Master Thesis

Double MSc Degree in Industrial Engineering and Energy Engineering

Dynamic Simulation and Stability Analysis of Power Systems: Development of RMS and EMT Tools within VeraGrid at eRoots

January 2026

Author: Maria Sans Esqué

Supervisors: Vinícius Albernaz Lacerda

Josep Fanals i Batllori

ETSEIB

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

ETSEIB

Maria Sans Esqué

Abstract

Resumen

Resum

Contents

List of Figures

List of Tables

Glossary

Symbols

$\boldsymbol{\theta}$	Voltage angle vector
λ	Eigenvalue
ζ	Damping ratio
$\boldsymbol{\omega}$	Angular frequency vector
$\boldsymbol{A}$	State matrix
$\boldsymbol{B}$	Input matrix
$\boldsymbol{C}$	Output matrix
$\boldsymbol{D}$	Feedthrough matrix
$\boldsymbol{x}$	State vector
$\boldsymbol{u}$	Input vector
$\boldsymbol{y}$	Output vector

n_{m_p}	example
n_{τ}	example

Acronyms

AC	Alternating Current
CIG	converter-interfaced generation
DAE	Differential Algebraic Equation
DC	Direct Current
EMT	Electro Magnetic Transient
ETSEIB	Escola Tècnica Superior d´Enginyeria Industrial de Barcelona
GUI	Guided User Interface
IGE	Induction Generator Effect
PF	Participation Factor
PSCAD	Power Systems Computer Aided Design
RMS	Root Mean Square
SSCI	Subsynchronous control interaction
SSR	Subsynchronous resonance

Preface

1 Introduction

1.1 Objectives

This thesis is part of the ongoing development of the dynamic simulation framework, with a focus on extending its capabilities to include small-signal stability analysis and foundational electromagnetic transient (EMT) modeling. The work is carried out within the Veragrid environment, where new modules and methodologies are being implemented. The main goal is to equip Veragrid with advanced tools for studying the dynamic behavior of power systems, integrating symbolic modeling, numerical routines, and graphical interfaces for analysis and visualization.

General Objectives

To develop and validate advanced methodologies for dynamic simulation and stability analysis of power systems by incorporating small-signal stability techniques and foundational EMT modeling, fully integrated into the Veragrid environment.

Specific Objectives

- To develop the small-signal analysis module for RMS models, including the formulation and linearization of differential-algebraic equations at the operating point, the computation of eigenvalues and participation factors from the Jacobian matrix, and its integration into Veragrid's graphical interface.

- To implement the foundational components for EMT simulation, modeling transmission lines and system elements in the abc domain, applying discretization techniques such as the Dommel algorithm and alternatives like the 2S-DIRK method, and validating the EMT solver using benchmark systems compared against commercial tools such as PSCAD.

- To extend symbolic system formulation to support custom models and control schemes, improving numerical routines, and ensuring consistent initialization of dynamic studies.

- To validate the developed methodologies through case studies, continuously comparing results with commercial tools to ensure model reliability and correctness.

1.2 Scope

This thesis is part of the ongoing development of Veragrid, a leading software platform for power system planning and simulation. The work focuses on improving dynamic simulation tools for modern grids, particularly in the context of small-signal stability and electromagnetic transient (EMT) modeling. Over a nine-month period—from September 2025 to May 2026—the project aims to build essential components that support symbolic formulation, numerical validation, and integration with existing simulation environments.

The first major area of focus is the implementation of small-signal stability analysis using RMS-based state-space models. This includes the computation of eigenvalues and participation factors, symbolic reduction of system equations, and integration of these routines into the VeraGrid graphical interface. The goal is to provide researchers and engineers with intuitive and accurate tools for identifying dominant modes and assessing system stability under varying conditions.

The second area involves the development of a foundational EMT solver in the abc domain. This includes modeling transmission lines and components, implementing discretization techniques such as the Dommel algorithm and two-stage diagonally implicit Runge-Kutta (2S-DIRK) methods, and benchmarking solver performance against commercial tools like PSCAD. Although the EMT module is not intended to be exhaustive, it serves as a proof of concept for future expansion and integration.

All development is conducted in Python, with an emphasis on code quality, symbolic computation, and reproducibility. The thesis also includes continuous benchmarking and validation using real-world data, including industrial cases. Technical supervision is provided by the eRoots team, ensuring alignment with architectural standards and long-term project goals.

1.3 State of the art

sdfsfgsfd

1.4 Veragrid

dfdfgassa

1.5 Previous requirements

Before starting this thesis, it was necessary to have a solid understanding of power system dynamics, numerical methods for differential equations, and programming in Python.

1.5.1 Dynamic framework in Veragrid

dfdg

1.5.2 Symbolic formulation

ewstrw

1.5.3 DAE

dsffg

2 Small-signal stability analysis

2.1 Power system stability

Power system stability is defined as that property of a power system that enables it to remain in a state of operating equilibrium under normal conditions and to regain an acceptable state of equilibrium after being subjected to a disturbance [**StabilityAndControlKundur**]. Other definitions state not only the state of equilibrium must be acceptable but also most system variables must be bounded so that practically the entire system remains intact[**kundurDef**].

Although the primary concern is the behavior of the interconnected system as a whole, the stability of individual components such as generators, motor loads, or regional subsystems; can be equally significant, particularly when localized instability does not propagate to the broader network. The system's dynamic behavior is governed by nonlinear interactions among its elements, and its response to perturbations is influenced by both the prevailing operating conditions and the specific nature of the disturbance. Stability is understood around an equilibrium point and is subject to change under small or large disturbances.

Power system stability is commonly classified depending on the pysical nature of the instability, the size of the disturbance considered and the devices, processes and time span that must be considered to assess stability [**kundurDef**]. The combination of these factors influence the methodologies, tools and considerations used in the analysis. The main categories of power system stability are described in the following enumeration.

- *Rotor angle stability*: The ability the ability of the interconnected synchronous machines in a power system to remain in synchronism under normal operating conditions and to regain synchronism after being subjected to a small or large disturbance [**StabilityAndControlKundur**]. A synchronous machine stays in synchronism when the electromagnetic torque exactly balances the mechanical torque from the prime mover, producing zero net accelerating torque. Stability therefore depends on the machine and its controls restoring that torque balance after a disturbance; failure to do so causes rotor acceleration or deceleration and loss of synchronism.

- *Voltage stability*: The capacity of the network to maintain acceptable voltage magnitudes at all buses during normal operation and following disturbances, such that voltages do not decrease sustainedly. Loss of this capacity manifests as progressive voltage drops and, ultimately, voltage collapse, which may force extensive load disconnection or generalised service interruption.

- *Frequency stability*: The capability of the system to preserve a near-nominal frequency following a major imbalance between generation and demand, through inertial response and secondary/tertiary control actions. Inadequate frequency stability results in sustained under-frequency or over-frequency transients that can damage equipment, trigger protective disconnections, and precipitate broader system failure.

Due to the incrising penetration of power electronics into the grid, two new categories of stability have been considered [**DefStabExtended**].

- *Converter-driven stability*: Refers to oscillatory behaviour caused by control interactions in converter-interfaced generation (CIG). Fast-interaction instabilities arise from high-frequency dynamics (hundreds of Hz to kHz) involving inner control loops and grid components. Slow-interaction instabilities occur at low frequencies (<10 Hz), driven by PLL and outer-loop controls, especially in weak grids. Synchronization issues and power transfer limits further compromise stability. These phenomena differ from classical generator dynamics and require tailored mitigation strategies

- *Ressonance stability*: Refers to the system's ability to withstand oscillatory energy exchange without magnifying voltage, current, or torque beyond safe limits. It includes subsynchronous resonance (SSR), which arises from interactions between series compensation and either mechanical shaft modes or electrical generator characteristics. The mechanical form leads to torsional resonance, while the electrical form—9 (Induction Generator Effect (IGE)) can cause self-excitation. Converter controls in DFIGs can exacerbate these effects, leading to subsynchronous control interaction (SSCI). These phenomena pose risks to both mechanical integrity and electrical equipment.

Rotor angle stability is inherent of classical power systems with synchronous machines. Modern power grids, with the increasing penetration of power electronics, have introduced new dynamics and interactions that can affect rotor angle stability. Converters decrease the inertia of the system and have an effect on the electromechanical modes. However, the fundamental principles of rotor angle stability remain intact and still takes a crucial role on stability analysis of power systems.

Therefore, understanding and analyzing rotor-angle stability remains essential for ensuring the overall reliability of power systems. Insufficient or negative synchronizing torque produces aperiodic, non-oscillatory transient instability that drives large rotor-angle deviations and is typically studied with time-domain numerical integration. In contrast, the absence of adequate damping torque gives rise to small-disturbance oscillatory instability.

In the context of this thesis, the focus is on rotor angle stability, particularly small-signal stability, its eigenvaule-based characterization, modal properties, and analysis methods. The following sections describe the theoretical background and methodologies used for small-signal stability analysis in power systems.

2.2 Stability of a dynamic system

A dynamic system is considered stable if, when subjected to a disturbance, it returns to its original state or to a new equilibrium state without exhibiting unbounded behavior. The equilibrium points are those states where all the derivatives $\dot{x}$ are zero, meaning the system is at rest or in a steady state.

Linearity affects on the stability of a system. The stability of a liner system is independent of the input and the initial conditions. However, for a non-linear system, stability depends on the magnitude of the input and initial conditions. Depending on the region of the state-space, stability is classified into the following categories:

- *Local stability*: the system is locally stable around an equilibrium point if when a small perturbation is applied, it remains around the equilibrium point. If as time increases it returns to the equilibrium point, it is locally asymptotically stable[**StabilityAndControlKundur**].
- *Finite stability*: the system is finitely stable if when a perturbation of finite size is applied, it remains bounded and does not diverge to infinity.
- *Global stability*: the system is globally stable if it returns to an equilibrium point for any initial condition in the whole state-space.

Therefore, linearizing a non-linear system around an equilibrium point allows to study its local stability as if it was a linear system.

2.3 Small-Signal stability

Small-signal stability refers to the ability of a power system to maintain synchronism when subjected to small disturbances [**StabilityAndControlKundur**], such as minor load changes or small faults. These small disturbances (typically within 1%)occur frequently in power systems and allow the linearization of non-linear system equations around a specific operating point in order to perform analysis. The resulting linear representation enables the use of standard control engineering tools to assess system stability and dynamic performance [**SmallSignalCheah**].

The resulting instability due to those small perturbations can have two forms: Non-oscillatory unstability defined as an increase in rotor angle due to insufficient synchronizing torque and oscillatory unstability, oscillations of increasing magnitude due to insufficient damping torque [**StabilityAndControlKundur**]. In practice, most of the instabilities come from insufficient damping torque. The following list summarizes the main oscillatory modes to consider:

- *Local modes*: Oscillations involving individual generators or small groups of units swinging against the rest of the system, typically localized near a generating station.

- *Inter area modes*: Low-frequency oscillations between large groups of generators in different regions, often linked by weak transmission corridors.

- *Control modes*: Oscillations arising from interactions between poorly tuned control systems—such as exciters, speed governors, HVDC converters, or static var compensators.

- *Torsional modes*: Oscillations associated with the mechanical shaft system of turbine-generators, which may become unstable due to interactions with control systems or series-compensated transmission lines.

Although small-signal analysis only applies to small variations around a fixed operating point, it remains a practical and widely used method for studying power system dynamics. By linearizing the system, it allows to apply control theory tools like eigenvalue analysis and state-space modeling. This helps identify poorly damped modes and assess how the system responds to small disturbances. Despite its limitations, it is a reliable approach for early detection of potential instabilities and for designing stabilizing controls.

2.4 State-space representation

Small-signal stability assessment methods are generally categorized into two main groups: state-space techniques and frequency-domain techniques. State-space techniques allow one to represent the system using a set of first order differential equations written in the following form:

$$\dot{x} = f(x, u, t) \tag{2.1}$$

Where x is the state column vector that stores the state variables, u is the input column vector that stores external signals that influence the system performance and t is time. The system can also not depend on time, that system is called time-invariant. It is also important to notw that state variables are the minimum amount of variables needed to represent the system and be able to compute its future behaviour.

Often the purpose of the state-space representation is to look at a set of the system variables, called outputs y. Then, a new expression is added to the state-space representation:

$$y = g(x, u, t) \tag{2.2}$$

Where y is the output column vector that stores the output variables.

A dynamic system can be described in many different ways depending on which variables are chosen as states, inputs, and outputs. These choices shape how the system behaves mathematically and how easily can it be analyzed. For example, using electrical quantities like voltage and current might be more practical for converter models, while mechanical variables such as rotor angle and speed are better suited for synchronous machines. The flexibility in selecting these variables allows to adapt the model to the specific goals of the study while still ensuring that the essencial dynamics of the system are captured.

