

## **PWF.jl and ControlPowerFlow.jl: Julia Packages for Performing Power Flow using ANAREDE files**

Iago Sichinel Chávarry

Projeto de Graduação



## **PWF.jl and ControlPowerFlow.jl: Julia Packages for Performing Power Flow using ANAREDE files**

**Aluno(s): Iago Sichinel Chávarry**

**Orientador: Alexandre Street de Aguiar**

**Coorientador: Davi Michel Valladão**

Trabalho apresentado com requisito parcial à conclusão do curso de Engenharia Elétrica na Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil.

# PWF.jl and ControlPowerFlow.jl: Julia Packages for Performing Power Flow using ANAREDE files

## Abstract

Real large-scale electrical power systems offer several difficulties when performing network analysis. The AC power flow model is a non-convex and non-linear problem which is difficult to solve quickly and reliably. Additionally, real power system often suffers from voltage instability and are composed with several flexible AC transmission system (FACTS) devices and controllable components to accommodate the adversities and guarantee convergence. For that reason, market consolidated power system software's possesses features that handle the characteristics of large-scale networks that are not accounted in tradition power flow formulations.

In this context, Brazil is a country of continental dimension and relies on controllable components to handle electrical instability. In Brazilian electric systems representation, it's common to use these components to get more realistic results, handle bad voltage profiles and guarantee convergence. Examples of control actions in Brazilian load flow analysis are: (i) consideration of reactive generation limits and voltage profile limits, (ii) usage of shunt controller and load tap changer transformers (LTC) to handle voltage instability, and (iii) usage phase shifting control to handle active power flow in transmission lines.

The present report focusses on two contributions to the power system community: (i) a Julia Language package for parsing the Brazilian power system file (ANAREDE file) into Julia - **PWF.jl** - and (ii) a Julia package for performing power flow analysis with control action options - **ControlPowerFlow.jl**, which are both available as open-source projects. The practical validation of those packages is demonstrated in this work through case studies.

**Keywords:** Power Flow; Optimization; Julia Language

# PWF.jl e ControlPowerFlow.jl: Pacotes em Julia para Análise de Fluxo de Potência com Ações de Controle

## Resumo

Realizar estudos e análises em sistemas elétricos de potência em larga escala é comumente um trabalho complexo. O modelo de fluxo de potência CA é não-convexo e não-linear, que o torna extremamente complicado de se resolver de forma rápida e confiável. Adicionalmente, sistemas elétricos de larga escala frequentemente sofrem com baixos perfis de tensão e, consequentemente, possuem diversos componentes controláveis para acomodar essas adversidades e garantir convergência. Nesse contexto, softwares comerciais consolidados possuem funcionalidades que lidam com essas características típicas de redes reais e que não estão presentes na formulação tradicional do problema.

O Sistema de potência brasileiro possui dimensão continental e depende de elementos controláveis para lidar com instabilidade elétrica. Na representação do sistema brasileiro, é comum inserir esses elementos para se ter análises mais realistas, aumentar os níveis de tensão e garantir convergência. Exemplo de ações de controles utilizadas no fluxo de carga são: (i) considerar limites de geração reativas e limites de tensões, (ii) uso de chaveamento automático de reatores e controle do tap de transformadores e (iii) uso de transformadores defasadores para controlar a potência ativa em linhas de transmissão.

Dessa forma, o presente trabalho foca em duas contribuições para a comunidade de sistema de potência: (i) um pacote Julia, **PWF.jl**, para leitura e conversão dos dados de rede Brasileiros (arquivo ANAREDE/PWF) e (ii) um pacote Julia, **ControlPowerFlow.jl**, para realizar análises de fluxo de potência considerando ações de controle. Ambos os pacotes estão disponíveis para uso na linguagem Julia de forma aberta. As validações práticas dos pacotes são demonstrada através de estudos de caso.

**Palavras-Chave:** Fluxo de Potência, Otimização, Julia Language.

## Table of Contents

1.	<b>Introduction</b>	7
2.	<b>AC Power Flow Review</b>	9
2.1.	Branch Equations	9
2.1.1.	AC Transmission Line ( $\pi$ -circuit)	9
2.1.2.	Simplified DC Line Model	11
2.2.	Bus Equations	11
2.3.	Power Flow Formulation	12
2.4.	Newton Raphson's Method	14
2.5.	Convergency Issues	15
2.6.	Open-Source Software	15
3.	<b>Brazilian Power System</b>	16
3.1.	Controllable Devices	16
4.	<b>PWF.jl</b>	18
4.1.	ANAREDE File	18
4.2.	Parser	18
4.3.	Converter	19
4.4.	Example	20
4.5.	Case Study	23
5.	<b>ControlPowerFlow.jl</b>	24
5.1.	Control Actions	24
5.1.1.	Reactive Generation Limits (QLIM)	24
5.1.2.	Voltage Magnitude Limits (VLIM)	24
5.1.3.	Shunt Reactor Control (CSCA)	25
5.2.	Case Study	25
5.2.1.	Shunt Reactor Control (CSCA)	26
6.	<b>Conclusion</b>	28
	Bibliography	29
	Appendix	31
A.	Power Flow Equations	31

## List of Figures

Figure 1: Generic branch ( $\pi$ -circuit) model. ....	9
Figure 2: Two linked ‘dummy’ generator model simplification for DC lines. ....	11
Figure 3: Network elements connected to a bus. ....	12
Figure 4: Unidimensional newton method ....	15
Figure 5: Brazilian Electric System. ....	16
Figure 6: PWF.jl features. ....	18
Figure 7: Example of a PWF file ....	20
Figure 8: Reading PWF file by feature 1. ....	20
Figure 9: Julia Dictionary by feature 1 ....	20
Figure 10: Information of bus inside DBAR by feature 1 ....	21
Figure 11: Julia Dictionary by feature 2 ....	21
Figure 12: Information of bus inside “bus” section. ....	21
Figure 13: Information of gen inside “gen” section ....	22
Figure 14: information of load inside “load” section ....	22
Figure 15: Running power flow in PowerModels with PWF data. ....	22
Figure 16: IEEE 118 bus system. ....	23
Figure 17: Mismatch between ANAREDE results and PowerModels ....	23
Figure 18: Activation of control options QLIM and CSCA. ....	25
Figure 19: Running a control power flow in Julia. ....	26
Figure 20: Modified IEEE 24 bus system. ....	26
Figure 21: CSCA voltage magnitude results. ....	27
Figure 22: CSCA shunt susceptance results ....	27
Figure 23: Kirchoff Current Law at the output of the transformer from-bus side. ....	31
Figure 24: Transformer at from-bus ....	31
Figure 25: Kirchhoff’s current law at to-bus side ....	33
Figure 25: Kirchhoff’s current law at to-bus side ....	33

## 1. Introduction

The power flow (PF) analysis is an essential tool for operation and planning of power systems and has resulted in millions of dollars in savings annually [1]. It is a nonlinear and non-convex problem formulated to analyze the power system operating point at a specified frequency, therefore, it's a steady state representation of the actual power system [2].

Due to the established overall good convergence properties, high efficiency in computational speed and memory storage, the Newton-Rapson (N-R) method [3] is popularly used as the standard solving algorithm to AC power flow studies. However, traditional N-R method might not converge properly in practical and large-scale systems applications [4].

Another approach to perform AC PF analysis is through optimization techniques, which have been empowered by the recent advances in Nonlinear Programming (NLP) [5] and open source cutting edge solvers [6]. This approach is commonly used when solving Optimum Power Flow. The OPF problem seeks to optimize a given preference by controlling power flow in electrical network while obeying electrical power flow constraints or systems operating limits. Like conventional power flow, OPF determines the system's state of operation, however, it works with an under constrained (inequality) system thus having multiple solutions. The OPF problem, therefore, performs multiple power flow iterations, modifying the control variables to optimize a certain objective [7].

Despite the differences, power flow solved through N-R and optimization methods presents similar convergence issues when performing real case applications. First, the nonlinear power balance equations present alternative and instable solutions (low voltage) [8], and eventually, no solution [9]. Second, numerical difficulties when the operation point is close to operation limit (ill-conditioned systems) [10]. Third, the region of convergence for these solutions is fractal [11], [12], [13]. Finally, the initialization of variables plays a decisive role in good convergence, where "reasonable" initial guesses might be in region of convergence for low voltage or a local infeasible solution region [4], [7], [9], [14].

To improve convergence robustness of power flow analysis, several methodologies have been proposed. In traditional N-R algorithm, adjust the step size computation helps the iterative process to reach a desired solution more regularly [14], [15]. In optimization problems, robustness is usually achieved by conducting the problem to a feasible region. This can be obtained by enforcing control variables limits, such as generation and voltage magnitude, or by minimizing the distance of the control variables to the nominal value in the objective function [16].

Additionally, modern power systems must provide reliability to adverse conditions due to increasing role of demand response, integration of renewable sources and automation in fault detection [1]. These large-scale reliable systems comprehend several control components to accommodate and guarantee stability [17]. In this context, modern system engineers must consider these control variables when performing power flow analysis to get a more realistic result and avoid system collapses. The representation of these control variables is particularly determinant when analyzing system variations, such as contingency analysis or scaling down generation/load in peak cases, since the control variables are fixed and won't adapt to system variation [18].

In this context, the Brazilian Electric System is an example of a large-scale complex power system. The installed capacity of the Brazilian system is of over 150GW, the same level as the PJM system in the United States and the German electric grid [19]. Hydro based assets are the main generation of Brazil, with close to 66% of the total generation capacity in 2021 [20]. Since reservoirs are often far from major consumption centers, the Brazilian transmission grid must be capable of flowing large amount of energy through long distances. This is achieved by extremely high-voltages transmission lines and, more recently, the usage of direct current (HVDC) transmission [21]. To handle instabilities and complexities of the system, the official commercial Brazilian power flow software, ANAREDE - CEPEL, have implemented several control features to accommodate poor conditioned cases with bad voltage profiles [22]. Additionally, ANAREDE power cases format (".pwf") are quite different from other formats used in commercial and open-source softwares, making conversions between these files a nontrivial task. Therefore, the Brazilian electric agents and researchers are compelled to use ANAREDE in power flow analysis.

The present work aims to describe the implementation of a Julia package, **ControlPowerFlow.jl** [23], for performing control power flow, that is, incorporate additional control components in the procedure of power flow analysis to improve accuracy and convergence. In addition, the present work also describes the implementation of a Julia data converter, **PWF.jl** [24], from Brazilian data format (PWF or ANAREDE format) to perform power flow analysis. This document is organized as follows. Section 2 provides the formulation of the traditional AC power flow equations used in the implementation of the packages, and discussions on how to improve convergence. Section 3 describes the Brazilian power



system and controllable devices present in the system. In section 4, the implementation of the PWF Parser is described. Section 5 describes the ControlPowerFlow.jl package foundation and implementation. Finally, the conclusions are made in Section 6 and appendix A provide detailed information of the power flow equations.



## 2. AC Power Flow Review

Traditional electric power systems are commonly divided into three sub-systems operating at different voltage levels: generation (11 – 35kV), transmission ( $\geq 110kV$ ) and distribution (11kV – 230/400V) systems. The generation subsystem converts primary energy sources such as fossil fuels, wind and hydro to electric energy using synchronous generators. These generation units inject alternating current (AC) to a 3-phase transmission system at a constant voltage magnitude ( $V$ ) and frequency ( $f$ ), usually at 50/60Hz. To reduce power losses in transmission lines due to Joule's effect ( $\approx i^2 \cdot r$ ), the voltage magnitude of generation units is usually increased using step up transformers to transmission level. Another way of reducing power losses is by using HVDC transmission lines, which are only financially viable in long distance applications. Transmission system provides a network of interconnected lines and substations to enable reliable flow of electric power to large consumers, connected directly at this system level, and small consumers, supplied at the distribution level. The distribution system is usually operated at lower voltages, which requires step down transformers [2].

