo básico utilizado é descrito no fluxograma da Figura 3.4.

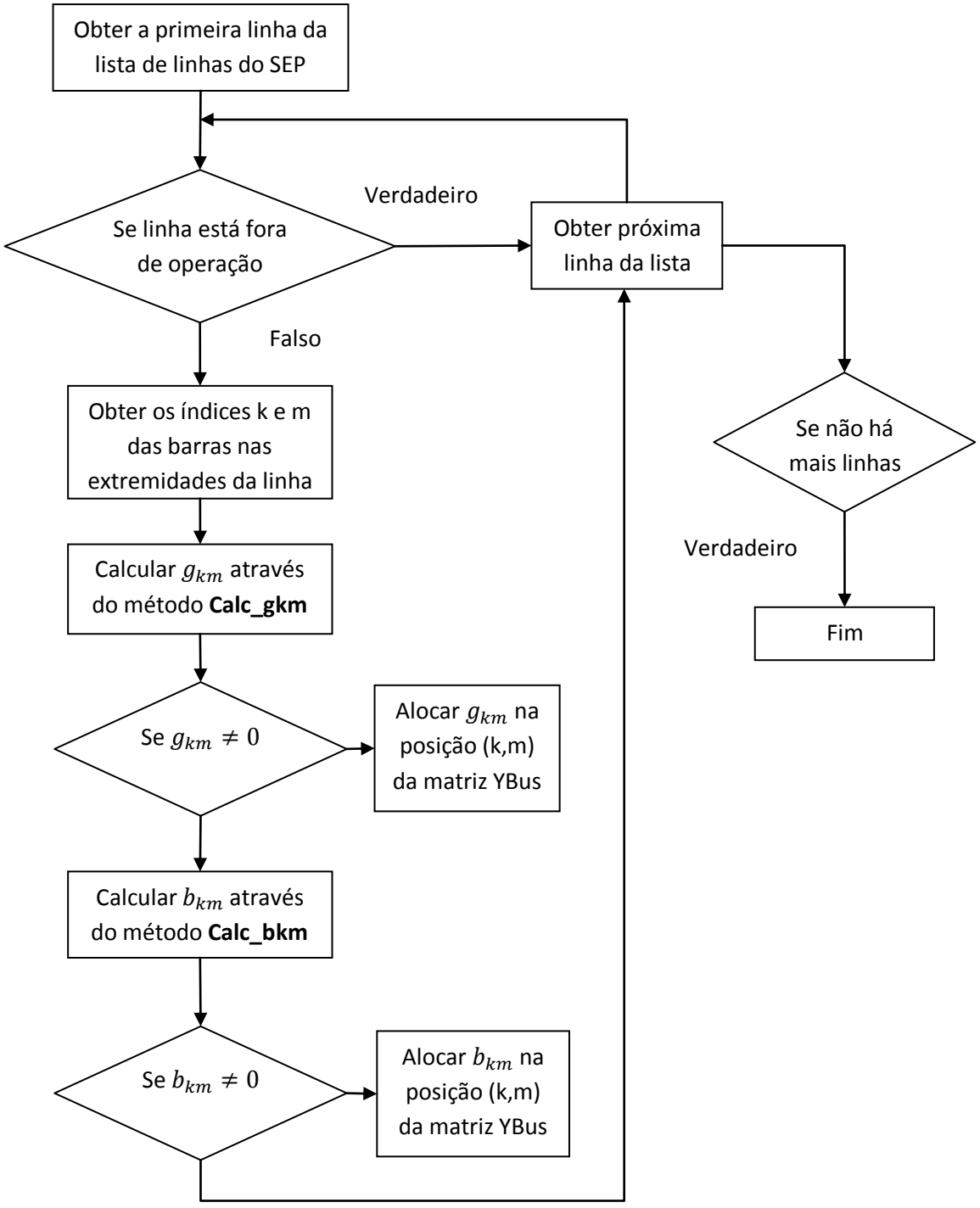

Figura 3.4: Algoritmo computacional implementado para o método Ykm da classe YBus.

- **void Ykk(*EPSD epsd)**

O método **Ykk** é responsável por calcular e alocar os elementos da diagonal da matriz YBus. O cálculo dos elementos da diagonal é feito como mostrado na equação (3.8). Para cada barra k do SEP obtem-se o valor de sua reatância *shunt* de barra. Além disso, é

necessário fazer varredura de todas as linhas adjacentes à barra k e obter seus parâmetros. Os valores inseridos na diagonal para a matriz ***G*** e a para a matriz ***B*** são calculados de forma distinta.

A classe EPSD possui um lista de todas as barras do SEP e também um método com o qual pode-se obter cada linha adjacente a uma barra. Da mesma forma que no método **Ykm**, a partir da varredura das linhas é possível obter-se os valores de resistência, reatância série e reatância *shunt*, bem como os índices das barras as quais a linha está conectada (k e m). O algoritmo básico utilizado é mostrado a seguir em linguagem estruturada.

1. Obter a k-ésima barra da lista de barras do SEP;
2. Obter o valor da reatância *shunt* da barra, $b_k{}^{sh}$;
3. Para cada linha adjacente à barra k:

   3.1. Se linha não está fora de operação:

      3.1.1. Calcular g_{km};

      3.1.2. Calcular b_{km};

      3.1.3. Obter $b_{km}{}^{sh}$;

      3.1.4. Acumular $g_{kk}{}^{\text{novo}} = g_{kk}{}^{\text{atual}} + g_{km}$;

      3.1.5. Acumular $b_{kk}{}^{\text{novo}} = b_{kk}{}^{\text{atual}} + b_{km} + b_{km}{}^{sh}$;

   3.2. Fim Se;
4. Fim Para;
5. Fazer $b_{kk} = b_{kk} + b_k{}^{sh}$;
6. Se g_{kk} é não nulo, alocar na posição (k,k) da matriz ***G***;
7. Se b_{kk} é não nulo, alocar na posição (k,k) da matriz ***B***;
8. Obter a próxima barra da lista de barras do SEP.

- **YBus(int size)**

O construtor da classe YBus é responsável por alocar dinamicamente as duas matrizes do tipo SpMatrix, ***G*** e ***B***, respectivamente, e apontá-las aos seus respectivos ponteiros, atributos da classe. Verifica-se a necessidade prévia de se conhecer o tamanho da matriz YBus para que esta possa ser criada.

- **~YBus()**

O destrutor desta classe tem a função de liberar o espaço alocado em memória pelo construtor, onde estão alocadas as matrizes ***G*** e ***B***.

3.6.2 Classe Newton-Raphson

A classe NewtonRaphson foi criada com o objetivo de executar o algoritmo do método de Newton-Raphson propriamente dito e também armazenar os vetores e matrizes inerentes. O Diagrama de Classe mostrado na Figura 3.5 mostra a estrutura da classe criada.

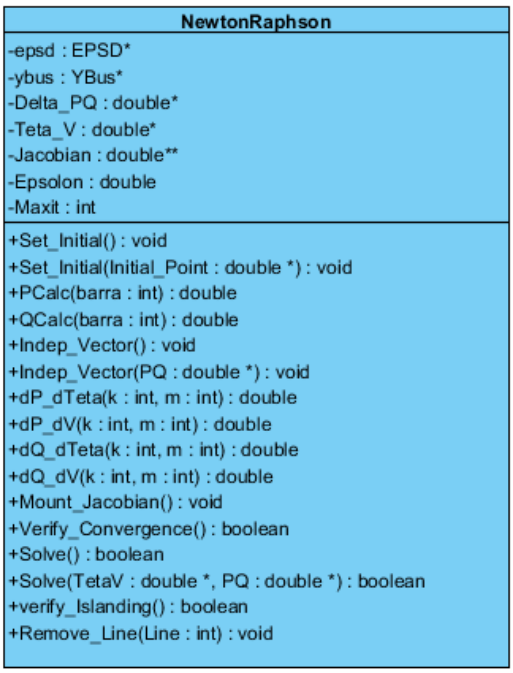

Figura 3.5: Diagrama de Classe da classe NewtonRaphson.

Como pode ser visto no diagrama da Figura 3.5, a classe NewtonRaphson agrega a classe YBus e a classe EPSD. Além disso, a classe NewtonRaphson também armazena o vetor independente, $g(z)$, o vetor de incógnitas, z, e a matriz Jacobiana, $J(z)$, representados respectivamente pelos atributos ***Delta_PQ***, ***Teta_V*** e ***Jacobian***. Também estão presentes os métodos necessários para montar estas estruturas e resolver o algoritmo de Newton-Raphson. Os atributos e métodos da classe são descritos nas subseções seguintes.

3.6.2.1 Atributos

A classe NewtonRaphson contém um ponteiro de objeto da classe EPSD, chamado ***epsd***, que recebe um objeto de mesma classe, responsável por armazenar os dados do SEP. O ponteiro ***epsd*** fica alocado de forma conveniente nesta classe, pois quase todos os seus métodos acessam os dados armazenados neste. A partir dos dados do SEP armazenados a classe pode então alocar dinamicamente os demais atributos durante a execução de seu construtor.

A classe EPSD armazena o número total de barras do sistema, bem como o número total de linhas existentes, os quais podem ser obtidos através de métodos específicos. A partir do número de barras pode-se definir o tamanho dos vetores ***Delta_PQ*** e ***Teta_V*** e também da matriz ***Jacobiana***. Estes atributos são alocados dinamicamente durante a execução do construtor da classe. Da mesma forma, sabendo-se o número total de linhas do sistema pode-se alocar dinamicamente o objeto da classe YBus.

A descrição de cada atributo é dada a seguir.

- ***EPSD * epsd***: ponteiro do tipo EPSD responsável por receber o objeto EPSD que contém os dados do SEP.
- ***YBus * ybus***: ponteiro para a classe YBus responsável por receber o objeto YBus alocado durante a construção da classe NewtonRaphson.
- ***double * Delta_PQ***: ponteiro do tipo *double* responsável por armazenar o vetor $g(z)$ alocado dinamicamente durante a construção da classe.
- ***double * Teta_V***: ponteiro do tipo *double* responsável por armazenar o vetor z alocado dinamicamente durante a construção da classe.
- ***double ** Jacobian***: vetor de ponteiros do tipo *double* por armazenar a matriz $J(z)$ alocada dinamicamente durante a construção da classe.
- ***double Epsilon***: valor da variável ϵ utilizada no critério de parada no algoritmo de Newton-Raphson.
- ***int Maxit***: número máximo de iterações para a convergência do método de Newton-Raphson.

3.6.2.2 Métodos

Os métodos pertencentes à classe NewtonRaphson são descritos abaixo, juntamente com seus algoritmos implementados.

- **void Set_Initial( )**

O método **Set_Initial** é responsável por armazenar a solução inicial do problema do fluxo de potência no vetor ***Teta_V***. Este método monta a solução inicial conhecida como *flat start*. Os valores de V e θ da barra *slack* e os valores de V para as barras PV são preenchidos no vetor ***Teta_V***. Os demais valores, incógnitas para o sistema, são preenchidos com 1 para V e 0 para θ. O algoritmo utilizado é mostrado abaixo em linguagem estruturada.

1. Obter NB, número total de barras do SEP;

    1.1. Para cada barra k do sistema:

        1.1.1. Se a barra k for do tipo **Vθ**, então:

            1.1.1.1. $\mathbf{Teta\_V}(\mathrm{k}) = V_k$

            1.1.1.2. $\mathbf{Teta\_V}(\mathrm{k} + \mathrm{NB}) = \theta_k$

        1.1.2. Fim Se

        1.1.3. Se a barra k for do tipo **PV**, então:

            1.1.3.1. $\mathbf{Teta\_V}(\mathrm{k}) = V_k$

            1.1.3.2. $\mathbf{Teta\_V}(\mathrm{k} + \mathrm{NB}) = 0$

        1.1.4. Fim Se;

        1.1.5. Se a barra k for do tipo **PQ**, então:

            1.1.5.1. $\mathbf{Teta\_V}(\mathrm{k}) = 1$

            1.1.5.2. $\mathbf{Teta\_V}(\mathrm{k} + \mathrm{NB}) = 0$

        1.1.6. Fim Se;

    1.2. Fim para;

    1.3. Fim.

Vale ressaltar que o vetor ***Teta_V*** tem dimensão $m = 2NB$. Os primeiros NB elementos do vetor ***Teta_V*** armazenam os valores de módulo de tensão para cada barra k do SEP. Os próximos NB elementos armazenam os valores de ângulo de tensão para cada barra k do SEP.

- **void Set_Initial(double * Initial_Point)**

Neste caso o método **Set_Initial** é sobrecarregado, sendo que o vetor de solução inicial é passado externamente por referência. O método então copia os valores fornecidos no vetor ***Inital_Point*** para o vetor ***Teta_V***. Desta forma, a solução pode ser fornecida externamente, não sendo necessária a execução do *flat start*.

- **double PCalc(int barra)**

O método **Pcalc** é uma função auxiliar responsável por calcular o valor de P_k utilizando a expressão (3.12) para a barra definida pelo índice ***barra***, argumento de entrada do método. O valor de P_k é calculaldo utilizando-se os dados das matrizes ***G*** e ***B***, obtidas na classe YBus, e os dados de ângulo e módulo de tensão armazenados no vetor ***Teta_V***. O

valor de P_k é retornado em formato double. O algoritmo utilizado para o calcula de P_k é descrito no fluxograma da Figura 3.6.

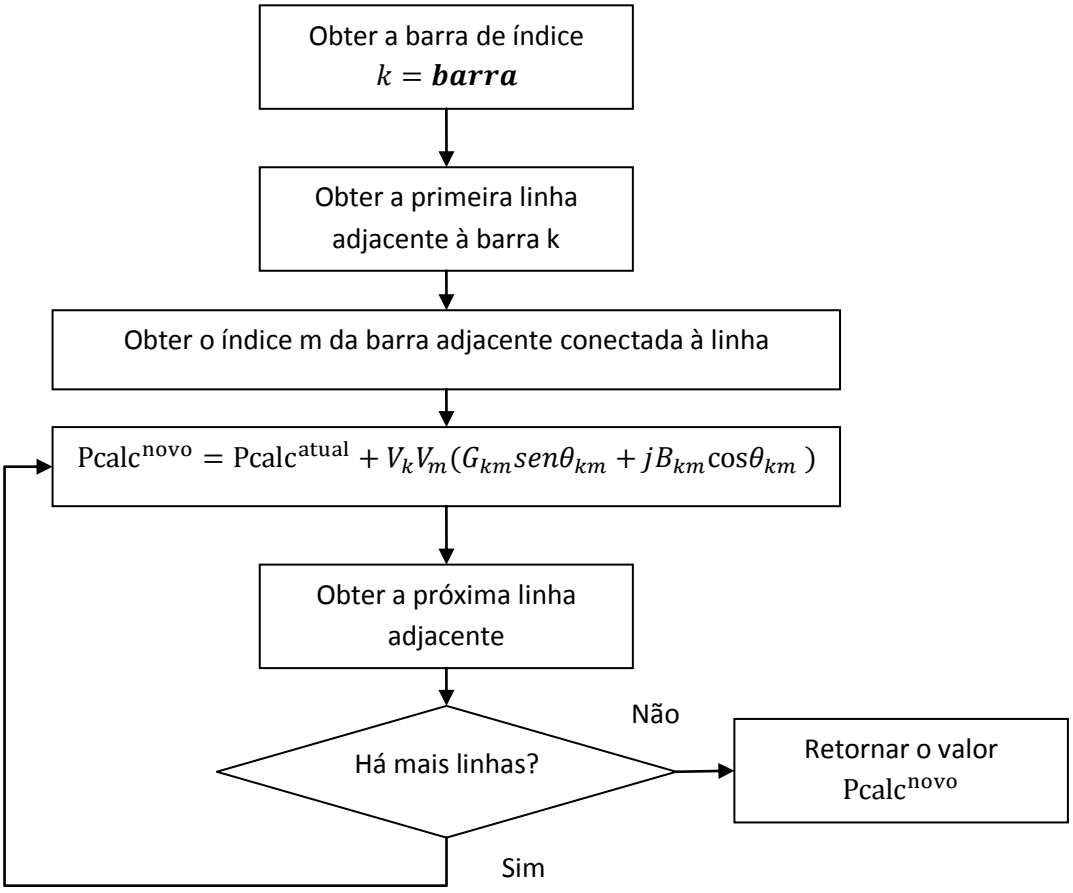

Figura 3.6: Algoritmo implementado para o método Pcalc(int barra)

- **double QCalc(int barra)**

Da mesma forma que no método **PCalc**, o método **QCalc** calcula o valor de Q_k utilizando a expressão (3.13). Também são utilizados os valores obtidos na classe YBus e no vetor ***Teta_V***. O algoritmo utilizado tem o mesmo formato que o descrito para o método **PCalc**, porém agora a expressão (3.13) é utilizada.

- **void Indep_Vector( )**

O método **Indep_Vector** é responsável por montar o vetor ***Delta_PQ*** utilizando os métodos **PCalc** e **QCalc** descritos anteriormente. Para todas as barras do SEP, os valores ΔP e ΔQ são calculados de acordo com (3.15) e (3.16). Logo após, os valores de ΔP para a barra *slack* e os valores de ΔQ para a barra *slack* e para todas as barras PV são tornados nulos. O algoritmo implementado é apresentado abaixo em linguagem estruturada.

1. Obter NB, número total de barras do SEP;

    1.1. Para cada barra k do sistema:

        1.1.1. Obter o valor esperado P_k^{esp} no objeto ***epsd***;

        1.1.2. Calcular **PCalc(k)**;

        1.1.3. Fazer $\boldsymbol{Delta\_PQ}(k) = P_k^{\text{esp}} - \text{PCalc(k)}$;

        1.1.4. Obter o valor esperado Q_k^{esp} no objeto ***epsd***;

        1.1.5. Calcular **QCalc(k)**;

        1.1.6. Fazer $\boldsymbol{Delta\_PQ}(k + NB) = Q_k^{\text{esp}} - \text{QCalc(k)}$;

    1.2. Fim para;

    1.3. Para cada barra k do sistema:

        1.3.1. Se barra k é do tipo **Vθ**, então:

            1.3.1.1. $\boldsymbol{Delta\_PQ}(k) = 0$

            1.3.1.2. $\boldsymbol{Delta\_PQ}(k + NB) = 0$

        1.3.2. Fim Se;

        1.3.3. Se barra k é do tipo **PV**, então:

            1.3.3.1. $\boldsymbol{Delta\_PQ}(k + NB) = 0$

        1.3.4. Fim Se;

    1.4. Fim para;

    1.5. Fim.

Vale ressaltar que o vetor ***Delta_PQ*** tem dimensão $m = 2NB$. Os primeiros NB elementos do vetor ***Delta_PQ*** armazenam os valores de ΔP_k para cada barra k do SEP. Os próximos NB elementos armazenam os valores de ΔQ_k para cada barra k do SEP.

- **void Indep_Vector(* double PQ)**

Neste caso o método **Indep_Vector** é sobrecarregado, recebendo agora um vetor ***PQ*** com todos os valores de P e Q para todas as barras. Os valores armazenados neste vetor são utilizados para definir valores de P_k^{esp} e Q_k^{esp}. O algoritmo se assemelha ao apresentado anteriormente, porém P_k^{esp} e Q_k^{esp} são lidos a partir do vetor ***PQ***.

- **double dP_dTeta(int k, int m)**

O método **dP_dTeta** é uma função auxiliar criada para calcular o valor da derivada $\frac{\partial P_k}{\partial \theta_m}$ em que os argumentos ***k*** e ***m*** de entrada da função são os índices k e m indicados. Utiliza-se o conjunto de expressões visto em (3.23).

Durante a execução do algoritmo, se os índices ***k*** e ***m*** obtidos forem iguais, executa-se o cálculo da primeira equação de (3.23), senão, executa-se o cálculo para a segunda equação. O algoritmo implementado é mostrado no fluxograma Figura 3.7.

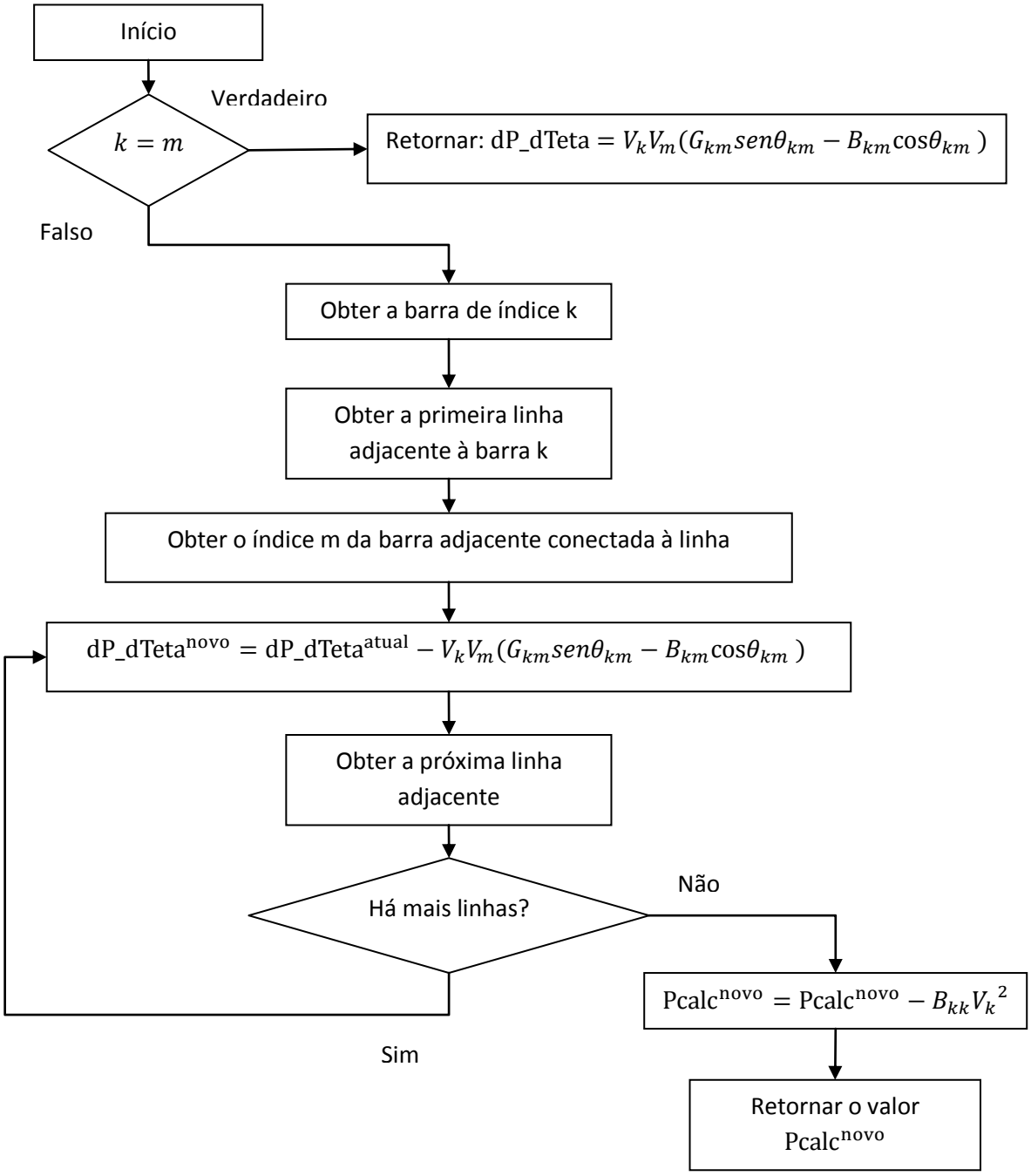

Figura 3.7: Algoritmo implementado para o método dP_dTeta(int k, int m).

- **double dP_dV(int k, int m)**

Da mesma forma que **dP_dTeta**, este método calcula o valor da derivada $\frac{\partial P_k}{\partial V_m}$ através do conjunto de equações (3.24). O algoritmo é semelhante ao algoritmo descrito para **dP_dTeta**.

- **double dQ_dTeta(int k, int m)**

Da mesma forma que os métodos descritos acima, este método calcula o valor da derivada $\frac{\partial Q_k}{\partial \theta_m}$ através do conjunto de equações (3.25). O algoritmo é semelhante ao algoritmo descrito para **dP_dTeta**.

- **double dQ_dV(int k, int m)**

Da mesma forma que os métodos descritos acima, este método calcula o valor da derivada $\frac{\partial Q_k}{\partial V_m}$ através do conjunto de equações (3.26). O algoritmo é semelhante ao algoritmo descrito para **dP_dTeta**.

- **void Mount_Jacobian( )**

O método **Mount_Jacobian** é responsável por montar a matriz Jacobiana utilizando os métodos **dP_dTeta**, **dP_dV**, **dQ_dTeta** e **dQ_dV** supracitados. A função calcula cada elemento da matriz Jacobiana e aloca o elemento na posição adequada de acordo com as considerações feitas na Seção 0. O algoritmo implementado é descrito a seguir.

1. Obter o número total de barras, NB, a partir do objeto EPSD;
2. Para toda barra k da lista de barras do objeto EPSD:

   2.1. Se a barra k é do tipo **Vθ**, então:

      2.1.1. $\boldsymbol{Jacobian}(k,k) = 1$

      2.1.2. $\boldsymbol{Jacobian}(k+NB, k+NB) = 1$

   2.2. Fim Se;

   2.3. Se a barra k é do tipo **PV**, então:

      2.3.1. $\boldsymbol{Jacobian}(k,k) = 1$

   2.4. Fim Se;

3. Fim para;
4. Para toda barra i da lista de barras do objeto EPSD:

   4.1. Se o tipo da barra i é diferente de **Vθ**, então:

4.1.1. Para toda barra j da lista de barras do objeto EPSD:

4.1.2. Se o tipo da barra j é **PV**:

4.1.2.1. Calcular **dP_dTeta**(i, j) e armazenar o valor em $\boldsymbol{Jacobian}(i,j)$;

4.1.2.2. Calcular **dQ_dTeta**(i, j) e armazenar o valor em $\boldsymbol{Jacobian}(i,j+NB)$;

4.1.3. Fim Se;

4.1.4. Se o tipo da barra j é **PQ**:

4.1.4.1. Calcular **dP_dTeta**(i, j) e armazenar o valor em $\boldsymbol{Jacobian}(i,j)$;

4.1.4.2. Calcular **dP_dV**(i, j) e armazenar o valor em $\boldsymbol{Jacobian}(i+NB,j)$;

4.1.4.3. Calcular **dQ_dTeta**(i, j) e armazenar o valor em $\boldsymbol{Jacobian}(i,j+NB)$;

4.1.4.4. Calcular **dQ_dV**(i, j) e armazenar o valor em $\boldsymbol{Jacobian}(i+NB,j+NB)$;

4.1.5. Fim Se;

4.2. Fim Se;

5. Fim para;

6. Fim.

- **bool Verify_Convergence( )**

O método é responsável por verificar o critério de convergência do método de Newton-Raphson dado por: $\left|g\left(z^{(i)}\right)\right| \leq \epsilon$. Vale ressaltar que o vetor $g\left(z^{(i)}\right)$ é representado pelo vetor ***Delta_PQ*** durante a i-ésima iteração e o valor de ϵ é armazenado na variável ***Epsilon***.