State-space models are commonly represented in their matrix formulation as follows:

$$\dot{x} = Ax + Bu \tag{2.3}$$

$$y = Cx + Du \tag{2.4}$$

Where:

- x : state variables vector
- u : system inputs vector
- y : outputs vector
- A : state matrix
- B : input matrix
- C : output matrix
- D : direct transmission matrix

2.4.1 Linearization of state-space models

In order to linearize a non-linear state-space model, a small perturbation is applied around the operationg point (equilibrium point).

$$\mathcal{X} \overset{\triangle}{=} x - x^* \tag{2.5}$$

$$\mathcal{Y} \overset{\triangle}{=} y - y^* \tag{2.6}$$

$$\mathcal{U} \overset{\triangle}{=} u - u^* \tag{2.7}$$

Where x^*, y^* and u^* are the state, output and input vectors at the equilibrium point respectively. The new linearized state-space model is given by:

$$\dot{\mathcal{X}} = A\mathcal{X} + B\mathcal{U} \tag{2.8}$$

$$\mathcal{Y} = C\mathcal{X} + D\mathcal{U} \tag{2.9}$$

Where the new matrices are computed as the Jacobian matrices of the non linear system evaluated at the equilibrium point:

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \tag{2.10}$$

$$B = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_n} \end{bmatrix} \tag{2.11}$$

$$C = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_n}{\partial x_1} & \cdots & \frac{\partial g_n}{\partial x_n} \end{bmatrix} \tag{2.12}$$

$$D = \begin{bmatrix} \frac{\partial g_1}{\partial u_1} & \cdots & \frac{\partial g_1}{\partial u_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_n}{\partial u_1} & \cdots & \frac{\partial g_n}{\partial u_n} \end{bmatrix} \tag{2.13}$$

For simplicity, the perturbation notation is often omitted, and the linearized state-space model is expressed as:

$$\Delta\dot{x} = A\Delta x + B\Delta u \tag{2.14}$$

$$\Delta y = C\Delta x + D\Delta u \tag{2.15}$$

2.4.2 DAE to state-space representation

In power systems, the dynamic behaviour is mathematically represented by a set of Differential-Algebraic Equations (DAEs) that capture the interaction between dynamic components and network constraints. This formulation naturally arises because power systems combine elements with both dynamic and instantaneous responses.

- *Differential equations*: describe the time-dependent evolution of state variables associated with components that possess energy storage or control dynamics. These include synchronous generators (rotor angle and speed dynamics), excitation systems, governors, power electronic converters, and various control loops such as voltage and frequency regulators.

- *Algebraic equations*: represent the instantaneous electrical relationships and constraints imposed by the network. They stem primarily from Kirchhoff's laws, ensuring power balance and voltage-current consistency at each bus, as well as from static components like loads, transmission lines, and transformers, which are assumed to reach steady-state conditions instantaneously.

The explicit formulation of the DAE system is given by:

$$T\dot{x} = f(x, y) \tag{2.16}$$

$$0 = g(x, y) \tag{2.17}$$

Which is linearized around an equilibrium point as follows:

$$T\Delta\dot{x} = \frac{\delta f}{\delta x}\Delta x + \frac{\delta f}{\delta y}\Delta y \tag{2.18}$$

$$0 = \frac{\delta g}{\delta x}\Delta x + \frac{\delta g}{\delta y}\Delta y \tag{2.19}$$

From the second equation, Δy can be expressed in terms of Δx:

$$g_y \Delta y = -g_x \Delta x \rightarrow \Delta y = -g_y^{-1} g_x \Delta x \tag{2.20}$$

And then substituted into the first equation:

$$T\Delta\dot{x} = f_x \Delta x + f_y(-g_y^{-1} g_x \Delta x) \tag{2.21}$$

Rearranging the equation gives the linearized state-space representation and the expression for the state matrix A:

$$\Delta\dot{x} = T^{-1}(f_x - f_y g_y^{-1} g_x)\Delta x \rightarrow A = T^{-1}(f_x - f_y g_y^{-1} g_x) \tag{2.22}$$

The A matrix encapsulates the dynamic interactions between the system's state variables, accounting for both the intrinsic dynamics of the components and the constraints imposed by the network. From the state matrix the stability assessment can be performed as explained in the next section.

2.5 Stability assessment: Liapunov's first method

The stability of a system can be studied in large-signal and small-signal therms. Stability *in the large* needs to study the whole non-linear system. This method is complex and requires a high computational effort. On the other hand, stability *in the small* studies the system behaviour around an equilibrium point. This method is simpler and less computationally intensive, but it only provides information about the local stability of the system. Computing the eigenvalues of the state matrix A allows to determine the small-signal stability of the system.

Eigenvalue analysis and participation factors (PFs) are key tools for identifying dominant modes and evaluating system stability. These methods are well established in conventional power systems and are increasingly being applied to power-electronics-based systems, where dynamic behavior is often more complex and sensitive to operating conditions.

The eigenvalues λ of the state matrix A (commonly referred to as the system's modes) characterize its small-signal stability according to the following criteria:

- All modes satisfy $Re(\lambda) < 0$: the system is asymptotically stable
- All modes satisfy $Re(\lambda) \leq 0$: the system is marginally stable
- At least one mode satisfies $Re(\lambda) > 0$: the system is unstable

When a linearized system has complex conjugate modes, they represent oscillatory modes in the dynamic response:

- The real part determines damping:
  - $Re(\lambda) < 0$: exponential decay
  - $Re(\lambda) = 0$: oscillations persist indefinitely
  - $Re(\lambda) > 0$: exponential growth
- The imaginary part determines the oscillation frequency defined as: $f = \frac{Im(\lambda)}{2\pi}$

An other way to look at the damping of a mode is through the damping ratio ζ. The damping ratio is a dimensionless measure that describes how oscillations in a system decay after a disturbance. It is defined as the ratio of actual damping to critical damping. The critical damping is the minimum amount of damping that prevents oscillations. The damping ratio is given by:

$$\zeta = -\frac{Re(\lambda)}{\sqrt{Re(\lambda)^2 + Im(\lambda)^2}} \tag{2.23}$$

Where $Re(\lambda)$ is the real part of the eigenvalue and $Im(\lambda)$ is the imaginary part of the eigenvalue. The interpretation of the damping ratio is described below.

- $\zeta < 0$: *Unstable oscillations.* The system exhibits modes that grow exponentially with time, caused by eigenvalues located in the right half of the complex plane.
- $\zeta = 0$: *Marginal stability.* The system produces undamped, sustained oscillations since the eigenvalues lie exactly on the imaginary axis.

- $0 < \zeta < 1$: *Stable oscillatory response.* The system returns to its equilibrium point through oscillations that gradually decay over time. In practical terms, a damping ratio of about $\zeta = 0.05$ is generally considered sufficient for well-damped behaviour.

- $\zeta = 1$: *Critical damping.* The system returns to equilibrium without oscillations, reaching the steady state in the shortest possible time without overshoot.

Finally, participation factors quantify the relative influence of each state variable on the different dynamic modes of the system. In essence, they indicate how much a given state contributes to a specific mode and, conversely, how strongly that mode affects the state. This dual interpretation makes participation factors a valuable tool for understanding the internal structure of system dynamics.

In the context of power systems, participation factors play a key role in identifying the physical origin of oscillations and instabilities. By analysing these factors, it is possible to determine which components—such as generators, controllers, or converter units—are most involved in poorly damped or unstable modes. This information supports targeted actions for control tuning, model validation, and stability improvement. Participation factors are calculated as follows:

$$PF_{i,k} = W_{i,k} \cdot V_{i,k} \tag{2.24}$$

Where:

- $PF_{i,k}$ is the participation factor of the k-th state variable to the i-th mode.

- $W_{i,k}$ is the left eigenvector of the k-th state variable to the i-th mode of matrix A. It satisifies $w^T A = \lambda w^T$.

- $V_{i,k}$ is the right eigenvector of the k-th state variable to the i-th mode of matrix A. It satisifies $Av = \lambda v$.

3 Grid Model

Once all the elements of the grid have been defined, the model of the power flow can be properly explained. The approach chosen is aligned with the models proposed by GridCal, which make use of the Universal Branch Model to describe the admittance matrices of the branches. The power flow model used during this project has been the Bus Injection Model (BIM), which will be calculated at each bus of the grid, and has to be equal to zero for the system to be in equilibrium. This model relates voltages, power transfer, tap variables, generation and demand, which are the variables of the optimization problem.

The notation used throughout this work is as follows: vectors are denoted in bold (can be both upper or lowercase), matrices are denoted in uppercase, and scalars are denoted in lowercase. Vectors that are transformed in diagonal matrices appear within square brackets. Vectors that slice other vectors using their values as indexes appear after the slice vector within curly brackets.

3.1 Power Flow Equations

The full construction of the power flow equations is shown, since some intermediate steps can be useful for later purposes. The first important equation is the power transfer that occurs between two nodes connected with branch, modelled using the Flexible Universal Branch Model for the k branch as in Equations (**??**) and (**??**). They contain the information of a single line. In the grid model used, y_{ff}, y_{ft}, y_{tf} and y_{tt} are vectors of length l that store these primitives of the admittance matrix for all the branches in the model.

Consider the connectivity matrices C_f and C_t, with size k x l that relate the index of the 'from' and 'to' buses to the index of their branch as follows:

$$C_{f_{ki}} = \begin{cases} 1, & \text{for } k == (i,j) \quad \forall j \in \boldsymbol{N} \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

$$C_{t_{kj}} = \begin{cases} 1, & \text{for } k == (i,j) \quad \forall i \in \boldsymbol{N} \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

Using these connectivity matrices and the primitives, $from$ and to admittance matrices can be constructed. The to admittance matrix contains the elements of the admittance matrix that are connected to the 'to' bus of the branch, while the $from$ admittance matrix contains the elements connected to the 'from' bus of the branch. The bus admittance matrix Y_{bus}, which contains all the information needed to calculate the power flow, can then be constructed.

$$\begin{aligned} Y_f &= [\boldsymbol{Y_{ff}}]\, C_f + [\boldsymbol{Y_{ft}}]\, C_t \\ Y_t &= [\boldsymbol{Y_{tf}}]\, C_f + [\boldsymbol{Y_{tt}}]\, C_t \\ Y_{bus} &= C_f^{\top} Y_f + C_t^{\top} \boldsymbol{Y_t} + [\boldsymbol{Y_{sh}}] \end{aligned} \tag{3.3}$$

where the brackets indicate that the vector is transformed into a diagonal matrix and the $[\boldsymbol{Y_{sh}}]$ matrix contains the shunt admittance of each bus, which is calculated for each bus by GridCal modeller. The resulting matrices have size *m* x *n*.

The model used to calculate the power balance of the system is the bus injection model, which is based on the equality between the power injected in a bus and power flowing out of it. The injection can come from generators connected at the bus or lines from buses while the outflow goes to demand connected to the bus or lines that transport energy to other buses.

The vector of complex power injections from lines $\boldsymbol{S_{bus}}$ is given by the following equations:

$$\boldsymbol{S_{bus}} = [\boldsymbol{V}](Y_{bus}\boldsymbol{V})^* \tag{3.4}$$

where $\boldsymbol{V}$ is the vector of the complex voltages of the buses calculated as:

$$\boldsymbol{\mathcal{V}} e^{j\boldsymbol{\theta}} \tag{3.5}$$

using element to element multiplication, and being $\mathcal{V}$ the vector of voltage modules and θ the vector of voltage angles. The complete balance can be now calculated as:

$$\boldsymbol{G^S_{bus}} = \boldsymbol{S_{bus}} - \boldsymbol{S^g_{bus}} + \boldsymbol{S^d_{bus}} = \boldsymbol{0} \quad \text{(equilibrium)} \tag{3.6}$$

where $\boldsymbol{S^g_{bus}}$ is the power generated at the buses, and $\boldsymbol{S^d_{bus}}$ the power consumed at the buses. The notation $\boldsymbol{G^S}$ will be useful later to refer to this balance as an equality constraint of the optimization problem.

One particularity to be considered is that there could be multiple generators or loads connected to the same bus. For the present work, there is no segregation of loads and the bus demand is obtained directly from GridCal, but it could be consider to model load flexibility and shedding. For generations, the separation has to be done as there could be differences in price or operational limits. To do so, $\boldsymbol{S^g}$ is obtained through the use of the *n* x *ng* connectivity matrix C_G:

$$\boldsymbol{S^{g}_{bus}} = C_G \boldsymbol{S^{g}} \tag{3.7}$$

For buses that are part of a DC link, the power balance has to account for the power transferred through the link.

$$\begin{aligned} \boldsymbol{G^{S}_{bus}}\{\boldsymbol{fdc}\} &\mathrel{+}= \boldsymbol{P_{dc}}\{\boldsymbol{dc}\} \\ \boldsymbol{G^{S}_{bus}}\{\boldsymbol{tdc}\} &\mathrel{-}= \boldsymbol{P_{dc}}\{\boldsymbol{dc}\} \end{aligned} \tag{3.8}$$

In addition to the bus power balance, the line power balance has to be considered to ensure that operational limits of the lines are not exceeded. The power flowing through a line can be calculated using the $from$ and to admittance matrices and the voltages of the buses connected to the line. Both sides of the lines have to be considered, since the powers at the ends are not equal and the direction of the flow can change depending on grid conditions.

$$\begin{aligned} \boldsymbol{V_f} &= C_f \boldsymbol{V} \\ \boldsymbol{V_t} &= C_t \boldsymbol{V} \\ \boldsymbol{S_f} &= [\boldsymbol{V_f}](Y_f \boldsymbol{V})^* \\ \boldsymbol{S_t} &= [\boldsymbol{V_t}](Y_t \boldsymbol{V})^* \end{aligned} \tag{3.9}$$

Here it is important to note that the vector $\boldsymbol{V_f}$ will contain the voltage value of the $from$ bus of each line, which means there can be some buses appearing more than one time.

3.2 Operational limits

Each element of the grid has its owns operational limits that have to be respected. The limits are stored in lists of length equal to the respective element count. These limits correspond to inequalities of the optimization problem. The notation $\boldsymbol{H^{I}}$ is used to identify to each different limit.