The power flow, or load flow, analysis consists essentially in determining an operating condition for an electrical power system. In the analysis, the subsystem in focus is usually the high voltage electric transmission system, which is the most crucial subsystem in the power supply chain. Therefore, the generation and distribution levels are usually represented in a simplified form. This section represents the power flow equation and formulation in a resumed form. For a step-by-step formulation visit Appendix A.

### 2.1.Branch Equations

A power transmission system is described by a directed graph  $G = (\mathcal{B}, \mathcal{L})$  consisting of a set of nodes  $\mathcal{B}$  and a set of branches  $\mathcal{L}$ , where each branch  $l \in \mathcal{L}$  represent a transmission line, a transformer or a DC line, and each node  $i \in \mathcal{B}$  a connection point between two or more buses. The set of AC branch elements is given by  $\mathcal{L}^{AC}$ , while the set of DC lines is given by  $\mathcal{L}^{DC}$ . At some buses power is injected into the network, while at others power is consumed by system loads. Power flow studies are predominantly considered in steady state operation, or sufficiently close that transitory effects can be discarded. For this matter, the three-phase transmission system is traditionally modelled as a balanced per phase equivalent system.

#### 2.1.1. AC Transmission Line ( $\pi$ -circuit)

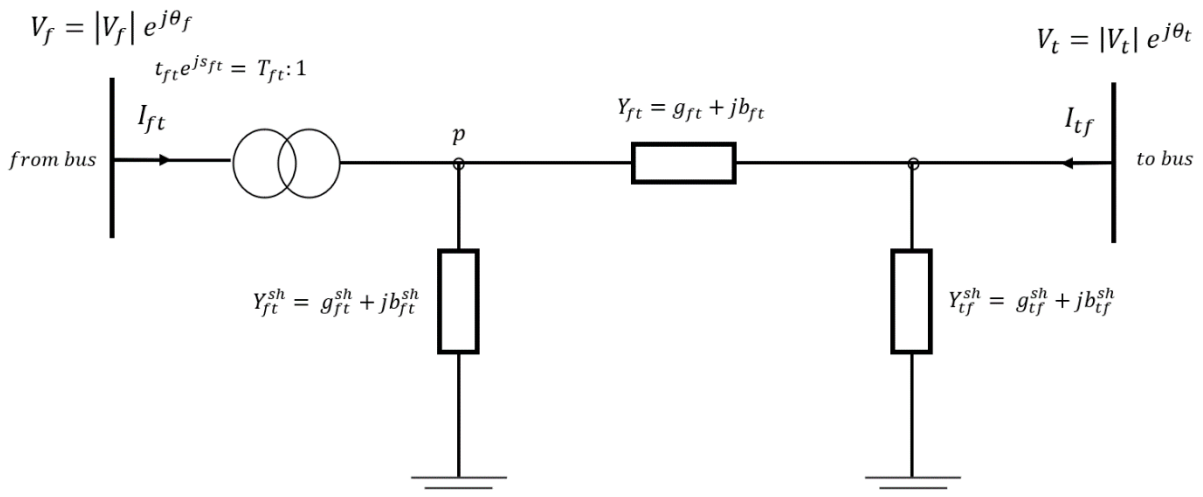


Figure 1: generic branch ( $\pi$ -circuit) model for: (i) transmission lines ( $t_{ft} = 0$  &  $s_{ft} = 0$ ); (ii) in-phase transformers ( $s_{ft} = 0$ ); (iii) phase-shifting transformers ( $s_{ft} \neq 0$ ). The transformer ratio is modelled at the from side of the bus.

Figure 2: generic branch ( $\pi$ -circuit) model for: (i) transmission lines ( $t_{ft} = 0$  &  $s_{ft} = 0$ ); (ii) in-phase transformers ( $s_{ft} = 0$ ); (iii) phase-shifting transformers ( $s_{ft} \neq 0$ ). The transformer ratio is modelled at the from side of the bus.

The basic properties<sup>1</sup> of transmission lines and transformers can be described by a generic branch ( $\pi$ -circuit) model illustrated in figure 1.  $f$  represents the from-side bus and  $t$  the to-side bus.  $T_{f,t}$  is the transformer ratio composed by  $t_{f,t}$ , tap ratio magnitude, and  $s_{f,t}$ , shift ratio angle;  $Y_{f,t} = Y_{t,f}$  is the line admittance composed by real  $g_{f,t}$ , line conductance, and imaginary  $b_{f,t}$ , line susceptance;  $Y_{f,t}^{sh}$  is the combination of line shunts and line charge admittance of transmission lines and it's composed by a real shunt conductance  $g_{f,t}^{sh}$  and a imaginary shunt susceptance  $b_{f,t}^{sh}$ . Note that  $Y_{f,t}^{sh}$  can be different from  $Y_{t,f}^{sh}$ .  $I_{f,l}$  and  $I_{t,f}$  represents the current injection from-bus and to-bus respectively. The  $V_f$  and  $V_t$  variables represent the voltage at bus  $f$  and  $t$  respectively and are composed by a magnitude  $|V_{t/f}|$  and an angle  $\theta_{f/f}$ . This voltage representation is called polar coordinates representation. Another equivalent formulation uses rectangular [25] representation of voltage variables.

From the  $\pi$  - circuit model it's possible to calculate the apparent flow equations for each branch of the system:

$$S_{f,t} = \frac{(Y_{f,t}^* + Y_{f,t}^{*sh})V_f V_f^*}{T_{f,t} T_{f,t}^*} - \frac{T_{f,t}^* Y_{f,t} V_f V_t^*}{T_{f,t} T_{f,t}^*}$$

Equation 1

$$S_{t,f} = (Y_{f,t}^* + Y_{t,f}^{*sh})V_t V_t^* - \frac{T_{f,t} Y_{f,t}^* V_t V_f^*}{T_{f,t} T_{f,t}^*}$$

Equation 2

Equation 1 and 2 can be separated into real and imaginary part<sup>2</sup>. The real part of the power is called **active power** and is given by:

$$p_{f,t} = \frac{(g_{f,t} + g_{f,t}^{sh})v_f^2}{t_{f,t}^2} + \frac{(-t_{f,t}^r g_{f,t} + t_{f,t}^i b_{f,t}) \cos(\theta_f - \theta_t) v_f v_t}{t_{f,t}^2} + \frac{(-t_{f,t}^r b_{f,t} - t_{f,t}^i g_{f,t}) \sin(\theta_f - \theta_t) v_f v_t}{t_{f,t}^2}$$

Equation 3

$$p_{t,f} = \frac{(g_{f,t} + g_{f,t}^{sh})v_t^2}{t_{f,t}^2} + \frac{(-t_{f,t}^r g_{f,t} - t_{f,t}^i b_{f,t}) \cos(\theta_t - \theta_f) v_t v_f}{t_{f,t}^2} + \frac{(-t_{f,t}^r b_{f,t} + t_{f,t}^i g_{f,t}) \sin(\theta_t - \theta_f) v_t v_f}{t_{f,t}^2}$$

Equation 4

The imaginary part of the power is called **reactive power** and is given by:

$$q_{f,t} = -\frac{(b_{f,t} + b_{f,t}^{sh})v_f^2}{t_{f,t}^2} - \frac{(-t_{f,t}^r b_{f,t} - t_{f,t}^i g_{f,t}) \cos(\theta_f - \theta_t) v_f v_t}{t_{f,t}^2} + \frac{(-t_{f,t}^r g_{f,t} + t_{f,t}^i b_{f,t}) \sin(\theta_f - \theta_t) v_f v_t}{t_{f,t}^2}$$

Equation 5

$$q_{t,f} = \frac{(b_{f,t} + b_{f,t}^{sh})v_t^2}{t_{f,t}^2} - \frac{(-t_{f,t}^r b_{f,t} + t_{f,t}^i g_{f,t}) \cos(\theta_t - \theta_f) v_t v_f}{t_{f,t}^2} + \frac{(-t_{f,t}^r g_{f,t} - t_{f,t}^i b_{f,t}) \sin(\theta_t - \theta_f) v_t v_f}{t_{f,t}^2}$$

Equation 6

Since there can be more than one branch element connecting the same two buses, the identification of a branch will be made by the addition of index  $l$ . Therefore, a generic branch is defined by the information of index  $l$ , a from-side bus  $f_{bus}(l)$  and a to-side bus  $t_{bus}(l)$ . For example, suppose branch  $l = 1$  has  $f_{bus}(1) = 1$  and  $t_{bus}(1) = 2$ . This means that branch 1 is connected between buses 1 (from-side) and 2 (to-side). Therefore, the active and reactive power flow is given by  $p_{1,1,2}$  and  $q_{1,1,2}$  flowing from-bus 1 to 2, and  $p_{1,2,1}$  and  $q_{1,2,1}$  flowing from bus 2 to 1. Note that the power flowing in direction from-to and to-from are different. The difference between them is called power loss.

<sup>1</sup> Variables in capital letters denote complex numbers and variables in lowercase represent real values.

<sup>2</sup> Detailed explanation can be obtained in Appendix A.

## 2.1.2. Simplified DC Line Model

To reduce costs and raise controllability, power system has been recently adopting HVDC transmission lines. HVDC lines work simplistically by rectifying AC power in one bus, transmitting DC power through the DC line and inverting this power in the other end. In our formulation, the DC transmission line will be approximated by the two linked “dummy” generator model [26]. This model illustrated by figure 2, where the dummy generators are connected in both ends of the DC link and the active power of the generators are specified. In one end, the generator consumes the power, representing the rectifier bus, and in the other the generation is injecting, representing the inverter side. The reactive power of these generators are free variables.

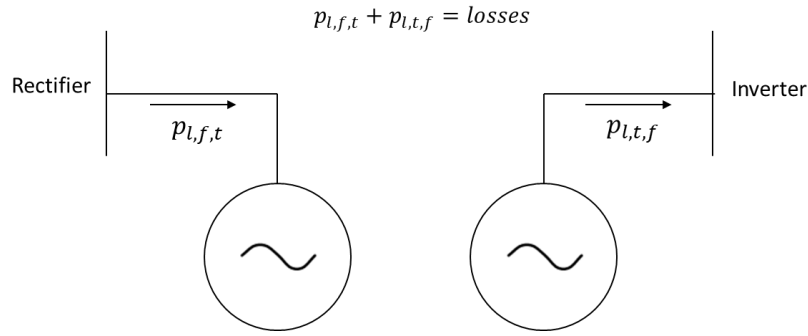


Figure 3: Two linked ‘dummy’ generator model simplification for DC lines.

## 2.2. Bus Equations

The formulation of power flow analysis consists in satisfying Kirchhoff’s current law for all buses of the system. That is, the power generated and consumed by the different elements connected at a bus must be equal to the sum of the power flowing in and out of the same bus. Figure 3 represents the different elements connected at the system buses that can generate or consume power.