- **bool Solve( )**

O método **Solve** tem como objetivo resolver o algoritmo de Newton-Raphson propriamente dito, utilizando todos outros métodos descritos anteriormente. Esta implementação do método tem a função de iniciar o algoritmo utilizando o *flat start*. O algoritmo implementado é mostrado no fluxograma da Figura 3.8.

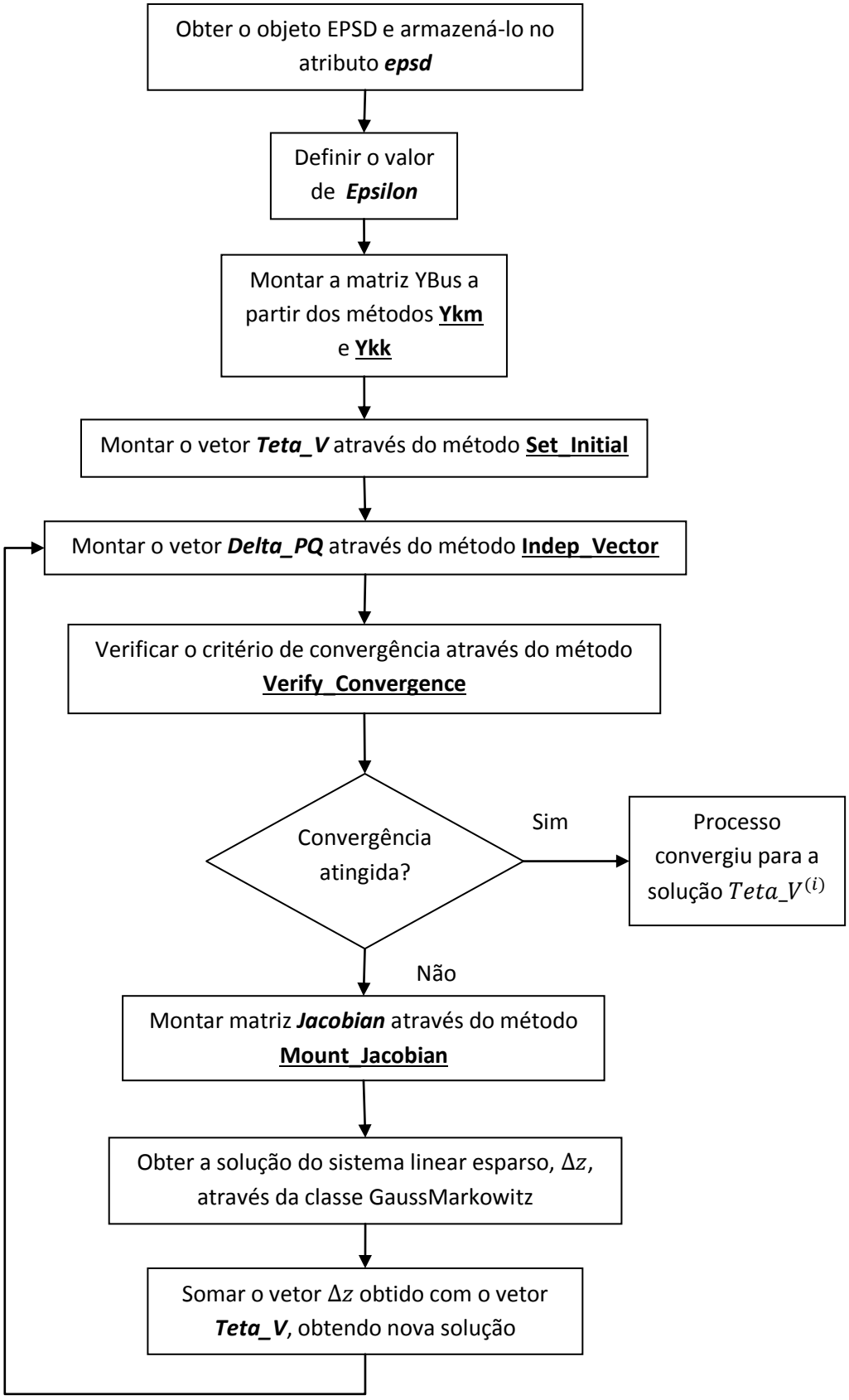

Figura 3.8: Algoritmo implementado para o método Solve( ).

- **bool Solve(double * Teta_V, double * PQ)**

Neste caso o método **Solve** foi sobrecarregado. Esta implementação foi desenvolvida para ser utilizada no algoritmo de otimização do método Gradiente Reduzido que será apresentado no Capítulo 3.

O método **Solve** recebe o vetor com a solução inicial ***TetaV*** e um vetor contendo todos os valores esperados de P para as barras PV e PQ e todos os valores esperados de Q para as barras PQ. O algoritmo é semelhante ao algoritmo mostrado para a primeira implementação do método **Solve**, descrita anteriormente. A Figura 3.9 mostra o algoritmo implementado na forma de fluxograma.

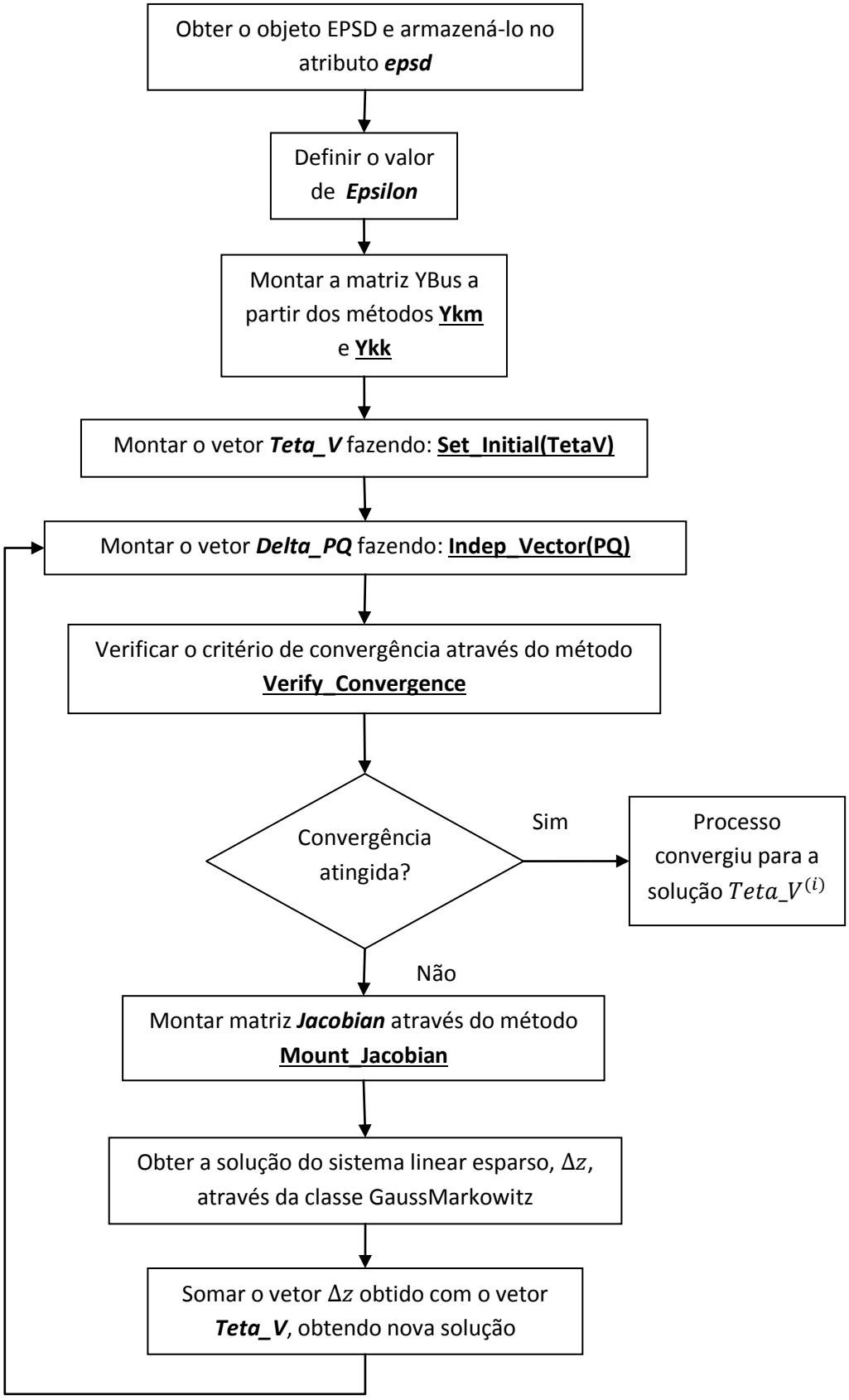

Figura 3.9: Algoritmo implementado para o método Solve(double * TetaV, double * PQ).

- **void Remove_Line(int Line)**

Este método tem a função de retirar de operação a linha do SEP definida pelo argumento de entrada ***Line***. A classe NewtonRaphson acessa seu objeto YBus e passa o valor contido em ***Line*** para o atributo ***SetLineFault*** , indicando que a linha está fora de operação.

- **bool verify_Islanding(void)**

O método é responsável por verificar se há ilhamento do SEP. Obtém-se o valor do atributo ***SetLineFault*** na classe YBus. Verifica-se então os índices k e m das barras conectadas nas extremidades da linha indicada por ***SetLineFault***. Para as barras indicadas, verifica-se se há pelo menos duas linhas conectadas a esta. Se houver apenas uma linha conectada à barra, isso significa que o sistema está ilhado, pois a linha é a própria linha que saiu de operação. Caso contrário, o SEP não está ilhado.

3.7 Fluxo de Potência – Exemplos e Resultados

Nesta seção serão utilizados dois SEPs como exemplo:

- Sistema sul reduzido brasileiro composto de 45 barras e 56 linhas (SANTOS, 2008);
- Sistema sul-sudeste-Mato Grosso reduzido composto de de 107 barras e 171 linhas (ver Anexo A).

Os dados para o sistema sul reduzido estão disponíveis na referência (SANTOS, 2008), enquanto que os dados do sistema sul-sudeste-Mato Grosso reduzido são apresentados no Anexo A.

As informações referentes a cada SEP estão armazenadas em formato XML e são lidas através dos métodos da classe XMLReader. Com isso, a estrutura EPSD é gerada e o método de Newton-Raphson é então executado através da classe NewtonRaphson.

Os resultados para o FP partindo do *flat start* para o sistema sul reduzido são mostrados na Tabela 3.2. A fim de validar os resultados obtidos através do *software* desenvolvido, os resultados partindo do *flat start* para o sistema sul-sudeste-Mato Grosso reduzido são comparados aos resultados obtidos através da ferramenta computacional ANAREDE na Tabela 3.3.

Os resultados são calculados no ANAREDE partindo do *flat start* e considerando que todos os tapes de transformadores estão com valor igual a 1 pu. Nos cálculos via ANAREDE são consideradas as impedâncias dos transformadores. Além disso, todas as potências obtidas através do ANAREDE foram convertidas em valores por unidade

utilizando uma base $S = 100\ MVA$. Os resultados completos obtidos através do ANAREDE são apresentados no Anexo B.

Tabela 3.2: Resultados para a solução do FP para o sistema sul reduzido brasileiro utilizando *flat start*.

Barra	Tipo	Injeção de Potência Ativa - P	Injeção de Potência Reativa - Q	Ângulo de Tensão - θ	Módulo de Tensão - V
		(p.u.)	(p.u.)	(graus)	(p.u.)
1	PQ	0,0000	0,0000	-6,9018	1,0338
2	PQ	0,0000	0,0000	-9,3030	1,0252
3	PV	6,5000	-0,3826	-6,6907	1,0200
4	PQ	-1,7700	-0,6800	-31,7958	0,9738
5	PQ	-1,9100	-0,4200	-33,6263	1,0202
6	PV	2,1500	0,5134	-13,9860	1,0400
7	PQ	-1,7100	-0,1850	-19,3264	1,0217
8	PQ	-1,2600	-0,4700	-16,0758	0,9986
9	PQ	-0,4600	-0,1470	-10,4790	0,9965
10	PV	8,9500	0,8887	3,0904	1,0200
11	PQ	-2,8100	-0,5650	-2,5653	1,0150
12	PQ	-2,7900	-0,6070	-10,8464	1,0163
13	PQ	-1,3000	-0,2940	-20,6077	1,0143
14	PQ	-4,2700	0,2500	-20,7927	1,0257
15	PQ	-3,1000	-1,4100	-27,1224	0,9719
16	PQ	-4,2400	-0,9060	-25,9202	1,0020
17	PQ	-1,1700	-0,5310	-28,2753	0,9926
18	Vθ	15,2939	-1,2993	0,0000	1,0220
19	PQ	0,0000	0,0000	-5,5570	1,0354
20	PQ	0,0000	0,0000	-18,5168	1,0290
21	PQ	-3,6800	-0,5960	-17,0373	1,0330
22	PQ	0,0000	0,0000	-23,9398	1,0105
23	PQ	-1,7400	0,0800	-11,4881	1,0287
24	PQ	0,0000	0,0000	-29,0642	1,0258
25	PQ	0,0000	0,0000	-22,6135	1,0267
26	PQ	0,0000	0,0000	-9,7073	1,0321
27	PV	13,2500	-0,5601	6,9310	1,0180
28	PQ	0,0000	0,0000	-1,3117	1,0350
29	PV	0,9000	0,3902	-24,0511	1,0300
30	PQ	-1,2500	-0,3980	-28,4163	0,9999
31	PV	1,2000	0,4126	-22,5791	1,0300
32	PV	2,4100	0,6932	-21,2269	1,0300
33	PQ	0,0000	0,0000	-27,2392	1,0052
34	PV	11,0000	-1,4058	2,7245	1,0200
35	PQ	0,0000	0,0000	-1,3504	1,0320
36	PQ	-8,1300	-1,1000	-33,1507	1,0286
37	PQ	-6,1200	4,5500	-32,4548	1,0386
38	PV	4,6000	1,0445	-14,3475	1,0200
39	PQ	-4,0400	-1,3500	-20,4479	1,0015
40	PQ	-3,9300	1,1100	-25,2260	1,0328
41	PQ	-2,6200	-0,1320	-15,3574	0,9924
42	PQ	-2,2900	-1,8300	-11,4512	1,0104
43	PQ	-1,8400	-0,6020	-15,1630	0,9815
44	PQ	-1,3900	-0,5370	-13,1199	0,9787
45	PQ	-0,9010	-0,5530	-31,5134	0,9722
Perdas		**1,5329**	**-7,0282**		

Tabela 3.3: Resultados comparativos para a solução do FP para o sistema sul-sudeste-Mato Grosso reduzido utilizando *flat start*.

Dados de Barra		Simulado no Programa Desenvolvido				Resultados ANAREDE			
Barra Nº	Tipo	Injeção de Potência Ativa - P	Injeção de Potência Reativa - Q	Ângulo de Tensão - θ	Módulo de Tensão - V	Injeção de Potência Ativa - P	Injeção de Potência Reativa - Q	Ângulo de Tensão - θ	Módulo de Tensão - V
		(p.u.)	(p.u.)	(graus)	(p.u.)	(p.u.)	(p.u.)	(graus)	(p.u.)
12	PV	3,0000	-1,7126	-24,2283	1,0000	3,0000	-1,7200	-24,2	1,000
16	PV	8,0000	-1,1366	-26,3193	1,0000	8,0000	-1,1310	-26,3	1,000
18	Vθ	10,0216	-3,6891	-24,0011	1,0200	10,0480	-3,6170	-24,0	1,020
20	PV	9,0000	-2,9874	-22,4230	1,0100	9,0000	-2,9750	-22,4	1,010
21	PV	1,4000	-0,3014	-59,1310	1,0000	1,4000	-0,2160	-62,9	1,000
22	PV	1,5000	-0,1724	-19,9291	1,0000	1,5000	-0,1730	-19,9	1,000
35	PV	2,0000	-0,8249	-26,9849	1,0000	2,0000	-0,8060	-27,0	1,000
48	PV	0,0000	-4,0777	-43,0228	1,0000	0,0000	-4,0400	-43,0	1,000
86	PQ	-0,6600	-0,0120	-43,0228	1,0291	-0,6600	-0,0120	-43,0	1,029
100	PQ	0,0000	0,0000	-28,5608	1,0534	0,0000	0,0000	-28,6	1,053
101	PQ	0,0000	0,0000	-36,4496	1,0637	0,0000	0,0000	-36,5	1,063
102	PQ	0,0000	0,0000	-43,2042	1,0530	0,0000	0,0000	-43,2	1,053
103	PQ	0,0000	0,0000	-43,5110	1,0647	0,0000	0,0000	-43,5	1,064
104	PQ	-9,1000	-2,3500	-52,0751	1,0535	-9,1000	-2,3500	-52,1	1,053
106	PQ	0,0000	0,0000	-52,9624	1,0419	0,0000	0,0000	-53,0	1,042
120	PQ	-1,8000	-0,9000	-41,4618	1,0364	-1,8000	-0,9000	-41,5	1,036
122	PQ	-2,0000	-0,3800	-41,9507	1,0586	-2,0000	-0,3800	-42,0	1,058
123	PQ	-4,5000	-1,7500	-46,3219	1,0283	-4,5000	-1,7500	-46,3	1,028
126	PQ	-2,9000	-0,9500	-43,7615	1,0330	-2,9000	-0,9500	-43,8	1,033
131	PQ	0,0000	0,0000	-27,3648	1,0238	0,0000	0,0000	-27,4	1,024
134	PQ	0,0000	0,0000	-26,4705	1,0236	0,0000	0,0000	-26,5	1,024
136	PQ	-0,5400	-0,2300	-33,2056	1,0249	-0,5400	-0,2300	-33,2	1,025
138	PQ	-0,7200	-0,3400	-44,3636	1,0305	-0,7200	-0,3400	-44,4	1,030
140	PQ	-7,0000	-2,5000	-54,0807	1,0151	-7,0000	-2,5000	-54,1	1,015
210	PQ	0,0000	0,0000	-27,5910	1,0462	0,0000	0,0000	-27,6	1,046
213	PQ	-0,9300	-0,3900	-28,8252	1,0442	-0,9300	-0,3900	-28,8	1,044
216	PQ	-0,5300	-0,2500	-27,8852	1,0403	-0,5300	-0,2500	-27,9	1,040
217	PQ	-3,6400	-0,5800	-32,3174	1,0459	-3,6400	-0,5800	-32,3	1,045
218	PQ	-6,0000	-2,0000	-40,1711	1,0197	-6,0000	-2,0000	-40,2	1,019
219	PQ	0,0000	0,0000	-39,0313	1,0231	0,0000	0,0000	-39,1	1,022
220	PQ	0,0000	0,0000	-31,9408	1,0410	0,0000	0,0000	-32,0	1,040
225	PQ	0,0000	0,0000	-34,6461	1,0532	0,0000	0,0000	-34,7	1,003
228	PQ	-0,8600	-0,3400	-40,7449	1,0106	-0,8600	-0,3400	-40,8	1,010
231	PQ	-0,8970	-0,3190	-48,0150	1,0584	-0,8970	-0,3190	-49,6	1,004
233	PQ	0,0000	0,0000	-36,2868	1,0348	0,0000	0,0000	-36,3	1,034
234	PQ	-10,0000	-3,5000	-39,1275	1,0221	-10,0000	-3,5000	-39,2	1,021
300	PV	7,0000	-1,7285	-18,9788	1,0200	7,0000	-1,7120	-19,0	1,020

Dados de Barra		Simulado no Programa Desenvolvido				Resultados ANAREDE			
Barra Nº	Tipo	Injeção de Potência Ativa - P	Injeção de Potência Reativa - Q	Ângulo de Tensão - θ	Módulo de Tensão - V	Injeção de Potência Ativa - P	Injeção de Potência Reativa - Q	Ângulo de Tensão - θ	Módulo de Tensão - V
		(p.u.)	(p.u.)	(graus)	(p.u.)	(p.u.)	(p.u.)	(graus)	(p.u.)
301	PV	3,0000	-1,1998	-19,4396	1,0100	3,0000	-1,2010	-19,5	1,010
302	PV	4,0000	-1,1796	-18,3075	1,0200	4,0000	-1,1750	-18,3	1,020
303	PV	2,0000	-2,7096	-24,3260	1,0200	2,0000	-2,6990	-24,3	1,020
305	PV	3,0000	-1,2519	-22,1871	1,0000	3,0000	-1,2000	-22,2	1,002
320	PQ	0,0000	0,0000	-24,0800	1,0471	0,0000	0,0000	-24,1	1,047
325	PQ	0,0000	0,0000	-23,7341	1,0442	0,0000	0,0000	-23,7	1,044
326	PQ	-2,7400	-1,0400	-25,9811	1,0292	-2,7400	-1,0400	-26,0	1,029
360	PQ	0,0000	0,0000	-22,4747	1,0452	0,0000	0,0000	-22,5	1,045
370	PQ	0,0000	0,0000	-25,4594	1,0483	0,0000	0,0000	-25,5	1,048
396	PQ	0,0000	0,0000	-25,8622	1,0297	0,0000	0,0000	-25,9	1,030
500	PV	8,0000	-1,0951	-21,6148	1,0200	8,0000	-1,0880	-21,6	1,020
535	PQ	0,0000	0,0000	-26,0734	1,0341	0,0000	0,0000	-26,1	1,034
536	PQ	-7,0000	-1,5000	-28,8717	1,0221	-7,0000	-1,5000	-28,9	1,022
800	PV	11,0000	1,0999	-6,7025	1,0200	11,0000	1,1030	-6,7	1,020
808	PV	11,5000	0,9463	4,4277	1,0200	11,5000	1,1520	4,5	1,020
810	PV	12,0000	0,6982	-3,2969	1,0200	12,0000	0,7220	-3,3	1,020
814	PQ	-7,3540	-1,9100	-37,7944	1,0128	-7,3540	-1,9100	-37,8	1,000
824	PQ	0,0000	0,0000	-16,9528	1,0181	0,0000	0,0000	-16,9	1,018
834	PQ	-0,1340	-0,0420	-28,6639	0,9855	-0,1340	-0,0420	-28,7	0,993
839	PQ	0,0000	0,0000	-5,6259	0,9808	0,0000	0,0000	-5,6	0,997
840	PQ	-1,5900	-0,3600	-8,7292	0,9673	-1,5900	-0,3600	-8,6	0,984
848	PQ	-0,9400	-0,1800	-4,7167	0,9782	-0,9400	-0,1800	-4,7	0,984
856	PQ	0,0000	0,0000	-10,2507	1,0203	0,0000	0,0000	-10,2	1,020
895	PQ	0,0000	0,0000	-35,4666	1,0256	0,0000	0,0000	-35,5	1,024
896	PQ	0,0000	0,0000	-3,3949	1,0046	0,0000	0,0000	-3,4	1,000
897	PQ	0,0000	0,0000	-2,0647	1,0171	0,0000	0,0000	-2,0	1,015
898	PQ	0,0000	0,0000	-1,1839	0,9918	0,0000	0,0000	-1,2	0,998
904	PV	7,0000	-1,3936	-14,7458	1,0200	7,0000	-1,4030	-14,7	1,020
915	PV	7,0000	-0,5036	-12,5936	1,0200	7,0000	-0,5210	-12,6	1,020
919	PV	7,0000	0,6098	6,7076	1,0000	7,0000	0,3520	6,6	1,000
925	PV	9,5000	0,3115	0,5623	1,0200	9,5000	0,3530	0,6	1,020
933	PQ	0,0000	0,0000	-17,3348	1,0181	0,0000	0,0000	-17,3	1,018
934	PQ	-2,3700	-0,5900	-17,5360	1,0000	-2,3700	-0,5900	-17,5	1,000
938	PQ	0,0000	0,0000	-37,6132	1,0191	0,0000	0,0000	-37,6	1,022
939	PQ	-11,4900	-0,5306	-40,1451	1,0145	-11,4900	-0,5310	-40,1	1,000
955	PQ	0,0000	0,0000	-23,4636	1,0418	0,0000	0,0000	-23,5	1,042
959	PQ	0,0000	0,0000	-35,1618	1,0139	0,0000	0,0000	-35,2	1,012
960	PQ	-8,4470	-4,6910	-37,7559	0,9863	-8,4470	-4,6910	-37,8	1,000
964	PQ	0,0000	0,0000	-31,0001	1,0210	0,0000	0,0000	-31,0	1,021

Dados de Barra		Simulado no Programa Desenvolvido				Resultados ANAREDE			
Barra Nº	Tipo	Injeção de Potência Ativa - P	Injeção de Potência Reativa - Q	Ângulo de Tensão - θ	Módulo de Tensão - V	Injeção de Potência Ativa - P	Injeção de Potência Reativa - Q	Ângulo de Tensão - θ	Módulo de Tensão - V
		(p.u.)	(p.u.)	(graus)	(p.u.)	(p.u.)	(p.u.)	(graus)	(p.u.)
965	PQ	-7,5560	-0,5624	-33,5478	1,0159	-7,5560	-0,5620	-33,5	1,000
976	PQ	0,0000	0,0000	-33,7667	0,9955	0,0000	0,0000	-33,8	0,996
995	PQ	0,0000	0,0000	-19,1175	1,0388	0,0000	0,0000	-19,1	1,039
1015	PQ	-0,7000	-0,0200	-40,0370	1,0054	-0,7000	-0,0200	-40,0	1,002
1030	PQ	0,0000	0,0000	-20,4280	1,0399	0,0000	0,0000	-20,4	1,040
1047	PQ	0,0000	0,0000	-0,1580	0,9968	0,0000	0,0000	-0,2	1,001
1060	PQ	0,0000	0,0000	-7,3492	1,0251	0,0000	0,0000	-7,3	1,025
1210	PQ	-12,2800	-4,2500	-36,6127	0,9755	-12,2800	-4,2500	-36,6	1,000
1503	PQ	0,0000	0,0000	-49,8539	1,0532	0,0000	0,0000	-49,9	1,053
1504	PQ	-1,4500	-0,6300	-53,8848	1,0184	-1,4500	-0,6300	-53,9	1,018
2458	PQ	-4,0300	-1,2600	-5,8645	0,9823	-4,0300	-1,2600	-5,9	1,000
4501	PQ	-0,3140	-0,0710	-58,4122	1,0642	-0,3140	-0,0710	-61,2	1,022
4521	PQ	0,0000	0,0000	-63,5149	1,0518	0,0000	0,0000	-66,9	1,032
4522	PQ	0,0000	0,0000	-65,5990	1,0566	0,0000	0,0000	-69,0	1,029
4523	PV	0,5000	-0,1792	-57,9209	1,0100	0,5000	-0,0820	-61,2	1,010
4530	PQ	0,0000	0,0000	-69,8608	1,0557	0,0000	0,0000	-73,6	1,020
4532	PQ	0,0000	0,0000	-69,8608	1,0557	0,0000	0,0000	-73,6	1,040
4533	PQ	-0,7540	-0,1610	-70,1852	1,0235	-0,7540	-0,1610	-74,0	1,014
4542	PQ	0,0000	0,0000	-69,0304	1,0354	0,0000	0,0000	-72,8	1,025
4552	PQ	-0,1260	-0,0120	-76,4823	1,0200	-0,1260	-0,0120	-80,4	1,006
4562	PQ	-0,2380	-0,0740	-84,5601	1,0276	-0,2380	-0,0740	-88,7	1,011
4572	PQ	-0,1800	-0,0640	-81,6984	1,0238	-0,1800	-0,0640	-85,8	1,008
4582	PQ	-0,6550	-0,1670	-87,3165	1,0347	-0,6550	-0,1670	-91,6	1,017
4592	PQ	0,0000	0,0000	-64,1547	1,0232	0,0000	0,0000	-67,9	1,018
4596	PV	2,3000	-0,5267	-65,3337	1,0000	2,3000	-0,2720	-69,1	1,000
4623	PQ	-1,2820	-0,4076	-68,4136	1,0416	-1,2820	-0,4080	-72,0	1,015
4703	PQ	-1,8210	-0,2975	-71,2763	1,0119	-1,8210	-0,2980	-75,1	1,002
4804	PV	0,5000	-0,2172	-71,4681	1,0000	0,5000	-0,1650	-75,4	1,000
4805	PQ	0,0000	0,0000	-75,1750	1,0311	0,0000	0,0000	-79,1	1,024
4807	PQ	-1,2890	-0,3630	-76,4638	1,0331	-1,2890	-0,3630	-80,4	1,024
4862	PQ	0,0000	0,0000	-74,7769	1,0566	0,0000	0,0000	-78,7	1,046
Perdas		**3,4046**	**-59,1953**			**3,4310**	**-58,4890**		

Os valores de perdas ativas e reativas apresentados na Tabela 3.2 e na Tabela 3.3 são calculados somando-se as potências ativas e reativas, respectivamente, calculadas para cada barra do SEP. Partindo da premissa de que toda a potência gerada deve ser consumida, a potência restante da diferença entre geração e consumo deve ser igual às perdas totais do sistema.