3.2.1 Nodal voltage limits

The voltage module limits of the buses are stored in the $\boldsymbol{V_{max}}$ and $\boldsymbol{V_{min}}$ arrays, with its value in per unit. The limits are checked using the following equations:

$$\begin{aligned} \boldsymbol{H^{v_u}} &= \boldsymbol{\mathcal{V}} - \boldsymbol{V_{max}} \leq \boldsymbol{0} \quad \text{(upper limit)} \\ \boldsymbol{H^{v_l}} &= \boldsymbol{V_{min}} - \boldsymbol{\mathcal{V}} \leq \boldsymbol{0} \quad \text{(lower limit)} \end{aligned} \tag{3.10}$$

3.2.2 Generation limits

The generation limits apply to both the active and reactive power of the generators. They are stored in the P_{max}, P_{min}, Q_{max} and Q_{min} lists, with its value in per unit. The limits are checked using the following equations:

$$\begin{aligned} \boldsymbol{H^{P_u}} &= \boldsymbol{P_g - P_{max} \leq 0} \quad \text{(upper limit)} \\ \boldsymbol{H^{P_l}} &= \boldsymbol{P_{min} - P_g \leq 0} \quad \text{(lower limit)} \\ \boldsymbol{H^{Q_u}} &= \boldsymbol{Q_g - Q_{max} \leq 0} \quad \text{(upper limit)} \\ \boldsymbol{H^{Q_l}} &= \boldsymbol{Q_{min} - Q_g \leq 0} \quad \text{(lower limit)} \end{aligned} \tag{3.11}$$

An additional constraint over the generation is the reactive power limitations due to the generator capability curves. These curves can be modelled by piecewise defined function, but in the formulation of the problem they have been considered much simpler, and by default they are not applied. The following constraint models the reactive power limits of the generators:

$$\boldsymbol{H^{ctQ}} = \boldsymbol{Q_g^2 - tanmax^2 P_g^2 \leq 0} \tag{3.12}$$

where $\boldsymbol{tanmax}$ is the maximum value of the tangent of the angle between the active and reactive power of the generator.

3.2.3 Transformer limits

The controllable transformers will have their limits for the controllable variables limited. The limits are similar to the previous:

$$\begin{aligned} \boldsymbol{H^{m_{p_u}}} &= \boldsymbol{m_p - m_{p_{max}} \leq 0} \quad \text{(upper limit)} \\ \boldsymbol{H^{m_{p_l}}} &= \boldsymbol{m_{p_{min}} - m_p \leq 0} \quad \text{(lower limit)} \\ \boldsymbol{H^{\tau_u}} &= \boldsymbol{\tau - \tau_{max} \leq 0} \quad \text{(upper limit)} \\ \boldsymbol{H^{\tau_l}} &= \boldsymbol{\tau_{min} - \tau \leq 0} \quad \text{(lower limit)} \end{aligned} \tag{3.13}$$

3.2.4 Line limits

The line limits will be checked only for the monitored branches, which is a subset of size m from the total number of branches l, considering the rating of each line in per unit. Since the direction of the flow is not known, and to avoid using absolute values, this constraint is considered quadratically.

$$\begin{aligned} \boldsymbol{H^{S_f}} &= \boldsymbol{S_f S_f^* - S_{max}^2 \leq 0} \\ \boldsymbol{H^{S_t}} &= \boldsymbol{S_t S_t^* - S_{max}^2 \leq 0} \end{aligned} \tag{3.14}$$

3.2.5 Bound slack variables

So far, all the constraints described has physical meaning and are related to the grid model. The last set of constraints are related to the optimization problem and are used to improve, or even make possible, the convergence of the solver. The bound slacks are used to allow the solution to surpass some operational limits in order to find a feasible solution. Of course, it should come at a cost and the objective function will be used to penalize this behavior.

These bound slacks are applied to the voltage limits and to the line limits, allowing slight over or undervoltages and some overloads. The constraint equations are modified, and the slack variables are imposed to be positive. The equations are as follows:

$$\begin{aligned} \boldsymbol{H^{v_u}} &= \boldsymbol{\mathcal{V} - V_{max} - sl_{v_{max}} \leq 0} \quad \text{(upper limit)} \\ \boldsymbol{H^{v_l}} &= \boldsymbol{V_{min} - \mathcal{V} - sl_{v_{min}} \leq 0} \quad \text{(lower limit)} \\ \boldsymbol{H^{S_f}} &= \boldsymbol{S_f S_f^* - S_{max}^2 - sl_{sf} \leq 0} \quad \text{(from limit)} \\ \boldsymbol{H^{S_t}} &= \boldsymbol{S_t S_t^* - S_{max}^2 - sl_{st} \leq 0} \quad \text{(to limit)} \\ \boldsymbol{-H^{sl_{v_{max}}}} &\boldsymbol{\leq 0} \\ \boldsymbol{-H^{sl_{v_{min}}}} &\boldsymbol{\leq 0} \\ \boldsymbol{-H^{sl_{sf}}} &\boldsymbol{\leq 0} \\ \boldsymbol{-H^{sl_{st}}} &\boldsymbol{\leq 0} \end{aligned} \tag{3.15}$$

The use of this bound slacks is optional and can be removed from the optimization problem if the user prefers to have a solution in compliance with the established limits.

4 Optimization Problem

This section describes the optimization model used to solve the AC Optimal Power Flow problem. The mathematical formulation of this problem is presented using abstract notation of the Karush-Kuhn-Tucker (KKT) conditions for optimization problems, which later on will regain a more practical form. The implementation of the solver is also described, as well as the pseudocode used to solve the problem. The general theory of optimization can be found in the book of Boyd and Vandenberghe [1], with a more detailed explanation of the IPM in Chapter 19 of Nocedal and Wright's book [2], and the adaptation of this theory to a more compact formulation can be reviewed in the Matpower Interior Point Solver documentation [3] [4].

4.1 Optimization problem

The general optimization problem involving equality and inequality constraints can be formulated as follows:

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & \boldsymbol{G}(\boldsymbol{x}) = \mathbf{0} \\ & \boldsymbol{H}(\boldsymbol{x}) \leq \mathbf{0} \end{aligned} \tag{4.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, being n the number of decision variables, $f(x)$ is the function to be minimized which typically accounts for the generation cost, $\boldsymbol{G}(\boldsymbol{x})$ is the set of n_e equality constraints of the problem, and $\boldsymbol{H}(\boldsymbol{x})$ is the set of n_i inequalities.

The problem will be solved using the barrier parameter method, which consists of solving the optimization problems sequencially with different values of the barrier parameter γ. The barrier parameter is an IPM methodology used to penalize the violation of the inequality constraints in the problem, which are turned into equalities of the form $\boldsymbol{H}(\boldsymbol{x}) + \boldsymbol{Z} = \mathbf{0}$ by the addition of the positive slack variables $\boldsymbol{Z}$.

The Lagrangian of the problem is defined by:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{Z}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^\top \boldsymbol{G}(\boldsymbol{x}) + \boldsymbol{\mu}^\top (\boldsymbol{H}(\boldsymbol{x}) + \boldsymbol{Z}) - \gamma \sum_{i=1}^{n_i} \log(z_i) \tag{4.2}$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are the Lagrange multipliers associated with the equality and inequality constraints respectively, the superindex $\top$ is used to indicate the transposition of a vector and the elements of the sum z_i correspond to each value of the vector $\boldsymbol{Z}$.

The KKT conditions are a set of equations that must be satisfied by the solution of the optimization problem. These conditions are a result of imposing that the partial derivatives of the Lagrangian in the (x, Z, λ, μ) variable space equals 0, as shown in Equation (4.4).

$$\nabla_{x,Z,\lambda,\mu}\mathcal{L}(\boldsymbol{x}, \boldsymbol{Z}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathbf{0} \tag{4.3}$$

$$\begin{aligned}
\boldsymbol{\mathcal{L}}_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{Z}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &= f_x + \boldsymbol{\lambda}^\top G_x + \boldsymbol{\mu}^\top H_x = \mathbf{0} \\
\boldsymbol{\mathcal{L}}_{\boldsymbol{Z}}(\boldsymbol{x}, \boldsymbol{Z}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &= \boldsymbol{\mu}^\top - \gamma \mathbf{1}_{n_i}^\top [\boldsymbol{Z}]^{-1} = \mathbf{0} \\
\boldsymbol{\mathcal{L}}_{\boldsymbol{\lambda}}(\boldsymbol{x}, \boldsymbol{Z}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &= \boldsymbol{G}^\top(\boldsymbol{x}) = \mathbf{0} \\
\boldsymbol{\mathcal{L}}_{\boldsymbol{\mu}}(\boldsymbol{x}, \boldsymbol{Z}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &= \boldsymbol{H}^\top(\boldsymbol{x}) + \boldsymbol{Z}^\top = \mathbf{0}
\end{aligned} \tag{4.4}$$

where $\mathbf{1}_{n_i}$ is a vector of ones of length n_i, and γ is the barrier parameter that will be updated in each iteration of the solver until reaching a value close to zero. The notation $f_{\boldsymbol{X}}$ indicates the gradient of the given function f with respect to the vector $\boldsymbol{X}$.

The first gradient corresponds to the stationary condition of the problem, the second gradient corresponds to the complementary slackness, and the other two gradients correspond to the primal feasibility conditions of the problem. These conditions, alongside the dual feasibility condition $Z_i, \mu_i \geq 0 \quad \forall i$, are necessary and sufficient for the solution of the optimization problem. The resulting set of equations will be solved computationally due to the non-convex nature of the problem. For this type of general problem, the approach chosen is the use of an Interior Point Method (IPM) solver.

4.2 Interior Point Method solver

The choice of this type of solver is regarded as the best option to deal with non-linear optimization, as shown in the dedicated chapter 19 in Nocedal and Wright's book. The IPM solver is a type of optimization algorithm that solves the KKT conditions of the problem iteratively. There are multiple algorithms that can be used, although the one presented in this project is the Newton-Raphson. Since this algorithm will be used by the Spanish TSO for country-sized grids, the scalability of the solver is a key factor to consider. In reference [5], this algorithm is compared with the Gauss-Seidel algorithm, yielding better scalability for the Newton-Raphson method.

Some optimization solver packages such as IPOPT were considered, but creating a tailored solver and integrating it in GridCal was considered the best option, as it avoids depending on third-party software, while also allowing the customization of its behavior.

There can be improvements to this method as presented in the literature, but for the purpose of this project, the IPM has been implemented directly using the Newton-Raphson method with a step-control mechanism.

The Newton-Raphson iterative process that will find the roots of the KKT conditions can be described as in Algorithm 1.

Algorithm 1 Newton-Raphson Iterative Process

```
Initialize $\boldsymbol{x}_0$, tolerance $\epsilon$, and set $k = 0$
while $||\boldsymbol{f}(\boldsymbol{x}_k)|| > \epsilon$ do
    Compute the Jacobian matrix $J(\boldsymbol{x}_k)$
    Solve $J(\boldsymbol{x}_k)\Delta\boldsymbol{x}_k = -\boldsymbol{f}(\boldsymbol{x}_k)$ for $\Delta\boldsymbol{x}_k$
    Apply step control to $\Delta\boldsymbol{x}_k$
    Update $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta\boldsymbol{x}_k$
    $k = k + 1$
end while
```

The algorithm will perform several Newton-Raphson steps until the convergence criteria is met, meaning that equations (4.4) are solved. The expanded formulation of the system solved at each step is:

$$
-\begin{pmatrix} \boldsymbol{\mathcal{L}_{xx}} & \boldsymbol{0} & \boldsymbol{G}_{\boldsymbol{x}}^{\top}(\boldsymbol{x}) & \boldsymbol{H}_{\boldsymbol{x}}^{\top}(\boldsymbol{x}) \\ \boldsymbol{0} & [\boldsymbol{\mu}] & \boldsymbol{0} & [\boldsymbol{Z}] \\ \boldsymbol{G}_{\boldsymbol{x}}(\boldsymbol{x}) & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{H}_{\boldsymbol{x}}(\boldsymbol{x}) & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\Delta x} \\ \boldsymbol{\Delta Z} \\ \boldsymbol{\Delta \lambda} \\ \boldsymbol{\Delta \mu} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mathcal{L}}_{x}^{\top} \\ [\boldsymbol{\mu}]\boldsymbol{Z} - \gamma\boldsymbol{1}_{n_i} \\ \boldsymbol{G}(\boldsymbol{x}) \\ \boldsymbol{H}(\boldsymbol{x}) + \boldsymbol{Z} \end{pmatrix} \tag{4.5}
$$

where $\boldsymbol{\Delta x}$ represents the variation of $\boldsymbol{x}$ (and similarly to $\boldsymbol{s}$, $\boldsymbol{y}$, $\boldsymbol{z}$), $\boldsymbol{I}$ is the identity matrix, and the term $\boldsymbol{\mathcal{L}_{xx}}$ is the derivative of the first KKT condition, which was already the gradient of the Lagrangian of the problem. This Hessian matrix is the element that has the most computational cost to be calculated. Its calculation will have to be the most optimized possible to ensure the solver's scalability.

Note that dual feasibility condition $Z_i, \mu_i \geq 0 \quad \forall i$ is not included in the system of equations. This condition will be enforced by the algorithm when choosing the step length of the iteration when updating the variables.