The generating units are the main source of active and reactive power of the system while storage and shunt devices are used to balance and control voltage abnormal values. Load units represents the power consumption of the system. The notation for the element sets is given by:  $\mathcal{Gen}$  represents the generating units,  $\mathcal{S}$  represents the storage devices,  $\mathcal{BSh}$  represents the bus shunt elements and  $\mathcal{Load}$  the load units.

In our representation, multiple elements can be found in each bus. Therefore, the active and reactive balance of bus  $b$  is given by:

$$\sum_{l \in \mathcal{L} \mid t_{bus(l)=b} p_{l,f,t} - \sum_{l \in \mathcal{L} \mid f_{bus(l)=b} p_{l,f,t} = \sum_{i \in \mathcal{Gen} \mid bus(i)=b} p_i^g + \sum_{i \in \mathcal{S} \mid bus(i)=b} p_i^s + \sum_{i \in \mathcal{BSh} \mid bus(i)=b} g_i^{bsh} v_b^2 - \sum_{i \in \mathcal{Load} \mid bus(i)=b} p_b^d \quad \text{Equation 7}$$

$$\sum_{l \in \mathcal{L} \mid t_{bus(l)=b} q_{l,f,t} - \sum_{l \in \mathcal{L} \mid f_{bus(l)=b} q_{l,f,t} = \sum_{i \in \mathcal{Gen} \mid bus(i)=b} q_i^g + \sum_{i \in \mathcal{S} \mid bus(i)=b} q_i^s + \sum_{i \in \mathcal{BSh} \mid bus(i)=b} b_i^{bsh} v_b^2 - \sum_{i \in \mathcal{Load} \mid bus(i)=b} q_b^d \quad \text{Equation 8}$$

Equation 7 and 8 states that all power flowing into bus b minus all power flowing away of that bus must be equal to the sum of all generation, plus the sum of all storage injection, plus all power supplied by the reactors, minus all the consumption in that bus.

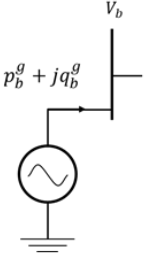
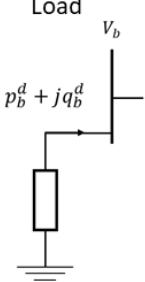
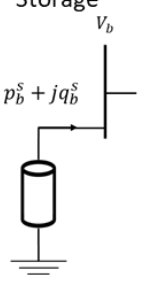
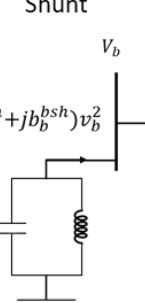
<p>Generation</p> 	<p>Injects electric power into the system by converting primary sources, such as oil, gas, coal, hydro, etc. Generating units usually control bus voltages magnitudes <math>v_b</math></p>
<p>Load</p> 	<p>Represents the electric power consumption. It can be by large consumers directly connected to the grid, or by total power consumption of local distributors.</p>
<p>Storage</p> 	<p>Represents the electric power consumption or generation by electric storage devices.</p>
<p>Shunt</p> 	<p>Represents shunt reactors bank or static compensators which are used to control abnormal bus voltages</p>

Figure 4: network elements connected to a bus, generator, load, storage devices and bus shunt devices.

## 2.3. Power Flow Formulation

The power flow formulation requires boundary conditions to solve the problem. These conditions represent part of the operating condition of the systems. In traditional power flow problems, each bus is classified on the prescribed boundary condition. Their definition is given by:

1. **Slack-Bus** (Reference Bus): voltage magnitude  $v_b$  and voltage angle  $\theta_b$  are specified at values  $v_b^{spec}$  and  $\theta_b^{spec}$ . Active power  $p_b$  and reactive power  $q_b$  are free variables. Slack buses are responsible to active and reactive balance for all system and are also responsible to supply the

power losses in the transmission lines. For that matter, they are usually connected to a large generation unit. The set notation of slack buses is  $\mathcal{B}^{ref}$

2. **PV-Bus** (Generation Bus): active power  $p_b$  and voltage magnitude  $v_b$  are specified at values  $p_b^{spec}$  and  $v_b^{spec}$ , voltage angle  $\theta_b$  and reactive power  $q_b$  are free variables. The set notation of slack buses is  $\mathcal{B}^{PV}$ . PV buses must have at least one active generation.
3. **PQ-Bus** (Load Bus): active power  $p_b$  and reactive power  $q_b$  are specified at values  $p_b^{spec}$  and  $q_b^{spec}$ . Voltage magnitude  $v_b$  and voltage angle  $\theta_b$  are free variables. The set notation of slack buses is  $\mathcal{B}^{PQ}$ . In our formulation, we assume that PQ buses don't have active generators.

Finally, the traditional power flow formulated as an optimization problem is given by:

$$\begin{aligned} \text{Min} \quad & 0 \\ \text{over} \quad & \theta, v, p, q \end{aligned}$$

s. t.

$$\sum_{l \in \mathcal{L} | t_{bus}(l) = b} p_{l,f,t} - \sum_{l \in \mathcal{L} | f_{bus}(l) = b} p_{l,f,t} = \sum_{i \in \mathcal{Gen} | bus(i) = b} p_i^g + \sum_{i \in \mathcal{S} | bus(i) = b} p_i^s + \sum_{i \in \mathcal{BSH} | bus(i) = b} g_i^{bsh} v_b^2 - \sum_{i \in \mathcal{Load} | bus(i) = b} p_b^d, \quad \forall b \in \mathcal{B} \quad (1)$$

$$\sum_{l \in \mathcal{L} | t_{bus}(l) = b} q_{l,f,t} - \sum_{l \in \mathcal{L} | f_{bus}(l) = b} q_{l,f,t} = \sum_{i \in \mathcal{Gen} | bus(i) = b} q_i^g + \sum_{i \in \mathcal{S} | bus(i) = b} q_i^s + \sum_{i \in \mathcal{BSH} | bus(i) = b} b_i^{bsh} v_b^2 - \sum_{i \in \mathcal{Load} | bus(i) = b} q_b^d, \quad \forall b \in \mathcal{B} \quad (2)$$

$$p_{l,f,t} = \frac{(g_{l,f,t} + g_{l,t,f}^{sh})v_f^2}{t_{l,f,t}^2} + \frac{(-t_{l,f,t}^r g_{l,f,t} + t_{l,t,f}^i b_{l,f,t}) \cos(\theta_f - \theta_t) v_f v_t}{t_{l,f,t}^2} + \frac{(-t_{l,f,t}^r b_{l,f,t} - t_{l,t,f}^i g_{l,t,f}) \sin(\theta_f - \theta_t) v_f v_t}{t_{l,f,t}^2}, \quad \forall l \in \mathcal{L}^{AC} : f = f_{bus}(l), t = t_{bus}(l) \quad (3)$$

$$p_{l,t,f} = -\frac{(b_{l,t,f} + b_{l,t,f}^{sh})v_f^2}{t_{l,t,f}^2} - \frac{(-t_{l,t,f}^r b_{l,t,f} - t_{l,t,f}^i g_{l,t,f}) \cos(\theta_f - \theta_t) v_f v_t}{t_{l,t,f}^2} + \frac{(-t_{l,t,f}^r g_{l,t,f} + t_{l,t,f}^i b_{l,t,f}) \sin(\theta_f - \theta_t) v_f v_t}{t_{l,t,f}^2}, \quad \forall l \in \mathcal{L}^{AC} : f = f_{bus}(l), t = t_{bus}(l) \quad (4)$$

$$q_{l,t,f} = \frac{(g_{l,t,f} + g_{l,t,f}^{sh})v_t^2}{t_{l,t,f}^2} + \frac{(-t_{l,t,f}^r g_{l,t,f} - t_{l,t,f}^i b_{l,t,f}) \cos(\theta_t - \theta_f) v_t v_f}{t_{l,t,f}^2} + \frac{(-t_{l,t,f}^r b_{l,t,f} + t_{l,t,f}^i g_{l,t,f}) \sin(\theta_t - \theta_f) v_t v_f}{t_{l,t,f}^2}, \quad \forall l \in \mathcal{L}^{AC} : f = f_{bus}(l), t = t_{bus}(l) \quad (5)$$

$$q_{l,t,f} = \frac{(b_{l,t,f} + b_{l,t,f}^{sh})v_t^2}{t_{l,t,f}^2} - \frac{(-t_{l,t,f}^r b_{l,t,f} + t_{l,t,f}^i g_{l,t,f}) \cos(\theta_t - \theta_f) v_t v_f}{t_{l,t,f}^2} + \frac{(-t_{l,t,f}^r g_{l,t,f} - t_{l,t,f}^i b_{l,t,f}) \sin(\theta_t - \theta_f) v_t v_f}{t_{l,t,f}^2}, \quad \forall l \in \mathcal{L}^{AC} : f = f_{bus}(l), t = t_{bus}(l) \quad (6)$$

$$\theta_b = \theta_b^{spec}, \quad \forall b \in \mathcal{B}^{ref} \quad (7)$$

$$v_b = v_b^{spec}, \quad \forall b \in \mathcal{B}^{ref} \cup \mathcal{B}^{PV} \quad (8)$$

$$p_b^g = p_b^{spec}, \quad \forall b \in \mathcal{B}^{PV}, i \in \mathcal{Gen} | bus(i) = b \quad (9)$$

The objective function of the power flow problem is null, i.e., the optimization problem only needs to find a feasible solution from the nonlinear feasible space. The decision variables of the problem are voltage angles and magnitude,  $\theta$  and  $v$ , the active and reactive power generation,  $p_g$  and  $q_g$ , and the power flow on both AC and DC transmission lines  $p$  and  $q$ . Constraints (1) and (2) represent the active and reactive power balance for all buses of the system. Constraints (3) - (6) yields the power flow for every branch of the system for both directions. This is necessary because the power flow in each direction of the branch is different. Constraints (7) - (11) deals with the boundary conditions of the system. The resulting problem is a nonlinear, nonconvex optimization model, which are usually effectively handled effective by nonlinear solvers [3].

Another equivalent formulation of the power flow can be done in a concise form. Let  $\mathcal{P}_b(\underline{v}, \underline{\theta}, \underline{\psi}^{fix})$  and  $\mathcal{Q}_b(\underline{v}, \underline{\theta}, \underline{\psi}^{fix})$  be the active and reactive power injection (generation or consumption) of a bus  $b$ .  $\mathcal{P}$  and  $\mathcal{Q}$  are functions of system voltages  $\underline{v}$ , angles  $\underline{\theta}$  and fixed branch parameters  $\underline{\psi}^{fix}$ . Since the problem have boundary conditions, an equivalent way of formulation the problem is by enforcing that all those conditions are respected. That is:

$$\Delta \mathcal{P}_b = \mathcal{P}_b(\underline{v}, \underline{\theta}, \underline{\psi}^{fix}) - \mathcal{P}_b^{esp} = 0 \quad \forall b \in \mathcal{B}^{PV} \cup \mathcal{B}^{PQ}$$

Equation 9

$$\Delta \mathcal{Q}_b = \mathcal{Q}_b(\underline{v}, \underline{\theta}, \underline{\psi}^{fix}) - \mathcal{Q}_b^{esp} = 0 \quad \forall b \in \mathcal{B}^{PQ}$$

Equation 10

Where  $\Delta \mathcal{P}_b$  and  $\Delta \mathcal{Q}_b$  are called active/reactive power mismatch and should be zero in a converged solution.