O método de Newton-Raphson convergiu em 5 iterações para o sistema sul reduzido e em 6 iterações para o sistema sul-sudeste-Mato Grosso para o programa desenvolvido neste trabalho.

Pode-se verificar que os resultados obtidos para a solução no ANAREDE são bastante próximos dos resultados obtidos no programa. Vale ressaltar que a modelagem implementada neste trabalho não considera as posições nem as variações dos tapes dos transformados, bem como não considera as impedâncias dos mesmos.

A **Tabela 3.**4 mostra o cálculo dos erros médios e máximos absolutos para os valores de módulo de tensão, ângulo de tensão, injeção de potência ativa e injeção de potência reativa, para cada barra, obtidos no programa quando comparados à solução obtida na ferramenta ANAREDE. Sçao apresentados também os números das barras onde o erro absoluto máximo ocorre e seus tipos correspondentes.

Tabela 3.4: Erros absolutos calculados para comparação entre os resultados obtidos do programa desenvolvido e através da ferramenta ANAREDE.

Resultado	Erro Absoluto		Barra Erro Máx	Tipo de Barra
	Médio	Máximo		
Módulo de Tensão (p.u.)	0,0055	0,0544	231	PQ
Ângulo de Tensão (graus)	0,7763	4,2835	4582	PQ
Injeção de Potência Ativa (pu)	0,0002	0,0264	18	Vθ
Injeção de Potência Reativa (pu)	0,0121	0,2578	919	PV

Pode-se verificar que o erro médio para o módulo de tensão é da ordem de 0,001 pu, o que representa aproximadamente 0,23 kV, para um sistema de 230 kV. Os erros para potência ativa são da ordem de 0,0001 pu, o que representa 10 kW, para uma carga de 100 MW. Os erros para potência ativa são da ordem de 0,01 pu, o que representa 1 Mvar, para uma carga de 100 Mvar. O máximo erro para o ângulo de tensão é aproximadamente 4 graus.

Vale ressaltar que não foram consideradas as impedâncias dos transformadores durante a modelagem. A ferramenta computacional ANAREDE oferece uma modelagem completa e uma solução diferenciada para a resuloção do fluxo de potência. Desta forma, conclui-se que uma modelagem mais completa pode reduzir os erros obtidos nas simulações do programa.

Os resultados obtidos se mostraram satisfatórios frente à modelagem desenvolvida. Com isso, passamos à modelagem do fluxo de potência ótimo que utiliza a solução do fluxo de potência em seu algoritmo.

Capítulo 4 Fluxo de Potência Ótimo

O fluxo de potência ótimo (FPO) visa determinar a solução ótima para o problema de fluxo de potência considerando uma função objetivo, podendo ser por exemplo de minimização de custos ou perdas em linhas de transmissão, que satisfizer as restrições e limites impostos pelo sistema elétrico. Dentre as restrições existentes podemos ressaltar as restrições de tensão nas barras de geração, bem como os limites de injeção de potência ativa por estas barras. Além disso, é necessário que a solução do FPO também satisfaça a resolução do fluxo de potência (WOOD; WOLLENBERG, 1996).

Um problema de otimização genérico é formado por uma ou mais funções objetivo, restrições de igualdade e restrições de desigualdade. A função objetivo é a função que tem seu valor minimizado ou maximizado na solução ótima do problema. As restrições de igualdade são as restrições representadas a partir de uma equação. Da mesma forma, as restrições de desigualdade são representadas por inequações (WOOD; WOLLENBERG, 1996).

As restrições com relação a limites de tensão e potência são restrições de desigualdade para o problema do FPO. A restrição de igualdade do FPO é dada pela necessidade de solução do fluxo de potência, fazendo com que a equação (3.18) seja satisfeita a cada iteração do problema de otimização.

A modelagem apresentada nas demais seções deste capítulo está relacionada à minimização de custos de geração. Porém, para o escopo deste trabalho, a função objetivo a será utilizada apenas para minimização de perdas totais do sistema elétrico. Desta forma, as perdas serão minimizadas a partir da minimização da potência ativa da barra *slack*. Isto será feito considerando o custo de geração nulo para todas as barras PV do sistema e custo de geração unitário para a barra *slack*.

Com relação às restrições de desigualdade, serão consideradas apenas as restrições de módulo de tensão e potência ativa líquida nas barras PV e de módulo de tensão na barras *slack*.

Existem vários algoritmos para implementação do FPO descritos na literatura (HAPP; WIRGAU, 1981; MOMOH *et al.*, 1999a; MOMOH *et al.*, 1999b; WOOD; WOLLENBERG, 1996), dentre eles: método de Newton e o método do Gradiente Reduzido. Para a finalidade deste trabalho o método do Gradiente Reduzido foi escolhido como algoritmo de otimização, devido à praticidade quanto a implementação.

Nesta seção é apresentado o método do Gradiente Reduzido, bem como seu algoritmo básico. Em seguida, é apresentada a modelagem do algoritmo em termos de programação orientada à objeto, utilizando as bibliotecas descritas no Capítulo 3.

4.1 Gradiente Reduzido - Método e Algoritmo

O Gradiente Reduzido é um método numérico de otimização que pode ser utilizado para solucionar o problema de fluxo de potência ótimo. Sua modelagem é uma das mais simples para este tipo de problema e é baseada nos multiplicadores de Lagrange.

O fluxo de potência ótimo pode então ser descrito por uma função de Lagrange, como segue:

$$L = f(P_i) + \lambda(P_{Load} + P_{Losses} - \sum P_i) \tag{4.1}$$

Sendo P_i a potência ativa gerada pela i-ésima barra geradora (PV ou *slack*), P_{Load} a potência ativa total da cargas do sistema, P_{Losses} a perda total do sistema, f a função objetivo do FPO e λ o vetor dos multiplicadores de Lagrange.

Na equação (4.1), a função objetivo, $f(P_i)$, pode representar o custo total de geração do SEP, sendo composta pela soma dos custos de geração para cada barra geradora (PV ou *slack*). Desta forma, a função objetivo pode ser escrita como na equação (4.2).

$$f(P_i) = \sum F_i(P_i) \tag{4.2}$$

Sendo $F_i(P_i)$ o custo de geração para a i-ésima barra geradora do SEP.

A restrição de igualdade é representada pela subtração da potência total gerada menos a potência total consumida mais perdas. De forma geral, a restrição de igualdade pode ser descrita como mostra a equação (4.2).

$$P_{Load} + P_{Losses} - \sum P_i = 0 \tag{4.3}$$

Pode-se verificar que a restrição imposta pela equação (4.3) também é uma condição necessária para a solução do fluxo de potência. Como será visto nas seções subsequentes esta restrição de igualde é satisfeita a cada iteração do FPO através da resolução do FP (WOOD; WOLLENBERG, 1996).

4.1.1 Definição de Variáveis

Neste equacionamento, o vetor z representará o conjunto das magnitudes e ângulos das tensões de todas as barras do SEP, sendo equivalente ao mesmo vetor definido na equação (3.17), no Capítulo 3. Serão definidos três novos tipos de variáveis: variáveis de estado, variáveis de controle e variáveis fixas.

As variáveis de estado serão representadas durante esta modelagem por um vetor x, as variáveis de controle são representadas como um vetor u e as variáveis fixas são representadas pelo vetor p.

As variáveis de estado são desconhecidas de início e serão conhecidas somente após a solução do fluxo de potência. O vetor x é formado então por todos os ângulos de tensão de nas barras PV e PQ e todos os módulos de tensão nas PQ. A representação do vetor x dada é pela expressão (4.4).

$$x = \begin{bmatrix} \theta_i \; em \; cada \; Barra \; PV \\ \theta_i \\ V_i \; em \; cada \; Barra \; PQ \end{bmatrix} \tag{4.4}$$

As variáveis de controle, ou ajustáveis, são as variáveis do sistema que podem ser alteradas para ajustar as condições operacionais do sistema, de forma a promover o balanço entre geração e carga e controlar os níveis de tensão nas barras de geração. Estas variáveis se mantém constantes a cada solução do fluxo de potência, porém se modificam a cada iteração do processo de otimização de forma a minimizar a função objetivo (WOOD; WOLLENBERG, 1996).

As variáveis ajustáveis são os módulos das tensões nas barras PV e *slack* e as potências nas barras PV. Portanto, o vetor u é representado pela expressão (4.5).

$$u = \begin{bmatrix} V_k \quad na \; Barra \; Slack \\ P_k \\ V_k \; em \; cada \; Barra \; PV \end{bmatrix} \tag{4.5}$$

Conforme descrito anteriormente, as variáveis fixas, ou constantes, permanecerão fixas durante todo o processo de otimização. O vetor p então é formado pelo ângulo de tensão na barra *slack* e pelas potência ativas e reativas nas barras PQ. Estas variáveis não serão utilizadas para minimizar a função objetivo, mas fazem parte do problema do fluxo de potência. O vetor p é representado através da expressão (4.6).

$$p = \begin{bmatrix} \theta_k \quad na \; Barra \; Slack \\ P_k \\ Q_k \; em \; cada \; Barra \; PQ \end{bmatrix} \tag{4.6}$$

As variáveis de controle podem se tornar fixas durante o problema de otimização devido aos limites de tensão ou potência reativa impostos na barra controlada. Se uma das variáveis de controle atinge seu limite, então ela se torna um parâmetro fixo para o algoritmo de otimização, passando a compor o vetor p (WOOD, 1996).

4.1.2 Equacionamento

A partir das variáveis definidas na seção 0 pode-se reescrever a expressão (3.18) considerando as variáveis x, u e p, como mostrado na expressão (4.7).

$$g(z) = g(x,u,p) = 0 \tag{4.7}$$

A função $g(x,u,p)$ é formada por *m* equações que regem o fluxo de potência. Assim, $g(x,u,p)$ é representada pelo conjunto de equações (4.8).

$$g(x,u,p) = \begin{bmatrix} P_k(V,\theta) - {P_k}^{esp}\ para\ cada\ Barra\ PV \\ P_i(V,\theta) - {P_i}^{esp} \\ Q_i(V,\theta) - {Q_i}^{esp}\ para\ cada\ Barra\ PQ \end{bmatrix} \tag{4.8}$$

Pode-se observar que a potência na barra de referência (*slack*) é uma função das demais variáveis do SEP e não pode ser ajustada ou conhecida sem que o fluxo de potência seja resolvido. Para que a função objetivo seja expressa apenas pelas variáveis de controle e suas respectivas funções de custo, podemos separar a mesma em duas partes. Assim, a primeira parte é o somatório dos custos de geração das barras PV, dependente apenas das variáveis de controle u, e a segunda parte é o custo de geração da barra de *slack*, dependente das variáveis de estado do SEP. Deste modo, a função objetivo é representada pela equação (4.9).

$$f = f(x,u) = \sum_{i=NPV} F(P_i) + F_{ref}\left(P_{ref}(V,\theta)\right) \tag{4.9}$$

Desta forma, a função de Lagrange para o método Gradiente Reduzido poderá ser expressa com relação às variáveis anteriormente definidas através da equação (4.10).

$$L(x,u,p) = f(x,u) + \lambda^t g(x,u,p) \tag{4.10}$$

Na expressão (4.10) incluem-se as expressões (4.8) e (4.9). A mesma pode ser também expandida, como pode ser visto na expressão (4.11). Vale ressaltar que o vetor λ tem a mesma dimensão do sistema de equações do fluxo de potência, isto é, possui dimensão $m = 2NPQ + NPV$

$$L(x,u,p) = \sum_{i=NPV} F(P_i) + F_{ref}\left(P_{ref}(V,\theta)\right) + [\lambda_1, \dots, \lambda_m] \begin{bmatrix} P_k(V,\theta) - {P_k}^{esp} \\ P_i(V,\theta) - {P_i}^{esp} \\ Q_i(V,\theta) - {Q_i}^{esp} \end{bmatrix} \tag{4.11}$$

No processo de otimização realizado pelo método gradiente reduzido, para que a função objetivo seja minimizada, considerando as restrições de igualdade, é necessário que o gradiente da Lagrangiana seja nulo no ponto ótimo do problema. Desta forma, aplica-se a equação (4.12).

$$\nabla L(x,u,p) = 0 \tag{4.12}$$

O vetor gradiente, $\nabla L(x,u,p)$, pode ser decomposto com relação às variáveis da função de Lagrange: x, u e λ. Desta forma, obtém-se um sistema linear de equações a ser resolvido, formado por (4.13), (4.14) e (4.15).

$$\nabla L_x = \frac{\partial L}{\partial x} = \frac{\partial f}{\partial x} + \left[\frac{\partial g}{\partial x}\right]^t \lambda \tag{4.13}$$

$$\nabla L_u = \frac{\partial L}{\partial u} = \frac{\partial f}{\partial u} + \left[\frac{\partial g}{\partial u}\right]^t \lambda \tag{4.14}$$

$$\nabla L_\lambda = \frac{\partial L}{\partial \lambda} = g(x,u,p) \tag{4.15}$$

Para que a equação (4.12) seja satisfeita é necessário que cada componente do vetor $\nabla L(x,u,p)$ seja nulo. Pode-se verificar que o componente ∇L_λ é igual ao conjunto de equações do fluxo de potência e deve ser nulo para que o FPO seja resolvido. Desta forma, a expressão (4.7) deve ser satisfeita no ponto ótimo (WOOD; WOLLENBERG, 1996).

O vetor $\frac{\partial f}{\partial x}$, na expressão (4.13), representa a derivada da função objetivo, f, definida em (4.9), com relação às variáveis de estado, x, pertencentes ao equacionamento do método do Gradiente Reduzido. Este vetor pode ser descrito levando em consideração que apenas a potência da barra de referência é dependente do vetor x. Assim, a equação (4.16) mostra a derivada da equação (4.9) com relação à variável x.

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial P_{ref}} \frac{\partial P_{ref}}{\partial x} \tag{4.16}$$

A matriz $\frac{\partial g}{\partial x}$, na expressão (4.13), é a derivada do sistema de equações do fluxo de potência, $g(z)$, com relação à variáveis de estado, x. Ela representa a própria matriz jacobiana obtida a partir da solução do fluxo de potência através do método de Newton-Raphson (WOOD; WOLLENBERG, 1996).

O vetor $\frac{\partial f}{\partial u}$, na expressão (4.14), é representado como a derivada da função objetivo, f, com relação às variáveis de controle, u. Considerando que as variáveis de controle podem ser representadas pelas potências nas barras PV e pelos módulos das tensões na barras *slack* e PV do SEP, podemos dividir o vetor $\frac{\partial f}{\partial u}$ em duas partes como mostra a expressão (4.17).

$$\frac{\partial f}{\partial u} = \begin{Bmatrix} se\ u = P_k \rightarrow \dfrac{\partial f}{\partial u} = \dfrac{\partial F_k(P_k)}{\partial P_k} \\ se\ u = V_k \rightarrow \dfrac{\partial f}{\partial u} = \dfrac{\partial F_k(P_k)}{\partial P_k} \dfrac{\partial P_k}{\partial V_k} \end{Bmatrix} \tag{4.17}$$

Como a função objetivo é modelada como função das potências ativas geradas pelas barras geradoras, o resultado de (4.17), quando $u = P_k$, é a derivada direta do

polinômio que representa a função f. Quando $u = V_k$ é necessário que se utilize da Regra da Cadeia para que se obtenha a função apropriada.

A matriz $\frac{\partial g}{\partial u}$, na expressão (4.14), é a derivada das equações do fluxo de carga, $g(z)$, com relação às variáveis de controle, u. Da mesma forma, como foi feito com o vetor $\frac{\partial f}{\partial u}$ em (4.17), podemos dividir a matriz $\frac{\partial g}{\partial u}$ em duas partes, como mostra a expressão (4.18).

$$\frac{\partial g}{\partial u} = \begin{Bmatrix} se\ u = P_k \rightarrow \ \dfrac{\partial g}{\partial u} = -1 \\ se\ u = V_k \rightarrow \dfrac{\partial g}{\partial u} = \dfrac{\partial P(V,\theta)}{\partial V}\ ou\ \dfrac{\partial Q(V,\theta)}{\partial V} \end{Bmatrix} \tag{4.18}$$

A cada iteração do método do Gradiente Reduzido as variáveis de controle, u, são consideradas constantes durante a execução do método de Newton-Raphson. Desta forma, se $u = P_k$, então este valor é o valor esperado de P para a barra PV correspondente. Neste caso, a derivada terá sempre valor $\frac{\partial g}{\partial u} = -1$. Se $u = V_k$, então é necessário calcular-se a derivada propriamente dita (WOOD; WOLLENBERG, 1996).

4.1.3 Algoritmo

Sendo $\left|\frac{\partial L}{\partial u}\right| < \epsilon$ o critério de convergência, ϵ um número real próximo de zero e c o passo de convergência para o método, e a partir das equações apresentadas na Seção 0, pode-se definir o seguinte algoritmo para o método do Gradiente Reduzido:

1. Resolver o fluxo de potência, fazendo $g(x, u, p) = 0$;
2. Encontrar o vetor λ resolvendo o sistema linear: $\lambda \left[\frac{\partial g}{\partial x}\right]^t = -\frac{\partial f}{\partial x}$;
3. Determinar: $\frac{\partial L}{\partial u} = \frac{\partial f}{\partial u} + \left[\frac{\partial g}{\partial u}\right]^t \lambda$;
4. Verificar a critério de convergência, $\left|\frac{\partial L}{\partial u}\right| < \epsilon$: se a condição for satisfeita o processo de otmização convergiu para $z^{(i)}$, senão o processo continua;
5. Modificar as variáveis de controle: $u_{NEW} = u_{OLD} - c\frac{\partial L}{\partial u}$;
6. Voltar ao passo 1.

O critério de convergência é definido segundo as condições de otimalidade de Karush-Kuhn-Tucker (KKT) visto na referência (BAZARAA, 1993). As condições de KKT implicam que é necessário que a equação (4.19) seja satisfeita no ponto ótimo.

$$\frac{\partial L}{\partial u} = 0 \tag{4.19}$$

Assim, o critério de convergência satisfaz as condições de KKT definindo-se ϵ suficientemente próximo a zero.

Vale ressaltar que o passo de convergência deve ser suficientemente pequeno para que o método possa convergir quando atingir a bacia de convergência. Um passo de convergência relativamente grande pode causar a não convergência do método.

4.2 Implementação Computacional – Método do Gradiente Reduzido

Nesta seção são apresentadas as classes desenvolvidas para a implementação do método do Gradiente Reduzido, bem como seus atributos e métodos. Também será apresentado o algoritmo computacional utilizado para o método do Gradiente Reduzido juntamente com as considerações e premissas adotadas.

4.2.1 Algoritmo Computacional

Levando em consideração o algoritmo descrito na Seção 0, foi desenvolvido um algoritmo computacional com base nas classes desenvolvidas no Capítulo 4. Algumas considerações serão adotadas:

- O sistema de equações que representam o fluxo de potência continuará a ter dimensão $m = 2NB$, em que NB é o número total de barras do SEP;
- A solução do sistema linear presente no passo 2 será feita através da classe GaussMarkowitz;
- As perdas serão otimizadas minimizando-se a potência ativa na barra slack, fazendo-se nulos os coeficientes de custo para as barras PV e tornando unitário o custo de geração da barra *slack*;
- Os dados referentes à limites de tensão e potência, bem como quais variáveis serão mantidas como variáveis de controle, serão armazenados em uma estrutura diferente da EPSD.

As considerações quanto à dimensão do sistema de equações do fluxo de potência e quanto à solução do sistema linear através da classe GaussMarkowitz são mantidas da mesma forma que no Capítulo 3.

Como a estrutura EPSD utilizada não armazena os dados necessários para a execução do algoritmo do gradiente é necessária a criação de uma nova estrutura à parte desta. Esta nova estrutura será representada pela classe GradientData. Mais detalhes sobre esta classe serão exibidos na seção 0.

O algoritmo computacional para o método Gradiente Reduzido foi desenvolvido baseando no algoritmo apresentado na Seção 0, nas considerações supracitadas e nas equações definidas na Seção 0. A Figura 4.1 representa o algoritmo implementado em forma de fluxograma.

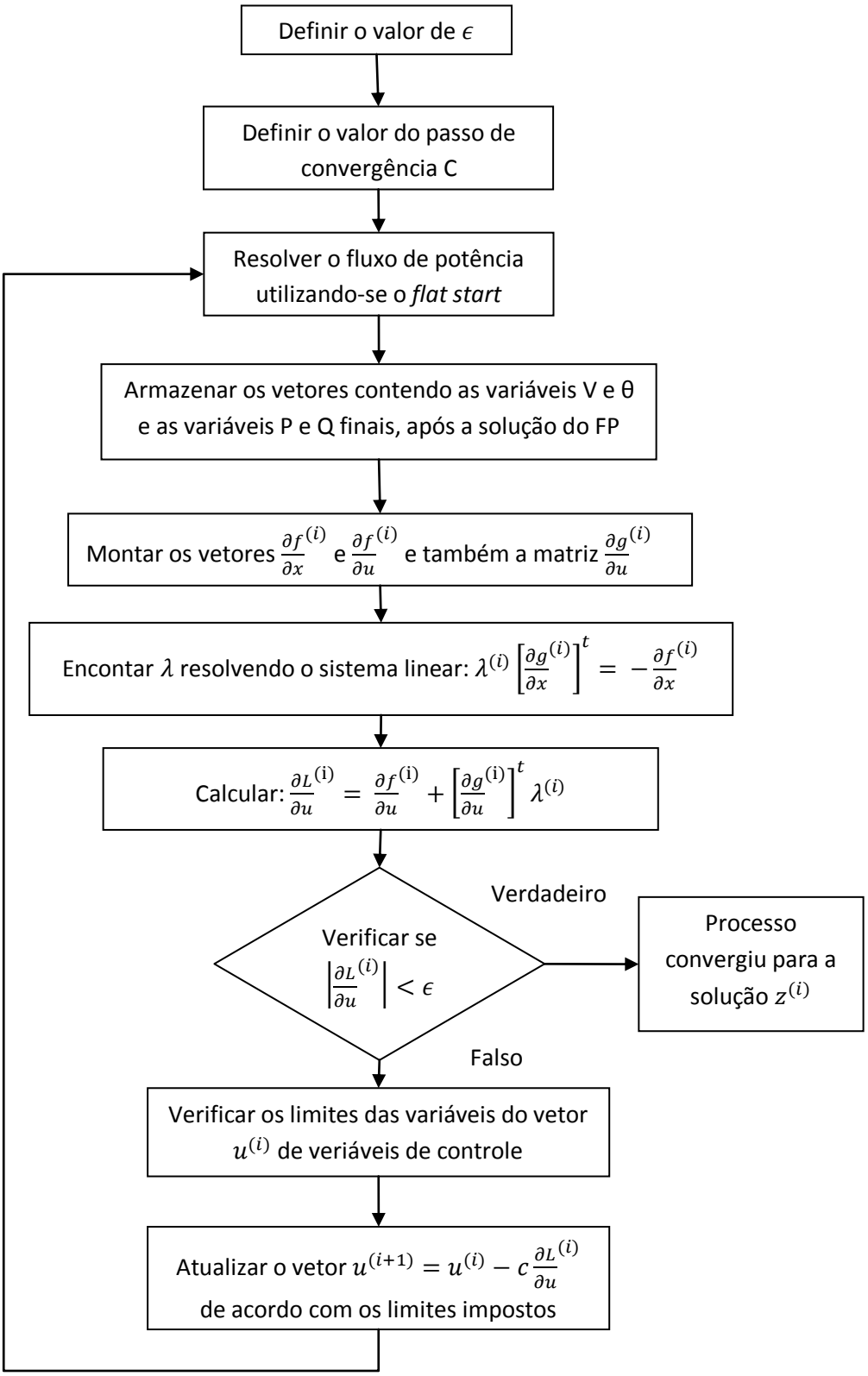

Figura 4.1: Algoritmo computacional implementado para o método do Gradiente Reduzido.

4.3 Classes – Método do Gradiente Reduzido

Foram criadas duas classes para o problema do Gradiente. A primeira classe apresentada nesta seção é a GradientData e, em seguida, a classe Gradient.

4.3.1 Classe GradientData

A classe GradientData tem por objetivo o armazenamento dinâmico de dados inerentes ao método do Gradiente Reduzido separadamente da classe EPSD. Para cada barra de geração cria-se um objeto do tipo GradientData, o qual armazenará as variáveis que serão mantidas fixas e as controladas, além dos coeficientes para a função de custo de geração da barra, os limites para tensão e potência na barra e seu índice no objeto EPSD.

O Diagrama de Classe para a classe GradientData é mostrado na Figura 4.2.

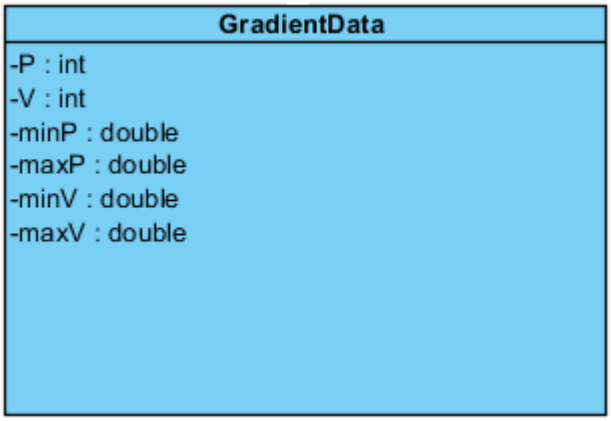

Figura 4.2: Diagrama de Classe da classe GradientData.