This system of equations can be further reduced using methods that can be found in [2], derived with detail in [3]. The relevant equations obtained during the reduction process are shown in Equations (4.6) and (4.7).

$$\begin{cases} [\boldsymbol{\mu}]\Delta \boldsymbol{Z} + [\boldsymbol{Z}]\Delta\boldsymbol{\mu} & = -[\boldsymbol{\mu}]\boldsymbol{Z} + \gamma \boldsymbol{1}_{n_i} \\ [\boldsymbol{Z}]\Delta\boldsymbol{\mu} & = -[\boldsymbol{Z}]\boldsymbol{\mu} + \gamma \boldsymbol{1}_{n_i} - [\boldsymbol{\mu}]\Delta \boldsymbol{Z} \\ \Delta\boldsymbol{\mu} & = -\boldsymbol{\mu} + [\boldsymbol{Z}]^{-1}(\gamma \boldsymbol{1}_{n_i} - [\boldsymbol{\mu}]\Delta \boldsymbol{Z}). \end{cases} \tag{4.6}$$

$$\begin{cases} \boldsymbol{H}_x \Delta \boldsymbol{X} + \Delta \boldsymbol{Z} & = -\boldsymbol{H}(\boldsymbol{X}) - \boldsymbol{Z} \\ \Delta \boldsymbol{Z} & = -\boldsymbol{H}(\boldsymbol{X}) - \boldsymbol{Z} - \boldsymbol{H}_x \Delta \boldsymbol{X}. \end{cases} \tag{4.7}$$

$$\begin{cases} \boldsymbol{M} & \equiv \boldsymbol{\mathcal{L}}_{\boldsymbol{xx}} + \boldsymbol{H}_{\boldsymbol{x}}^{\top}[\boldsymbol{Z}]^{-1}[\boldsymbol{\mu}]\boldsymbol{H}_{\boldsymbol{x}} \\ & = \boldsymbol{f}_{\boldsymbol{xx}} + \boldsymbol{G}_{\boldsymbol{xx}}(\boldsymbol{\lambda}) + \boldsymbol{H}_{\boldsymbol{xx}}(\boldsymbol{\mu}) + \boldsymbol{H}_{\boldsymbol{x}}^{\top}[\boldsymbol{Z}]^{-1}[\boldsymbol{\mu}]\boldsymbol{H}_{\boldsymbol{x}}. \end{cases} \tag{4.8}$$

$$\begin{cases} \boldsymbol{N} & \equiv \boldsymbol{\mathcal{L}}_{\boldsymbol{x}}^{\top} + \boldsymbol{H}_{\boldsymbol{x}}^{\top}[\boldsymbol{Z}]^{-1}(\gamma \boldsymbol{1}_{\boldsymbol{n_i}} + [\boldsymbol{\mu}]\boldsymbol{H}(\boldsymbol{x})) \\ & = \boldsymbol{f}_{\boldsymbol{x}}^{\top} + \boldsymbol{G}_{\boldsymbol{x}}^{\top}\boldsymbol{\lambda} + \boldsymbol{H}_{\boldsymbol{x}}^{\top}\boldsymbol{\mu} + \boldsymbol{H}_{\boldsymbol{x}}^{\top}[\boldsymbol{Z}]^{-1}(\gamma \boldsymbol{1}_{\boldsymbol{n_i}} + [\boldsymbol{\mu}]\boldsymbol{H}(\boldsymbol{x})). \end{cases} \tag{4.9}$$

This transformation will make possible to reduce the size of Equation (4.5) to a system of equations over $(\boldsymbol{x}, \boldsymbol{\lambda})$ instead, and then calculating the terms corresponding to the inequalities using the expressions shown in Equation (4.6). This smaller size reduces the computational time of computing such a large system. The resulting reduced system is the following:

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{G}_{\boldsymbol{x}}^{\top} \\ \boldsymbol{G}_{\boldsymbol{x}} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{N} \\ -\boldsymbol{G}(\boldsymbol{x}) \end{bmatrix} \tag{4.10}$$

The solver will firstly solve this smaller matricial problem to obtain the variation of the decision variables and the Lagrange multipliers associated with the equality constraints, and then it will compute the inequality terms using Equations (4.6) and (4.7).

Before proceeding to update the values of the variables, there will be two additional steps to be taken. The first one is to ensure that the dual feasibility condition

$$\boldsymbol{Z_i}, \boldsymbol{\mu_i} \geq 0 \quad \forall i$$

is met. This will be done by using the maximum step length such that the values of the multipliers remain positive. The calculation of this step length is shown at Equation (4.11).

$$
\begin{aligned}
\alpha_p &= \min\left(\tau \cdot \min\left(-\frac{\boldsymbol{Z}}{\Delta \boldsymbol{Z}}\right), 1\right) \\
\alpha_d &= \min\left(\tau \cdot \min\left(-\frac{\boldsymbol{\mu}}{\Delta \boldsymbol{\mu}}\right), 1\right)
\end{aligned}
\tag{4.11}
$$

where τ is a parameter that will be set to a value really close to 1, without exceeding it (i.e. 0.9995).

This step length will be tested to ensure that it is not too large as it could lead to divergence of the algorithm. This will be done by using a step control mechanism that will reduce the step length if the conditions are not met. The step control mechanism will update just the decision variables x, using the Lagrangian of the system as a reference. The step control mechanism is described in Algorithm 2.

Algorithm 2 Step Control

```
Initialize $\mathcal{L}_0 = f(\boldsymbol{x}) + \boldsymbol{\lambda}^\top \boldsymbol{G}(\boldsymbol{x}) + \boldsymbol{\mu}^\top \boldsymbol{H}(\boldsymbol{x}) - \gamma \sum_{i=1}^{n_i} \log(Z_i)$, $\alpha = 1$, use its gradient $\boldsymbol{\mathcal{L}_x}$ and hessian $\boldsymbol{\mathcal{L}_{xx}}$ previously used for the Jacobian, and set $\alpha = trust$
while $i < step\_control\_iterations$ do
    $\Delta \boldsymbol{x}_1 = \alpha \Delta \boldsymbol{x}$
    $\boldsymbol{x}_1 = \boldsymbol{x} + \Delta \boldsymbol{x}_1$
    Compute $\mathcal{L}_1$ and $\mathcal{L}_{x_1}$
    $\rho = \frac{\mathcal{L}_1 - \mathcal{L}}{\boldsymbol{\mathcal{L}_x} \Delta \boldsymbol{x_1} + 0.5 \Delta \boldsymbol{x}_1^\top \boldsymbol{\mathcal{L}_{xx}} \Delta \boldsymbol{x_1}}$
    if $\rho \leq \rho_{lower}$ or $\rho \geq \rho_{upper}$ then
        $i = i + 1$
        $\alpha = \alpha / 2$
    end if
end while
```

This will reduce all the step sizes if the new Lagrangian does not accomplish the condition. If this condition is not activated, then $\alpha = 1$. Now, the values of all the variables and multipliers can be updated with the final values of the step length:

$$
\begin{aligned}
\boldsymbol{x} &\leftarrow \boldsymbol{x} + \alpha \alpha_p \Delta \boldsymbol{x} \\
\boldsymbol{Z} &\leftarrow \boldsymbol{Z} + \alpha \alpha_p \Delta \boldsymbol{Z} \\
\boldsymbol{\lambda} &\leftarrow \boldsymbol{\lambda} + \alpha \alpha_d \Delta \boldsymbol{\lambda} \\
\boldsymbol{\mu} &\leftarrow \boldsymbol{\mu} + \alpha \alpha_d \Delta \boldsymbol{\mu}
\end{aligned}
\tag{4.12}
$$

The barrier parameter is updated after this new point is calculated using:

$$\gamma \leftarrow \sigma \frac{\boldsymbol{Z}^{\top} \boldsymbol{\mu}}{n_i} \tag{4.13}$$

where σ is a parameter that will be between 0 and 1. The value of σ will be set to 0.1, as in [3], although Nocedal and Wright propose some different approaches with different values of this parameter when the solver approaches the solution.

Finally, the last step of the solver will be to calculate the convergence criteria to determine if the algorithm can be stopped. There are three criteria that will be used to determine if the algorithm: the feasibility condition, the gradient condition and the gamma value. All of them have to be below a tolerance ϵ to consider the problem solved. The later condition can be checked performing direct comparison, while the former two can be calculated as [4]:

$$\begin{aligned} feascond &= \frac{\max\left([\|\mathbf{G}\|_{\infty}, \max(\boldsymbol{H})]\right)}{1 + \max\left([\|\mathbf{x}\|_{\infty}, \|\mathbf{Z}\|_{\infty}]\right)} \\ gradcond &= \frac{\|\mathcal{L}_{\mathbf{x}}\|_{\infty}}{1 + \max\left([\|\lambda\|_{\infty}, \|\mu\|_{\infty}]\right)} \end{aligned} \tag{4.14}$$

which monitor the compliance of the equality and inequality constraints for the $feascond$ and the gradient of the Lagrangian for the $gradcond$. The algorithm will continue to iterate until the convergence criteria is met, although there is a maximum number of iterations parameter to avoid infinite loops.

4.3 Python implementation

The resulting algorithm is implemented in Python as a stand-alone optimization solver. The solver is capable of solving any kind of optimization problem as long as the objective function, equality constraints, and inequality constraints are provided alongside their gradients and Hessians. The user will have to provide a function that returns an object containing these equations, as well as the desired initialization and tolerance values. The objective of this project is to solve electrical systems, but any other type of system correctly modelled can also be solved using this solver.

The algorithm is described by the block diagram shown in Figure 4.1, representing the complete algorithm including all the steps. This solver will be called by a parent function that is responsible to model the whole problem and contains the initialization of the variables and the constraints and the functions that compute the value of the objective function, equalities, inequalities, and their Jacobians and Hessians.

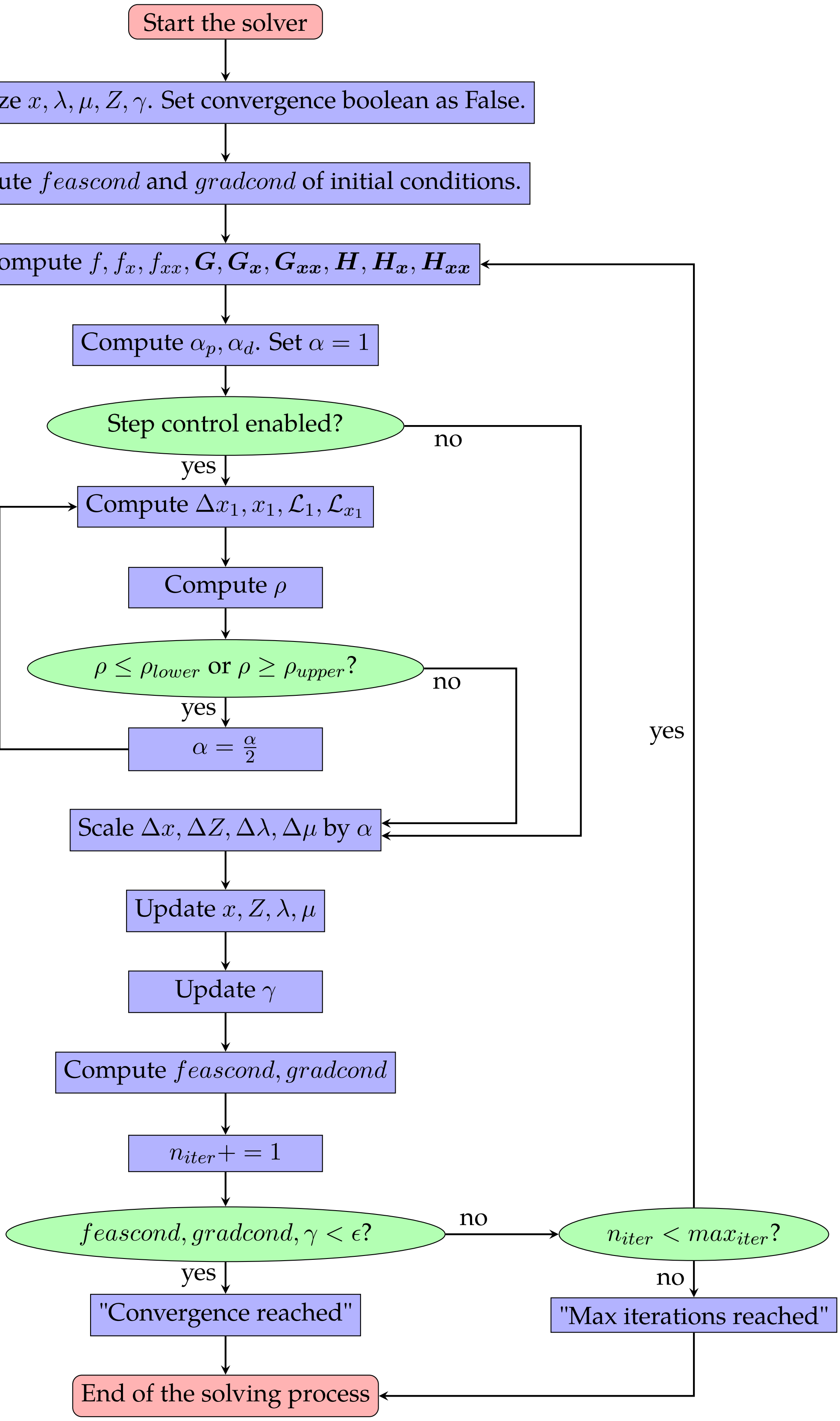

Start the solver
Initialize $x, \lambda, \mu, Z, \gamma$. Set convergence boolean as False.
Compute $feascond$ and $gradcond$ of initial conditions.
Compute $f, f_x, f_{xx}, \boldsymbol{G}, \boldsymbol{G_x}, \boldsymbol{G_{xx}}, \boldsymbol{H}, \boldsymbol{H_x}, \boldsymbol{H_{xx}}$
Compute α_p, α_d. Set $\alpha = 1$
Step control enabled?
no
yes
Compute $\Delta x_1, x_1, \mathcal{L}_1, \mathcal{L}_{x_1}$
Compute ρ
$\rho \leq \rho_{lower}$ or $\rho \geq \rho_{upper}$?
no
yes
$\alpha = \frac{\alpha}{2}$
yes
Scale $\Delta x, \Delta Z, \Delta\lambda, \Delta\mu$ by α
Update x, Z, λ, μ
Update γ
Compute $feascond, gradcond$
$n_{iter} += 1$
$feascond, gradcond, \gamma < \epsilon$?
no
$n_{iter} < max_{iter}$?
yes
no
"Convergence reached"
"Max iterations reached"
End of the solving process

Figure 4.1: Block diagram of the solver.