The free variables in the problem are  $n^{PV} + n^{PQ}$  variables for voltage angles ( $\theta$ ) and  $n^{PQ}$  variables for voltage magnitudes ( $v$ ), resulting in a total of  $2n^{PQ} + n^{PV}$  variables. The number of restrictions due to active and reactive power injections are also  $2n^{PQ} + n^{PV}$ . Therefore, equations 9 and 10 are sufficient to formulate the power flow problem in a mathematical point of view. These power mismatches variables can be rearranged in a vectorial form:

$$\underline{f}(\underline{x}) = \begin{bmatrix} \Delta \mathcal{P} \\ \Delta \mathcal{Q} \end{bmatrix} = 0$$

Therefore, the power flow problem can be fit into the form,  $\underline{f}(\underline{x}) = 0$ , where  $\underline{f}(\underline{x})$  is a system of nonlinear algebraic equations, which many algorithms can effectively handle.

## 2.4. Newton Raphson's Method

The traditional method for solving the power flow algorithm is the Newton-Raphson method. This algorithm consists in solving iteratively a system of equations by approximating the original function by its first order Taylor series [27]. Figure 4 illustrates the geometric representation of the newton's method iteration procedure.

Let  $f(x)$  be a unidimensional system where  $f(x)$  and  $x$  are scalars. The algorithm consists in:

- i)  $i = 0$ , choose an initial solution for the variable  $x^0$
- ii) Compute  $\varepsilon = |f(x^i)|$ .
  - a. If  $\varepsilon < \text{tolerance}$ , the algorithm stops and  $x^i$  is the root for the function  $f(x)$
  - b. Else proceed to 3)
- iii) Compute  $x^{i+1} = x^i + \Delta x^i$ , where:
  - a.  $f(x^i + \Delta x^i) \approx f(x^i) + \frac{f(x^i)}{dx} \Delta x^i = 0 \rightarrow \Delta x^i = -\frac{f(x^i)}{f'(x^i)}$
- iv)  $i = i + 1$ , then go to step ii)

To generalize the method for a  $n$ -dimension system of equations, the Jacobian of  $f(x)$  must be calculated. The Jacobian of a  $n$ -dimension system of equations is given by:

$$J(x) = \frac{\partial f(x)}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

Equation 11

A common variation in the Newton Raphson method is the Jacobian Constant version. In this variation, the Jacobian is computed only in the initialization procedure, and it's kept constant throughout the iterations.

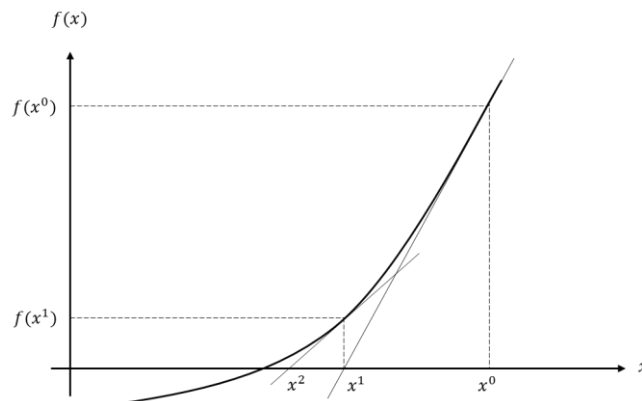


Figure 5: unidimensional newton method for  $f(x)$

## 2.5. Convergency Issues

Due to the nonlinearities and nonconvexities, the solution of the power flow problem might not always be an easy task to find. The main convergency issues that arises in power flow problems are: (i) the system of nonlinear equations can have multiple solutions, with low voltage and electrically instable ones [8]; (ii) for stressed systems where the operating point is close the operation limit, the Jacobian is usually ill-conditioned [10]; the convergence region in newton's method is fractal for the power flow problem [11], [12], [13]; (iv) large-scale electrical systems rely on a good starting point [4], [7], [9], [14], [18].

When using Newton's method to solve the problem, the improvements usually consist in: (i) correcting the steps to guarantee that the power mismatch is declining in each iteration (optimum multiplier methods), (ii) improve the nonlinear approximation by adding higher orders in the Taylor expansion; (iii) exploring polar-rectangular variations in the formulation [14], [15].

In recent years, the usage of optimization problems for solving AC power flow has increased [5]. The main reason for that is the improvements in nonlinear programming solvers which have been handling more robustly large power system cases. The improvements using optimization problems can be made by reinforcing bounds on system variables, such as voltage magnitudes and reactive power generation, to avoid bad convergence areas. Additionally, another way of improving the performance AC power system models is by neutralizing the effect of initial values. Studies have shown that using homotopy-base power flow methods can achieved that goal [16]. These methods seek to replace the original problem by sequential subproblems, which are easier to solve, but are iteratively conducting the problem to the real one. The last sequential subproblem must be equivalent to the original AC power flow problem.

## 2.6. Open-Source Software

Several open-source software packages for power flow analysis have been published. The most recognized among the community are: MATPOWER [26], and its python variation PYPOWER [28]; and more recently, PandaPower [29] and PowerModels [30]. Matpower, PandaPower and PowerModels are known for being able to handle large-scale systems. Additionally, Julia optimization, power by JuMP.jl [31], have been receiving a great attention by the industry. For that reason, the proposed package ControlPowerFlow.jl to perform power flow with control actions was implemented in Julia on top of the PowerModels and JuMP packages.



## 3. Brazilian Power System

The Brazilian power system is among the largest electrical systems in the world. The installed capacity of the country is 150GW and the total extension of transmission lines is more than 100.000 Km, which puts the Brazilian grid asides with PJM, in United States, and the Germany system [19]. Nevertheless, the generation profile of the country is composed by 64% of hydropower plants, 25% of thermal and 11% of renewable, which bring interesting characteristics to the operation of the system. First, the transmission lines must be able to flow high amounts of energy through long distances due to the distances between large hydro reservoir and large consumption centers, as illustrated in Figure 5. In this context, Brazil uses HVDC links improve power transferring in the system, while reducing costs.

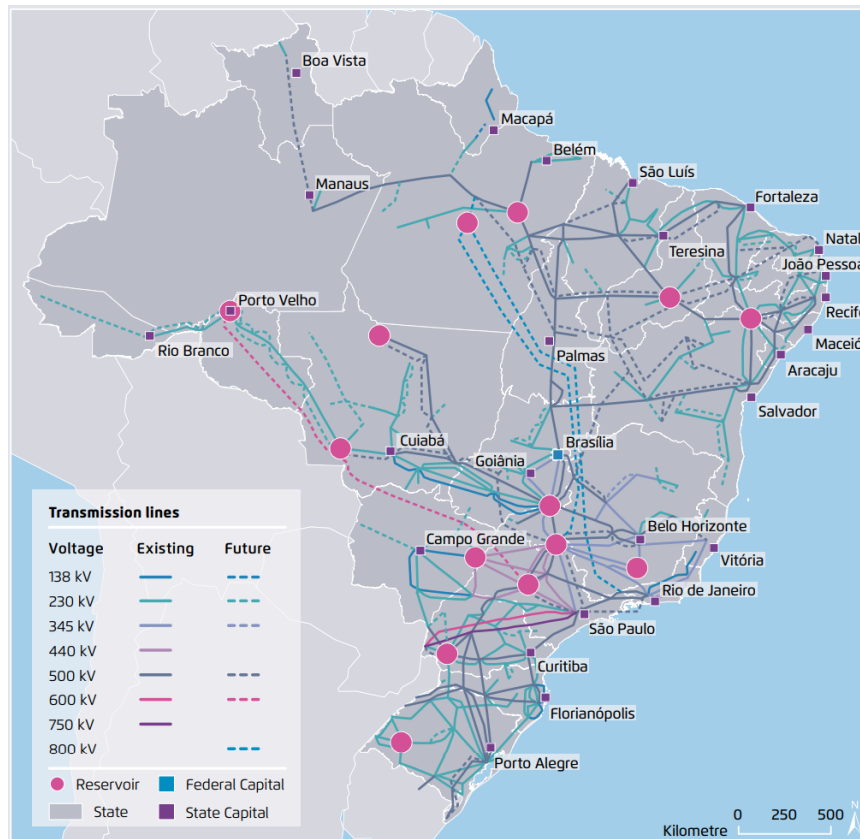


Figure 6: Brazilian Electric System [33]

The official software for power flow studies is ANAREDE (acronymous for Network Analysis (*portuguese: Análise de Rede*)). The ANAREDE software is develop by Eletrobrás Cepel, and the most common users are: (i) brazilian sectorial companies such as ONS (System Operator), EPE (Energy Research Company), MME (Ministry of Energy) (ii) generation and distribution companies; (iii) large consumers [22]. ANAREDE software is adapted to the Brazilian necessities and characteristics, implementing important features of the system in the power flow algorithm, such as, HVDC transmission lines and controllable devices. Additionally, ANAREDE software allows the user to choose which control to perform while running power flow analysis.

### 3.1. Controllable Devices

Control elements are added networks to accommodate stressed systems and improve system robustness against contingencies or heavy variations. In the Brazilian System, several controllable elements are modeled. In general, the most used control elements in power systems are [32]:

1. **Static Var Compensators (SVC):** Shunt electronic device capable of managing power transfer stability. The SVC can vary the shunt susceptance to keep the voltage magnitude in the controlled bus in a specified level.



2. **Shunt Reactors:** Equipment's composed by capacitor and/or inductions capable of controlling the voltage magnitude of the system. The reactors can be switched in different levels to control the controlled bus voltage.
3. **Thyristor Controlled Series Capacitor (CSC):** Series electronic device capable of managing transmission stability. The CSC can be used to control the active power flow in transmission lines.
4. **On-load tap changer (OLTC):** Transformer capable of automatically adjust its tap value to control the voltage magnitude of the controlled bus
5. **Phase-Shifting:** Transformer capable of automatically adjust its shift angle to control the active power flow in a transmission line.

#### 4. PWF.jl

To execute power flow analysis with the Brazilian cases, it's necessary to create a data parser and a converter to the software which will perform the study. Since we are interested in performing load flow analysis within the Julia Language environment, we develop a Julia package called PWF.jl <sup>3</sup>[24] to read and prepare the data. The parser implementation was built on ANAREDE user manual guide v0.9.

##### 4.1. ANAREDE File

The ANAREDE data format (PWF) defines all the information regarding the power system operation. That is, the buses and lines identification, voltages and angles, active and reactive generation/consumption, and much more. The PWF file is design to accommodate many special features of the Brazilian electric power system, such as DC lines, Var Compensators and Controllable Series Compensators. Additionally, PWF files contain the necessary information for the ANAREDE software perform the power flow analysis considering control variables.

The structure of the PWF file is composed by **execution codes**. Each execution code defines a section in the data file. The execution codes considered in our parser are represented in Table 1.

##### 4.2. Parser

The PWF.jl package has two features: (i) read the PWF file into a Julia dictionary without any transformation or data corrections; (ii) convert and treat the PWF data to PowerModels package format. Figure 5 illustrate the package features.

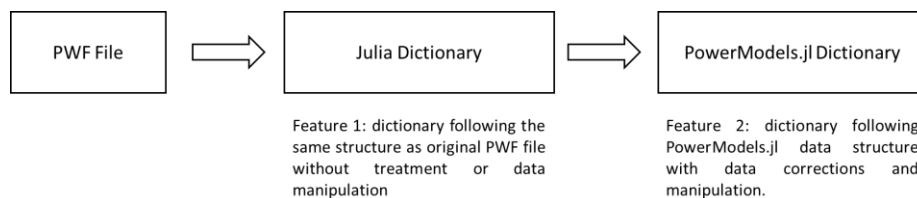


Figure 7: PWF.jl features

The first feature is intended to applications that and don't necessarily want to perform load flow analysis or use PowerModels. These applications might be data verification or visualization. The second is intended to users that want to perform power flow studies by PowerModels convention. Since many open-source packages have adopted that convention (included our power flow package ControlPowerFlow.jl) feature 2 is the main functionality of PWF.jl.