4.3.1.1 Atributos

Os atributos da classe GradientData são os principais componentes da classe, visto que esta tem a função de armazenar dados. O atributo ***Bus*** é um número inteiro que armazena o índice da barra k lida no objeto EPSD.

Os inteiros ***P*** e ***V*** têm função de *flag* e armazenam valor 0 ou 1, indicando se as variáveis P ou V de uma barra k estão sendo utilizados como variável de controle. Valor 0 indica que a variável é um parâmetro fixo e valor 1 indica que a variável é um parâmetro de controle para o FPO. Estes valores são armazenados para todas as barras PV e também para a barra *slack.*

Os limites mínimos e máximos de módulo de tensão e potência ativa são indicados, respectivamente, por: ***minV***, ***maxV***, ***minP***, ***maxP***.

Em resumo, os atributos da classe podem ser descritos como mostrado abaixo:

- ***int P***: inteiro responsável por indicar se a variável ***P*** será utilizada como variável de controle ou como variável fixa;
- ***int V***: inteiro responsável por indicar se a variável ***V*** será utilizada como variável de controle ou como variável fixa;
- ***int Bus***: inteiro que armazena o índice da barra no objeto EPSD;
- ***double Coef [4]***: vetor tipo *double* responsável por armazenar os coeficientes da função de custo de geração para a barra indicada por ***Bus***;
- ***int minV***: limite inferior para o módulo de tensão na barra indicada por ***Bus***;
- ***int maxV***: limite superior para o módulo de tensão na barra indicada por ***Bus***;
- ***int minP***: limite inferior para a potência ativa na barra indicada por ***Bus***;
- ***int maxP***: limite superior para a potência ativa na barra indicada por ***Bus***.

4.3.1.2 Métodos

Os métodos desta classe tem a função de receber ou retornar os valores armazenados nas variáveis tidas como atributos da classe e não serão apresentados.

4.3.2 Classe Gradient

A classe Gradient é responsável pela execução do método do Gradiente Reduzido propriamente dito. Ela possui os vetores e matrizes apresentados na Seção 0, bem como as funções de montagem e cálculo necessárias para a execução do algoritmo apresentado na Seção 0. O Diagrama de Classe para a classe Gradient é mostrado na Figura 4.3.

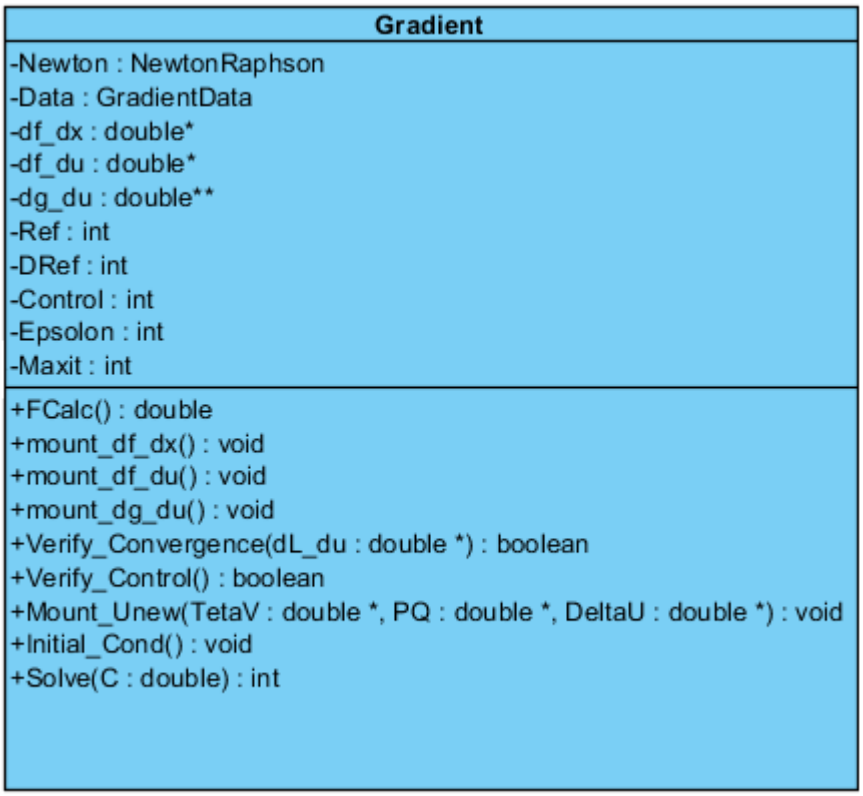

Figura 4.3: Diagrama de Classe da classe Gradient.

4.3.2.1 Atributos

Os atributos da classe Gradient tem a função de armazenar os vetores e matrizes inerentes ao método do Gradiente Reduzido, bem como outras variáveis de controle necessárias para a execução do algoritmo. A classe também armazena um ponteiro da classe NewtonRaphson, ***Newton***, e um ponteiro da classe GradientData, ***Data***.

O ponteiro ***Newton*** é responsável por armazenar o objeto EPSD referente ao SEP e também por resolver o FP a cada iteração do método do Gradiente Reduzido. Este atributo é mantido como ponteiro devido à necessidade de alocar-se dinamicamente o objeto da classe. A alocação é feita durante a execução do construtor da classe Gradient.

Além disso, cria-se um vetor de objetos da classe GradientData, para todas as barras geradoras, incluindo a barra *slack*, durante a execução do construtor da classe Gradient. Esse vetor é então armazenado na classe Gradient em seu respectivo ponteiro, ***Data***. A alocação de memória é feita dinamicamente.

Ainda durante a execução do construtor da classe Gradient, o índice da barra *slack* no objeto EPSD é armazenado na variável inteira ***ref***, o índice da barra *slack* no vetor ***Data*** é armazenado na variável inteira ***Dref*** e o número total de variáveis de controle é armazenado na variável inteira ***control***, em que ***control*** $= 2NPV + 1$.

Os vetores $\frac{\partial f}{\partial x}$ e $\frac{\partial f}{\partial u}$ bem como a matriz $\frac{\partial g}{\partial u}$, representados, respectivamente, pelos atributos ***df_dx***, ***df_du*** e ***dg_du*** também são alocados dinâmicamente durante a execução do construtor da classe Gradient.

Em resumo, os atributos da classe podem ser descritos como mostrado abaixo:

- ***NewtonRaphson * Newton***: ponteiro da classe NewtonRaphson responsável por armazenar o objeto de mesmo tipo alocado dinamicamente durante a execução do construtor da classe Gradient.
- ***GradientData * Data***: ponteiro da classe GradientData responsável por armazenar o vetor de objetos de mesmo tipo alocado dinamicamente durante a execução do construtor da classe Gradient.
- ***double *df_dx***: ponteiro do tipo *double* responsável por armazenar o vetor $\frac{\partial f}{\partial x}$ alocado dinamicamente durante a execução do construtor da classe Gradient.
- ***double *df_du***: ponteiro do tipo *double* responsável por armazenar o vetor $\frac{\partial f}{\partial u}$ alocado dinamicamente durante a execução do construtor da classe Gradient.
- ***double **dg_du***: ponteiro do tipo *double** responsável por armazenar a matriz $\frac{\partial g}{\partial u}$ alocada dinamicamente durante a execução do construtor da classe Gradient.
- ***int ref***: inteiro que indica o índice da barra *slack* no objeto EPSD armazenado no objeto ***Newton***.
- ***int Dref***: inteiro que indica o índice da barra *slack* no vetor ***Data***.
- ***int control***: inteiro que armazena o número total máximo de variáveis de controle.

4.3.2.2 Métodos

Os métodos pertencentes à classe Gradient são descritos abaixo, juntamente com seus algoritmos implementados.

- **double FCalc(void)**

O método calcula o valor total do custo de geração para o SEP. A partir do vetor ***Data*** tem-se acesso a todas as informações das barras de geração do SEP. Para cada elemento do vetor ***Data*** recebe-se o índice ***Bus*** da barra e calcula-se o valor de P_{Bus} gerado pela barra após a resolução do FP. A partir do valor de P_{Bus} pode-se calcular o valor do custo de geração para a barra correspondente através dos coeficientes armazenados no elemento do vetor ***Data*** utilizando a equação (4.21). Este valor é acumulado para cada

barra, resultando no custo final de geração para o SEP. O algoritmo é mostrado abaixo em linguagem estruturada.

1. Criar acumulador $Fcalc$;
2. Fazer $Fcalc = 0$;
3. Para cada elemento i do vetor ***Data***:
   3.1. Receber o índice armazenado no atributo ***Bus***;
   3.2. Calcular P_{Bus} utilizando o método **PCalc** da classe NewtonRaphson;
   3.3. Calcular o valor de $F(P_{Bus})$ utilizando os coeficientes do i-ésimo elemento de Data segundo a equação (4.18);
   3.4. Fazer $Fcalc = Fcalc + F(P_{Bus})$;
4. Fim Para;
5. Retornar $Fcalc$.

- **void mount_df_dx (void)**

Este método tem a função de montar o vetor $\frac{\partial f}{\partial x}$ utilizando a expressão (4.17). De acordo com a expressão é necessário que a derivada $\frac{\partial f(P_{ref})}{\partial P_{ref}}$ seja calculada. Isto é feito a partir dos coeficientes armazenados para a barra *slack* no vetor ***Data***. O cálculo da derivada $\frac{\partial P_{ref}}{\partial x}$ é feito através dos métodos **dP_dV** e **dP_dTeta** da classe NewtonRaphson.

O algoritmo implementado é mostrado no fluxograma da Figura 4.4.

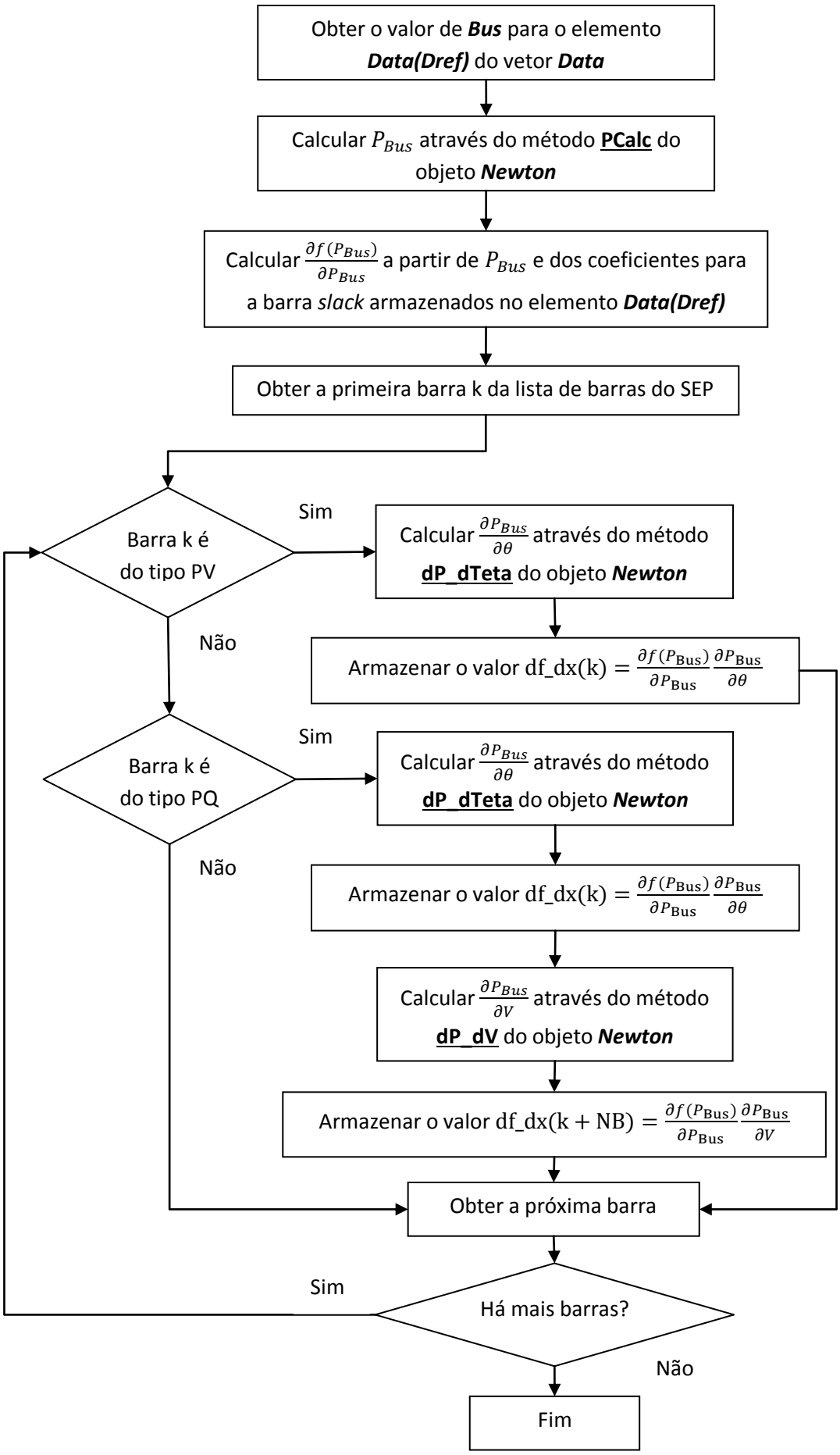

Figura 4.4: Algoritmo implementado para o método df_dx(void).

- **void mount_df_du (void)**

Este método tem a função de montar o vetor $\frac{\partial f}{\partial u}$ utilizando a expressão (4.18). O primeiro elemento do vetor é sempre o módulo da tensão na barra *slack*, seguido dos demais módulos de tensão e os valores de potência ativa para todas as barras PV do SEP. Se a variável armazenada no vetor não é utilizada como variável de controle, seu valor é anulado.

O algoritmo apresentado para este método é similar ao algoritmo apresentado para o método **mount_df_dx**. O algoritmo é mostrado abaixo em linguagem estruturada.

1. Obter o elemento ***Data(Dref)*** do vetor ***Data***;
2. Obter a variável ***Bus*** para ***Data(Dref)***;
3. Calcular P_{Bus} através do método **PCalc** do objeto ***Newton***;
4. Calcular $\frac{\partial F_{Bus}(P_{Bus})}{\partial P_{Bus}}$ a partir de P_{Bus} e dos coeficientes para a barra *slack* armazenados no elemento ***Data(Dref)*** do vetor ***Data***;
5. Calcular $\frac{\partial P_{Bus}}{\partial V}$ através do método **dP_dV** do objeto ***Newton***;
6. Calcular e armazenar o valor $\mathrm{df\_du}(\boldsymbol{Dref}) = \frac{\partial F_{Bus}(P_{Bus})}{\partial P_{Bus}} \frac{\partial P_{Bus}}{\partial V_{Bus}} \cdot \boldsymbol{V}$;
7. Para cada elemento i do vetor ***Data*** diferente de ***Dref***, fazer:
   7.1. Obter a variável ***Bus*** para ***Data(i)***;
   7.2. Calcular P_{Bus} através do método **PCalc** do objeto ***Newton***;
   7.3. Calcular $\frac{\partial F_{Bus}(P_{Bus})}{\partial P_{Bus}}$ a partir de P_{Bus} e dos coeficientes para a barra ***Bus*** armazenados no elemento ***Data(i)*** do vetor ***Data***;
   7.4. Calcular $\frac{\partial P_{Bus}}{\partial V}$ através do método **dP_dV** do objeto ***Newton***;
   7.5. Calcular e armazenar o valor $\mathrm{df\_du}(i) = \frac{\partial F_{Bus}(P_{Bus})}{\partial P_{Bus}} \cdot \boldsymbol{P}$
   7.6. Calcular e armazenar o valor $\mathrm{df\_du}(control - 1 + i) = \frac{\partial F_{Bus}(P_{Bus})}{\partial P_{Bus}} \frac{\partial P_{Bus}}{\partial V_{Bus}} \cdot \boldsymbol{V}$
8. Fim Para;
9. Fim.

- **void mount_dg_du(void)**

Da mesma forma que os métodos de montagem descritos anteriormente, o método **mount_dg_du** é responsável por montar a matriz $\frac{\partial g}{\partial u}$ segundo a expressão (4.19). Vale ressaltar que as dimensões da matriz são MxN, em que: $M = 2NB$ e $N = 2(control + 1) - 1$. O algoritmo implementado é apresentado a seguir em linguagem estruturada.

1. Obter o número total de barras do SEP, NB, através do objeto EPSD;
2. Para cada barra k do SEP:
   2.1. Se o tipo da barra k é diferente de **Vθ**, então:
      2.1.1. Para elemento i do vetor ***Data***:
         2.1.1.1. Obter a variável ***Bus*** para o elemento ***Data(i)***;
         2.1.1.2. Obter o indicador ***P*** para o elemento ***Data(i)***;
         2.1.1.3. Obter o indicador ***V*** para o elemento ***Data(i)***;
         2.1.1.4. Se $\boldsymbol{Bus} = k$, então: $dg\_du(k, i) = \boldsymbol{P}$;
         2.1.1.5. Senão, fazer: $dg\_du(k, i) = dP\_dV(k, Bus) \cdot \boldsymbol{V}$;
      2.1.2. Fim Para;
   2.2. Fim Se;
   2.3. Se o tipo da barra k é igual a **PQ**, então:
      2.3.1. Para elemento i do vetor ***Data***:
         2.3.1.1. Obter a variável ***Bus*** para o elemento ***Data(i)***;
         2.3.1.2. Obter o indicador ***P*** para o elemento ***Data(i)***;
         2.3.1.3. Obter o indicador ***V*** para o elemento ***Data(i)***;
         2.3.1.4. Fazer $dg\_du(k, i + NB) = dQ\_dV(k, Bus) \cdot \boldsymbol{V}$;
      2.3.2. Fim Para;
   2.4. Fim Se;
3. Fim Para;
4. Fim.

- **int verify_Convergence(double* dL_du)**

Este método faz a verificação do critério de convergência do método do Gradiente Reduzido, dado por $\left|\frac{\partial L}{\partial u}\right| < \gamma$. Como o vetor $\frac{\partial L}{\partial u}$ é calculado para cada iteração e não é

armazenado como atributo, então este deve ser passado por referência para este método. Vale ressaltar que γ é um valor muito próximo de zero. Quando o critério de convergência é atingido o método retorna 1, senão retorna 0.

- **int verify_ Control( )**

Este método é responsável por verificar se ainda existem variáveis de controle disponíveis para a continuidade do algoritmo. Verifica-se, para cada elemento do vetor ***Data***, se os indicadores ***P*** e ***V*** são nulos. Caso todos sejam nulos, então o método retorna 1, senão, retorna 0.

- **void Mount_Unew(double* TetaV, double* PQ, double* Delta_U)**

O método é responsável por atualizar o vetor de variáveis de controle utilizando o valor calculado, $\Delta u = -c\frac{\partial L}{\partial u}$. O vetor Δu é passado por referência através da variável ***Delta_U***. Os valores correspondentes às variáveis de controle nos vetores ***TetaV*** e ***PQ*** são então atualizados segundo as restrições armazenadas no vetor ***Data***.

O algoritmo utilizado é mostrado abaixo em linguagem estruturada.

1. Obter o número total de barras, NB, através do objeto EPSD;
2. Definir as variáveis de teste V e P do tipo *double*;
3. Para cada elemento i do vetor ***Data***;
   3.1. Obter a variável ***Bus*** para o elemento ***Data(i)***;
   3.2. Fazer $V = \boldsymbol{TetaV}(\boldsymbol{Bus} + NB) + \boldsymbol{Delta\_U}(i)$
   3.3. Fazer $P = \boldsymbol{PQ}(\boldsymbol{Bus}) + \boldsymbol{Delta\_U}(i + \boldsymbol{control} - 1)$
   3.4. Se V está dentro dos limites armazenados no elemento ***Data(i)***, então:
      3.4.1. $\boldsymbol{TetaV}(\boldsymbol{Bus} + NB) = V$
   3.5. Senão, a variável de controle não é atualizada e passa a fazer parte do vetor p;
   3.6. Se P está dentro dos limites armazenados no elemento ***Data(i)***, então:
      3.6.1. $\boldsymbol{PQ}(\boldsymbol{Bus}) = P$
   3.7. Senão, a variável de controle não é atualizada e passa a fazer parte do vetor p;
4. Fim Para;
5. Fim.

- **void Initial_Cond( )**

Neste método são especificadas as condições iniciais para o controle do fluxo de potência ótimo. Para cada elemento do vetor ***Data*** são definidos os valores de seus atributos: número da barra no objeto EPSD, ***Bus***, coeficientes para custo de geração, ***Coef***, e limites de tensão e potência ativa, ***minV***, ***maxV***, ***minP*** e ***maxP***.

- **int solve(double C)**

O método **solve** tem a função de executar o algoritmo do método do Gradiente Reduzido propriamente dito utilizando os métodos já implementados. A implementação deste algoritmo exige que dois vetores sejam criados para armazenar os valores de P e Q e de V e θ a cada iteração do FPO. Estes vetores são atualizados ao final de cada iteração e então passados ao método **Solve** da classe NewtonRaphson. O algoritmo implementado é apresentado no fluxograma da Figura 4.5.

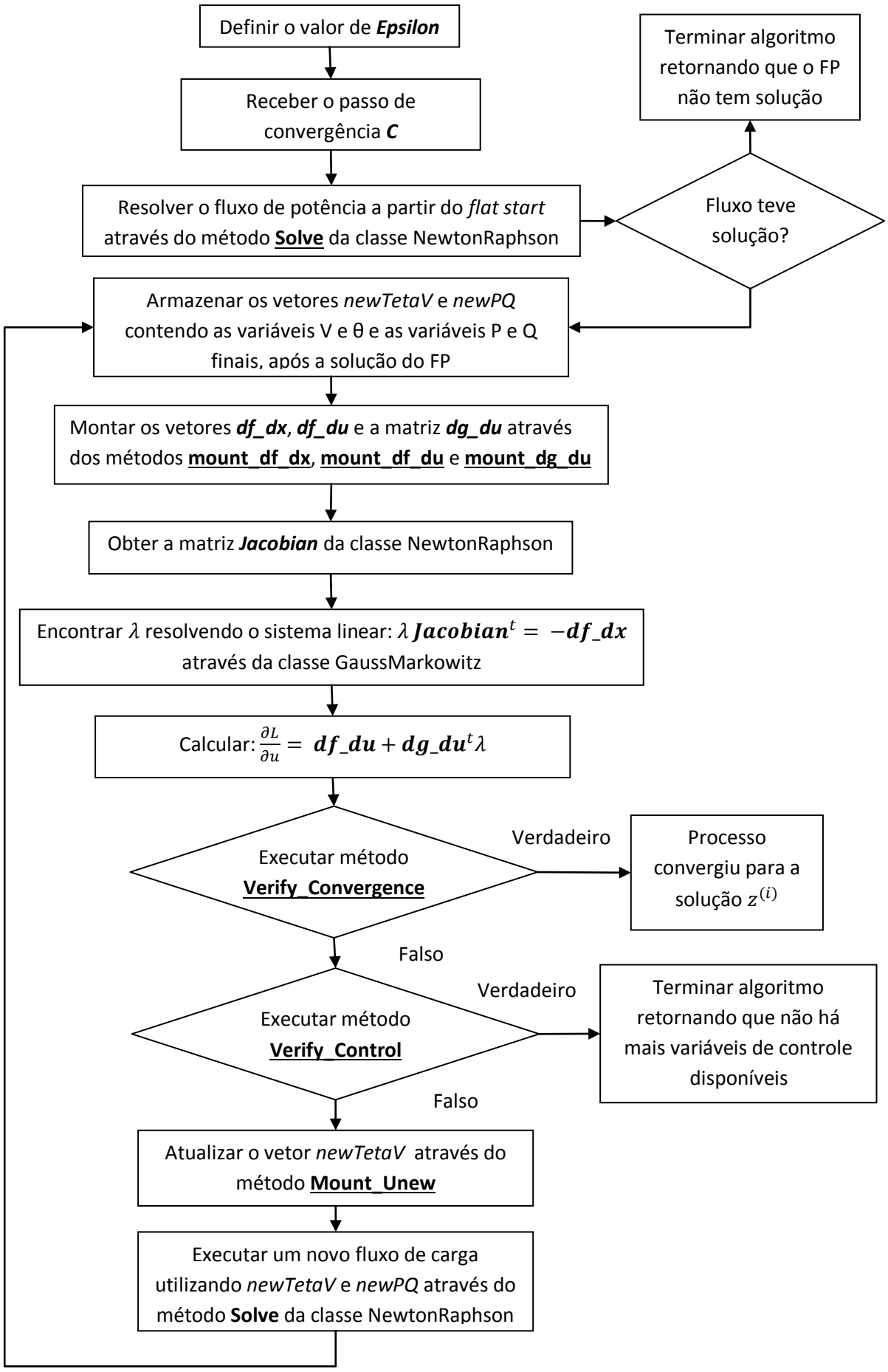

Figura 4.5: Algoritmo implementado para o método Solve(double C).

4.4 Método do Gradiente – Simulação e Resultados

Nesta seção serão apresentados os resultados da aplicação do método do Gradiente Reduzido aos sistemas sul reduzido e sul-sudeste-Mato Grosso descritos no Capítulo 3.

Para cada SEP foi realizada a otimização com restrição nas variáveis de controle considerando-se o *flat start* como solução inicial. Para ambos os casos foi minimizada a geração de potência ativa na barra *slack* do SEP, fazendo assim $f(x,u,p) = P_{slack}$. Apenas as tensões nas barras de geração foram mantidas como variáveis de controle. O controle das potência ativas geradas não foi considerado nos casos analisados. As variáveis de controle foram mantidas dentro dos limites especificados na Tabela 4.1.

Tabela 4.1: Limites e controle de variáveis para a execução do método do Gradiente.

Limites	Injeção de Potência Ativa - P	Tensão nas Barras Geradoras - V
	(p.u.)	(p.u.)
Inferior	-	0,95
Superior	-	1,05
Controlável?	Não	Sim

Para cada simulação foram obtidos os resultados do fluxo de potência para o ponto ótimo de operação resultante do método do Gradiente. Os resultados obtidos para o *flat start* dos SEPs analisados foram apresentados nas tabelas Tabela 3.2 e Tabela 3.3 da seção 0.

As tabelas Tabela 4.2 e Tabela 4.3 mostram os resultados referentes à otimização restrita nas variáveis de controle, respectivamente, para os sistemas sul reduzido e sul-sudeste-Mato Grosso reduzido.

Tabela 4.2: Resultados para a otimização restrita nas variáveis de controle para o sistema sul reduzido partindo do *flat start*.