4.3.1 Dealing with sparsity

A small comment to be made is that the solver will work with sparse matrices. The usage of this type of matrices available in the SciPy package [6] will allow the solver to be more efficient when dealing with the large systems that result when modelling power grids that are composed of thousands of buses and lines.

This type of matrix is used to store only the non-zero elements of the matrix, not wasting memory in storing the zeros. For instance, the Jacobian obtained in the IEEE14 test case has a size of 129 x 129, although only 794 of the elements are non-zero, representing 4.8% of the total elements.

The larger the system, the lower this fraction of non-zero values is. A case of 300 buses used as a benchmark contains 0.3% of non-zero values in the Jacobian. For this reason, the usage of sparse matrices is extremely important to solve country-sized power grids.

5 AC-OPF

In this section, the development of the AC-OPF model for the IPM solver is presented. As explained earlier, the aim of the project is to solve the power flow without simplification of any equation that can give unfeasible results, while giving the grid operator all the information that can be important to decide the operation of the grid, as for instance how the generation should be distributed among the dispatchable generators in order to have an optimal solution while accomplishing limits, or the setpoints that can be given to the transformers that have controllability.

The development of the model is crucial in order to have a solid model that encapsulates all the constraints of the grid, allowing the easy addition of new constraints that can be considered such as the power curves of the generators, that limit the reactive power generation depending on the active power generation.

The model is developed starting from Matpower's derivation for the traditional power flow variables in [7], and then the additional features developed in this project are introduced using a similar framework.

5.1 Decision variables

The power flow is modelled using polar form, since it is more compact to use compared to the rectangular form. The first group of variables correspond to the traditional power flow variables. Then, the bound slack variables are introduced for the line flow limits and the voltage limits. The transformer variables are added next, and finally the DC links $from$ powers. The resulting variable vector is:

$$\boldsymbol{x} = [\boldsymbol{\theta}^T, \boldsymbol{\mathcal{V}}^T, \boldsymbol{P_g}^T, \boldsymbol{Q_g}^T, \boldsymbol{sl_{sf}}^T, \boldsymbol{sl_{st}}^T, \boldsymbol{sl_{vmax}}^T, \boldsymbol{sl_{vmin}}^T, \boldsymbol{m_p}^T, \boldsymbol{\tau}^T, \boldsymbol{P_{DC}}^T]^T \tag{5.1}$$

where $\boldsymbol{\theta}$ is the vector of size n of voltage angles, $\boldsymbol{\mathcal{V}}$ is the vector of size n of voltage magnitudes, $\boldsymbol{P_g}$ and $\boldsymbol{Q_g}$ are the vectors of size n_g of active and reactive power generation, $\boldsymbol{sl_{sf}}$ and $\boldsymbol{sl_{st}}$ are the vectors of size m of slack variables for the line flow limits, $\boldsymbol{sl_{vmax}}$ and $\boldsymbol{sl_{vmin}}$ are the vectors of size n_{pq} of slack variables for the voltage limits, $\boldsymbol{m_p}$ and $\boldsymbol{\tau}$ are the vectors of transformer tap positions, sized n_{m_p} and n_τ respectively, and $\boldsymbol{P_{DC}}$ is the vector of DC link powers, sized n_{DC}.

In total, the number of decision variables can be calculated as:

$$NV = 2n + 2n_g + 2m + 2n_{pq} + n_{m_p} + n_\tau + n_{DC} \tag{5.2}$$

5.2 Objective function

Firstly, the objective function is described as the minimization of cost of operation of the grid. This cost is modelled as a quadratic cost with respect to the active power generation, and linear with respect to the limit violation slacks. This means that the generation has to be as cheap as the limitations allow.

These limitations can be set to be not-strict using slack variables, meaning they can be surpassed at a greater cost in order to be able to solve complex grid states, which can be the case of more realistic scenarios. For instance, a grid with a high demand state can have some lines slightly overloaded for short periods of time, and exchanging this overload for a penalty can be preferred to not being able to solve the grid state. Of course, the cost has to be such that the overloads are not extreme or common.

The resulting cost function is:

$$\begin{aligned} f(\boldsymbol{x}) = &\sum_{i \in \mathrm{ig}} c_{i_0} + c_{i_1} P_{g_i} + c_{i_2} P_{g_i}^2 \\ &+ \sum_{k \in \mathrm{M}} c_{s_k} (sl_{sf_k} + sl_{st_k}) + \sum_{i \in \mathrm{pq}} c_{sl_{v}i} (sl_{vmax_i} + sl_{vmin_i}) \end{aligned} \tag{5.3}$$

where c_{i_0}, c_{i_1} and c_{i_2} are the base, linear and quadratic coefficients of the cost function for the active power generation, c_{s_k} is the cost of the slack variables for the line flow limits, $c_{sl_{v_i}}$ is the cost of the slack variables for the voltage limits and ig, M and pq are the sets of dispatchable generators, monitored lines and PQ buses respectively.

5.2.1 Objective function gradient

The gradient of the objective function results in the following vector:

$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{0_n} \\ \boldsymbol{0_n} \\ \boldsymbol{c_1} + 2\boldsymbol{c_2 P_g} \\ \boldsymbol{0_{n_g}} \\ \boldsymbol{c_s} \\ \boldsymbol{c_s} \\ \boldsymbol{c_{sl_v}} \\ \boldsymbol{c_{sl_v}} \\ \boldsymbol{0_{n_{mp}}} \\ \boldsymbol{0_{n_\tau}} \\ \boldsymbol{0_{n_{DC}}} \end{bmatrix} \tag{5.4}$$

where the subindex $\boldsymbol{0_n}$ indicates a vector of zeros of size n.

5.2.2 Objective function hessian

The hessian of the objective function is a diagonal matrix, with the following structure:

$$\nabla^2 f(\boldsymbol{x}) = \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 2[\boldsymbol{c_2}] & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \tag{5.5}$$

with only the diagonal elements corresponding to the active power generation variables are non-zero.

5.3 Equality constraints

The vector of equality constraints G for the AC-OPF problem has the following structure:

$$\boldsymbol{G}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{G_P^S} \\ \boldsymbol{G_Q^S} \\ \boldsymbol{G_{slack}^{Th}} \\ \boldsymbol{G_{pv}^V} \end{bmatrix} \tag{5.6}$$

where $\boldsymbol{G_P^S}$ and $\boldsymbol{G_Q^S}$ are the active and reactive nodal power balances, $\boldsymbol{G_{slack}^{Th}}$ is the slack phase reference, and $\boldsymbol{G_{pv}^V}$ are the voltage equalities for pv buses.

The power balance is calculated in complex form as in Equation (3.6) since the calculations are much more compact, and then it is split into active and reactive power balances, since the values have to be real numbers for the solver.

To obtain the Jacobian and Hessian matrices, the G vector needs to be derived twice, including the associated multiplier in the second derivative term. The derivatives are obtained following the Matpower documentation [7], directly using the derivatives with respect to the voltage and power variables, and developing with a similar philosophy the derivatives with respect to the additional variables that are introduced in the model.

5.3.1 First derivatives of G

The vector of partial derivatives of the equality constraints G_X^S is:

$$G_X^S = \frac{\partial G^s}{\partial X} = \begin{bmatrix} G_\theta^S & G_V^S & G_P^S & G_Q^S & 0_{n\,x\,sl} & G_{m_p}^S & G_\tau^S & G_{P_{DC}}^S \end{bmatrix} \tag{5.7}$$

The first derivatives of the power balance equations are obtained form Matpower [7] as follows:

$$\begin{aligned}
G_\theta^S &= \frac{\partial \boldsymbol{S_{bus}}}{\partial \boldsymbol{\theta}} = [\boldsymbol{I^*_{bus}}]\frac{\partial \boldsymbol{V}}{\partial \boldsymbol{\theta}} + [\boldsymbol{V}]\frac{\partial \boldsymbol{I^*_{bus}}}{\partial \boldsymbol{\theta}} \\
&= \left[\boldsymbol{I^*_{bus}}\right] j[\boldsymbol{V}] + \left[(j\boldsymbol{V})\boldsymbol{I^*_{bus}}\right] \\
&= j[\boldsymbol{V}]\left([\boldsymbol{I^*_{bus}}] - Y^*_{bus}[\boldsymbol{V^*}]\right) \\
G_{\mathcal{V}}^S &= \frac{\partial \boldsymbol{S_{bus}}}{\partial \boldsymbol{\mathcal{V}}} = \left[\boldsymbol{I^*_{bus}}\frac{\partial \boldsymbol{V}}{\partial \boldsymbol{\mathcal{V}}} + [\boldsymbol{V}]\frac{\partial \boldsymbol{I^*_{bus}}}{\partial \boldsymbol{\mathcal{V}}}\right] \\
&= \left[\boldsymbol{I^*_{bus}}\right][\boldsymbol{E}] + [\boldsymbol{V}]Y^*_{bus}[\boldsymbol{E^*}] \\
&= [\boldsymbol{V}]([\boldsymbol{I^*_{bus}}] + Y^*_{bus}[\boldsymbol{V^*}])[\boldsymbol{\mathcal{V}}]^{-1} \\
G_{P_g}^S &= -C_g \\
G_{Q_g}^S &= -jC_g
\end{aligned} \tag{5.8}$$

Where $[\boldsymbol{E}] = [\boldsymbol{V}]\boldsymbol{\mathcal{V}}^{-1}$. The derivatives with respect to the tap variables have been derived with the following expressions, starting from the admittance primitives which are the quantities that depend on the tap variables as seen in Equations (**??**) and (**??**). These primitives are used to calculate the $from$ and *to* power flowing through a line and its primitives with the following expression:

$$\begin{aligned}
\boldsymbol{S_f} &= \boldsymbol{V_f I_f^*} \\
\boldsymbol{S_t} &= \boldsymbol{V_t I_t^*} \\
\boldsymbol{I_f} &= y_{ff}\boldsymbol{V_f} + y_{ft}\boldsymbol{V_t} \\
\boldsymbol{I_t} &= y_{tf}\boldsymbol{V_f} + y_{tt}\boldsymbol{V_t}
\end{aligned} \tag{5.9}$$

To obtain the power injection per bus, we compose the S vector using the connectivity matrices as follows:

$$\boldsymbol{S_{bus}} = C_f^T \boldsymbol{S_f} + C_t^T \boldsymbol{S_t} \tag{5.10}$$

The equality constraints for the power flow that have to be derived are the ones described by Equation (3.6). Deriving the expressions written above for the admittance primitives, we obtain the following expressions:

$$\begin{aligned}
\frac{\partial \boldsymbol{G^S}}{\partial \boldsymbol{m_p}} &= \frac{\partial \boldsymbol{S_{bus}}}{\partial \boldsymbol{m_p}} = Cf^T \frac{\partial \boldsymbol{S_f}}{\partial \boldsymbol{m_p}} + Ct^T \frac{\partial \boldsymbol{S_t}}{\partial \boldsymbol{m_p}} \\
\frac{\partial \boldsymbol{G^S}}{\partial \boldsymbol{\tau}} &= \frac{\partial \boldsymbol{S_{bus}}}{\partial \boldsymbol{\tau}} = Cf^T \frac{\partial \boldsymbol{S_f}}{\partial \boldsymbol{\tau}} + Ct^T \frac{\partial \boldsymbol{S_t}}{\partial \boldsymbol{\tau}} \\
\frac{\partial S_{f_k}}{\partial m_{p_i}} &= V_{f_k}(\frac{-2(y_{s_k}V_{f_k})^*}{m_{p_i}^3} + \frac{(y_{s_k}V_{t_k})^*}{m_{p_i}^2 e^{j\tau}}) \\
\frac{\partial S_{t_k}}{\partial m_{p_i}} &= V_{t_k}\frac{(y_{s_k}V_{f_k})^*}{m_{p_i}^2 e^{-j\tau_i}} \\
\frac{\partial S_{f_k}}{\partial \tau_i} &= V_{f_k}\frac{j(y_{s_k}V_{t_k})^*}{m_{p_i} e^{j\tau_i}} \\
\frac{\partial S_{t_k}}{\partial \tau_i} &= V_{t_k}\frac{-j(y_{s_k}V_{f_k})^*}{m_{p_i} e^{-j\tau_i}}
\end{aligned} \tag{5.11}$$

Each of the individual S_{f_k} and S_{t_k} derivative with respect to each i transformer variable will be stored in the position of their corresponding matrix as $\frac{dS_f}{dm_p}_{ki}$ and $\frac{dS_f}{dm_p}_{ki}$, with size m x n_{m_p} and m x n_τ respectively, with m being the total number of branches of the grid. These matrices will have very few values different from zero, since the transformer control is only present in a few transformers of the grid.