<sup>3</sup> The PWF.jl package documentation provides detailed information about the usage of the package and the ANAREDE file. It can be found at documentation [webpage](#)

Execution Code	Description
DOCP	Execution and Control Options Data
DCTE	Constants and Parameters used in the solution
DBAR	General information about buses inside the AC circuit
DLIN	General information about branch data, both for transmission lines and transformers.
DGER	Active generation bounds and participation factors data
DGBT	Bus base voltage data
DGLT	Bus voltage bounds data
DSHL	Line shunt data
DCSC	Controllable series compensator data
DBSH	Controllable capacitor or reactor banks connected to AC buses or branches.
DCER	Static reactive compensator data
DELO	DC link data
DCBA	DC bus data
DCLI	DC line data
DCNV	AC-DC converter data
DCCV	AC-DC converter control data

Table 1

### 4.3. Converter

To convert the PWF data to PowerModels structure and be able to perform power flow analysis using PowerModels formulations some adaptations from the PWF file were made:

- 1. DC Line Approximation:** DC lines are modeled with two 'dummy' generators model, described in 2.1.2. Therefore, the DC line, links and bus data were adapted to the simplifications.
- 2. Static Var Compensators (SVC):** are not modeled in PowerModels. Therefore, we approximate its functioning by a shunt reactor.
- 3. Thyristor Controlled Series Capacitor (CSC):** are not modeled in PowerModels. Therefore, we approximate its functioning by an AC transmission line with fixed impedance.

#### 4.4.Example

Figure 7 gives an example of ANAREDE file. Each section starts with the execution code and ends with the number '99999'. In this example, the file is composed by sections, *DOCP*, *DCTE*, *DBAR* and *DLIN*. Additionally, the line under *TITU* represents the file name and *FIM* represents the file end. Each section has different elements which are separated by fixed delimiters. The description of each section elements and delimiters are given in Appendix B.

```

TITU
2 Bus generic system
DOCP IMPR
(Op) E (Op) E (Op) E (Op) E (Op) E (Op) E (Op) E (Op) E
QLIM L
99999
DCTE
(Mn) ( Val) (Mn) ( Val) (Mn) ( Val) (Mn) ( Val) (Mn) ( Val)
BASE 100. DASE 100. TEPA .1 EXST .4 TETP 5. TBPA 5.
99999
DBAR
(Num)OETGb( nome )Gl( V)( A)( Pg)( Qg)( Qn)( Qm)(Bc )( Pl)( Ql)( Sh)Are(Vf)M(1)(2)(3)(4)(5)(6)(7)(8)(9)(10
1 L2 ABUS 1 11029 0. 38.9.396-130.130.4 3. 11000
2 L ABUS 2 21000-.44 50.39 30. 11000
99999
DLIN
(De )d O d(Pa )NcEP ( R% )( X% )(Mvar)(Tap)(Tmn)(Tmx)(Phs)(Bc )(Cn)(Ce)Ns(Cq)(1)(2)(3)(4)(5)(6)(7)(8)(9)(10
1 2 1 18.1 17.4 99999999 9999
99999
FIM

```

Figure 8: Example of a PWF file

The Julia code in Figure 8 demonstrate how to read the PWF file using Feature 1. The user must provide the path to the PWF file, and additionally, the 'pm' parameters indicates if the parser must convert the dictionary to PowerModels convention.

```

using PWF
file = "2busexample.PWF"
pwf_dict = parse_file(file; pm = false)

```

Figure 9: Reading PWF file by feature 1

The resulting dictionary using feature 1 is given by Figure 9. Note that each key of the dictionary is related to one section of the PWF file. Figure 10 illustrate bus 1 data inside the DBAR section.

```

julia> pwf_dict = parse_file(file; pm = false)
Dict{String, Any} with 6 entries:
  "name"      => "2busfrank"
  "DOCP IMPR" => Dict{String, Any}{"QLIM"=>'L'}
  "DBAR"      => Dict{String, Any}{"1"=>Dict{String, ...
  "DCTE"      => Dict{String, Any}{"TLVC"=>0.5, "APAS"...
  "DLIN"      => Dict{String, Any}{"1"=>Dict{String, ...

```

Figure 10: Julia Dictionary by feature 1

```
julia> pwf_dict["DBAR"]["1"]
Dict{String, Any} with 20 entries:
  "ANGLE" => 0.0
  "MINIMUM REACTIVE GENERATION" => -130.0
  "BASE VOLTAGE GROUP" => " A"
  "ACTIVE CHARGE" => 0.0
  "REACTIVE GENERATION" => 9.396
  "NAME" => "Bus 1"
  "VISUALIZATION" => 0
  "CONTROLLED BUS" => 1
  "ACTIVE GENERATION" => 38.0
  "VOLTAGE LIMIT GROUP" => " 1"
  "AREA" => 1
  "NUMBER" => 1
  "STATUS" => 'L'
  "VOLTAGE" => 1.029
  "TYPE" => 2
  "OPERATION" => 'A'
```

Figure 11: Information of bus 1 inside DBAR by feature 1

Converting the PWF file directly into PowerModels convention, using feature 2, is like the previous case. The difference is that 'pm' parameter can be omitted since its default is true. Figure 11 illustrate the PWF file converted into PowerModels format. The resulting dictionary sections are: "name", "baseMVA", "per\_unit", "bus", "branch", "dcline", "switch", "gen", "load" and "shunt". All the PWF original sections are converted into one of those and their data are manipulated to fit the convention.

```
julia> pwf_dict = parse_file(file)
Dict{String, Any} with 13 entries:
  "bus" => Dict{String, Any}{"1"=>Dict{String, Any}{"zone"=>1, ...
  "source_type" => "pwf"
  "name" => "2busfrank"
  "dcline" => Dict{String, Any}{}
  "source_version" => "09"
  "branch" => Dict{String, Any}{"1"=>Dict{String, Any}{"br_r"=>0.181...
  "gen" => Dict{String, Any}{"1"=>Dict{String, Any}{"pg"=>0.38, ...
  "storage" => Dict{String, Any}{}
  "switch" => Dict{String, Any}{}
  "baseMVA" => 100.0
  "per_unit" => true
  "shunt" => Dict{String, Any}{"1"=>Dict{String, Any}{"source_id" ...
  "load" => Dict{String, Any}{"1"=>Dict{String, Any}{"source_id"=> ...
```

Figure 12: Julia Dictionary bu feature 2. PowerModels.jl convention

Figure 12 shows the bus 1 information inside "bus" section of the resulting dictionary. Note that, since PowerModels formulations model individually load and generation units, the information of active and reactive power are not inside the bus element.

```
julia> pwf_dict["bus"]["1"]
Dict{String, Any} with 12 entries:
  "zone" => 1
  "bus_i" => 1
  "bus_type" => 3
  "name" => "Bus 1"
  "vmax" => 1.1
  "source_id" => ["bus", "1"]
  "area" => 1
  "vmin" => 0.9
  "index" => 1
  "va" => 0.0
  "vm" => 1.029
  "base_kv" => 1.0
```

Figure 13: Information of bus 1 inside "bus" section

Figures 13 and 14 show the "gen" and "load" section of the resulting dictionary. In the original file, bus 1 has active and reactive generation information, and no consumption data, while bus 2 has consumption data and no generation. Therefore, a generating unit was created associated to bus 1, Figure 13, and a load unit was created associated to bus 2, Figure 14.

```
julia> pwf_dict["gen"]["1"]
Dict{String, Any} with 17 entries:
  "gen_bus"    => 1
  "pg"         => 0.38
  "pmin"       => 0.0
  "pmax"       => 999.99
  "qg"         => 0.09396
  "qmin"       => -1.3
  "qmax"       => 1.304
  "model"      => 2
  "gen_status" => 1
  ...
```

*Figure 14: Information of gen "1" inside "gen" section*

```
julia> pwf_dict["load"]["1"]
Dict{String, Any} with 6 entries:
  "load_bus"  => 2
  "pd"        => 0.5039
  "qd"        => 0.3
  "status"    => 1
  ...
```

*Figure 15: information of load 1 inside "load" section*

The resulting dictionary from feature 2 can be used to perform power flow analysis on the PowerModels.jl package, which contains several exact AC power flow formulations and innumerable relaxations and linear approximations. Figure 15 illustrate how to perform power flow analysis reading ANAREDE files. For more information on the PowerModels.jl package visit the PowerModels website on GitHub [30].

```
using PWF, PowerModels, Ipopt
file = "2busexample.pwf"
pwf_dict = parse_file(file)
results = run_ac_pf(pwf_dict, Ipopt.Optimizer)
```

*Figure 16: Running power flow in PowerModels with PWF data.*

The ANAREDE file have more information then PowerModels convention expects, mostly related to control information. Therefore, we also added a parameter in the parse\_file function to convert additional data to PowerModels convention structure. Therefore, the command to parser the additional information from the PWF to PowerModels structure is:

```
# PWF file path
file = "Example.PWF"
# Reading PWF file with control_data
pwf_file = PWF.parse_file(file; add_control_data = true)
```

Therefore, to parse the necessary information to use control options, the user must set the parameter add\_control\_data equal to true.

## 4.5. Case Study

In this chapter we will perform a case study to verify the correct functioning of the parser. The case study consists in comparing the AC power flow results obtained by the ANAREDE software and the results obtained by the PowerModels software using the PWF parser to read and prepare the data. The expectation is that if the PWF file is being read correctly, the results from both software should be the approximately the same. The power system case used in this comparison is the IEEE 118 [33] Bus illustrated by figure 16.

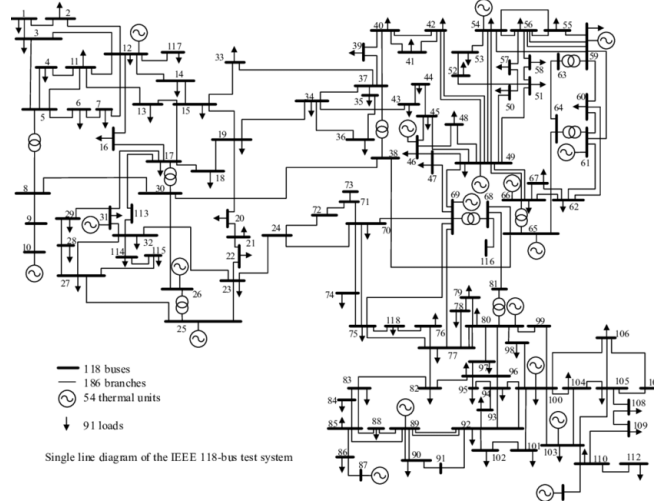


Figure 17: IEEE 118 bus system used to verify PWF parser.

The ANAREDE results were obtained by running the IEEE 118 Bus in the academic version (v11.5.5-Jul21) without any control options, that is, the traditional power flow algorithm. To obtain the PowerModels results, we used the commands in figure 15.