Barra	Tipo	Injeção de Potência Ativa - P	Injeção de Potência Reativa - Q	Ângulo de Tensão - θ	Módulo de Tensão - V
		(p.u.)	(p.u.)	(graus)	(p.u.)
1	PQ	0,0000	0,0000	-6,4607	1,0714
2	PQ	0,0000	0,0000	-8,6895	1,0646
3	PV	6,5000	-0,8670	-6,2190	1,0483
4	PQ	-1,7700	-0,6800	-29,4312	1,0039
5	PQ	-1,9100	-0,4200	-31,2740	1,0582
6	PV	2,1500	0,2067	-12,6213	1,0494
7	PQ	-1,7100	-0,1850	-17,7981	1,0446
8	PQ	-1,2600	-0,4700	-14,7763	1,0256

Barra	Tipo	Injeção de Potência Ativa - P	Injeção de Potência Reativa - Q	Ângulo de Tensão - θ	Módulo de Tensão - V
		(p.u.)	(p.u.)	(graus)	(p.u.)
9	PQ	-0,4600	-0,1470	-9,5490	1,0258
10	PV	8,9500	0,6537	3,2137	1,0471
11	PQ	-2,8100	-0,5650	-2,1392	1,0445
12	PQ	-2,7900	-0,6070	-10,1222	1,0520
13	PQ	-1,3000	-0,2940	-19,2020	1,0571
14	PQ	-4,2700	0,2500	-19,3718	1,0687
15	PQ	-3,1000	-1,4100	-25,1793	1,0179
16	PQ	-4,2400	-0,9060	-24,0765	1,0457
17	PQ	-1,1700	-0,5310	-26,1609	1,0280
18	Vθ	15,1915	-2,5303	0,0000	1,0484
19	PQ	0,0000	0,0000	-5,2109	1,0690
20	PQ	0,0000	0,0000	-17,2737	1,0716
21	PQ	-3,6800	-0,5960	-15,9076	1,0749
22	PQ	0,0000	0,0000	-22,2594	1,0544
23	PQ	-1,7400	0,0800	-10,7366	1,0627
24	PQ	0,0000	0,0000	-27,0591	1,0656
25	PQ	0,0000	0,0000	-21,0474	1,0675
26	PQ	0,0000	0,0000	-9,0768	1,0713
27	PV	13,2500	-0,9245	6,5108	1,0494
28	PQ	0,0000	0,0000	-1,2260	1,0692
29	PV	0,9000	0,3422	-22,0343	1,0500
30	PQ	-1,2500	-0,3980	-26,2143	1,0243
31	PV	1,2000	0,3086	-20,6397	1,0494
32	PV	2,4100	0,5148	-19,3440	1,0488
33	PQ	0,0000	0,0000	-25,0948	1,0319
34	PV	11,0000	-2,2853	2,6056	1,0461
35	PQ	0,0000	0,0000	-1,2497	1,0634
36	PQ	-8,1300	-1,1000	-30,8404	1,0683
37	PQ	-6,1200	4,5500	-30,1982	1,0780
38	PV	4,6000	0,9602	-13,1884	1,0500
39	PQ	-4,0400	-1,3500	-18,9296	1,0336
40	PQ	-3,9300	1,1100	-23,4714	1,0730
41	PQ	-2,6200	-0,1320	-14,2699	1,0336
42	PQ	-2,2900	-1,8300	-10,6737	1,0506
43	PQ	-1,8400	-0,6020	-14,0845	1,0231
44	PQ	-1,3900	-0,5370	-12,1379	1,0179
45	PQ	-0,9010	-0,5530	-29,1607	1,0020
Perdas		**1,4305**	**-10,9439**		

Tabela 4.3: Resultados para a otimização restrita nas variáveis de controle para o sistema sul-sudeste-Mato Grosso reduzido partindo do *flat start*.

Dados de Barra		Simulado no Programa Desenvolvido			
Barra Nº	Tipo	Injeção de Potência Ativa - P	Injeção de Potência Reativa - Q	Ângulo de Tensão - θ	Módulo de Tensão - V
		(p.u.)	(p.u.)	(graus)	(p.u.)
12	PV	3,0000	-1,9986	-24,0565	1,0494
16	PV	8,0000	-2,4479	-25,9228	1,0470
18	Vθ	9,6958	-5,6377	-24,0011	1,0465
20	PV	9,0000	-4,0566	-22,3726	1,0488
21	PV	1,3999	-0,5024	-54,9966	1,0494
22	PV	1,5000	-0,2553	-20,1730	1,0496
35	PV	2,0000	-0,8593	-26,7661	1,0495
48	PV	0,0000	-6,1869	-40,5578	1,0462
86	PQ	-0,6600	-0,0120	-40,5578	1,0884
100	PQ	0,0000	0,0000	-28,0161	1,1030
101	PQ	0,0000	0,0000	-35,0262	1,1327
102	PQ	0,0000	0,0000	-40,8833	1,1354
103	PQ	0,0000	0,0000	-41,0840	1,1450
104	PQ	-9,1000	-2,3500	-48,4143	1,1514
106	PQ	0,0000	0,0000	-49,1489	1,1425
120	PQ	-1,8000	-0,9000	-39,3509	1,1110
122	PQ	-2,0000	-0,3800	-39,5884	1,1302
123	PQ	-4,5000	-1,7500	-43,5328	1,1094
126	PQ	-2,9000	-0,9500	-41,2507	1,0971
131	PQ	0,0000	0,0001	-26,8946	1,0785
134	PQ	0,0000	0,0000	-26,0900	1,0755
136	PQ	-0,5400	-0,2300	-32,1084	1,0892
138	PQ	-0,7200	-0,3400	-41,8806	1,1205
140	PQ	-7,0000	-2,5000	-50,0623	1,1172
210	PQ	0,0000	0,0000	-27,2670	1,0842
213	PQ	-0,9300	-0,3900	-28,2718	1,0945
216	PQ	-0,5300	-0,2500	-27,4307	1,0905
217	PQ	-3,6400	-0,5800	-31,6024	1,0917
218	PQ	-6,0000	-2,0000	-38,7796	1,0722
219	PQ	0,0000	0,0000	-37,7492	1,0752
220	PQ	0,0000	0,0004	-31,2758	1,0897
225	PQ	0,0000	0,0000	-33,7105	1,1031
228	PQ	-0,8600	-0,3400	-39,2989	1,0633
231	PQ	-0,8970	-0,3190	-45,7061	1,1271
233	PQ	0,0000	0,0000	-35,2611	1,0851
234	PQ	-9,9999	-3,5000	-37,8364	1,0741
300	PV	7,0000	-2,4368	-19,1763	1,0490
301	PV	3,0000	-1,3895	-19,6429	1,0498
302	PV	4,0000	-1,6535	-18,5386	1,0495

Dados de Barra		Simulado no Programa Desenvolvido			
Barra Nº	Tipo	Injeção de Potência Ativa - P	Injeção de Potência Reativa - Q	Ângulo de Tensão - θ	Módulo de Tensão - V
		(p.u.)	(p.u.)	(graus)	(p.u.)
303	PV	2,0000	-3,2970	-24,1313	1,0496
305	PV	2,9999	-1,3667	-22,2262	1,0493
320	PQ	0,0000	0,0000	-23,9658	1,0843
325	PQ	0,0000	0,0003	-23,6101	1,0873
326	PQ	-2,7400	-1,0400	-25,6502	1,0779
360	PQ	0,0000	0,0003	-22,4483	1,0825
370	PQ	0,0000	0,0005	-25,1974	1,0830
396	PQ	0,0000	0,0007	-25,5655	1,0798
500	PV	8,0000	-1,9141	-21,5374	1,0478
535	PQ	0,0000	0,0000	-25,7342	1,0694
536	PQ	-7,0000	-1,5000	-28,3488	1,0578
800	PV	11,0000	-0,6323	-7,7681	1,0458
808	PV	11,5000	0,0514	2,6966	1,0456
810	PV	12,0000	-0,8637	-4,5794	1,0469
814	PQ	-7,3540	-1,9100	-35,7912	1,0911
824	PQ	0,0000	0,0000	-17,2675	1,0707
834	PQ	-0,1340	-0,0420	-27,7671	1,0577
839	PQ	0,0000	0,0000	-6,7882	1,0226
840	PQ	-1,5900	-0,3600	-9,6393	1,0098
848	PQ	-0,9400	-0,1800	-6,0361	1,0273
856	PQ	0,0000	0,0000	-11,0841	1,0624
895	PQ	0,0000	0,0000	-33,7820	1,1029
896	PQ	0,0000	0,0000	-4,6870	1,0412
897	PQ	0,0000	0,0000	-3,4297	1,0512
898	PQ	0,0000	0,0000	-2,8286	1,0401
904	PV	7,0000	-3,3153	-15,0826	1,0455
915	PV	7,0000	-1,6512	-13,0960	1,0483
919	PV	7,0000	0,5371	4,3346	1,0481
925	PV	9,5000	-0,5380	-0,9369	1,0470
933	PQ	0,0000	0,0000	-17,6166	1,0715
934	PQ	-2,3700	-0,5900	-17,7753	1,0559
938	PQ	0,0000	0,0000	-35,6186	1,0968
939	PQ	-11,4899	-0,5306	-37,8012	1,0928
955	PQ	0,0000	0,0000	-23,1397	1,0994
959	PQ	0,0000	0,0000	-33,4977	1,0918
960	PQ	-8,4470	-4,6910	-35,7243	1,0665
964	PQ	0,0000	0,0000	-29,8881	1,0845
965	PQ	-7,5560	-0,5624	-32,1446	1,0798
976	PQ	0,0000	0,0000	-32,3358	1,0616
995	PQ	0,0000	0,0000	-19,1660	1,0849
1015	PQ	-0,7000	-0,0200	-37,7052	1,0852

Dados de Barra		Simulado no Programa Desenvolvido			
Barra Nº	Tipo	Injeção de Potência Ativa - P	Injeção de Potência Reativa - Q	Ângulo de Tensão - θ	Módulo de Tensão - V
		(p.u.)	(p.u.)	(graus)	(p.u.)
1030	PQ	0,0000	0,0000	-20,3680	1,0896
1047	PQ	0,0000	0,0000	-1,9080	1,0455
1060	PQ	0,0000	0,0000	-8,3621	1,0637
1210	PQ	-12,2799	-4,2500	-34,8313	1,0431
1503	PQ	0,0000	0,0000	-46,5496	1,1481
1504	PQ	-1,4500	-0,6300	-49,9210	1,1168
2458	PQ	-4,0300	-1,2600	-6,9893	1,0228
4501	PQ	-0,3140	-0,0710	-54,7814	1,1380
4521	PQ	0,0000	0,0001	-59,0709	1,1139
4522	PQ	0,0000	0,0000	-61,0009	1,1267
4523	PV	0,5000	-0,3035	-53,9902	1,0497
4530	PQ	0,0000	0,0000	-64,6592	1,1218
4532	PQ	0,0000	0,0000	-64,6592	1,1218
4533	PQ	-0,7540	-0,1605	-64,9390	1,0837
4542	PQ	0,0000	0,0000	-63,9187	1,1055
4552	PQ	-0,1260	-0,0120	-70,4604	1,1079
4562	PQ	-0,2380	-0,0740	-77,3427	1,1328
4572	PQ	-0,1800	-0,0640	-74,9285	1,1233
4582	PQ	-0,6549	-0,1670	-79,6500	1,1450
4592	PQ	0,0000	0,0004	-59,5162	1,0834
4596	PV	2,2999	-0,8569	-60,5761	1,0499
4623	PQ	-1,2819	-0,4076	-63,4809	1,1122
4703	PQ	-1,8209	-0,2975	-65,9110	1,0728
4804	PV	0,5000	-0,3168	-65,9527	1,0497
4805	PQ	0,0000	0,0001	-69,2873	1,0918
4807	PQ	-1,2889	-0,3630	-70,5066	1,0974
4862	PQ	0,0000	0,0000	-69,0122	1,1231
Perdas		**3,0792**	**-77,8622**		

Os valores de perdas ativas e reativas apresentados na Tabela 4.2 e na Tabela 4.3 são calculados somando-se as potências ativas e reativas, respectivamente, calculadas para cada barra do SEP.

Pode-se verificar na Tabela 4.2 que a potência ativa líquida calculada para a barra *slack* é $P = 15{,}1915\ pu$. Este valor é menor que o valor obtido para o *flat start* apresentado na Tabela 3.2, em que $P = 15{,}2939\ pu$. A redução é pouco expressiva devido aos limites impostos às variáveis de controle.

Na Tabela 4.3, é possível verificar que a potência ativa líquida para a barra *slack* é $P = 9{,}6958\ pu$, também menor que o valor apresentado para a mesma barra, na Tabela 3.3, em que $P = 10{,}0216\ pu$. Neste caso a diferença é um pouco mais expressiva do que no caso do sistema sul reduzido.

É possível verificar na Tabela 4.2 e na Tabela 4.3 que os valores de perdas de potência ativa foram reduzidos. Em consequência disso, houve aumento do consumo de reativos pelo sistema.

É possível verificar também que os módulos de tensão para algumas barras têm valores maiores do que os limites estipulados pela Tabela 4.1. Isto ocorre pois os limites são impostos apenas às variáveis de controle e não a todas as barras do SEP. Desta forma, as barras do tipo PQ podem ter seus valores de módulo de tensão acima do especificado para as variáveis de controle.

Os resultados obtidos para o método do Gradiente Reduzido não foram comparados a nenhum resultado obtido através de *software* comercial. A partir da validade do método para cálculo de fluxo de potência apresentado e da diminuição das perdas ativas para ambos os casos simulados, considera-se que o método obteve resultados satisfatórios para a aplicação em questão.

Capítulo 5 Análise de Contingências

Uma contingência pode ser definida como um evento aleatório no qual um ou mais equipamentos ficam fora de operação de forma inesperada, resultando assim em uma mudança de estados de um ou mais elementos do SEP. Como exemplo de contingência pode-se citar a saída de uma linha de transmissão devido à ocorrência de uma falta no sistema (QUINTELA, 2002).

As contingências podem se classificadas como:

- **Contingência simples**: contingência em que apenas um equipamento sai de operação.
- **Contingência múltipla**: caso em que mais de um equipamento sai de operação.

A análise de contingências visa verificar a operabilidade do sistema em situações de contingências múltiplas ou simples. Em um SEP de grande porte existe um elevado número de contingências a ser analisado. Desta forma, é comum que se faça a análise de todas as contingências simples do sistema, chamada de análise “n-1”. As contingências múltiplas com maior probabilidade de ocorrência são listadas e também são analisadas.

Este trabalho apenas envolve a análise “n-1” de contingências simples, com a finalidade de verificar o ponto ótimo de operação do SEP para estes casos. Não serão avaliadas as contingências de geradores e de cargas, pois estes não fazem parte da modelagem proposta. Apenas serão consideradas contingências nas linhas de transmissão do SEP.

Desta forma, podem haver 4 tipos de resultados envolvendo a análise de contingências:

- Fluxo de potência não converge: neste caso o fluxo de potência não tem solução, resultando em uma solução não factível ou chegando ao número máximo de iterações imposto ao algoritmo do FP.
- Ilhamento do sistema: há ilhamento do sistema quando uma ou mais partes do SEP ficam desconectadas.
- Método do Gradiente Reduzido converge: neste caso tem-se o ponto ótimo de operação para o cenário de operação atual.
- Método do Gradiente Reduzido não converge: neste caso o método do Gradiente Reduzido chega ao seu limite de iterações sem antes convergir.

O fluxo de potência pode não atingir a convergência devido a diversos fatores, como: geração menor que a demanda ou mesmo a não existência de um ponto de operação

factível. O método do Gradiente Reduzido pode não convergir devido à necessidade de um grande número de iterações para que este chegue ao ponto ótimo. O ilhamento do sistema gera erros na solução do programa, o que traz a necessidade de verificação prévia do mesmo, antes de resolver o fluxo de potência. A verificação é feita através do método **verify_Islanding** já implementado na classe NewtonRaphson.

5.1 Implementação Computacional

Nesta seção será apresentado o programa de aplicação deste trabalho que utiliza das classes desenvolvidas nos Capítulos 3 e 4 e também dos recursos da computação paralela utilizando openMP. Será apresentado o algoritmo utilizado no programa de aplicação e em seguida serão apresentados os resultados obtidos para a análise de contingências para os SEPs escolhidos.

O Diagrama de Classes final obtido neste trabalho é apresentado na Figura 5.1.

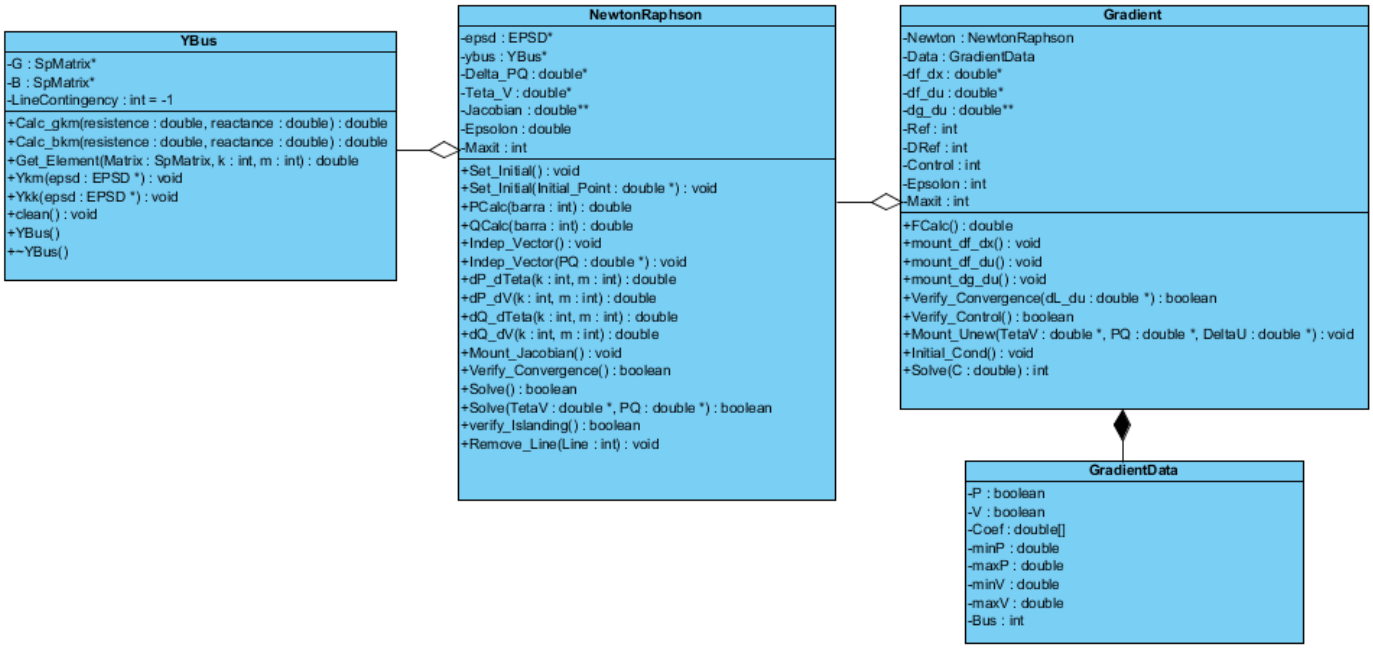

Figura 5.1: Diagrama de Classes geral das classes desenvolvidas neste trabalho.

Pode-se verificar que as classes YBus, NewtonRaphson e Gradient tem relação de agregação entre si, pois uma classe contém a outra sem que o sentido da existência de cada uma seja perdido. No caso da classe GradientData pode-se verificar uma relação de composição, visto que a classe GradientData não tem sentido existindo sozinha.

5.2 Algoritmo

O algoritmo computacional desenvolvido para o programa de aplicação é apresentado no fluxograma da Figura 5.2.

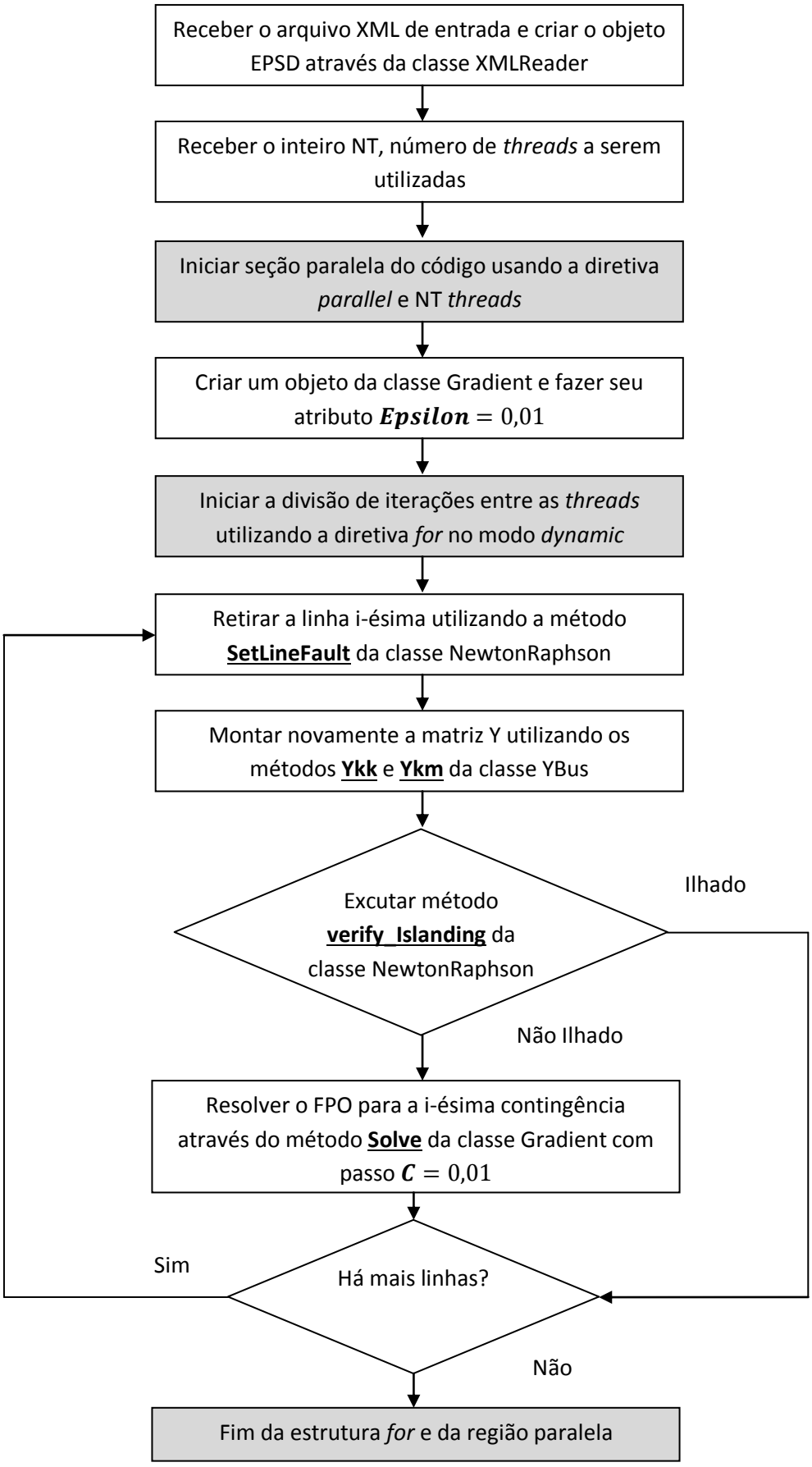

Figura 5.2: Algoritmo implementado para análise de contingência utilizando openMP.

5.3 Simulação e Resultados

Nesta seção são apresentados os resultados da aplicação do programa de análise de contingências aos sistemas sul reduzido de 45 barras e sul-sudeste-Mato Grosso de 107 barras, já avaliados com relação ao fluxo de potência e fluxo de potência ótimo nos capítulos anteriores. Para cada SEP foi considerada a otimização restrita segundo as mesmas limitações impostas pela Tabela 4.1 para cada caso de contingência.

Para cada SEP foi executada a análise de contingências utilizando apenas 1 *thread* (tornando a análise sequencial) e, em seguida, utilizando 4 *threads*. As análises foram feitas utilizando-se um computador pessoal de dois núcleos físicos de 2,34 GHz cada e duas *threads* por núcleo. Os resultados foram classificados em:

- "Sistema Ilhado": neste caso o ilhamento do sistema é acusado pelo método ***verify_Islanding***.
- "Não convergência do Fluxo de Potência": neste caso não houve convergência do fluxo de potência para a solução do *flat start*.
- "Convergência do método do Gradiente Reduzido": o método do gradiente reduzido convergiu sem problemas.

Os resultados obtidos para cada contingência são apresentados nas tabelas Tabela 5.1 e Tabela 5.2, respectivamente para os casos de 45 e 107 barras. Vale ressaltar que os resultados obtidos na execução com 1 ou com 4 *threads* são idênticos.

Tabela 5.1: Resultados para a análise de contingências para o sistema sul reduzido brasileiro composto de de 45 barras e 57 linhas.

Linha Afetada	Resultado da Análise de Contigências
1	Não convergência do Fluxo de Potência
2	Convergência do método do Gradiente Reduzido
3	Convergência do método do Gradiente Reduzido
4	Não convergência do Fluxo de Potência
5	Sistema Ilhado
6	Convergência do método do Gradiente Reduzido
7	Convergência do método do Gradiente Reduzido
8	Convergência do método do Gradiente Reduzido
9	Convergência do método do Gradiente Reduzido
10	Convergência do método do Gradiente Reduzido
11	Sistema Ilhado
12	Convergência do método do Gradiente Reduzido
13	Convergência do método do Gradiente Reduzido
14	Convergência do método do Gradiente Reduzido
15	Convergência do método do Gradiente Reduzido

Linha Afetada	Resultado da Análise de Contigências
16	Convergência do método do Gradiente Reduzido
17	Sistema Ilhado
18	Convergência do método do Gradiente Reduzido
19	Convergência do método do Gradiente Reduzido
20	Convergência do método do Gradiente Reduzido
21	Convergência do método do Gradiente Reduzido
22	Convergência do método do Gradiente Reduzido
23	Convergência do método do Gradiente Reduzido
24	Não convergência do Fluxo de Potência
25	Convergência do método do Gradiente Reduzido
26	Convergência do método do Gradiente Reduzido
27	Não convergência do Fluxo de Potência
28	Convergência do método do Gradiente Reduzido
29	Sistema Ilhado
30	Não convergência do Fluxo de Potência
31	Convergência do método do Gradiente Reduzido
32	Convergência do método do Gradiente Reduzido
33	Convergência do método do Gradiente Reduzido
34	Convergência do método do Gradiente Reduzido
35	Não convergência do Fluxo de Potência
36	Não convergência do Fluxo de Potência
37	Não convergência do Fluxo de Potência
38	Não convergência do Fluxo de Potência
39	Não convergência do Fluxo de Potência
40	Não convergência do Fluxo de Potência
41	Sistema Ilhado
42	Convergência do método do Gradiente Reduzido
43	Sistema Ilhado
44	Não convergência do Fluxo de Potência
45	Sistema Ilhado
46	Sistema Ilhado
47	Convergência do método do Gradiente Reduzido
48	Sistema Ilhado
49	Não convergência do Fluxo de Potência
50	Convergência do método do Gradiente Reduzido
51	Sistema Ilhado
52	Convergência do método do Gradiente Reduzido
53	Convergência do método do Gradiente Reduzido
54	Convergência do método do Gradiente Reduzido
55	Convergência do método do Gradiente Reduzido
56	Convergência do método do Gradiente Reduzido
57	Convergência do método do Gradiente Reduzido

Tabela 5.2: Resultados para a análise de contingências para o sistema sul-sudeste-Mato Grosso reduzido brasileiro de 107 barras e 171 linhas.