The derivative of $G_{P_{DC}}^S$ can be obtained using the indexing of the buses participant of a DC link, which are listed in the lists fdc and tdc. Each link DC is also indexed in the list DC, of length n_{DC}. The derivative is obtained as follows:

$$\begin{cases} G^S_{P_{DC}}\{fdc, link\} = 1 \quad \forall link = (fdc, tdc) \in DC \\ G^S_{P_{DC}}\{tdc, link\} = -1 \quad \forall link = (fdc, tdc) \in DC \\ 0 \quad \text{otherwise} \end{cases} \tag{5.12}$$

The complete matrix G_X can be obtained concatenating the following submatrices:

$$G_X = \begin{bmatrix} \boldsymbol{\mathcal{R}(G^{S^\top}_X)} \\ \boldsymbol{\mathcal{I}(G^{S^\top}_X)} \\ \boldsymbol{G^{Th^\top}_X} \\ \boldsymbol{G^{\mathcal{V}^\top}_X} \end{bmatrix} \tag{5.13}$$

where $\boldsymbol{G^S_X}0$ is separated into its real $(\mathcal{R})$ and imaginary $(\mathcal{I})$ part since the solver needs real valued matrices, G^{Th} and $G^{\mathcal{V}}$ are the slack phase and voltage magnitude equalities, and their derivatives have 1 in the fixed buses positions and 0 otherwise.

5.3.2 Second derivatives of G

The Hessian matrix associated to the equality constraints has the following structure:

$$G^S_{XX} = \begin{bmatrix} G_{\theta\theta} & G_{\theta\mathcal{V}} & G_{\theta P_g} & G_{\theta Q_g} & G_{\theta sl} & G_{\theta m_p} & G_{\theta\tau} & G_{\theta P_{DC}} \\ G_{\mathcal{V}\theta} & G_{\mathcal{V}\mathcal{V}} & G_{\mathcal{V}P_g} & G_{\mathcal{V}Q_g} & G_{\mathcal{V}sl} & G_{\mathcal{V}m_p} & G_{\mathcal{V}\tau} & G_{\mathcal{V}P_{DC}} \\ G_{P_g\theta} & G_{P_g\mathcal{V}} & G_{P_gP_g} & G_{P_gQ_g} & G_{P_gsl} & G_{P_gm_p} & G_{P_g\tau} & G_{P_gP_{DC}} \\ G_{Q_g\theta} & G_{Q_g\mathcal{V}} & G_{Q_gP_g} & G_{Q_gQ_g} & G_{Q_gsl} & G_{Q_gm_p} & G_{Q_g\tau} & G_{Q_gP_{DC}} \\ G_{sl\theta} & G_{sl\mathcal{V}} & G_{slP_g} & G_{slQ_g} & G_{slsl} & G_{slm_p} & G_{sl\tau} & G_{slP_{DC}} \\ G_{m_p\theta} & G_{m_p\mathcal{V}} & G_{m_pP_g} & G_{m_pQ_g} & G_{m_psl} & G_{m_pm_p} & G_{m_p\tau} & G_{m_pP_{DC}} \\ G_{\tau\theta} & G_{\tau\mathcal{V}} & G_{\tau P_g} & G_{\tau Q_g} & G_{\tau sl} & G_{\tau m_p} & G_{\tau\tau} & G_{\tau P_{DC}} \\ G_{P_{DC}\theta} & G_{P_{DC}\mathcal{V}} & G_{P_{DC}P_g} & G_{P_{DC}Q_g} & G_{P_{DC}sl} & G_{P_{DC}m_p} & G_{P_{DC}\tau} & G_{P_{DC}P_{DC}} \end{bmatrix} \tag{5.14}$$

The first set of second derivatives corresponding to the 4 x 4 submatrix block of the Hessian (those that involve only derivation with respect to Power Flow variables $((\theta, \mathcal{V}, P_g, Q_g))$, noted $\boldsymbol{X_{PF}}$) is obtained from [7].

$$
\begin{aligned}
G^S_{XX_{PF}}(\boldsymbol{\lambda}) &= \frac{\partial}{\partial \boldsymbol{X_{PF}}}(\boldsymbol{G^S}^{\top}_{\boldsymbol{X_{PF}}}\boldsymbol{\lambda}) \\
&= \begin{bmatrix} G^S_{\boldsymbol{\theta\theta}}(\boldsymbol{\lambda}) & G^S_{\boldsymbol{\theta}V}(\boldsymbol{\lambda}) & 0 & 0 \\ G^S_{V\boldsymbol{\theta}}(\boldsymbol{\lambda}) & G^S_{VV}(\boldsymbol{\lambda}) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
G^S_{\boldsymbol{\theta\theta}}(\boldsymbol{\lambda}) &= \frac{\partial}{\partial \boldsymbol{\theta}}({G^S}^{\top}_{\boldsymbol{\theta}}\boldsymbol{\lambda}) \\
&= \left[\boldsymbol{V^*}\right]\left(Y^{*\top}_{bus}\left[\boldsymbol{V}\right]\left[\boldsymbol{\lambda}\right] - \left[Y^{*\top}_{bus}[\boldsymbol{V}]\boldsymbol{\lambda}\right]\right) \\
&\quad + \left[\boldsymbol{\lambda}\right]\left[\boldsymbol{V}\right]\left(Y^{*}_{bus}\left[\boldsymbol{V^*}\right] - \left[\boldsymbol{I^*_{bus}}\right]\right) \\
G^S_{\boldsymbol{\mathcal{V}\theta}}(\boldsymbol{\lambda}) &= \frac{\partial}{\partial \boldsymbol{\theta}}({G^S}^{\top}_{\boldsymbol{\mathcal{V}}}\boldsymbol{\lambda}) \\
&= j[\boldsymbol{\mathcal{V}}]^{-1}([\boldsymbol{V^*}](Y^{*}_{bus}[\boldsymbol{V}][\boldsymbol{\lambda}] - [Y^{*\top}_{bus}[\boldsymbol{V}]\boldsymbol{\lambda}]) \\
&- [\boldsymbol{\lambda}][\boldsymbol{V}](Y^{*}_{bus}[\boldsymbol{V^*}] - [\boldsymbol{I^*_{bus}}])) \\
G^S_{\boldsymbol{\mathcal{V}\mathcal{V}}}(\boldsymbol{\lambda}) &= \frac{\partial}{\partial \boldsymbol{\mathcal{V}}}({G^S}^{\top}_{\boldsymbol{\mathcal{V}}}\boldsymbol{\lambda}) \\
&= [\boldsymbol{\mathcal{V}}]^{-1}\left([\boldsymbol{\lambda}][\boldsymbol{V}]Y^{*}_{bus}[\boldsymbol{V^*}] + [\boldsymbol{V^*}]Y^{*}_{bus}[\boldsymbol{V}][\boldsymbol{\lambda}]\right)[\boldsymbol{\mathcal{V}}]^{-1}
\end{aligned}
\tag{5.15}
$$

The rest of the second derivatives have been obtained following a similar strategy to add the multipliers. The derivation starts again with the $from$ and to expressions, with the expressions for a given branch k and its connected buses (f_k, t_k) as follows:

$$
\begin{aligned}
\frac{\partial^2 S_{f_k}}{\partial m_{p_i} \partial m_{p_i}} &= V_{f_k} \Big(\frac{6(y_{s_k} V_{f_k})^*}{m_{p_i}^4} - \frac{2(y_{s_k} V_{t_k})^*}{m_{p_i}^3 e^{j\tau_i}} \Big) \\
\frac{\partial^2 S_{t_k}}{\partial m_{p_i} \partial m_{p_i}} &= V_{t_k} \frac{-2(y_{s_k} V_{f_k})^*}{m_{p_i}^3 e^{-j\tau_i}} \\
\frac{\partial^2 S_{f_k}}{\partial \tau_i \partial \tau_i} &= V_{f_k} \frac{(y_{s_k} V_{t_k})^*}{m_{p_i} e^{j\tau_i}} \\
\frac{\partial^2 S_{t_k}}{\partial \tau_i \partial \tau_i} &= V_{t_k} \frac{(y_{s_k} V_{f_k})^*}{m_{p_i} e^{-j\tau_i}} \\
\frac{\partial^2 S_{f_k}}{\partial m_{p_i} \partial \tau_i} &= V_{f_k} \frac{-j(y_{s_k} V_{t_k})^*}{m_{p_i}^2 e^{j\tau_i}} \\
\frac{\partial^2 S_{t_k}}{\partial m_{p_i} \partial \tau_i} &= V_{t_k} \frac{j(y_{s_k} V_{f_k})^*}{m_{p_i}^2 e^{-j\tau_i}} \\
\frac{\partial S_{f_k}}{\partial m_{p_i} \partial v_{f_k}} &= \frac{V_{f_k}}{v_{f_k}} \Big(\frac{-4(y_{s_k} V_{f_k})^*}{m_{p_i}^3} + \frac{(y_{s_k} V_{t_k})^*}{m_{p_i}^2 e^{j\tau_i}} \Big) \\
\frac{\partial S_{t_k}}{\partial m_{p_i} \partial v_{f_k}} &= \frac{V_{t_k}}{v_{f_k}} \frac{(y_{s_k} V_{f_k})^*}{m_{p_i}^2 e^{-j\tau_i}} \\
\frac{\partial S_{f_k}}{\partial m_{p_i} \partial v_{t_k}} &= \frac{V_{f_k}}{v_{f_t}} \Big(\frac{-2(y_{s_k} V_{f_k})^*}{m_{p_i}^3} + \frac{(y_{s_k} V_{t_k})^*}{m_{p_i}^2 e^{j\tau_i}} \Big) \\
\frac{\partial S_{t_k}}{\partial m_{p_i} \partial v_{t_k}} &= \frac{V_{t_k}}{v_{f_t}} \frac{(y_{s_k} V_{f_k})^*}{m_{p_i}^2 e^{-j\tau_i}} \\
\frac{\partial S_{f_k}}{\partial \tau_i \partial v_{f_k}} &= \frac{V_{f_k}}{v_{f_k}} \frac{j(y_{s_k} V_{t_k})^*}{m_{p_i} e^{j\tau_i}} \\
\frac{\partial S_{t_k}}{\partial \tau_i \partial v_{f_k}} &= \frac{V_{t_k}}{v_{f_k}} \frac{-j(y_{s_k} V_{f_k})^*}{m_{p_i} e^{-j\tau_i}} \\
\frac{\partial S_{f_k}}{\partial \tau_i \partial v_{f_k}} &= \frac{V_{f_k}}{v_{f_t}} \frac{j(y_{s_k} V_{t_k})^*}{m_{p_i} e^{j\tau_i}} \\
\frac{\partial S_{t_k}}{\partial \tau_i \partial v_{f_k}} &= \frac{V_{t_k}}{v_{f_t}} \frac{-j(y_{s_k} V_{f_k})^*}{m_{p_i} e^{-j\tau_i}} \\
\frac{\partial S_{f_k}}{\partial m_{p_i} \partial \theta_{f_k}} &= V_{f_k} \frac{j(y_{s_k} V_{t_k})^*}{m_{p_i}^2 e^{j\tau_i}} \\
\frac{\partial S_{t_k}}{\partial m_{p_i} \partial \theta_{f_k}} &= V_{t_k} \frac{-j(y_{s_k} V_{f_k})^*}{m_{p_i}^2 e^{-j\tau_i}} \\
\frac{\partial S_{f_k}}{\partial m_{p_i} \partial \theta_{t_k}} &= V_{f_k} \frac{-j(y_{s_k} V_{t_k})^*}{m_{p_i}^2 e^{j\tau_i}} \\
\frac{\partial S_{t_k}}{\partial m_{p_i} \partial \theta_{t_k}} &= V_{t_k} \frac{j(y_{s_k} V_{f_k})^*}{m_{p_i}^2 e^{-j\tau_i}} \\
\frac{\partial S_{f_k}}{\partial \tau_i \partial \theta_{f_k}} &= V_{f_k} \frac{-(y_{s_k} V_{t_k})^*}{m_{p_i} e^{j\tau_i}} \\
\frac{\partial S_{t_k}}{\partial \tau_i \partial \theta_{f_k}} &= V_{t_k} \frac{-(y_{s_k} V_{f_k})^*}{m_{p_i} e^{-j\tau_i}} \\
\frac{\partial S_{f_k}}{\partial \tau_i \partial \theta_{f_k}} &= V_{f_k} \frac{(y_{s_k} V_{t_k})^*}{m_{p_i} e^{j\tau_i}} \\
\frac{\partial S_{t_k}}{\partial \tau_i \partial \theta_{f_k}} &= V_{t_k} \frac{(y_{s_k} V_{f_k})^*}{m_{p_i} e^{-j\tau_i}}
\end{aligned}
\tag{5.16}
$$

Crossed partial derivative $\frac{\partial^2}{\partial m_{p_i} \partial \tau_i}$ are considered for those lines with transformers that controls both variables. To compose the submatrix, the following expression is used for all of these transformers:

$$\begin{aligned} \frac{dS_{bus}}{dm_p dm_p}_{ii} &= \Re(\frac{\partial^2 S_{f_k}}{\partial m_{p_i} \partial m_{p_i}} \lambda_f + \frac{\partial^2 S_{t_k}}{\partial m_{p_i} \partial m_{p_i}} \lambda_t) \\ &+ \Im(\frac{\partial^2 S_{f_k}}{\partial m_{p_i} \partial m_{p_i}} \lambda_{f+N} + \frac{\partial^2 S_{t_k}}{\partial m_{p_i} \partial m_{p_i}} \lambda_{t+N}) \end{aligned} \tag{5.17}$$

Since we have to take into account the active and reactive power constraints, we have to divide the expression into real and imaginary and then use the corresponding multiplier. We compose the resulting hessian by concatenating the complete matrix as in the previous case.