The variables compared were voltage magnitudes, voltage angles, active and reactive power flow in one direction of the branch. The results mismatches were calculated and the errors statistics comparing all elements from the system is displayed in figure 17.

Results Mismatch								
Variable	Minimum	Quantile 5%	Quantile 25%	Mean	Median	Quantile 75%	Quantile 95%	Maximum
Magnitude (p.u.)	-0,0050	-0,0050	-0,0032	-0,0007	0,0000	0,0017	0,0038	0,0050
Angle (Deg)	-0,0048	-0,0042	-0,0021	0,0002	-0,0001	0,0029	0,0046	0,0049
Active Power Flow (p.u.)	-0,0001	-0,0001	0,0000	0,0000	0,0000	0,0000	0,0001	0,0001
Reactive Power Flow (p.u.)	-0,0001	-0,0001	0,0000	0,0000	0,0000	0,0000	0,0001	0,0001

Figure 18: Error statistics of the results mismatch between ANAREDE results and PowerModels with PWF parser. Statistics are: Minimum, Mean, Maximum and Quantiles of 5%, 25%, 50%, 75% and 95%.

From figure 16, we can observe that mismatches are lower than 2 decimal points, which is the precision used in ANAREDE for power flow calculations.

## 5. ControlPowerFlow.jl

ControlPowerFlow.jl package objective is to create a flexible framework for performing power flow with control options. The package was built on top of the open-source package PowerModels.jl [30], which provides a traditional power flow model implementation through optimization using JuMP package. JuMP allow writing and modifying optimization problems easily. The implications of the control actions to the power flow model can be resumed in three forms: **(i)** create a new decision variable from a fixed parameter (e.g., shunt susceptance); **(ii)** create a new constraint (e.g., reactive generation limits); **(iii)** modify an existing constraint by insert a slack variable and creating a quadratic penalization of it in the objective function (e.g. keep a variable as close as the specified level as possible).

In this context, the ControlPowerFlow package provides the user a pre-existing menu of actions, which were inspired by the ANAREDE software.

### 5.1. Control Actions

This section provides explanation of each one of the pre-existing control actions. The ANAREDE files contain all the necessary information to handle control actions, which are obtained through PWF.jl. The control package contains the actions: **reactive generation limits** (QLIM), **voltage magnitude limits** (VLIM), **shunt reactor control** (CSCA), **transformer's tap control** (CTAP), **transformer's tap control with voltage limits** (CTAF), and **phase-shifting control** (CPHS). In this work, we are only describing QLIM, VLIM and CSCA.

#### 5.1.1. Reactive Generation Limits (QLIM)

The QLIM control enforces the limits for the reactive power of the generating units. If the voltage magnitude of a PV bus can't be kept in the specified levels, due to reactive generation limits, the model must be able to adjust it. This can be obtained by inserting a slack variable in the voltage magnitude setpoint constraint and penalizing the squared of slack variable in the objective function. That way, the model will only adjust the bus voltage if it is necessary.

The model implications are:

- New variables:

$$sl_i^v, \quad \forall b \in \mathcal{B}^{PV} \quad (1.1)$$

- New Constraints:

$$q_i^{gMin} \leq q_i^g \leq q_i^{gMax}, \quad \forall i \in \mathcal{Gen} \quad (1.2)$$

- Modified Constraints:

$$v_b = v_b^{spec} + sl_b^v, \quad \forall b \in \mathcal{B}^{PV} \quad (1.3)$$

Where  $sl_i^v$  is the slack variable for the voltage magnitude setpoint constraint,  $q_i^g$  is the reactive generation of active generator  $i$ .

#### 5.1.2. Voltage Magnitude Limits (VLIM)

The VLIM control enforces voltage magnitude limits of PQ buses. The voltage magnitude of load buses is directly related to the reactive consumption of that bus and neighborhoods. For that, the reactive load of that bus is turned into a decision variable which will be kept as close as possible the specified level. This is achieved by inserting a slack variable in the load setpoint constraint and penalizing it in the objective function.

The model implications are:

- New variables:

$$q_i^d, \quad \forall i \in \mathcal{Load} \quad (2.1)$$

$$sl_i^d, \quad \forall i \in \mathcal{Load} \quad (2.2)$$



- New Constraints:

$$v_i^{MIN} \leq v_b \leq v_i^{MAX}, \quad \forall b \in \mathcal{B}^{PV} \quad (2.3)$$

$$q_i^d = q_i^{dspec} + sl_i^d, \quad \forall i \in Load \mid bus(i) \in \mathcal{B}^{PQ} \quad (2.4)$$

### 5.1.3. Shunt Reactor Control (CSCA)

The CSCA control vary the controllable shunts of the system to maintain a voltage setpoint or limits specification. The controllable shunts can have 2 modes of operation: (i) continuous and (ii) discrete. The **continuous shunt** varies the susceptance within its limits to maintain the voltage magnitude at the controlled bus **fixed**. The **discrete shunt** automatically switches the shunt banks to keep the voltage magnitude at the controlled bus **within bounds**. It's important to notice that all controllable shunts in the same bus must have the same control mode. In addition, the model always tries to keep the shunts in the original specified value, that is, the shunt is only adjusted if the controlled bus voltage magnitude requires. This is accomplished by setting the shunt variable to a specified level and adding a slack variable to that equation. Finally, shunt control is only valid for PQ buses, since PV buses already have the voltage magnitude fixed.

This version of the package approximate discrete variables of the discrete mode as continuous variables. Therefore, the difference between continuous and discrete mode, in our implementation, is the voltage type control (fixed or within bounds).

Additional modelling structures are added to help the formulation: if a shunt  $i \in \mathcal{B}^{Sh}$  is controllable, then  $control^{bsh}(i) = true$ . If a controllable shunt has discrete mode, then  $mode^{bsh}(i) = discrete$ . In the same manner, if a bus  $b$  has shunt control, then  $control^{bsh}(b) = true$ . The same is valid with the bus mode of control.

The model implications are:

- New variables:

$$bs_i^{bsh}, \quad \forall i \in \mathcal{B}^{Sh} \mid control^{bsh}(i) = true \quad (3.1)$$

$$sl_i^{vUpp}, sl_i^{vLow} \quad \forall i \in \mathcal{B}^{Sh} \mid control^{bsh}(i) = true \& mode^{bsh}(b) = discrete \quad (3.2)$$

$$sl_i^v, \quad \forall i \in \mathcal{B}^{Sh} \mid control^{bsh}(i) = true \& mode^{bsh}(b) = continuous \quad (3.2)$$

- New Constraints:

$$bs_i^{bshMIN} \leq bs_i^{bsh} \leq bs_i^{bshMAX}, \quad i \in \mathcal{B}^{Sh} \mid control(i) = true \quad (3.3)$$

$$bs_i^{bsh} = bs_i^{bshSpec} + sl_i^{bsh}, \quad i \in \mathcal{B}^{Sh} \mid control(i) = true \quad (3.4)$$

$$-sl_i^{vUpp} + v_b^{MIN} \leq v_b \leq v_b^{MAX} + sl_i^{vLow}, \quad \forall b \in \mathcal{B}^{PQ} \mid control^{bsh}(b) = true \& mode^{bsh}(b) = discrete \quad (3.4)$$

$$v_b = v_b^{spec} + sl_i^v, \quad \forall b \in \mathcal{B}^{PQ} \mid control^{bsh}(b) = true \& mode^{bsh}(b) = continuous \quad (3.6)$$

Where  $bs_i^{bsh}$  are the new decision variables due to the controllable shunts.  $sl_i^{vUpp}$  and  $sl_i^{vLow}$  are slack variables to the voltage magnitude limits of discrete shunt control and  $sl_i^v$  are the slack variables to the voltage magnitude setpoint for the continuous shunt control.

### 5.2. Case Study

This section provides a case study which objectives are: (i) serve as a user-guide; (ii) validate the control action formulations. To choose a control action, the user only needs to insert in DOPC IMPR section of the PWF file the desired controls. Figure 18 illustrate the activation of controls QLIM and CSCA.

```
DOPC IMPR
(0p) E (0p) E (0p) E (0p) E (0p) E (0p) E (0p) E (0p) E (0p) E
QLIM L CSCA L
```

Figure 19: Activation of control options QLIM and CSCA.

To read a PWF file and perform a control power flow, the user must do the following steps: (i) insert the desired control actions in the DOPC IMPR section; (ii) read the file with PWF.jl package setting `add_control_info` equals to `true` and (iii) call the control power flow function. These steps are represented in figure 19:

```
# Include Packages
using PWF, ControlPowerFlow, Ipopt

# PWF file path
file = "Example.PWF"

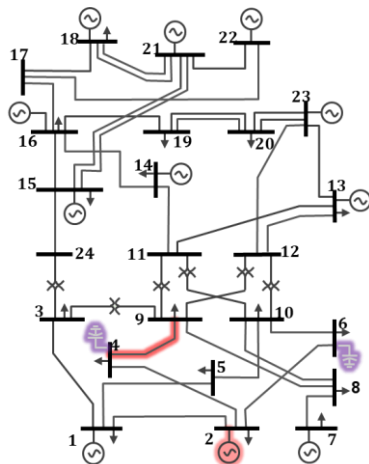
# Reading PWF file with control_data
pwf_file = PWF.parse_file(file; add_control_data = true)

# Perform control power flow
results = run_control_pf(file, optimizer = Ipopt.Optimizer)
```

Figure 20: Running a control power flow in Julia.

## 5.2.1. Shunt Reactor Control (CSCA)

Two controllable shunt reactors were inserted in the IEEE 24bus system [33]. The shunts specification and the system illustration are given by figure 20. To evaluate the control of the reactor devices, a severe contingency was applied in the original network. The contingency is composed by loss of transmission line between buses 9 and 4, and loss of generator in bus 2.



Bus	Controllable Reactors					
	Susceptance Specified	Susceptance Minimum	Susceptance Maximum	Control Mode	Voltage Control Minimum	Voltage Control Maximum
4	0,0	-0,1	0,1	Continuos	0,938	0,938
6	0,0	0,2	0,2	Discrete	0,952	0,962

Figure 21: Modified IEEE 24 bus system. Severe contingency: Loss of line between buses 9 and 4 and loss of the generator of bus 2.

The results are shown in figures 21 and 22, where voltage magnitudes of buses 3, 4, 5, 6 and 10, and shunt susceptance values were analyzed. In figure 21, the pre-contingency state is represented by the blue bar. The post contingency state performed without any control actions, i.e., original PowerModels formulation, is represented by the orange bar. Finally, the post-contingency state with CSCA control options performed by the ControlPowerFlow package is represented by the grey line. In figure 22, the shunt susceptance results are demonstrated using the same colors description.

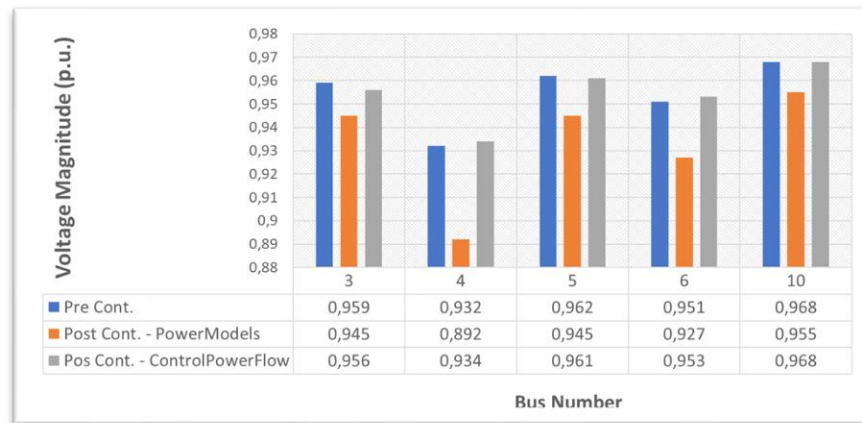


Figure 22: CSCA voltage magnitude results.