Linha Afetada		Resultado da Análise de Contigências
De Barra	Para Barra	
86	122	Sistema Ilhado
86	122	Convergência do método do Gradiente Reduzido
100	20	Convergência do método do Gradiente Reduzido
100	101	Sistema Ilhado
100	101	Convergência do método do Gradiente Reduzido
100	210	Convergência do método do Gradiente Reduzido
100	213	Convergência do método do Gradiente Reduzido
100	535	Convergência do método do Gradiente Reduzido
101	102	Convergência do método do Gradiente Reduzido
101	103	Convergência do método do Gradiente Reduzido
102	120	Convergência do método do Gradiente Reduzido
102	1503	Convergência do método do Gradiente Reduzido
103	123	Não convergência do Fluxo de Potência
104	103	Convergência do método do Gradiente Reduzido
104	1503	Não convergência do Fluxo de Potência
106	104	Não convergência do Fluxo de Potência
106	104	Convergência do método do Gradiente Reduzido
106	140	Convergência do método do Gradiente Reduzido
106	140	Convergência do método do Gradiente Reduzido
86	48	Convergência do método do Gradiente Reduzido
122	103	Convergência do método do Gradiente Reduzido
123	120	Convergência do método do Gradiente Reduzido
126	86	Convergência do método do Gradiente Reduzido
126	86	Convergência do método do Gradiente Reduzido
126	120	Convergência do método do Gradiente Reduzido
126	120	Convergência do método do Gradiente Reduzido
131	22	Sistema Ilhado
134	12	Sistema Ilhado
134	131	Convergência do método do Gradiente Reduzido
134	396	Convergência do método do Gradiente Reduzido
136	16	Sistema Ilhado
136	120	Convergência do método do Gradiente Reduzido
136	120	Convergência do método do Gradiente Reduzido
136	131	Convergência do método do Gradiente Reduzido
136	134	Convergência do método do Gradiente Reduzido
136	138	Convergência do método do Gradiente Reduzido
136	138	Convergência do método do Gradiente Reduzido
140	138	Convergência do método do Gradiente Reduzido
140	138	Convergência do método do Gradiente Reduzido

Linha Afetada		Resultado da Análise de Contigências
De Barra	Para Barra	
210	18	Sistema Ilhado
210	217	Convergência do método do Gradiente Reduzido
210	217	Convergência do método do Gradiente Reduzido
210	370	Convergência do método do Gradiente Reduzido
213	216	Convergência do método do Gradiente Reduzido
216	396	Convergência do método do Gradiente Reduzido
217	216	Convergência do método do Gradiente Reduzido
217	218	Convergência do método do Gradiente Reduzido
217	218	Convergência do método do Gradiente Reduzido
218	234	Convergência do método do Gradiente Reduzido
218	234	Convergência do método do Gradiente Reduzido
219	234	Convergência do método do Gradiente Reduzido
219	234	Convergência do método do Gradiente Reduzido
220	35	Sistema Ilhado
220	217	Convergência do método do Gradiente Reduzido
220	219	Convergência do método do Gradiente Reduzido
225	217	Convergência do método do Gradiente Reduzido
225	217	Convergência do método do Gradiente Reduzido
225	231	Convergência do método do Gradiente Reduzido
225	231	Não convergência do Fluxo de Potência
228	219	Sistema Ilhado
231	4501	Convergência do método do Gradiente Reduzido
231	4501	Não convergência do Fluxo de Potência
233	210	Convergência do método do Gradiente Reduzido
233	320	Convergência do método do Gradiente Reduzido
234	233	Convergência do método do Gradiente Reduzido
234	233	Convergência do método do Gradiente Reduzido
320	210	Convergência do método do Gradiente Reduzido
320	300	Sistema Ilhado
320	360	Convergência do método do Gradiente Reduzido
325	301	Sistema Ilhado
325	326	Convergência do método do Gradiente Reduzido
325	326	Convergência do método do Gradiente Reduzido
325	360	Convergência do método do Gradiente Reduzido
325	370	Convergência do método do Gradiente Reduzido
326	134	Convergência do método do Gradiente Reduzido
326	396	Convergência do método do Gradiente Reduzido
360	302	Sistema Ilhado
370	303	Sistema Ilhado
370	535	Convergência do método do Gradiente Reduzido
396	305	Sistema Ilhado
535	500	Sistema Ilhado
536	535	Convergência do método do Gradiente Reduzido

Linha Afetada		Resultado da Análise de Contigências
De Barra	Para Barra	
814	895	Convergência do método do Gradiente Reduzido
814	895	Convergência do método do Gradiente Reduzido
824	800	Convergência do método do Gradiente Reduzido
824	933	Sistema Ilhado
824	933	Convergência do método do Gradiente Reduzido
834	934	Convergência do método do Gradiente Reduzido
839	840	Convergência do método do Gradiente Reduzido
839	840	Convergência do método do Gradiente Reduzido
839	898	Convergência do método do Gradiente Reduzido
839	1047	Convergência do método do Gradiente Reduzido
839	2458	Convergência do método do Gradiente Reduzido
839	2458	Convergência do método do Gradiente Reduzido
856	810	Convergência do método do Gradiente Reduzido
856	933	Sistema Ilhado
856	1060	Não convergência do Fluxo de Potência
895	122	Convergência do método do Gradiente Reduzido
895	122	Convergência do método do Gradiente Reduzido
896	897	Convergência do método do Gradiente Reduzido
897	808	Convergência do método do Gradiente Reduzido
898	848	Sistema Ilhado
898	1047	Sistema Ilhado
933	895	Convergência do método do Gradiente Reduzido
933	955	Não convergência do Fluxo de Potência
933	959	Convergência do método do Gradiente Reduzido
934	933	Não convergência do Fluxo de Potência
934	1047	Convergência do método do Gradiente Reduzido
934	1047	Convergência do método do Gradiente Reduzido
938	955	Convergência do método do Gradiente Reduzido
938	959	Não convergência do Fluxo de Potência
939	938	Convergência do método do Gradiente Reduzido
939	938	Convergência do método do Gradiente Reduzido
939	938	Convergência do método do Gradiente Reduzido
939	1015	Convergência do método do Gradiente Reduzido
939	1015	Convergência do método do Gradiente Reduzido
955	964	Convergência do método do Gradiente Reduzido
959	895	Não convergência do Fluxo de Potência
960	834	Convergência do método do Gradiente Reduzido
960	959	Convergência do método do Gradiente Reduzido
960	959	Convergência do método do Gradiente Reduzido
960	1015	Convergência do método do Gradiente Reduzido
960	1015	Convergência do método do Gradiente Reduzido
964	976	Convergência do método do Gradiente Reduzido
965	964	Não convergência do Fluxo de Potência

Linha Afetada		Resultado da Análise de Contigências
De Barra	Para Barra	
965	964	Convergência do método do Gradiente Reduzido
976	995	Convergência do método do Gradiente Reduzido
995	904	Não convergência do Fluxo de Potência
995	964	Sistema Ilhado
995	1030	Não convergência do Fluxo de Potência
995	1060	Convergência do método do Gradiente Reduzido
1030	915	Convergência do método do Gradiente Reduzido
1030	955	Sistema Ilhado
1047	919	Convergência do método do Gradiente Reduzido
1060	897	Sistema Ilhado
1060	925	Não convergência do Fluxo de Potência
1210	976	Sistema Ilhado
1210	976	Convergência do método do Gradiente Reduzido
1210	976	Convergência do método do Gradiente Reduzido
1503	1504	Convergência do método do Gradiente Reduzido
2458	896	Sistema Ilhado
4501	4522	Convergência do método do Gradiente Reduzido
4501	4522	Convergência do método do Gradiente Reduzido
4521	4523	Convergência do método do Gradiente Reduzido
4522	4521	Sistema Ilhado
4522	4532	Não convergência do Fluxo de Potência
4522	4532	Convergência do método do Gradiente Reduzido
4522	4623	Convergência do método do Gradiente Reduzido
4522	4623	Convergência do método do Gradiente Reduzido
4532	4530	Convergência do método do Gradiente Reduzido
4532	4533	Sistema Ilhado
4532	4533	Convergência do método do Gradiente Reduzido
4532	4533	Convergência do método do Gradiente Reduzido
4532	4542	Convergência do método do Gradiente Reduzido
4533	4596	Não convergência do Fluxo de Potência
4542	4552	Sistema Ilhado
4552	4572	Não convergência do Fluxo de Potência
4562	4572	Não convergência do Fluxo de Potência
4562	4582	Não convergência do Fluxo de Potência
4592	21	Sistema Ilhado
4592	4542	Sistema Ilhado
4623	4533	Não convergência do Fluxo de Potência
4703	4533	Convergência do método do Gradiente Reduzido
4703	4533	Convergência do método do Gradiente Reduzido
4805	4804	Convergência do método do Gradiente Reduzido
4805	4807	Sistema Ilhado
4805	4807	Convergência do método do Gradiente Reduzido
4862	4532	Convergência do método do Gradiente Reduzido

Linha Afetada		Resultado da Análise de Contigências
De Barra	**Para Barra**	
4862	4532	Convergência do método do Gradiente Reduzido
4862	4807	Convergência do método do Gradiente Reduzido
536	535	Não convergência do Fluxo de Potência

Os resultados do fluxo de potência ótimo para os casos onde houve convergência do método Gradiente Reduzido não serão apresentados devido à quantidade de informação envolvida. Um resumo dos resultados obtidos para ambos os SEPs analisados é mostrado na **Tabela** 5.3.

Tabela 5.3: Resumo dos casos resultantes obtidos para a análise de contingência dos sistemas sul reduzido e sul-sudeste-Mato Grosso reduzido.

Caso Resultante	Número de Contigências	
	Sul Reduzido	**Sul-Sudeste-Mato Grosso Reduzido**
Sistema Ilhado	29	10
Não convergência do Fluxo de Potência	21	13
Convergência do método do Gradiente	121	34
Total	**171**	**57**

A **Tabela** 5.3 mostra que 70,8% dos casos obtiveram convergência para o método do Gradiente Reduzido para o sistema sul reduzido. Já para o sistema sul-sudeste-Mato Grosso 59,6% dos casos obtiveram convergência para o método.

A avaliação do tempo de processamento é mostrada na Tabela 5.4.

Tabela 5.4: Avaliação do tempo de processamento total em função do número de *threads* utilizadas para a simulação.

SEP	Número de *Threads*	Tempo de Processamento Total (s)	Redução no tempo de processamento
Sul Reduzido	1	9	66,7%
	4	3	
Sul-Sudeste-Mato Grosso Reduzido	1	488	34,6%
	4	319	

Através da Tabela 5.4 pode-se verificar que a execução da análise com quatro *threads* reduziu consideravelmente o tempo de processamento para ambos os SEPs frente aos casos em que se utilizou apenas 1 *thread*. A diminuição encontrada é significativa quando considera-se que, embora tenha 4 *threads,* o processador utilizado tem apenas 2 núcleos físicos.

É possível verificar também que a redução no tempo de processamento para o sistema sul reduzido foi maior que a redução para o sistema sul-sudeste-Mato Grosso Reduzido. A tendência natural é que a redução no tempo de processamento fosse maior para um sistema de maior porte. Porém, deve-se levar em consideração que a computação paralela depende de vários fatores que podem se contrapor à este resultado. São eles:

- Alocação de memória;
- Tipo de divisão de tarefas entre as *threads* definido pelas diretivas de compilação;
- Gerenciamento de tarefas pelo sistema operacional;
- Outras atividades em processo no sistema operacional;

Os resultados obtidos mostram a vantagem da utilização da computação paralela quando há uma quantidade de informação muito grande a ser analisada. Quando utilizado um sistema de computadores em rede do tipo *cluster*, pode-se utilizar de vários processadores para a divisão do trabalho na computação paralela. Desta forma, a redução temporal será ainda mais significante, sendo uma ótima alternativa para análise das contingências de um SEP a nível “n-1” ou superior.

Capítulo 6 Conclusões

Através do presente trabalho pôde-se verificar a vantagem oferecida pela computação paralela, em termos de tempo de processamento, quando aplicada à análise de contingências, onde se têm um grande volume de dados a ser processado. Os resultados mostram uma diminuição no tempo de processamento quando comparada a execução do programa utilizando 1 e 4 *threads*. Quando utilizado um sistema *cluster* podem ser encontrados resultados mais expressivos com relação à diminuição de tempo de processamento.

A implementação do código em C++ utilizando os preceitos da orientação a objetos é capaz de criar um programa modular e de fácil manutenabilidade. Pôde-se verificar a relação de agregação e composição das classes geradas, bem como a interdependência entre elas. Além disso, a programação orientada a objetos torna a estrutura do programa menos repetitiva e mais favorável à extensibilidade.

Os resultados obtidos nas simulações para o método de Newton-Raphson implementado foram comparados à solução obtida na ferramenta computacional ANAREDE. Os resultados se apresentaram satisfatórios frente à modelagem desenvolvida no trabalho. Conclui-se que uma modelagem mais completa do problema pode trazer resultados mais precisos.

O método do Gradiente Reduzido implementado ofereceu resultados satisfatórios com relação à otimização do fluxo de potência. Pôde-se observar que houve diminuição na potência ativa líquida na barra *slack* e consequente diminuição das perdas totais para os SEPs analisados. Apesar do Gradiente Reduzido ser um método de lenta convergência, os resultados com relação à tempo de processamento total na análise de contingências foram favorecidos com a utilização de programação paralela.

Assim, os conceitos e análises desenvolvidos neste trabalho mostram que a utilização de recursos como a orientação a objetos e a computação paralela, quando aplicados à uma modelagem mais completa do problema, utilizando métodos de otimização de mais rápida convergência, podem trazer resultados expressivos para a análise de contingências de um SEP, em termos de tempo de processamento, quando se trata de análise de segurança em tempo real.

Referências Bibliográficas

ANAREDE. Disponível em: <http://www.anarede.cepel.br>. Acesso em 05 de novembro de 2012.

BALU, N.; BERTRAM, T.; BOSE, A.; BRANDWAJN, V.; CAULEY, G.; CURTICE, D.; FOUAD, A.; FINK, L.; LAUBY, M. G.; WOLLENBERG, B. F.; WRUBEL, J. N. Online power system security analysis. **Proceedings of The IEEE**, [S.l.], v.80, p.262–282, 1992.

BAZARAA, M. S. **Nonlinear programming theory and algorithms**. New York: Wiley, 1993.

DEITEL, H. M. DEITEL, P. J. **C++: como programar**. Porto Alegre: Bookman, 2001 .

FARINELLI, F. **Conceitos Básicos de Programação Orientada a Objetos**. 2007.

GRAMA, A. **Introduction to Parallel Computing**. 2. ed. Addison-Wesley, 2003.

HAPP ,H. H. WIRGAU , K. A. **A review of the optimal power flow**. Journal of the Franklin Institute, 1981.

LLNL – *Lawrence Livermore National Laboratory*. Disponível em: <https://computing.llnl.gov/tutorials/openMP>. Acesso em 05 de novembro de 2012.

MANSOUR, M. R. GERALDI JUNIOR, E. L. RAMOS R. A. ALBERTO, L. F. C. 2011. Ferramenta Computacional para o Cálculo de Fluxo de Carga Continuado em Sistemas de Potência Utilizando uma Programação Orientada a Objetos. In: XIV ERIAC - Encuentro Regional Ibero-Americano de CIGRÉ, 2011, Ciudad del Este. XIV ERIAC - Encuentro Regional Ibero-Americano de CIGRÉ, 2011.

MEDEIROS, E. **Desenvolvendo Software Com Uml 2.0 – Definitivo**. Makron Books, 2004.

MOMOH, J.A. EL-HAWARY, M.E. ADAPA, R. A Review of Selected Optimal Power Flow Literature to 1993 Part I: Non-Linear and Quadratic Programming Approaches. **IEEE Transactions on Power Systems**, Vol. 14, 1999a.

MOMOH, J.A. EL-HAWARY, M.E. ADAPA, R. A Review of Selected Optimal Power Flow Literature to 1993 Part II: Newton, Linear Programming and Interior Point Methods**. IEEE Transactions on Power Systems**, Vol. 14, 1999b.

MONTICELLI, A. **Fluxo de Carga em Redes de Energia Elétrica**. Edgar Blucher, 1983.

OMG – *Object Management Group*. Disponível em: <www.omg.org>. Acesso em 05 de novembro de 2012.

QUINTELA, A. S. **Estudo de indices de proximidade ao limite de estabilidade de tensão e aplicação na seleção de contingencias**. 2002. Dissertação (Mestrado em Engenharia Elétrica) – UNICAMP, Campinas, SP.

RICARTE, I. L. M. Programação Orientada a Objetos com C++. 1995, DCA/FEE/UNICAMP.

SANTOS, C. J. R. **Método rápido para avaliação da margem de estabilidade de tensão considerando os limites de potência reativa dos geradores**. 2002. Dissertação (Mestrado em Engenharia Elétrica) – EESC/USP, São Carlos, SP.

UBUNTU. Disponível em: <http://www.ubuntu.com>. Acesso em 05 de novembro de 2012.

WOOD, ALLEN J. WOLLENBERG, BRUCE F. **Power Generation Operation and Control**. New York: John Wiley and Sons, 1996.

XML. Diponível em: <http://www.w3.org/xml>. Acesso em 05 de novembro de 2012

Apêndice A – Dados do Sistema-Teste Sul-Sudeste-Mato Grosso Reduzido de 107 barras

A.1 Características Elétricas do Sistema-Teste de 107 barras

Este sistema-teste representa uma versão reduzida dos sistemas elétricos das regiões Sul, Sudeste e Mato Grosso. É composto do sistema teste de 65 barras acrescido da malha de 345 kV da região Sudeste e da malha de suprimento ao Mato Grosso em 230 kV. O sistema-teste é dividido em três áreas, denominadas: Sul, Sudeste e Mato Grosso.

Os dados elétricos das barras do sistema são mostrados na **Tabela A-1**.

Tabela A-1: Dados elétricos das barras do sistema-teste Sul-Sudeste-Mato Grosso de 107 barras.

Barra Nº	Nome da Barra	Tipo	Tensão (kV)	Limites de Tensão		Área
				Max (pu)	Min (pu)	
12	Luiz Carlos Barreto	PV	13,8	1,05	0,95	1
16	Furnas	PV	13,8	1,05	0,95	1
18	Itumbiara	Vθ	13,8	1,05	0,95	1
20	Marimbondo	PV	13,8	1,05	0,95	1
21	Manso	PV	13,8	1,05	0,95	3
22	Mascarenhas de Moraes	PV	13,8	1,05	0,95	1
35	Corumbá	PV	13,8	1,05	0,95	1
48	Ibiúna	PV	13,8	1,05	0,95	1
86	Ibiúna	PQ	345	1,07	0,95	1
100	Marimbondo	PQ	500	1,1	0,95	1
101	Araraquara	PQ	500	1,1	0,95	1
102	Poços de Caldas	PQ	500	1,1	0,95	1
103	Campinas	PQ	500	1,1	0,95	1
104	Cachoeira Paulista	PQ	500	1,1	0,95	1
106	Adrianópolis	PQ	500	1,1	0,95	1
120	Poços de Caldas	PQ	345	1,07	0,95	1
122	Ibiúna	PQ	500	1,1	0,95	1
123	Campinas	PQ	345	1,07	0,95	1
126	Guarulhos	PQ	345	1,07	0,95	1
131	Mascarenhas de Moraes	PQ	345	1,07	0,95	1
134	Luiz Carlos Barreto	PQ	345	1,07	0,95	1
136	Furnas	PQ	345	1,07	0,95	1
138	Itutinga	PQ	345	1,07	0,95	1
140	Adrianópolis	PQ	345	1,07	0,95	1
210	Itumbiara	PQ	500	1,1	0,95	1
213	Marimbondo	PQ	345	1,07	0,95	1
216	Porto Colômbia	PQ	345	1,07	0,95	1
217	Itumbiara	PQ	345	1,07	0,95	1

Barra Nº	Nome da Barra	Tipo	Tensão (kV)	Limites de Tensão		Área
				Max (pu)	Min (pu)	
218	Bandeirantes	PQ	345	1,07	0,95	1
219	Brasília Sul	PQ	345	1,07	0,95	1
220	Corumbá	PQ	345	1,07	0,95	1
225	Itumbiara	PQ	230	1,07	0,95	1
228	Brasília Sul	PQ	230	1,07	0,95	1
231	Rio Verde	PQ	230	1,07	0,95	1
233	Samambaia	PQ	500	1,1	0,95	1
234	Samambaia	PQ	345	1,07	0,95	1
300	Emborcação	PV	13,8	1,05	0,95	1
301	Jaguara	PV	13,8	1,05	0,95	1
302	Nova Ponte	PV	13,8	1,05	0,95	1
303	São Simão	PV	13,8	1,05	0,95	1
305	Volta Grande	PV	13,8	1,05	0,95	1
320	Emborcação	PQ	500	1,1	0,95	1
325	Jaguara	PQ	500	1,1	0,95	1
326	Jaguara	PQ	345	1,07	0,95	1
360	Nova Ponte	PQ	500	1,1	0,95	1
370	São Simão	PQ	500	1,1	0,95	1
396	Volta Grande	PQ	345	1,07	0,95	1
500	Água Vermelha	PV	13,8	1,05	0,95	1
535	Água Vermelha	PQ	500	1,1	0,95	1
536	Água Vermelha	PQ	440	1,07	0,95	1
800	Governador Bento Munhoz	PV	13,8	1,05	0,95	2
808	Salto Caxias	PV	13,8	1,05	0,95	2
810	Salto Segredo	PV	13,8	1,05	0,95	2
814	Bateias	PQ	230	1,05	0,95	2
824	Governador Bento Munhoz	PQ	500	1,09	0,95	2
834	São Mateus	PQ	230	1,05	0,95	2
839	Cascavel	PQ	230	1,05	0,95	2
840	Cascavel	PQ	138	1,05	0,95	2
848	Foz do Chopin	PQ	138	1,05	0,95	2
856	Segredo	PQ	500	1,09	0,95	2
895	Bateias	PQ	500	1,09	0,95	2
896	Cascavel do Oeste	PQ	500	1,09	0,95	2
897	Salto Caxias	PQ	500	1,09	0,95	2
898	Foz do Chopin	PQ	230	1,05	0,95	2
904	Itá	PV	13,8	1,05	0,95	2
915	Machadinho	PV	13,8	1,05	0,95	2
919	Salto Osório	PV	13,8	1,05	0,95	2
925	Salto Santiago	PV	13,8	1,05	0,95	2
933	Areia	PQ	500	1,09	0,95	2

Barra Nº	Nome da Barra	Tipo	Tensão (kV)	Limites de Tensão		Área
				Max (pu)	Min (pu)	
934	Areia	PQ	230	1,05	0,95	2
938	Blumenau	PQ	500	1,09	0,95	2
939	Blumenau	PQ	230	1,05	0,95	2
955	Campos Novos	PQ	500	1,09	0,95	2
959	Curitiba	PQ	500	1,09	0,95	2
960	Curitiba	PQ	230	1,05	0,95	2
964	Caxias	PQ	500	1,09	0,95	2
965	Caxias	PQ	230	1,05	0,95	2
976	Gravataí	PQ	500	1,09	0,95	2
995	Itá	PQ	500	1,09	0,95	2
1015	Joinville	PQ	230	1,05	0,95	2
1030	Machadinho	PQ	500	1,09	0,95	2
1047	Salto Osório	PQ	230	1,05	0,95	2
1060	Salto Santiago	PQ	500	1,09	0,95	2
1210	Gravataí	PQ	230	1,05	0,95	2
1503	Itajubá	PQ	500	1,1	0,95	1
1504	Itajubá	PQ	138	1,05	0,95	1
2458	Cascavel	PQ	230	1,05	0,95	2
4501	Barra do Peixe	PQ	230	1,07	0,95	3
4521	Itiquira	PQ	230	1,07	0,95	3
4522	Rondonópolis	PQ	230	1,07	0,95	3
4523	Itiquira	PV	13,8	1,05	0,95	3
4530	Coxipó	PV	12	1,05	0,95	3
4532	Coxipó	PQ	230	1,07	0,95	3
4533	Coxipó	PQ	138	1,05	0,95	3
4542	Nobres	PQ	230	1,07	0,95	3
4552	Nova Mutun	PQ	230	1,07	0,95	3
4562	Sorriso	PQ	230	1,07	0,95	3
4572	Lucas do Rio Verde	PQ	230	1,07	0,95	3
4582	Sinop	PQ	230	1,07	0,95	3
4592	Manso	PQ	230	1,07	0,95	3
4596	Cuiabá	PV	13,8	1,05	0,95	3
4623	Rondonópolis	PQ	138	1,05	0,95	3
4703	Cuiabá	PQ	138	1,05	0,95	3
4804	Guaporé	PV	13,8	1,05	0,95	3
4805	Guaporé	PQ	138	1,05	0,95	3
4807	Jauru	PQ	138	1,05	0,95	3
4862	Jauru	PQ	230	1,07	0,95	3

Os dados elétricos das linhas do sistema são mostrados na **Tabela A-2**.

Tabela A-2: Dados elétricos das linhas do sistema-teste Sul-Sudeste-Mato Grosso de 107 barras.