5.4 Inequality constraints

We proceed similarly to the equality constraints. The vector of inequality constraints H for the AC-OPF problem has the following structure:

$$\boldsymbol{H}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{H}^{\boldsymbol{S_f}} \\ \boldsymbol{H}^{\boldsymbol{S_t}} \\ \boldsymbol{H}^{\boldsymbol{V_u}} \\ \boldsymbol{H}^{\boldsymbol{P_u}} \\ \boldsymbol{H}^{\boldsymbol{Q_u}} \\ \boldsymbol{H}^{\boldsymbol{V_l}} \\ \boldsymbol{H}^{\boldsymbol{P_l}} \\ \boldsymbol{H}^{\boldsymbol{Q_l}} \\ \boldsymbol{H}^{\boldsymbol{sl}} \\ \boldsymbol{H}^{\boldsymbol{m_{p_u}}} \\ \boldsymbol{H}^{\boldsymbol{m_{p_l}}} \\ \boldsymbol{H}^{\boldsymbol{\tau_u}} \\ \boldsymbol{H}^{\boldsymbol{\tau_l}} \\ \boldsymbol{H}^{\boldsymbol{Q_{max}}} \\ \boldsymbol{H}^{\boldsymbol{P_{DC_u}}} \\ \boldsymbol{H}^{\boldsymbol{P_{DC_l}}} \end{bmatrix} \tag{5.18}$$

The first derivatives of the power flows are the most complex ones to get, since they will be derived from the $from$ and to powers, and the constraints have their value squared.

$$\boldsymbol{H_f^S} = [\boldsymbol{S_f^*}]\boldsymbol{S_f} - \boldsymbol{S_{max}^2} \leq 0 \tag{5.19}$$

$$\begin{aligned} H_X^f &= 2(\Re([\boldsymbol{S_f}])\Re(S_X^f) + \Im([\boldsymbol{S_f}])\Im(S_X^f)) \\ H_{XX}^f(\boldsymbol{\mu}) &= 2\Re(S_{XX}^f([\boldsymbol{S_f^*}]\boldsymbol{\mu}) + {S_X^f}^T[\boldsymbol{\mu}]S_X^f) \end{aligned} \tag{5.20}$$

And similarly for the H^t *to* constraints. The derivatives with respect of the PF variables are obtained from [7] again:

$$\begin{aligned} S_{\boldsymbol{\theta}}^f &= \left[\boldsymbol{I_f^*}\right] \frac{\partial \boldsymbol{V_f}}{\partial \boldsymbol{\theta}} + \left[\boldsymbol{V_f}\right] \frac{\partial \boldsymbol{I_f^*}}{\partial \boldsymbol{\theta}} \\ &= \left[\boldsymbol{I_f^*}\right] jC_f[\boldsymbol{V}] + [C_f\boldsymbol{V}](jY_f^{\overline{\ }}[\boldsymbol{V}])^* \\ &= j\left([\boldsymbol{I_f^*}]C_f[\boldsymbol{V}] - [C_f\boldsymbol{V}]Y_f^*[\boldsymbol{V^*}]\right) \\ S_{\boldsymbol{\mathcal{V}}}^f &= \left[\boldsymbol{I_f^*}\right] \frac{\partial \boldsymbol{V_f}}{\partial \boldsymbol{\mathcal{V}}} + \left[\boldsymbol{V_f}\right] \frac{\partial \boldsymbol{I_f^*}}{\partial \boldsymbol{\mathcal{V}}} \\ &= \left[\boldsymbol{I_f^*}\right] C_f[\boldsymbol{E}] + [C_f\boldsymbol{V}]Y_f^*[\boldsymbol{E^*}] \\ S_{\boldsymbol{P_g}}^f &= 0 \\ S_{\boldsymbol{Q_g}}^f &= 0 \end{aligned} \tag{5.21}$$

$$\begin{aligned} S_{\boldsymbol{\theta\theta}}^f(\boldsymbol{\mu}) &= \frac{\partial}{\partial \boldsymbol{\theta}} \left(S_{\boldsymbol{\theta}}^f\right)^T \boldsymbol{\mu} \\ &= [\boldsymbol{V^*}]{Y_f^*}^\top[\boldsymbol{\mu}]C_f[\boldsymbol{V}] + [\boldsymbol{V}]C_f^\top[\boldsymbol{\mu}]Y_f^*[\boldsymbol{V^*}] \\ &- [{Y_f^*}^\top[\boldsymbol{\mu}]C_f\boldsymbol{V}][\boldsymbol{V^*}] - [C_f^T[\boldsymbol{\mu}]Y_f^*\boldsymbol{V^*}][\boldsymbol{V}] \\ S_{\boldsymbol{\mathcal{V}\theta}}^f(\boldsymbol{\mu}) &= \frac{\partial}{\partial \boldsymbol{\theta}} \left(S_{\boldsymbol{\mathcal{V}}}^f\right)^\top \boldsymbol{\mu} \\ &= j[\boldsymbol{\mathcal{V}}]^{-1}([\mathbf{V^*}]{Y_f^*}^\top[\boldsymbol{\mu}]C_f[\mathbf{V}] - [\mathbf{V}]C_f^\top[\boldsymbol{\mu}]Y_f^*[\mathbf{V^*}] \\ &- [{Y_f^*}^\top[\boldsymbol{\mu}]C_f\mathbf{V}][\mathbf{V^*}] + [C_f^\top[\boldsymbol{\mu}]Y_f^*\mathbf{V^*}][\mathbf{V}]) \\ S_{\boldsymbol{\mathcal{V}\mathcal{V}}}^f(\boldsymbol{\mu}) &= \frac{\partial}{\partial \boldsymbol{\mathcal{V}}} \left(S_{\boldsymbol{\mathcal{V}}}^f\right)^\top \boldsymbol{\mu} \\ &= [\boldsymbol{\mathcal{V}}]^{-1}\left([\mathbf{V^*}]{Y_f^*}^\top[\boldsymbol{\mu}]C_f[\mathbf{V}] + [\mathbf{V}]C_f^\top[\boldsymbol{\mu}]Y_f^*[\mathbf{V^*}]\right)[\boldsymbol{\mathcal{V}}]^{-1} \end{aligned} \tag{5.22}$$

The derivatives with respect to the tap variables have been derived using $\boldsymbol{H}$, the expressions for $\boldsymbol{S_f}$, $\boldsymbol{S_t}$ and its derivatives obtained in the power balance derivatives in Equations (5.11).

6 Results

After implementing the software in the GridCal package, now it is time to test how this solver performs when compared to already available solutions. The benchmark used for this comparison will be the Matpower solver [8], an open-source package that has been used in many works as a reference for the AC-OPF problem. The objective is to have a similar runtime and number of iterations for the base AC-OPF problem (the model already used by Matpower's solver), and to see how well does it scale when the new features are added.

The grids used for this benchmark are a combination of small grids with some interesting configurations that allow some testing of the new features, and larger grids that will be used to see if the solver is capable to solve them and what is the computational cost of doing so. The largest grid is the Great Britain national grid (an adaptation), which can give some insight on how the solver performs in a real-world scenario.

The results obtained have been obtained in a computer with the following specifications:

- CPU: Intel Core i7-1165G7 @ 2.80GHz
- RAM: 16GB DDR4
- GPU: Intel Iris Xe Graphics
- OS: Windows 11
- IDE: PyCharm
- Python version: 3.10
- Matpower version - Python version: PyPower 5.1.16 [8] [9]

And the tolerance requested for the convergence conditions is 10^{-6}.

There are two possible initializations for the solver, which are the following:

- **Power flow initialization:** The solver uses GridCal's built-in power flow with the default values introduced in the grid model as the initial values for the variables. The results of the power flow solver are used as the starting point for the optimization variables.
- **Basic initialization:** This is the initialization used by Matpower, which sets the variables at the center of the domain. Although it seems simpler, for some cases it could be better to ensure all the variables are not too close to some of their limits.

The use of bound slacks has also been tested for both initialization options, yielding a total of 4 different combinations per case. The results will be shown in the following subsections.

6.1 Error evolution

The first comparison carried out is the error evolution for the different cases. The graphs portray the error at each iteration for the different initialization settings, and will be compared to the error obtained by Matpower's solver. The error is the maximum value of the expressions (4.14) at each iteration. This error corresponds to the maximum value of the convergence conditions, which are calculated for each derivative of the Lagrangian, and including the γ value. The error should always tend to decrease, although sometimes it does so really slowly, or it even goes up for some iterations. This could mean that the path taken by the interior point method is exceeding some limits, and the resulting new point is worse than the previous. Excessive increases of the error are normally avoided by the step control implemented in the solver, although it does not prevent these increases in problems that are ill-conditioned or even unsolvable.

6.1.1 Case 9-bus

The first case is a 9-bus system [10] which can be useful to see the behavior of the algorithm in a small system, while it can be portrayed as shown in Figure 6.1 with sufficient clarity in this project using the GridCal GUI.

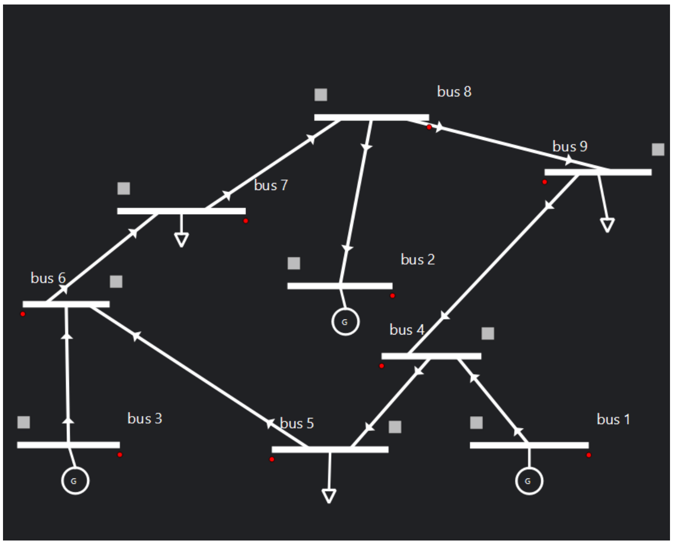


Figure 6.1: Schematic of the 9-bus grid.

It can be seen in Figure 6.2 that for such a small grid, there is not a significant difference between the different initialization options. The error evolution is very similar for all of them, and the error obtained is very close to the one obtained by Matpower, being the option without bound slacks slightly better as the vector of variables is considerably smaller. If there are no issues with the grid, the initialization with bound slacks is not necessary for this case.

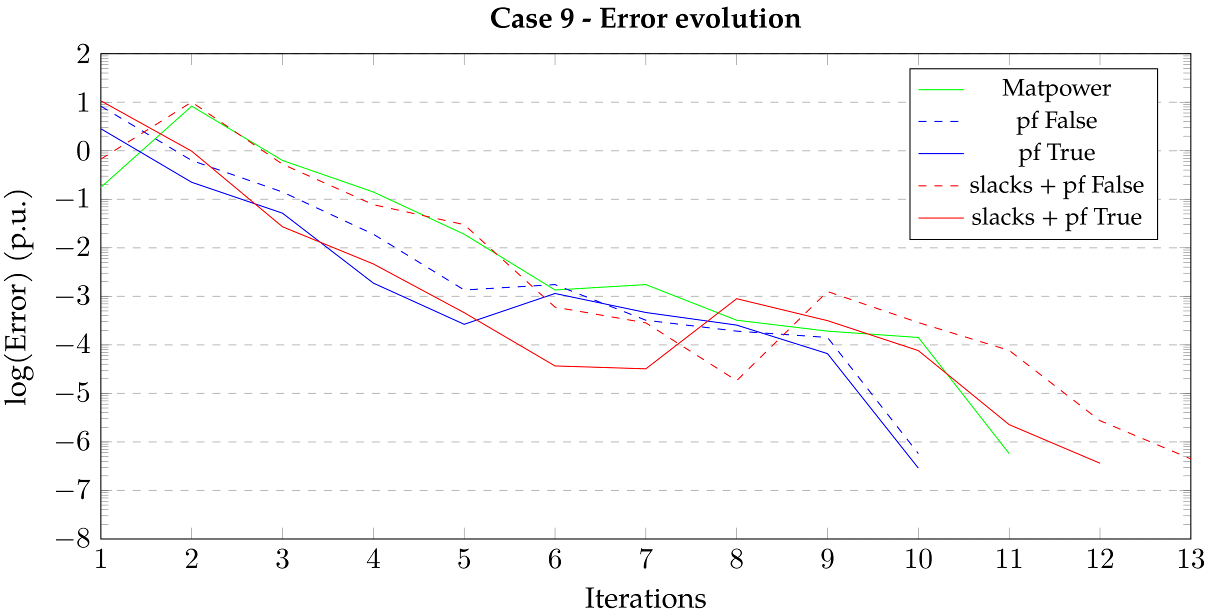
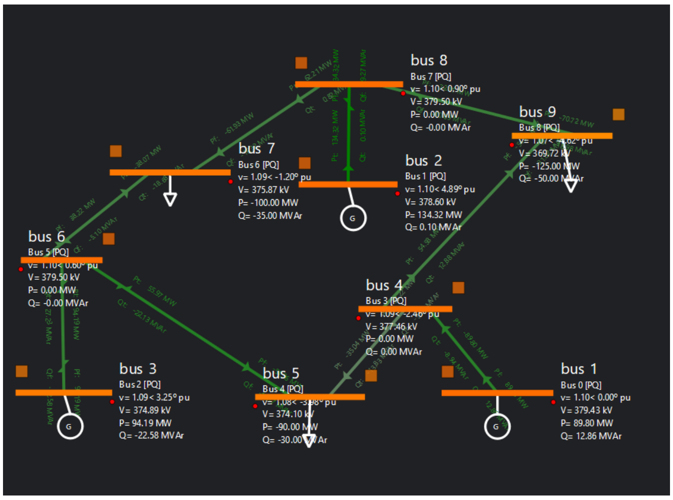


Figure 6.2: Error evolution for the Case 9 system for different initialization options.