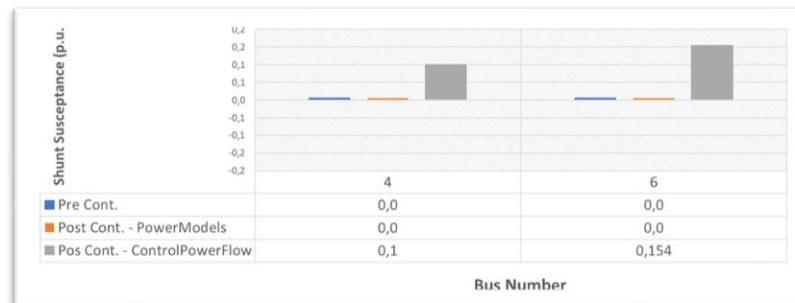


Figure 23: CSCA shunt susceptance results

The results show that the controllable elements in the system can handle the voltage magnitude instabilities by adjusting the susceptance of the reactors in the system. The control power flow model kept the voltage magnitude at the pre-contingency values, thus providing a validation for the CSCA control<sup>4</sup>.

<sup>4</sup> Future versions of the document will provide more case studies validations.

## 6. Conclusion

The present work described the implementation of two Julia Packages, PWF.jl and ControlPowerFlow.jl, for performing power flow using ANAREDE files. First, it's presented a literature review of power flow equations and common issues when solving the power flow problem. Then, a brief resume of Brazilian Power System and its controllable devices are given. After, the implementation and validation of both packages were made.

The PWF.jl package can successfully parse ANAREDE files to Julia structures and convert those structure to an open-source package, PowerModels.jl, convention to perform power flow analysis. We also showed through a case study that performing AC power flow analysis with ANAREDE files in PowerModels.jl using PWF.jl package returns the same results as running traditional power flow in ANAREDE software.

Nevertheless, the ControlPowerFlow.jl software allows the users to perform traditional power flow with control actions. In this work, the controls reactive generation limits, voltage magnitude limits, automatic reactor control are described. This work also provides validation to automatic reactor control option (CSCA) by performing a case study.

Finally, it is important to say that the ultimate objective of the packages is to empower the Brazilian electric system agents to run and perform AC analysis/studies in open-source environments without the dependency of commercial softwares. For that, both PWF.jl [24] and ControlPowerFlow.jl [23] packages are available in GitHub as open-sources projects and can be used and contributed in Julia Language.

- [1] P. V. H. Carleton Coffrin, "A Linear-Programmin Approximation of AC Power Flows," *INFORMS Journal on Computing*, pp. 718-734, Fakk 2014.
- [2] K. A. Pambour, "Modelling, simulation and analysis of security of supply scenarios in integrated gas and electricity transmission networks," University of Groningen, 20018.
- [3] C. E. H. William F. Tinney, "Power Flow Solution by Newton's Method," *IEEE Transactions on Power Apparatus and Systems*, p. 86, 1967.
- [4] Y. H. Zhijun Qin, "Application of Non-linear Programming for Large-Scale AC-DC Power Flow Analysis," *IEEE Power and Energy Society General Meeting*, July 2012.
- [5] T. T. G. H. Dimitry Shchetinin, "On the Construction of Linear Approximations of," *IEEE Transactions on Power Systems*, October 2018.
- [6] L. T. B. Andreas Wachter, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Society for Industrial and Applied Mathematics*, pp. 877-900, 2005.
- [7] S. R. Stephen Frank, "Optimal power flow: a bibliographic survey I," *Energy Systems*, pp. 221-258, 24 April 2012.
- [8] B. K. Johnson, "Extraneous and False Load Flow Solutions," *IEEE transactions on Power Apparatus and Systems*, 2 Aprix 1977.
- [9] T. X. T. J. O. Adam B. Birchfield, "Power Flow Convergence and Reactive Power Planning in the Creation of Large Synthetic Grids," *IEEE Transactions on Power Systems*, pp. 6667-6674, 6 Nov 2018.
- [10] C. A. C. C. A. F. M. Luciana M. C. Braz, "A Critical Evaluation of Step Size Optimization Based Load Flow Methods," *IEEE Transactions on Power Systems*, pp. 202-207, 15 Feb 2000.
- [11] S. A. N. James S. Thorp, "Load-Flow Fractals Draw Clues to Erratic Behavior," *IEEE Computer Applications in Power*, January 1997.
- [12] T. J. O. Christopher L. DeMarco, "Low Voltage Power Flow Solutions and Their Role in Exit Time Based Security Measures for Voltage Collapse," *27th Conference on Decision and Control*, pp. 2127-2137, December 1988.
- [13] G. Sanderson, "Newton's Fractal," 3Blue1Brown, 12 October 2021. [Online]. Available: <https://www.youtube.com/watch?v=-RdOwhmqP5s>. [Accessed 25 November 20221].
- [14] T. J. O. Raymond P. Klump, "Techniques for Improving Power Flow Convergence," *Power Engineering Society Summer Meeting*, 2000 July 2000 .
- [15] P. J. Lagace, "Power Flow Methods for Improcing Convergence," in *38th Annual Conference on IEEE Industrial Electronics Society*, Montreal, Canada, 2012.
- [16] T. T. D. R. A. W. Walter Murray, "Improving The Robustness of Newton-Based Power Flow Methods to Cope with Poor Initial Points," in *North American Power Symposium (NAPS)*, Manhattan, KS, USA, 2019.
- [17] Y. H. S. Y. Z. S. Ying Xiao, "Versatile Model for Power Flow Control Using FACTS :Devices," in *Third International Power Electronics and Motion Control Conference (IEEE Cat.)*, Beijing, China, 2000.

- [18] X. C. Y. S. Thomas J. Overbye, "A Comparison of the AC and DC Power Flow Models for LMP Calculations," in *37th Hawaii International Conference on System Sciences*, Big Island, HI, 2004.
- [19] L. B. Ribeiro, "Technical and economical aspects of WEMs: an international comparison and contributions for improvements in Brazil," PUC-Rio, 2020.
- [20] ONS, "ONS Histórico da Operação," [Online]. Available: [http://www.ons.org.br/Paginas/resultados-da-operacao/historico-da-operacao/capacidade\\_instalada.aspx](http://www.ons.org.br/Paginas/resultados-da-operacao/historico-da-operacao/capacidade_instalada.aspx). [Accessed 04 12 2021].
- [21] ABRATE, "TRANSMISSÃO EM CORRENTE CONTÍNUA: IMPACTOS DO SEU USO," [Online]. Available: <https://abrate.org.br/transmissao-em-corrente-continua-impactos-do-seu-uso/>. [Accessed 04 12 2021].
- [22] E. CEPEL, "ANAREDE Software," [Online]. Available: [http://srvlumis02.cepel.br/pt\\_br/produtos/anarede-analise-de-redes-eletricas.htm](http://srvlumis02.cepel.br/pt_br/produtos/anarede-analise-de-redes-eletricas.htm). [Accessed 04 12 2021].
- [23] I. S. Chávarry, "ControlPowerFlow.jl," LAMPS PUC-Rio, 04 12 2021. [Online]. Available: <https://github.com/LAMPSPUC/ControlPowerFlow.jl>. [Accessed 04 12 2021].
- [24] P. H. Iago S. Chávarry, "PWF.jl: a Julia package for converting ANAREDE data format (".pwf") into a Julia dictionary," LAMPS PUC-Rio, 11 2021. [Online]. Available: <https://github.com/LAMPSPUC/PWF.jl>. [Accessed 04 12 2021].
- [25] R. P. O. A. C. Mary . B Cain, "History of Optimall Power Flow and formulations," Federal Energy Regulatory Commission , 2012.
- [26] C. E. M.-S. R. J. T. Ray Daniel Zimmerman, "MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education," *IEEE TRANSACTIONS ON POWER SYSTEMS*, February 2011.
- [27] A. J. Monticelli, *Fluxo de Carga em Redes de Energia Elétrica*, São Paulo: Cepel, 1983.
- [28] R. Lincoln, "PYPOWER," [Online]. Available: <https://github.com/rwl/PYPOWER>. [Accessed 02 12 2021].
- [29] A. S. F. S. J.-H. M. J. D. F. M. Leon Thurner, "Pandapower—An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems," *IEEE TRANSACTIONS ON POWER SYSTEMS*, November 2018.
- [30] R. B. K. S. Y. N. M. L. Carleton Coffrin, "PowerModels.jl: An Open-Source Framework for Exploring Power Flow Formulations," *Power Systems Computation Conference (PSCC)*, June 2018.
- [31] J. H. M. L. Iain Dunning, "JuMP: A Modeling Language for Mathematical Optimization," *SIAM Review*, pp. 295-320, 2017.
- [32] Cepel, "Manual Anatem," [Online]. Available: <https://dre.cepel.br/manual/anatem/index.html>. [Accessed 04 12 2021].
- [33] A. E. & I. E. Diálogos, "Report on the Brazilian Power System," 2019.

## Appendix

### A. Power Flow Equations

From the ( $\pi$  - circuit) branch model, it's possible to calculate for each branch  $l \in \mathcal{L}$  an admittance matrix  $Y_l^b$ , which relates the complex from and to bus injection  $I_{fl}$  and  $I_{tl}$  to the voltages  $V_f$  and  $V_t$ . Figure 2 illustrate Kirchhoff current law at node  $p$ , the output of the transformer in the from-bus side. The current arriving at node  $p$  is  $I_{ft}^p$ , and the voltage at this node is  $V_f^p$ .