Dados da Linha			Sequência Positiva		Susceptância Shunt (Mvar)
De Barra	Para Barra	Nome	Resistência Equivalente (%)	Reatância Equivalente (%)	
100	101	Marimbondo-Araraquara	0,1720	2,7200	231,40
100	101	Marimbondo-Araraquara	0,1710	2,7000	230,20
100	210	Marimbondo-Itumbiara	0,2090	2,9350	254,60
100	535	Marimbondo-A.Vermelha	0,1530	2,4000	203,80
101	102	Araraquara-P.Caldas	0,1560	2,4600	208,50
101	103	Araraquara-Campinas	0,1520	2,3900	202,60
102	1503	Poços de Caldas-Itajubá	0,1100	1,9100	161,85
103	104	Campinas-C.Paulista	0,1960	3,1000	264,90
104	1503	Cachoeira Paulista-Itajubá	0,0500	0,8200	69,36
103	122	Campinas-Ibiúna	0,1050	1,6190	136,35
210	370	Itumbiara-São Simão	0,1470	2,3200	196,60
210	233	Itumbiara-Samambaia	0,2800	3,9900	355,36
233	320	Samambaia-Emborcação	0,2700	3,8700	344,03
210	320	Itumbiara-Emborcação	0,1250	1,9370	149,96
320	360	Emborcação-N.Ponte	0,0820	1,2560	98,99
325	360	Jaguara-Nova Ponte	0,1000	1,5190	119,67
325	370	Jaguara-São Simão	0,2800	4,8400	419,50
370	535	S.Simão-Água Vermelha	0,0931	1,3758	112,30
824	933	Gov.Bento Munhoz-Areia	0,0100	0,1240	15,20
824	933	Gov.Bento Munhoz-Areia	0,0100	0,1260	15,43
834	934	São Mateus-Areia	2,4440	12,6520	21,71
839	898	Cascavel-F.Chopin	1,1300	6,9900	12,62
839	1047	Cascavel-S.Osório	1,2200	7,6900	13,81
839	2458	Cascavel-Cascavel Oeste	0,2200	1,0900	1,86
839	2458	Cascavel-Cascavel Oeste	0,1700	1,0300	2,05
856	933	Segredo-Areia	0,0520	0,6540	80,49
856	1060	Segredo-S.Santiago	0,0560	0,6970	85,75
122	895	Ibiúna-Bateias	0,3080	3,9580	444,84
122	895	Ibiúna-Bateias	0,3080	3,9580	444,84
896	897	Cascavel Oeste-S.Caxias	0,0500	0,7300	78,06
898	1047	F.Chopin-S.Osório	0,1500	0,8900	1,63
933	895	Areia-Bateias	0,2000	2,5500	312,72
933	955	Areia-Campos Novos	0,1620	2,0480	250,17
933	959	Areia-Curitiba	0,2000	2,6900	336,40
934	1047	Areia-Salto Osório	3,0450	15,7380	27,12
934	1047	Areia-Salto Osório	3,0410	15,7180	27,09
938	955	Blumenau-C.Novos	0,2556	2,9224	360,40
938	959	Blumenau-Curitiba	0,1270	1,6030	195,89

Dados da Linha			Sequência Positiva		Susceptância Shunt (Mvar)
De Barra	Para Barra	Nome	Resistência Equivalente (%)	Reatância Equivalente (%)	
955	964	Campos Novos-Caxias	0,1877	2,3467	287,24
959	895	Curitiba-Bateias	0,0500	0,4400	47,58
964	976	Caxias-Gravataí	0,0733	0,9164	112,17
976	995	Gravataí-Itá	0,2820	3,8520	493,70
995	964	Itá-Caxias	0,1643	3,0339	354,88
995	1030	Itá-Machadinho	0,0730	0,9200	112,26
995	1060	Itá-Salto Santiago	0,1720	2,1700	265,16
1030	955	Machadinho-C.Novos	0,0470	0,5900	71,82
1060	897	S.Santiago-S.Caxias	0,0760	1,1710	124,58
939	1015	Blumenau-Joinville	1,2710	6,5620	11,31
939	1015	Blumenau-Joinville	1,2830	6,5640	11,52
1015	960	Joinville-Curitiba	1,8920	9,7760	16,85
1015	960	Joinville-Curitiba	1,8950	9,7040	17,03
960	834	Curitiba-São Mateus	2,2110	1,4750	19,69
126	86	Guarulhos-Ibiúna	0,1090	1,8260	51,18
126	86	Guarulhos-Ibiúna	0,1090	1,8260	51,18
126	120	Guarulhos-P.Caldas	0,6000	5,9500	92,80
126	120	Guarulhos-P.Caldas	0,6060	6,0200	93,80
134	131	L.C.Barreto-M.Moraes	0,0920	1,0100	16,90
134	396	L.C.Barreto-V.Grande	0,3200	3,5090	59,24
136	120	Furnas-Poços de Caldas	0,4360	4,3000	66,60
136	120	Furnas-Poços de Caldas	0,4360	4,3000	66,60
136	131	Furnas-M.Moraes	0,3480	3,4200	52,80
136	134	Furnas-L.C.Barreto	0,3750	4,1300	69,90
136	138	Furnas-Itutinga	0,6490	6,4600	100,80
136	138	Furnas-Itutinga	0,5580	6,1900	105,70
140	138	Adrianópolis-Itutinga	0,6520	6,5000	101,40
140	138	Adrianópolis-Itutinga	0,5580	6,1900	105,70
213	216	Marimbondo-P.Colômbia	0,2190	2,4200	40,70
216	396	P.Colômbia-V. Grande	0,1290	1,4140	23,77
217	216	Itumbiara-P.Colômbia	0,5650	6,2480	106,73
217	218	Itumbiara-Bandeirantes	0,5070	5,6100	95,60
217	218	Itumbiara-Bandeirantes	0,5070	5,6100	95,60
218	234	Bandeirantes-Samambaia	0,4300	4,7990	82,20
218	234	Bandeirantes-Samambaia	0,4300	4,7990	82,20
219	234	Brasília Sul-Samambaia	0,0350	0,4330	7,34
219	234	Brasília Sul-Samambaia	0,0350	0,4330	7,34
220	217	Corumbá-Itumbiara	0,2260	2,3960	43,24
220	219	Corumbá-Brasília Sul	0,7260	7,7040	138,01
225	231	Itumbiara-Rio Verde	4,1000	19,7600	36,08

Dados da Linha			Sequência Positiva		Susceptância Shunt (Mvar)
De Barra	Para Barra	Nome	Resistência Equivalente (%)	Reatância Equivalente (%)	
225	231	Itumbiara-Rio Verde	1,2700	13,6200	49,47
231	4501	Rio Verde-Barra do Peixe	4,5100	21,6900	40,25
231	4501	Rio Verde-Barra do Peixe	1,4900	16,0900	55,40
326	134	Jaguara-L.C.Barreto	0,0700	0,7600	12,29
326	396	Jaguara-V.Grande	0,2400	2,7400	45,47
106	104	Adrianópolis-C.Paulista	0,1520	2,3900	202,70
106	104	Adrianópolis-C.Paulista	0,1520	2,3900	203,10
123	120	Campinas-P.Caldas	0,3590	3,9450	66,68
4501	4522	B.do Peixe-Rondonópolis	3,7600	20,6800	35,66
4501	4522	B.do.Peixe-Rondonópolis	1,6400	12,4600	61,50
4522	4521	Rondonópolis-Itiquira	1,5300	7,6000	14,25
4522	4532	Rondonópolis-Coxipó	3,2500	17,9200	32,75
4522	4532	Rondonópolis-Coxipó	3,2500	17,9200	32,75
4532	4542	Coxipó-Nobres	1,6200	9,6800	19,15
4542	4552	Nobres-N.Mutum	1,8300	10,9300	18,60
4552	4572	N.Mutum-Lucas R.Verde	1,4000	8,3800	17,00
4562	4572	Sorriso-Lucas R.Verde	0,9400	5,5900	10,64
4562	4582	Sorriso-Sinop	1,2400	7,3800	13,28
4592	4542	Manso-Nobres	1,0000	6,1700	12,60
4623	4533	Rondonópolis-Coxipó	17,0600	45,5000	11,39
4703	4533	Cuiabá-Coxipó	0,9000	2,3100	0,58
4703	4533	Cuiabá-Coxipó	0,9000	2,3100	0,58
4805	4807	Guaporé-Jauru	3,0890	8,1340	2,09
4805	4807	Guaporé-Jauru	3,0890	8,1340	2,09
4862	4532	Jauru-Coxipó	2,5700	23,6800	97,42
4862	4532	Jauru-Coxipó	2,5700	23,6800	97,42

Os dados elétricos dos transformadores do sistema são mostrados na **Tabela A-3**.

Tabela A-3: Dados elétricos dos transformadores do sistema-teste Sul-Sudeste-Mato Grosso de 107 barras.

Dados da Unidade Transformadora					Sequência Positiva		Tap min	
De Barra	Para Barra	Nome	Relação de Transformação	Potência Nominal (MVA)	Resistência Equivalente (%)	Reatância Equivalente (%)	Min	Max
895	814	Bateias	500/230	600	0,0320	1,1460	0,900	1,100
895	814	Bateias	500/230	600	0,0300	1,1651	0,900	1,100
800	824	G.B.Munhoz	16/500	465	0,0000	3,3600	0,950	1,050
800	824	G.B.Munhoz	16/500	465	0,0000	3,3600	0,950	1,050

Dados da Unidade Transformadora					Sequência Positiva		Tap min	
De Barra	Para Barra	Nome	Relação de Transformação	Potência Nominal (MVA)	Resistência Equivalente (%)	Reatância Equivalente (%)	Min	Max
800	824	G.B.Munhoz	16/500	465	0,0000	3,3600	0,950	1,050
800	824	G.B.Munhoz	16/500	465	0,0000	3,3600	0,950	1,050
839	840	Cascavel	230/138	150	0,0000	6,6400	0,881	1,136
839	840	Cascavel	230/138	150	0,0000	6,2900	0,881	1,136
810	856	Salto Segredo	13,8/500	333	0,0000	4,2000	0,950	1,050
810	856	Salto Segredo	13,8/500	333	0,0000	4,2000	0,950	1,050
810	856	Salto Segredo	13,8/500	333	0,0000	4,2000	0,950	1,050
810	856	Salto Segredo	13,8/500	333	0,0000	4,2000	0,950	1,050
897	808	Salto Caxias	13,8/500	345	0,0000	4,0800	0,950	1,050
897	808	Salto Caxias	13,8/500	345	0,0000	4,0800	0,950	1,050
897	808	Salto Caxias	13,8/500	345	0,0000	4,0800	0,950	1,050
897	808	Salto Caxias	13,8/500	345	0,0000	4,0800	0,950	1,050
898	848	Foz do Chopin	230/138	150	0,0000	6,3600	0,881	1,136
933	934	Areia	500/230	672	0,0310	1,2070	0,900	1,100
938	939	Blumenau	500/230	672	0,0310	1,1500	0,900	1,100
938	939	Blumenau	500/230	672	0,0320	1,1630	0,900	1,100
938	939	Blumenau	500/230	672	0,0000	1,2770	0,900	1,100
959	960	Curitiba	500/230	672	0,0320	1,1630	0,900	1,100
959	960	Curitiba	500/230	672	0,0310	1,1660	0,900	1,100
964	965	Caxias	500/230	672	0,0200	1,2110	0,900	1,100
964	965	Caxias	500/230	672	0,0200	1,2330	0,900	1,100
904	995	Itá	16/500	305	0,0500	4,6150	0,950	1,050
904	995	Itá	16/500	305	0,0500	4,6150	0,950	1,050
904	995	Itá	16/500	305	0,0500	4,6150	0,950	1,050
904	995	Itá	16/500	305	0,0500	4,6150	0,950	1,050
904	995	Itá	16/500	305	0,0500	4,6150	0,950	1,050
915	1030	Machadinho	16/500	420	0,0000	4,1310	0,950	1,050
915	1030	Machadinho	16/500	420	0,0000	4,1310	0,950	1,050
915	1030	Machadinho	16/500	420	0,0000	4,1310	0,950	1,050
919	1047	Salto Osório	13,8/230	196	0,0800	6,8090	0,950	1,050
919	1047	Salto Osório	13,8/230	196	0,0800	6,8090	0,950	1,050
919	1047	Salto Osório	13,8/230	196	0,0800	6,8090	0,950	1,050
919	1047	Salto Osório	13,8/230	196	0,0800	6,8090	0,950	1,050
925	1060	S.Santiago	19/500	415	0,0400	4,5450	0,950	1,050
925	1060	S.Santiago	19/500	415	0,0400	4,5450	0,950	1,050
925	1060	S.Santiago	19/500	415	0,0400	4,5450	0,950	1,050
925	1060	S.Santiago	19/500	415	0,0400	4,5450	0,950	1,050
976	1210	Gravataí	500/230	672	0,0300	1,2190	0,900	1,100
976	1210	Gravataí	500/230	672	0,0390	1,1380	0,900	1,100
976	1210	Gravataí	500/230	672	0,0360	1,2170	0,900	1,100
896	2458	Cascavel Oeste	500/230	600	0,0000	1,2700	0,900	1,100

Dados da Unidade Transformadora					Sequência Positiva		Tap min	
De Barra	Para Barra	Nome	Relação de Transformação	Potência Nominal (MVA)	Resistência Equivalente (%)	Reatância Equivalente (%)	Min	Max
103	123	Campinas	500/345	560	0,0000	2,4190	0,950	1,110
102	120	P. de Caldas	500/345	560	0,0000	2,4030	0,950	1,110
1503	1504	Itajubá	500/138	300	0,0000	5,2000	0,950	1,100
100	213	Marimbondo	500/345	560	0,0000	2,3570	0,950	1,110
325	326	Jaguara	500/345	400	0,0000	2,1600	0,950	1,110
325	326	Jaguara	500/345	400	0,0000	2,1600	0,950	1,110
210	217	Itumbiara	500/345	560	0,0000	1,7200	0,950	1,110
210	217	Itumbiara	500/345	560	0,0000	1,7200	0,950	1,110
233	234	Samambaia	500/345	1050	0,0000	1,1130	0,950	1,110
233	234	Samambaia	500/345	1050	0,0000	1,0000	0,950	1,110
20	100	Marimbondo	13,8/500	190	0,0000	6,3200	1,000	1,100
20	100	Marimbondo	13,8/500	190	0,0000	6,3200	1,000	1,100
20	100	Marimbondo	13,8/500	190	0,0000	6,3200	1,000	1,100
20	100	Marimbondo	13,8/500	190	0,0000	6,3200	1,000	1,100
20	100	Marimbondo	13,8/500	190	0,0000	6,3200	1,000	1,100
20	100	Marimbondo	13,8/500	190	0,0000	6,3200	1,000	1,100
20	100	Marimbondo	13,8/500	190	0,0000	6,3200	1,000	1,100
20	100	Marimbondo	13,8/500	190	0,0000	6,3200	1,000	1,100
18	210	Itumbiara	13,8/500	400	0,0000	4,0000	1,000	1,100
18	210	Itumbiara	13,8/500	400	0,0000	4,0000	1,000	1,100
18	210	Itumbiara	13,8/500	400	0,0000	4,0000	1,000	1,100
18	210	Itumbiara	13,8/500	400	0,0000	4,0000	1,000	1,100
18	210	Itumbiara	13,8/500	400	0,0000	4,0000	1,000	1,100
18	210	Itumbiara	13,8/500	400	0,0000	4,0000	1,000	1,100
48	86	Ibiúna	13,8/500	350	0,0000	2,8590	0,900	1,100
48	86	Ibiúna	13,8/500	350	0,0000	2,8590	0,900	1,100
48	86	Ibiúna	13,8/500	350	0,0000	2,8590	0,900	1,100
48	86	Ibiúna	13,8/500	350	0,0000	2,8590	0,900	1,100
300	320	Emborcação	13,8/500	300	0,0000	4,0700	0,950	1,050
300	320	Emborcação	13,8/500	300	0,0000	4,0700	0,950	1,050
300	320	Emborcação	13,8/500	300	0,0000	4,0700	0,950	1,050
300	320	Emborcação	13,8/500	300	0,0000	4,0700	0,950	1,050
301	325	Jaguara	13,8/500	120	0,0000	10,5300	0,950	1,050
301	325	Jaguara	13,8/500	120	0,0000	10,5300	0,950	1,050
301	325	Jaguara	13,8/500	120	0,0000	10,5300	0,950	1,050
301	325	Jaguara	13,8/500	120	0,0000	10,5300	0,950	1,050
302	360	Nova Ponte	13,8/500	179	0,0000	5,8100	0,950	1,050
302	360	Nova Ponte	13,8/500	179	0,0000	5,8100	0,950	1,050
302	360	Nova Ponte	13,8/500	179	0,0000	5,8100	0,950	1,050
303	370	São Simão	13,8/500	290	0,0000	4,2300	0,950	1,050
303	370	São Simão	13,8/500	290	0,0000	4,2300	0,950	1,050

Dados da Unidade Transformadora					Sequência Positiva		Tap min	
De Barra	Para Barra	Nome	Relação de Transformação	Potência Nominal (MVA)	Resistência Equivalente (%)	Reatância Equivalente (%)	Min	Max
303	370	São Simão	13,8/500	290	0,0000	4,2300	0,950	1,050
303	370	São Simão	13,8/500	290	0,0000	4,2300	0,950	1,050
303	370	São Simão	13,8/500	290	0,0000	4,2300	0,950	1,050
303	370	São Simão	13,8/500	290	0,0000	4,2300	0,950	1,050
500	535	A. Vermelha	13,8/500	250	0,0000	4,1000	0,950	1,050
500	535	A. Vermelha	13,8/500	250	0,0000	4,1000	0,950	1,050
500	535	A. Vermelha	13,8/500	250	0,0000	4,1000	0,950	1,050
500	535	A. Vermelha	13,8/500	250	0,0000	4,1000	0,950	1,050
500	535	A. Vermelha	13,8/500	250	0,0000	4,1000	0,950	1,050
500	535	A. Vermelha	13,8/500	250	0,0000	4,1000	0,950	1,050
535	536	A. Vermelha	500/440	700	0,0000	1,5330	0,950	1,100
535	536	A. Vermelha	500/440	750	0,0000	1,4200	0,950	1,100
122	86	Ibiúna	500/345	750	0,0000	1,9130	0,950	1,110
122	86	Ibiúna	500/345	750	0,0000	1,9130	0,950	1,110
106	140	Adrianópolis	500/345	560	0,0000	2,9230	0,950	1,110
106	140	Adrianópolis	500/345	560	0,0000	2,6680	0,950	1,110
217	225	Itumbiara	345/230	225	0,0000	2,7210	0,950	1,110
217	225	Itumbiara	345/230	225	0,0000	2,9380	0,950	1,110
219	228	Brasília Sul	345/230	225	0,0000	3,5950	0,950	1,050
16	136	Furnas	15/345	160	0,0000	7,6800	0,999	1,046
16	136	Furnas	15/345	160	0,0000	7,6800	0,999	1,046
16	136	Furnas	15/345	160	0,0000	7,6800	0,999	1,046
16	136	Furnas	15/345	160	0,0000	7,6800	0,999	1,046
16	136	Furnas	15/345	160	0,0000	7,6800	0,999	1,046
16	136	Furnas	15/345	160	0,0000	7,6800	0,999	1,046
16	136	Furnas	15/345	160	0,0000	7,6800	0,999	1,046
16	136	Furnas	15/345	160	0,0000	7,6800	0,999	1,046
22	131	M. Moraes	13,8/345	63	0,0000	26,5000	0,972	1,083
22	131	M. Moraes	13,8/345	63	0,0000	26,5000	0,972	1,083
22	131	M. Moraes	13,8/345	63	0,0000	26,5000	0,972	1,083
22	131	M. Moraes	13,8/345	63	0,0000	26,5000	0,972	1,083
22	131	M. Moraes	13,8/345	63	0,0000	26,5000	0,972	1,083
22	131	M. Moraes	13,8/345	63	0,0000	26,5000	0,972	1,083
12	134	L.C.Barreto	13,8/345	184	0,0000	5,3400	0,999	1,046
12	134	L.C.Barreto	13,8/345	184	0,0000	5,3400	0,999	1,046
12	134	L.C.Barreto	13,8/345	184	0,0000	5,3400	0,999	1,046
12	134	L.C.Barreto	13,8/345	184	0,0000	5,3400	0,999	1,046
12	134	L.C.Barreto	13,8/345	184	0,0000	5,3400	0,999	1,046
12	134	L.C.Barreto	13,8/345	184	0,0000	5,3400	0,999	1,046
35	220	Corumbá	13,8/345	139	0,0000	8,9930	0,950	1,050
35	220	Corumbá	13,8/345	139	0,0000	8,9930	0,950	1,050

Dados da Unidade Transformadora					Sequência Positiva		Tap min	
De Barra	Para Barra	Nome	Relação de Transformação	Potência Nominal (MVA)	Resistência Equivalente (%)	Reatância Equivalente (%)	Min	Max
35	220	Corumbá	13,8/345	139	0,0000	8,9930	0,950	1,050
305	396	V.Grande	13,8/345	115	0,0000	8,8000	0,950	1,050
305	396	V.Grande	13,8/345	115	0,0000	8,8000	0,950	1,050
305	396	V.Grande	13,8/345	115	0,0000	8,8000	0,950	1,050
305	396	V.Grande	13,8/345	115	0,0000	8,8000	0,950	1,050
4522	4623	Rondonópolis	230/138	100	0,0000	7,9500	0,900	1,100
4522	4623	Rondonópolis	230/138	100	0,0000	7,9500	0,900	1,100
4532	4533	Coxipó	230/138	100	0,0000	8,6000	0,900	1,100
4532	4533	Coxipó	230/138	100	0,0000	8,6000	0,900	1,100
4532	4533	Coxipó	230/138	100	0,0000	8,6000	0,900	1,100
4862	4807	Jauru	230/138	300	0,0000	4,0500	0,900	1,100
21	4592	Manso	13,8/230	65	0,0000	19,2000	0,950	1,050
21	4592	Manso	13,8/230	65	0,0000	19,2000	0,950	1,050
21	4592	Manso	13,8/230	65	0,0000	19,2000	0,950	1,050
21	4592	Manso	13,8/230	65	0,0000	19,2000	0,950	1,050
4523	4521	Itiquira	13,8/230	40	0,0000	41,4200	0,950	1,050
4523	4521	Itiquira	13,8/230	40	0,0000	41,4200	0,950	1,050
4804	4805	Guaporé	13,8/230	48	0,0000	26,6670	0,950	1,050
4804	4805	Guaporé	13,8/230	48	0,0000	26,6670	0,950	1,050
4804	4805	Guaporé	13,8/230	48	0,0000	26,6670	0,950	1,050
4596	4533	Cuiabá	13,8/138	180	0,0000	7,5270	0,950	1,050
4596	4533	Cuiabá	13,8/138	180	0,0000	7,5270	0,950	1,050
4530	4532	Coxipó	13,8/230	100	0,0000	14,3000	0,950	1,050

Os dados elétricos das cargas do sistema são mostrados na Tabela A-4.

Tabela A-4: Dados elétricos das cargas do sistema-teste Sul-Sudeste-Mato Grosso de 107 barras.

Barra	Nome da Barra	Tensão (kV)	Carga	
			MW	Mvar
234	Samambaia	345	1000	350
217	Itumbiara	345	364	58
326	Jaguara	345	274	104
536	Água Vermelha	440	700	150
213	Marimbondo	345	93	39
123	Campinas	345	450	175
122	Ibiúna	500	200	38
120	Poços de Caldas	345	180	90
1504	Itajubá	138	145	63
104	Cachoeira Paulista	500	910	235

Barra	Nome da Barra	Tensão (kV)	Carga	
			MW	Mvar
140	Adrianópolis	345	700	250
228	Brasília Sul	230	86	34
86	Ibiúna	345	66	1,2
126	Guarulhos	345	290	95
218	Bandeirantes	345	600	200
216	Porto Colômbia	345	53	25
138	Itutinga	345	72	34
136	Furnas	345	54	23
814	Bateias	230	735,4	191
960	Curitiba	230	844,7	469,1
939	Blumenau	230	1149	53,1
965	Caxias	230	755,6	56,2
1210	Gravataí	230	1228	425
934	Areia	230	237	59
2458	Cascavel do Oeste	230	403	126
840	Cascavel	138	159	36
848	Foz do Chopin	138	94	18
834	São Mateus	230	13,4	4,2
1015	Joinville	230	70	2
231	Rio Verde	230	89,7	31,9
4501	Barra do Peixe	230	31,4	7,1
4623	Rondonópolis	138	128,2	40,8
4533	Coxipó	138	75,4	16,1
4703	Cuiabá	138	182,1	29,8
4807	Jauru	138	128,9	36,3
4552	Nova Mutum	230	12,6	1,2
4572	Lucas do Rio Verde	230	18	6,4
4562	Sorriso	230	23,8	7,4
4582	Sinop	230	65,5	16,7

Os dados elétricos dos geradores do sistema são mostrados na Tabela A-5.

Tabela A-5: Dados elétricos dos geradores do sistema-teste Sul-Sudeste-Mato Grosso de 107 barras.

Barra	Nome	Geração de Potência Ativa Máxima Total (MW)	Geração de Potência Reativa Máxima Total (Mvar)	Absorção de Potência Reativa Máxima Total (Mvar)
12	Luiz Carlos Barreto	1 104	420	540
16	Furnas	1 312	480	720
18	Itumbiara	2 280	600	546
20	Marimbondo	1 488	640	640

Barra	Nome	Geração de Potência Ativa Máxima Total (MW)	Geração de Potência Reativa Máxima Total (Mvar)	Absorção de Potência Reativa Máxima Total (Mvar)
21	Manso	216	84	80
22	Mascarenhas de Moraes	324	126	120
35	Corumbá	381	180	180
300	Emborcação	1 192	392	440
301	Jaguara	400	140	140
302	Nova Ponte	510	150	150
303	São Simão	1 680	600	600
305	Volta Grande	380	120	120
500	Água Vermelha	1 396,2	540	540
800	Gov. Bento Munhoz	1 674	800	800
808	Salto Caxias	1 240	600	600
810	Salto Segredo	1 260	532	400
904	Itá	1 450	475	475
915	Machadinho	1 140	465	516
919	Salto Osório	728	220	148
925	Salto Santiago	1 420	420	440
4523	Itiquira	60,8	30,4	42,2
4596	Cuiabá	320	160	160
4804	Guaporé	124,2	59,4	86,4

O diagrama unifilar do sistema é mostrado na Figura A-1.

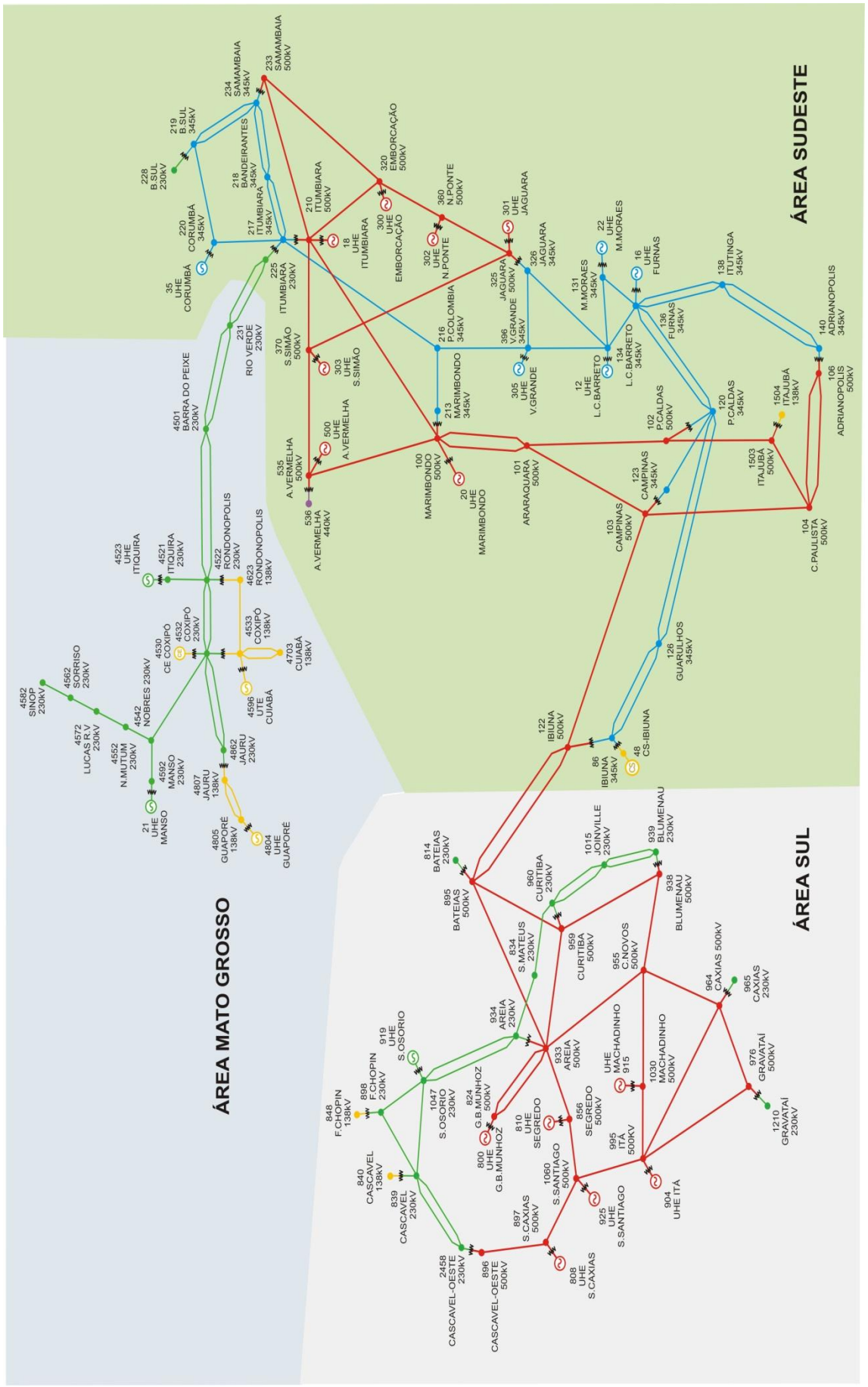

Figura A-1: Diagrama Unifilar para o sistema-teste Sul-Sudeste-Mato Grosso de 107 barras.

A-2 Ponto de Operação Proposto para o Sistema-Teste

A Tabela A-6 mostra os resultados obtidos para a solução do fluxo de potência, para um possível ponto de operação do sistema, através da ferramenta computacional ANAREDE desenvolvida pelo CEPEL. Os resultados obtidos consideram as posições dos tapes dos transformadores, bem como as impedâncias dos mesmos.

Tabela A-6: Dados de barra obtidos para o ponto de operação do sistema-teste através da ferramenta computacional ANAREDE.

Barra			Tensão		Geração		Carga		Shunt de Barra
Número	Nome	Tipo	Módulo [pu]	Ângulo [°]	MW	Mvar	MW	Mvar	Atual [pu]
12	LCBARRET-4GR	PV	1	-23,92	300	-202,6	0	0	0
16	FURNAS---5GR	PV	1	-25,95	800	-133,89	0	0	0
18	ITUMBIAR-6GR	Vθ	1,02	-23,77	995,76	-399,6	0	0	0
20	MARIMBON-5GR	PV	1,01	-22,13	900	-321,02	0	0	0
21	MANSO----3GR	PV	1	-62,09	140	-22,09	0	0	0
22	M.MOR.A--3GR	PV	1	-19,62	150	-20,57	0	0	0
35	CORUMBA--2GR	PV	1	-26,68	200	-49,63	0	0	0
48	IBIUNA---4CS	PV	1	-42,49	0	-461,07	0	0	0
86	IBIUNA---345	PQ	1,033	-42,49	0	0	66	1,2	0
100	MARIMBON-500	PQ	1,056	-28,25	0	0	0	0	0
101	ARARAQUA-500	PQ	1,069	-36,05	0	0	0	0	-228,47
102	POCOS----500	PQ	1,059	-42,72	0	0	0	0	-112,18
103	CAMPINAS-500	PQ	1,072	-43	0	0	0	0	0
104	C.PAULIS-500	PQ	1,061	-51,47	0	0	910	235	0
106	ADRIANO--500	PQ	1,05	-52,35	0	0	0	0	-110,17
120	P.CALDAS-345	PQ	1,041	-40,99	0	0	180	90	0
122	IBIUNA---500	PQ	1,067	-41,42	0	0	200	38	0
123	CAMPINAS-345	PQ	1,035	-45,78	0	0	450	175	0
126	GUARULHOS345	PQ	1,037	-43,23	0	0	290	95	0
131	M.MORAES-345	PQ	1,027	-27,04	0	0	0	0	0
134	LBARRETO-345	PQ	1,027	-26,16	0	0	0	0	0
136	FURNAS---345	PQ	1,028	-32,81	0	0	54	23	0
138	ITUTINGA-345	PQ	1,036	-43,87	0	0	72	34	0
140	ADRIANO--345	PQ	1,023	-53,46	0	0	700	250	0
210	ITUMBIARA500	PQ	1,048	-27,33	0	0	0	0	0
213	MARIMBON-345	PQ	1,05	-28,52	0	0	93	39	0
216	PCOLOMBIA345	PQ	1,049	-27,61	0	0	53	25	0
217	ITUMBIARA345	PQ	1,05	-32,03	0	0	364	58	0
218	BANDEIRA-345	PQ	1,025	-39,82	0	0	600	200	0
219	B.SUL----345	PQ	1,028	-38,69	0	0	0	0	0
220	CORUMBA--345	PQ	1,052	-31,71	0	0	0	0	0

Barra			Tensão		Geração		Carga		Shunt de Barra
Número	Nome	Tipo	Módulo [pu]	Ângulo [°]	MW	Mvar	MW	Mvar	Atual [pu]
225	ITUMBIARA230	PQ	1,008	-34,36	0	0	0	0	0
228	B.SUL----230	PQ	1,016	-40,38	0	0	86	34	0
231	R.VERDE-230	PQ	1,01	-49,08	0	0	89,7	31,9	0
233	SAMAMBAI-500	PQ	1,039	-35,97	0	0	0	0	0
234	SAMAMBAI-345	PQ	1,027	-38,78	0	0	1.000,00	350	0
300	EMBORCAC-3GR	PV	1,02	-18,73	700	-183,59	0	0	0
301	JAGUARA--4GR	PV	1,01	-19,17	300	-128,48	0	0	0
302	N.PONTE--3GR	PV	1,02	-18,05	400	-124,94	0	0	0
303	S.SIMAO--4GR	PV	1,02	-24,05	200	-279,14	0	0	0
305	V.GRANDE-4GR	PV	1	-21,89	300	-60,37	0	0	0
320	EMBORCAC-500	PQ	1,049	-23,82	0	0	0	0	0
325	JAGUARA--500	PQ	1,046	-23,46	0	0	0	0	0
326	JAGUARA--345	PQ	1,033	-25,69	0	0	274	104	0
360	NPONTE---500	PQ	1,047	-22,21	0	0	0	0	0
370	SSIMAO---500	PQ	1,049	-25,18	0	0	0	0	0
396	VGRANDE--345	PQ	1,041	-25,62	0	0	0	0	0
500	A.VERMEL-4GR	PV	1,02	-21,33	800	-118,07	0	0	0
535	AVERMELHA500	PQ	1,035	-25,78	0	0	0	0	0
536	AVERMELH-440	PQ	1,023	-28,58	0	0	700	150	0
800	GBMUNHOZ-2GR	PV	1,02	-6,89	1.100,00	138,38	0	0	0
808	SCAXIAS--4GR	PV	1,02	3,73	1.150,00	114,39	0	0	0
810	SSEGREDO-4GR	PV	1,02	-3,8	1.200,00	-72,2	0	0	0
814	BATEIAS--230	PQ	0,996	-37,31	0	0	735,4	191	0
824	GBMUNHOZ-500	PQ	1,038	-17,18	0	0	0	0	0
834	S.MATEUS-230	PQ	0,991	-28,56	0	0	13,4	4,2	0
839	CASCAVEL-230	PQ	1	-6,17	0	0	0	0	0
840	CASCAVEL-138	PQ	0,986	-9,16	0	0	159	36	0
848	FCHOPIM--138	PQ	0,999	-5,29	0	0	94	18	0
856	SEGREDO--500	PQ	1,035	-10,66	0	0	0	0	0
895	BATEIAS--500	PQ	1,044	-35,07	0	0	0	0	0
896	CASCAVELO500	PQ	1,028	-4,05	0	0	0	0	0
897	SCAXIAS--500	PQ	1,04	-2,78	0	0	0	0	0
898	FCHOPIM--230	PQ	1,012	-1,9	0	0	0	0	0
904	ITA------4GR	PV	1,02	-14,89	700	-236,4	0	0	0
915	MACHADIN-2GR	PV	1,02	-12,75	700	-109,43	0	0	0
919	SOSOR1A4-4GR	PV	1	5,98	700	89,06	0	0	0
925	SSANTIAG-3GR	PV	1,02	0,11	950	73,05	0	0	0
933	AREIA----500	PQ	1,038	-17,55	0	0	0	0	0
934	AREIA----230	PQ	0,998	-17,72	0	0	237	59	0
938	BLUMENAU-500	PQ	1,043	-37,11	0	0	0	0	0

Barra			Tensão		Geração		Carga		Shunt de Barra
Número	Nome	Tipo	Módulo [pu]	Ângulo [°]	MW	Mvar	MW	Mvar	Atual [pu]
939	BLUMENAU-230	PQ	0,996	-39,52	0	0	1.149,00	53,06	0
955	CNOVOS---500	PQ	1,058	-23,47	0	0	0	0	0
959	CURITIBA-500	PQ	1,033	-34,77	0	0	0	0	106,76
960	CURITIBA-230	PQ	0,996	-37,29	0	0	844,7	469,1	0
964	CAXIAS---500	PQ	1,037	-30,79	0	0	0	0	0
965	CAXIAS---230	PQ	1,003	-33,26	0	0	755,6	56,24	0
976	GRAVATAI-500	PQ	1,012	-33,47	0	0	0	0	0
995	ITA------500	PQ	1,05	-19,22	0	0	0	0	0
1015	JOINVILLE230	PQ	0,998	-39,47	0	0	70	2	0
1030	MACHADIN-500	PQ	1,052	-20,5	0	0	0	0	0
1047	SOSORIO--230	PQ	1,017	-0,92	0	0	0	0	0
1060	SSANTIAG-500	PQ	1,043	-7,85	0	0	0	0	0
1210	GRAVATAI-230	PQ	1,003	-36,22	0	0	1.228,00	425	0
1503	ITAJUBA--500	PQ	1,061	-49,28	0	0	0	0	0
1504	ITAJUBA--138	PQ	1,026	-53,25	0	0	145	63	0
2458	CASCAVEL-230	PQ	1,001	-6,4	0	0	403	126	0
4501	B.PEIXE--230	PQ	1,026	-60,51	0	0	31,4	7,1	-47,36
4521	ITIQUIRA-230	PQ	1,034	-66,19	0	0	0	0	0
4522	RONDONOP-230	PQ	1,032	-68,28	0	0	0	0	-21,3
4523	ITIQUIR--2GR	PV	1,01	-60,49	50	-9,08	0	0	0
4530	COXIPO-CE-12	PQ	1,02	-72,85	0	0	0	0	0
4532	COXIPO-230	PQ	1,041	-72,85	0	0	0	0	0
4533	COXIPO-138	PQ	1,015	-73,19	0	0	75,4	16,1	0
4542	NOBRES-230	PQ	1,025	-72,05	0	0	0	0	0
4552	N.MUTUM-230	PQ	1,007	-79,65	0	0	12,6	1,2	-20,28
4562	SORRISO-230	PQ	1,012	-87,94	0	0	23,8	7,4	0
4572	LUCAS-RV230	PQ	1,009	-84,99	0	0	18	6,4	0
4582	SINOP-230	PQ	1,018	-90,77	0	0	65,5	16,7	31,09
4592	MANSO-230	PQ	1,018	-67,14	0	0	0	0	0
4596	CBA--GAS-2GR	PV	1	-68,29	230	-28,66	0	0	0
4623	RONDONOP-138	PQ	1,018	-71,22	0	0	128,2	40,76	0
4703	CUIABA-138	PQ	1,003	-74,3	0	0	182,1	29,75	0
4804	GUAPORE--2GR	PV	1	-74,63	50	-16,77	0	0	0
4805	GUAPORE--138	PQ	1,025	-78,36	0	0	0	0	0
4807	JAURU-138	PQ	1,025	-79,62	0	0	128,9	36,3	0
4862	JAURU-230	PQ	1,046	-77,91	0	0	0	0	-32,83

Tabela A-7: Dados de linha obtidos para o ponto de operação do sistema-teste através da ferramenta computacional ANAREDE.

DA Barra	PARA Barra	Circuito	Tap	Fluxo		Potência	Corrente	Carregamento	Perdas
Número	Número	Número	(pu)	MW	Mvar	MVA	kA	Percentual	MW
86	122	1	1	-107,51	-183,48	212,66	118,86	29,29	0
86	122	2	1	-107,51	-183,48	212,66	118,86	29,29	0
100	20	1	1	-900	434,15	999,24	546,22	65,74	0
100	101	1	-	560,58	-175,06	587,28	321,02	35,27	4,88
100	101	2	-	564,72	-174,79	591,15	323,14	35,5	4,92
100	210	1	-	-58	-108,4	122,94	67,2	10,5	0,08
100	213	1	1	21,73	28,18	35,59	19,45	6,36	0
100	535	1	-	-189,03	-4,08	189,07	103,35	16,93	0,65
101	102	1	-	536,72	-80,05	542,65	293,13	32,59	3,95
101	103	1	-	578,78	-131,99	593,64	320,67	35,65	4,46
102	120	1	1	-138,72	80,3	160,29	87,37	28,62	0
102	1503	1	-	671,48	-98,86	678,72	369,97	40,76	4,42
103	123	1	1	222,82	169,41	279,91	150,74	49,98	0
104	103	1	-	-538,15	-112,3	549,74	299,08	33,28	5,06
104	1503	1	-	-520,84	11,04	520,96	283,42	31,63	1,22
106	104	1	-	-74,39	-157,38	174,08	95,75	10,46	0,11
106	104	2	-	-74,39	-157,6	174,28	95,86	10,47	0,11
106	140	1	1	71	97,74	120,8	66,45	21,57	0
106	140	2	1	77,78	107,08	132,35	72,8	23,63	0
86	48	1	1	0	476,27	476,27	266,2	45,36	0
122	103	1	-	192,07	-120,12	226,53	122,56	13,61	0,36
123	120	1	-	-227,18	-22,08	228,25	127,31	38,62	1,74
126	86	1	-	-74,41	1,48	74,42	41,43	6,05	0,06
126	86	2	-	-74,49	1,52	74,5	41,47	6,06	0,06
126	120	1	-	-70,96	-48,72	86,08	47,91	14,43	0,28
126	120	2	-	-70,14	-49,28	85,72	47,72	14,37	0,28
131	22	1	1	-150	40,82	155,45	87,41	41,13	0
134	12	1	0,999	-300	220,1	372,08	209,21	32,75	0
134	131	1	-	158,86	-21,68	160,34	90,15	22,36	0,22
134	396	1	-	-32,17	-69,17	76,28	42,89	10,91	0,08
136	16	1	1	-800	234,94	833,79	468,3	65,14	0
136	120	1	-	349,77	-77,56	358,26	201,22	59,91	5,12
136	120	2	-	349,77	-77,56	358,26	201,22	59,91	5,12
136	131	1	-	-305,49	22,41	306,31	172,04	43,55	3,16
136	134	1	-	-291,94	9,62	292,1	164,06	50,02	3,1

DA Barra	PARA Barra	Circuito	Tap	Fluxo		Potência	Corrente	Carregamento	Perdas
Número	Número	Número	(pu)	MW	Mvar	MVA	kA	Percentual	MW
136	138	1	-	314,85	-67,5	322,01	180,86	44,23	6,1
136	138	2	-	329,04	-67,37	335,87	188,64	43,85	5,72
140	138	1	-	-268,61	-24,82	269,75	152,3	37,87	4,55
140	138	2	-	-282,61	-28,48	284,04	160,37	37,75	4,3
210	18	1	1	-995,76	473,37	1.102,55	607,32	45,94	0
210	217	1	1	524,47	9,08	524,55	288,94	93,85	0
210	217	2	1	524,47	9,08	524,55	288,94	93,85	0
210	370	1	-	-177,28	-97,96	202,54	111,57	12,6	0,42
213	216	1	-	-71,27	-11,08	72,13	39,66	13,12	0,1
216	396	1	-	-259,7	75,4	270,43	148,85	39,52	0,88
217	216	1	-	-134,39	-39,32	140,02	76,98	21,11	0,94
217	218	1	-	263,54	-11,13	263,78	145,02	34,74	3,27
217	218	2	-	263,54	-11,13	263,78	145,02	34,74	3,27
218	234	1	-	-39,73	-44,43	59,6	33,58	9,33	0,06
218	234	2	-	-39,73	-44,43	59,6	33,58	9,33	0,06
219	234	1	-	42,56	20,5	47,24	26,52	7,98	0,01
219	234	2	-	42,56	20,5	47,24	26,52	7,98	0,01
220	35	1	1,025	-200	68,73	211,48	116,07	50,71	0
220	217	1	-	26,86	-18,74	32,76	17,98	5,15	0,02
220	219	1	-	173,14	-49,99	180,21	98,91	24,55	2,01
225	217	1	0,955	-165,66	25,32	167,59	95,96	74,48	0
225	217	2	0,955	-153,43	23,45	155,21	88,87	27,72	0
225	231	1	-	128,68	-29,06	131,91	75,53	66,96	6,72
225	231	2	-	190,41	-19,71	191,43	109,61	97,17	4,53
228	219	1	1	-86	-34	92,48	52,56	41,61	0
231	4501	1	-	91,24	-37,39	98,6	56,36	50,05	3,81
231	4501	2	-	126,89	-37,16	132,22	75,57	67,12	2,36
233	210	1	-	-407,39	-155,66	436,11	242,29	17,14	4,34
233	320	1	-	-587,09	-106,91	596,75	331,54	23,58	8,77
234	233	1	1	-470,65	-99,84	481,12	270,44	46,36	0
234	233	2	1	-523,83	-111,12	535,49	301	51,6	0
320	210	1	-	347,08	-91,9	359,04	197,69	18,43	1,37
320	300	1	1	-700	251,88	743,94	409,62	61,99	0
320	360	1	-	-242,94	-17,76	243,59	134,12	12,39	0,45
325	301	1	1	-300	155,97	338,12	186,56	67,62	0
325	326	1	1	194,36	68,24	205,99	113,65	51,5	0
325	326	2	1	194,36	68,24	205,99	113,65	51,5	0
325	360	1	-	-156,38	-53,98	165,44	91,28	7,69	0,22
325	370	1	-	67,66	-238,47	247,88	136,77	11,24	0,12

DA Barra	PARA Barra	Circuito	Tap	Fluxo		Potência	Corrente	Carregamento	Perdas
Número	Número	Número	(pu)	MW	Mvar	MVA	kA	Percentual	MW
326	134	1	-	121,88	68,43	139,78	78,11	16,94	0,13
326	396	1	-	-7,15	-52,7	53,18	29,72	8,54	0,02
360	302	1	1	-400	157,63	429,94	237,2	80,06	0
370	303	1	1	-200	291,12	353,2	194,37	20,3	0
370	535	1	-	89,85	40,48	98,55	54,23	8,32	0,16
396	305	1	1,025	-300	80,98	310,74	172,37	69,05	0
535	500	1	1	-800	182,5	820,55	457,73	54,7	0
536	535	2	1	-363,38	-77,87	371,63	209,75	50,13	0
814	895	1	0,965	-370,83	-95,89	383,03	222,11	64,62	0,44
814	895	2	0,965	-364,57	-95,11	376,77	218,48	63,56	0,4
824	800	1	1,024	-1.100,00	60,09	1.101,64	612,83	66,15	0
824	933	1	-	554,36	-30,51	555,2	308,85	25,44	0,29
824	933	2	-	545,64	-29,58	546,44	303,98	25,04	0,28
834	934	1	-	-140,24	24,41	142,35	82,96	40,86	5,2
839	840	1	1	77,35	21,81	80,36	46,42	53,58	0
839	840	2	1	81,65	23,02	84,83	49,01	56,56	0
839	898	1	-	-107,15	-2,65	107,18	61,91	57,39	1,3
839	1047	1	-	-120,74	-4,75	120,83	69,8	64,83	1,78
839	2458	1	-	33,04	-18,72	37,97	21,94	11,9	0,03
839	2458	2	-	35,85	-18,71	40,44	23,36	11,36	0,03
856	810	1	1	-1.200,00	218,06	1.219,65	680,46	96,8	0
856	933	1	-	1.963,40	-124,58	1.967,35	1.097,62	86,55	18,75
856	1060	1	-	-763,4	-93,48	769,1	429,09	35,25	3,06
895	122	1	-	306,32	-309,56	435,5	240,8	33,53	2,78
895	122	2	-	306,32	-309,56	435,5	240,8	33,53	2,78
896	897	1	-	-334,17	-177,02	378,16	212,4	23,1	0,62
897	808	1	1,024	-1.150,00	16,55	1.150,12	638,81	85,99	0
898	848	1	1	94	23,84	96,98	55,33	64,65	0
898	1047	1	-	-202,45	-21,77	203,62	116,18	63,1	0,61
933	895	1	-	1.284,21	-98,83	1.288,00	716,67	61,04	30,73
933	955	1	-	543,29	-250	598,06	332,77	28,34	4,64
933	959	1	-	1.187,89	-73,88	1.190,18	662,24	54,55	26,43
934	933	1	0,975	-28,65	-113,03	116,6	67,44	17,58	0,04
934	1047	1	-	-176,78	36,44	180,5	104,39	58,75	10,31
934	1047	2	-	-177,01	36,52	180,74	104,53	58,82	10,32
938	955	1	-	-878,61	-65,91	881,08	487,92	44,43	18,55
938	959	1	-	-267,64	-18,96	268,31	148,59	25,55	0,93

DA Barra	PARA Barra	Circuito	Tap	Fluxo		Potência	Corrente	Carregamento	Perdas
Número	Número	Número	(pu)	MW	Mvar	MVA	kA	Percentual	MW
939	938	1	0,959	-396,48	-9,38	396,59	229,8	59,19	0,45
939	938	2	0,959	-392,05	-9,05	392,15	227,23	58,53	0,46
939	938	3	0,959	-356,82	-18,07	357,28	207,02	53,33	0
939	1015	1	-	-1,83	-8,23	8,43	4,89	2,76	0
939	1015	2	-	-1,83	-8,33	8,53	4,94	2,79	0
955	964	1	-	602,07	-79,88	607,35	331,58	36,52	6,19
959	895	1	-	95,79	-293,04	308,3	172,27	14,61	0,38
960	834	1	-	-123,13	28,43	126,37	73,27	40,76	3,7
960	959	1	0,992	-398,5	-231,41	460,82	267,17	70,56	0,67
960	959	2	0,992	-397,28	-231,18	459,65	266,49	70,38	0,65
960	1015	1	-	36,98	-17,36	40,86	23,69	12,81	0,28
960	1015	2	-	37,23	-17,58	41,17	23,87	12,91	0,28
964	976	1	-	555,97	190,16	587,6	327,02	36,65	2,53
965	964	1	0,972	-381,2	-28,32	382,25	219,99	57,16	0,27
965	964	2	0,972	-374,4	-27,92	375,44	216,07	56,14	0,26
976	995	1	-	-676,57	-216,62	710,4	405,12	42,09	12,63
995	904	1	1	-700	296,93	760,37	418,2	46,79	0
995	964	1	-	724,14	-119,05	733,86	403,62	33,63	7,91
995	1030	1	-	265,11	-102,98	284,41	156,42	13,03	0,48
995	1060	1	-	-978,45	61	980,35	539,19	47,68	15,61
1030	915	1	1	-700	209,09	730,56	401,03	58,26	0
1030	955	1	-	964,63	-194,14	983,97	540,14	45,1	4,05
1047	919	1	1,025	-700	-4,3	700,01	397,47	89,55	0
1060	897	1	-	-810,52	56,35	812,48	449,56	34,76	4,69
1060	925	1	1,024	-950	59,14	951,84	526,68	67,96	0
1210	976	1	1,011	-399	-140,32	422,95	243,41	64,18	0,54
1210	976	2	1,011	-428,71	-146,13	452,93	260,66	68,73	0,81
1210	976	3	1,011	-400,29	-138,55	423,59	243,77	64,28	0,66
1503	1504	1	1	145	75,34	163,41	88,95	54,47	0
2458	896	1	0,994	-334,17	-159,83	370,43	213,72	63,03	0
4501	4522	1	-	67,29	-29,28	73,39	41,3	25,57	1,66
4501	4522	2	-	113,27	-44,41	121,67	68,47	50,91	2,02
4521	4523	1	1	-50	14,32	52,01	29,05	0,52	0
4522	4521	1	-	-49,64	0,93	49,64	27,78	26,01	0,36
4522	4532	1	-	45,6	-29,01	54,05	30,24	18,83	0,68
4522	4532	2	-	45,6	-29,01	54,05	30,24	18,83	0,68
4522	4623	1	1	67,66	20,24	70,62	39,51	70,62	0

DA Barra	PARA Barra	Circuito	Tap	Fluxo		Potência	Corrente	Carregamento	Perdas
Número	Número	Número	(pu)	MW	Mvar	MVA	kA	Percentual	MW
4522	4623	2	1	67,66	20,24	70,62	39,51	70,62	0
4532	4530	1	1	0	15,26	15,26	8,46	0,15	0
4532	4533	1	1	7,33	32,08	32,9	18,25	32,9	0
4532	4533	2	1	7,33	32,08	32,9	18,25	32,9	0
4532	4533	3	1	7,33	32,08	32,9	18,25	32,9	0
4532	4542	1	-	-12,23	8,62	14,96	8,3	20,76	0,08
4533	4596	1	1	-230	48,87	235,14	133,82	2,35	0
4542	4552	1	-	125,73	-5,34	125,84	70,86	83,89	2,76
4552	4572	1	-	110,37	-24,07	112,97	64,77	75,31	1,71
4562	4572	1	-	-89,87	17,29	91,52	52,22	62,43	0,79
4562	4582	1	-	66,07	-24,69	70,53	40,25	47,02	0,57
4592	21	1	1	-140	34,95	144,3	81,83	1,44	0
4592	4542	1	-	140	-34,95	144,3	81,83	60,37	1,97
4623	4533	1	-	7,12	-7,72	10,5	5,96	10,5	0,09
4703	4533	1	-	-91,05	-14,87	92,26	53,12	93,24	0,76
4703	4533	2	-	-91,05	-14,87	92,26	53,12	93,24	0,76
4805	4804	1	1	-50	20,48	54,03	30,45	39,15	0
4805	4807	1	-	25	-10,24	27,02	15,22	31,41	0,21
4805	4807	2	-	25	-10,24	27,02	15,22	31,41	0,21
4862	4532	1	-	-39,66	-44,93	59,93	33,08	12,62	0,39
4862	4532	2	-	-39,66	-44,93	59,93	33,08	12,62	0,39
4862	4807	1	1	79,32	57,03	97,69	53,92	32,56	0
536	535	1	1	-336,6	-72,13	344,24	194,29	49,76	0