Kirchoff current law states that the total current entering a junction is equal to the total current leaving

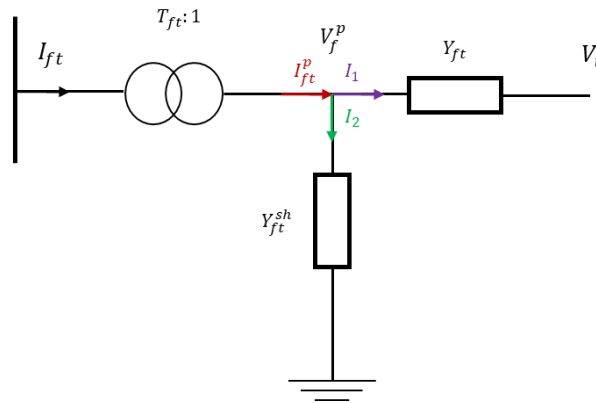


Figure 24: Kirchhoff Current Law at the output of the transformer from-bus side.

the same junction. From figure 23,

$$I_{ft}^p = I_1 + I_2$$

Equation 12

From the relationship between current and voltage,  $\Delta V = Z \cdot I$ , and since  $Y = Z^{-1}$ , we have  $I = \Delta V Y$ . Therefore:

$$I_{ft}^p = (V_f^p - V_t) Y_{ft} + V_f^p Y_{ft}^{sh}$$

Equation 13

Rearranging equation 13 we arrive at:

$$I_{ft}^p = V_f^p (Y_{ft} + Y_{ft}^{sh}) - V_t Y_{ft}$$

Equation 14

The relationship between  $I_{ft}^p / I_{ft}$  and  $V_f^p / V_f$  are given by the transformer equation illustrated in figure 3. The ideal transformer converts the energy without any loss. Therefore,

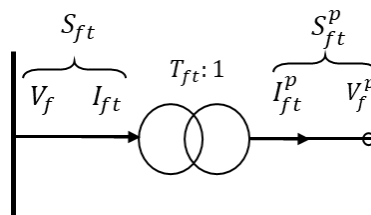


Figure 25: Transformer at from-bus

$$S_{ft} = S_{ft}^p$$

Equation 15

And from aparent power equation,  $S = VI^*$ , we arrive at:

$$V_f I_{ft}^* = V_f^p I_{ft}^{p*}$$

Equation 16

The relationship between  $V_f^p$  and  $V_f$  is given by:  $V_f^p = \frac{V_f}{T_{ft}}$ . Applying into Equation 16:

$$V_f I_{ft}^* = \frac{V_f}{T_{ft}} I_{ft}^{p*}$$

Equation 17

Manipulating equation 17, we arrive ate the relationship between  $I_{ft}^p$  and  $I_{ft}$ :

$$I_{ft} = \frac{I_{ft}^p}{T_{ft}^*}$$

Equation 18

Applying  $V_f^p / V_f$  relationship and Equation 18 into Equation 14, we arrive at the  $I_{ft}$  expression as function of  $V_f$  and  $V_t$ :

$$I_{ft} = \frac{V_f (Y_{ft} + Y_{ft}^{sh})}{T_{ft}^* T_{ft}} - \frac{V_t Y_{ft}}{T_{ft}^*}$$

Equation 19

The same procedure can be applyied to derive the  $I_{tf}$  expression. Figure 25 illustrate Kirchoff's current law ate to-bus side. The  $I_{tf}$  equation can be obtained by the following equations:

$$I_{tf} = I_1' + I_2'$$

Equation 20

$$I_{tf} = V_t (Y_{ft} + Y_{tf}^{sh}) - V_f^p Y_{ft}$$

Equation 21

$$I_{tf} = V_t (Y_{ft} + Y_{tf}^{sh}) - \frac{V_f}{T_{ft}} Y_{ft}$$

Equation 22



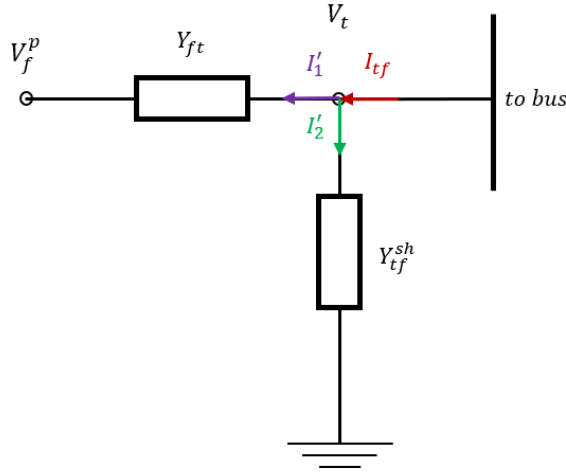


Figure 26: Kirchhoff's current law at to-bus side

Figure 27: Kirchhoff's current law at to-bus side

The relationship of current flowing between from-to sides are given by Equations 19 and 22. The matrix formulation of this relationship for a specific branch  $l$  can be expressed by,  $I_l = Y_l^b V_l$ , where:

Table 2

$$\begin{bmatrix} I_{ft} \\ I_{tf} \end{bmatrix} = \begin{bmatrix} \frac{(Y_{ft} + Y_{ft}^{sh})}{T_{ft}^* T_{ft}} & \frac{Y_{tf}}{-T_{ft}^*} \\ \frac{Y_{tf}}{-T_{ft}} & (Y_{tf} + Y_{tf}^{sh}) \end{bmatrix} \begin{bmatrix} V_f \\ V_t \end{bmatrix}$$

The elements of branch admittance matrices  $Y_l^b$ , can be used to assemble the system bus admittance,  $Y_{bus}$ , which describes complex bus current injections  $I$  relation with the complex bus voltages  $V$  for the entire network:

$$I = Y_{bus} \cdot V, \quad Y_{bus} = [Y_{ft}]^{n^B \times n^B}$$

Equation 23

Finally, by applying the apparent power equation,  $S = VI^*$ , into current flow equations 19 and 22, we arrive at the power flow equations:

$$S_{ft} = \frac{(Y_{ft}^* + Y_{ft}^{sh*}) V_f V_f^*}{T_{ft} T_{ft}^*} - \frac{Y_{ft}^* V_f V_t^*}{T_{ft}}$$

Equation 24

$$S_{tf} = (Y_{ft}^* + Y_{ft}^{sh*}) V_t V_t^* - \frac{Y_{ft}^* V_t V_f^*}{T_{ft}^*}$$

Equation 25

Equations 24 and 25 are equivalent to:

$$S_{ft} = \frac{(Y_{ft}^* + Y_{ft}^{sh*}) V_f V_f^*}{T_{ft} T_{ft}^*} - \frac{T_{ft}^* Y_{ft}^* V_f V_t^*}{T_{ft} T_{ft}^*}$$

Equation 26

$$S_{tf} = (Y_{ft}^* + Y_{tf}^{sh})V_t V_t^* - \frac{T_{ft} Y_{ft}^* V_t V_f^*}{T_{ft} T_{ft}^*}$$

Equation 27

Equations 26 and 27 describes the generic apparent power relation between complex network elements: line admittance  $Y_{ft}$ , combination of line shunts and line susceptance  $Y_{ft}^{sh}$ , and transformer  $T_{ft}$ , and complex voltages  $V_f$  and  $V_t$ . From them, it is possible to derive the real and imaginary power flow equations. First, we start by decomposing the complex values into real and imaginary parts.  $T_{ft} = t_{ft} e^{js_{ft}}$ , can be rewritten as:

$$T_{ft} = t_{ft} e^{js_{ft}} = t_{ft} \cos(s_{ft}) + j t_{ft} \sin(s_{ft}) = t_{ft}^r + j t_{ft}^i$$

Equation 28

The complex voltage multiplication  $V_t V_f^*$ , results in:

$$V_t V_f^* = v_f v_t e^{j(\theta_f - \theta_t)} = v_f v_t [\cos(\theta_f - \theta_t) + j \sin(\theta_f - \theta_t)]$$

Equation 29

Applying Equation 28 and 29 in 26, and expanding complex variables to real and imaginary parts:

$$S_{ft} = \frac{[(g_{ft} - jb_{ft}) + (g_{ft}^{sh} - jb_{ft}^{sh})]v_f^2}{t_{ft}^2} - \frac{(t_{ft}^r - j t_{ft}^i)(g_{ft} - jb_{ft})[\cos(\theta_f - \theta_t) + j \sin(\theta_f - \theta_t)]v_f v_t}{t_{ft}^2}$$

Equation 30

Separating Equation 30 in part 1 and part 2:

$$part\ 1 = \frac{[(g_{ft} - jb_{ft}) + (g_{ft}^{sh} - jb_{ft}^{sh})]v_f^2}{t_{ft}^2}$$

Equation 31

$$part\ 2 = \frac{-(t_{ft}^r - j t_{ft}^i)(g_{ft} - jb_{ft})[\cos(\theta_f - \theta_t) + j \sin(\theta_f - \theta_t)]v_f v_t}{t_{ft}^2}$$

Equation 32

Manipulating part 1, and separating in real and imaginary parts, we arrive at:

$$part\ 1\ real = \frac{(g_{ft} + g_{ft}^{sh})v_f^2}{t_{ft}^2}$$

Equation 33

$$part\ 1\ imaginary = -\frac{(b_{ft} + b_{ft}^{sh})v_f^2}{t_{ft}^2}$$

Equation 34

Manipulating part 2, we arrive at:

$$part\ 2 = \frac{[(-t_{ft}^r g_{ft} + t_{ft}^i b_{ft}) - j(-t_{ft}^r b_{ft} - t_{ft}^i g_{ft})][\cos(\theta_f - \theta_t) + j \sin(\theta_f - \theta_t)]v_f v_t}{t_{ft}^2}$$

Equation 35

Separating in real and imaginary parts, we arrive at:

$$part\ 2\ real = \frac{(-t_{ft}^r g_{ft} + t_{ft}^i b_{ft}) \cos(\theta_f - \theta_t) v_f v_t}{t_{ft}^2} + \frac{(-t_{ft}^r b_{ft} - t_{ft}^i g_{ft}) \sin(\theta_f - \theta_t) v_f v_t}{t_{ft}^2}$$

Equation 36

$$\text{part 2 imaginary} = -\frac{(-t_{ft}^r b_{ft} - t_{ft}^i g_{ft}) \cos(\theta_f - \theta_t) v_f v_t}{t_{ft}^2} + \frac{(-t_{ft}^r g_{ft} + t_{ft}^i b_{ft}) \sin(\theta_f - \theta_t) v_f v_t}{t_{ft}^2}$$

Equation 37

The real part of the apparent power  $S_{ft}$  is called **active power** and is represented by the variable  $p_{ft}$ . The imaginary part of  $S_{ft}$  is called **reactive power** and is represented by the variable  $q_{ft}$ .

Finally, merging part 1 and part 2 equations, the final generic expression of active and reactive power flow of from-bus side is given by:

$$p_{ft} = \frac{(g_{ft} + g_{ft}^{sh}) v_f^2}{t_{ft}^2} + \frac{(-t_{ft}^r g_{ft} + t_{ft}^i b_{ft}) \cos(\theta_f - \theta_t) v_f v_t}{t_{ft}^2} + \frac{(-t_{ft}^r b_{ft} - t_{ft}^i g_{ft}) \sin(\theta_f - \theta_t) v_f v_t}{t_{ft}^2}$$

Equation 38

$$q_{ft} = -\frac{(b_{ft} + b_{ft}^{sh}) v_f^2}{t_{ft}^2} - \frac{(-t_{ft}^r b_{ft} - t_{ft}^i g_{ft}) \cos(\theta_f - \theta_t) v_f v_t}{t_{ft}^2} + \frac{(-t_{ft}^r g_{ft} + t_{ft}^i b_{ft}) \sin(\theta_f - \theta_t) v_f v_t}{t_{ft}^2}$$

Equation 39

The active and reactive power equations for to-bus side can be obtained following the same steps. They are given by:

$$p_{tf} = \frac{(g_{tf} + g_{tf}^{sh}) v_t^2}{t_{ft}^2} + \frac{(-t_{ft}^r g_{ft} - t_{ft}^i b_{ft}) \cos(\theta_t - \theta_f) v_t v_f}{t_{ft}^2} + \frac{(-t_{ft}^r b_{ft} + t_{ft}^i g_{ft}) \sin(\theta_t - \theta_f) v_t v_f}{t_{ft}^2}$$

Equation 40

$$q_{tf} = \frac{(b_{tf} + b_{tf}^{sh}) v_t^2}{t_{ft}^2} - \frac{(-t_{ft}^r b_{ft} + t_{ft}^i g_{ft}) \cos(\theta_t - \theta_f) v_t v_f}{t_{ft}^2} + \frac{(-t_{ft}^r g_{ft} - t_{ft}^i b_{ft}) \sin(\theta_t - \theta_f) v_t v_f}{t_{ft}^2}$$

Equation 41