The result can be visualized in GridCal's GUI as shown in Figure 6.3.


bus 8
Bus 7 [PQ]
v= 1.10< 0.90° pu
V= 379.50 kV
P= 0.00 MW
Q= -0.00 MVAr

bus 9
Bus 8 [PQ]
v= 1.07< -4.62° pu
V= 369.72 kV
P= -125.00 MW
Q= -50.00 MVAr

bus 7
Bus 6 [PQ]
v= 1.09< -1.20° pu
V= 375.87 kV
P= -100.00 MW
Q= -35.00 MVAr

bus 2
Bus 1 [PQ]
v= 1.10< 4.89° pu
V= 378.60 kV
P= 134.32 MW
Q= 0.10 MVAr

bus 6
Bus 5 [PQ]
v= 1.10< 0.60° pu
V= 379.50 kV
P= 0.00 MW
Q= -0.00 MVAr

bus 4
Bus 3 [PQ]
v= 1.09< -2.46° pu
V= 377.46 kV
P= 0.00 MW
Q= 0.00 MVAr

bus 3
Bus 2 [PQ]
v= 1.09< 3.25° pu
V= 374.89 kV
P= 94.19 MW
Q= -22.58 MVAr

bus 5
Bus 4 [PQ]
v= 1.08< -3.98° pu
V= 374.10 kV
P= -90.00 MW
Q= -30.00 MVAr

bus 1
Bus 0 [PQ]
v= 1.10< 0.00° pu
V= 379.43 kV
P= 89.80 MW
Q= 12.86 MVAr

Figure 6.3: Plot of the solution of Case 9.

Something that can catch the eye is the orange color of the bus, which means they are near their upper voltage limit. This is due to the optimization trying to raise the voltage value as high as the limit allows to reduce the current flowing through the lines, which reduces quadratically the losses.

6.1.2 Case Pegase89

The next grid studied is a case with 89 buses obtained from the Pegase database [11], with an interesting result shown in Figure 6.4.

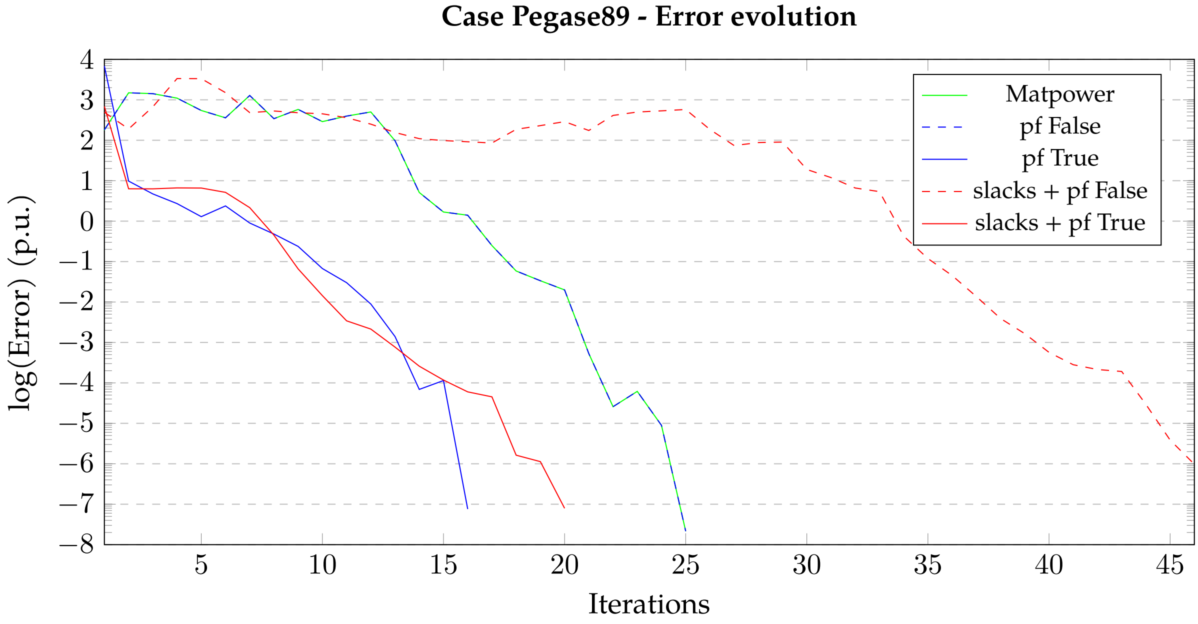


Figure 6.4: Error evolution for the Pegase89 system for different initialization options.

Firstly, it should be noted that the solution without bound slacks and with the power flow initialization disabled follows the exact same evolution as the Matpower solver. This did not happen in the previous case, but it seems to be due to the GridCal parser processing of the model. For some cases during the developing of the software, some grids were unsolvable due to some of its parameters not being directly specified in the model. For instance, some line ratings were not specified, and while the Matpower solver would then ignore some of these limits, the GridCal parser would not be able to process it in a correct way. To solve this, the parser has been adjusted to set a default value in case the lines are monitored but the rating is specified. Many other small nuances during the model loading could be the source of this difference in the convergence.

The other relevant observation that can be done is that the power flow initialization is doing a great job in improving the convergence of the problem. In a real scenario situation, the user would input a grid model that is already in a feasible state, although it may not be optimal, but it might be close to what an optimum generation profile would be. In this case, the OPF using an initialization with power flow takes almost half of the iterations in the case without bound slacks, and a third in the case with bound slacks.

One last observation is that bound slacks are not useful in this case either. The error evolution is very similar to the one without bound slacks, and the number of variables is considerably larger, which will make the problem a bit heavier to run as it will be shown in a later subsection.

6.1.3 Case 300

The next case is a 300-bus system [12], which is a considerably larger grid than the previous ones. The error evolution for each initialization is shown in Figure 6.5.

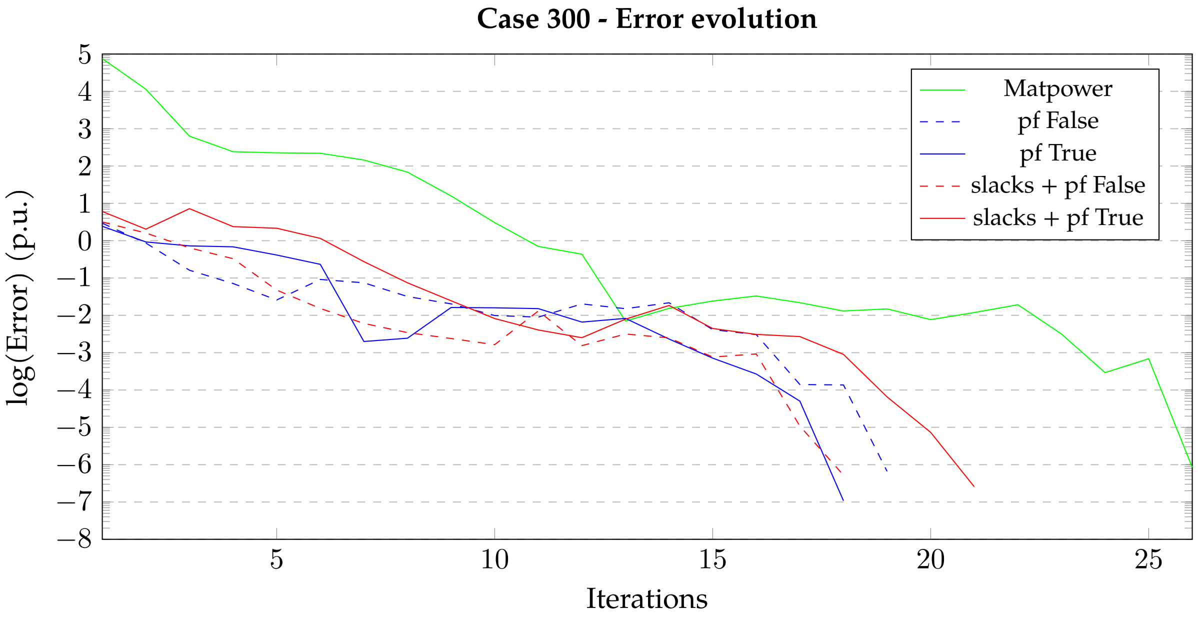


Figure 6.5: Error evolution for the 300-bus system for different initialization options.

The performance of the solver developed in this work outclasses the one of Matpower in this case. Considering that the solver is based on the same principles, this means that the GridCal modelled problem is better conditioned than the one modelled by Matpower, which is why the solver has been developed in such an environment.

6.1.4 Case GB

Last case shown is a country-sized 2223-bus system that has been provided directly by Redeia, which models the Great Britain grid. Error evolutions are shown in Figure 6.6.

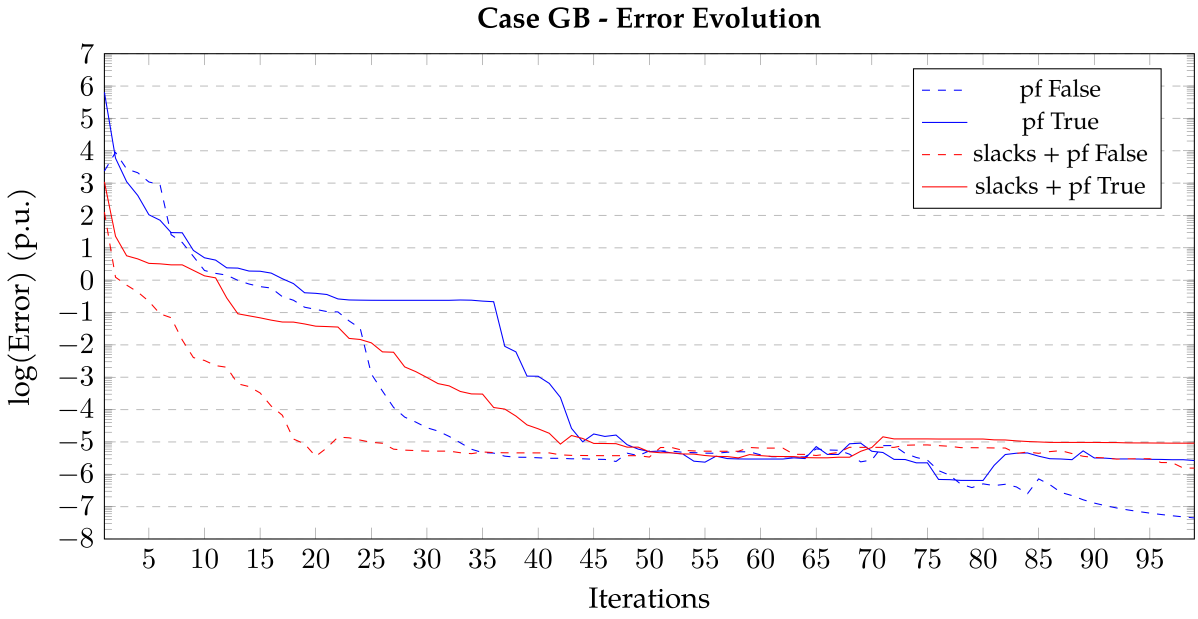


Figure 6.6: Error evolution for the Great Britain grid for different initialization options.

In this case, there is no comparative with the Matpower solver since the data is stored in a file that can only be managed in GridCal, and the model is not public. The results show a really promising evolution for a real use case, obtaining a low error in a few iterations. The slack initialization option can be preferred in this case, which is to be expected since it is a case that has not been tested before, and it might correspond to a difficult grid state (meaning the limits could be a bit strict, some generators/lines could be disabled or the load profile could be high).

Despite the good performance during the first steps, it stagnates when asking for lower tolerance values, a side effect of adding a lot of variables to the problem.

6.2 Effect of tap variables control - Case IEEE14

Once having seen the performance of the solver in some study cases, it is time to analyze the effect of the tap variables control. This control is a feature that allows the solver to manage the tap variables of the transformers in the grid, which are the ones that can be modified to control the voltage levels or the power transfer. This control can be done in two ways: by setting the tap variables as fixed variables or by setting them as optimization variables. The first option is the one that has been used in the previous subsections, and the second one is the one that is going to be tested in this one.

The grid used is the IEEE14 model grid [13], which is a 14-bus grid that includes transformers. Firstly, the error evolution is compared in Figures 6.7 and 6.8 to see the effect of adding this tap variables.

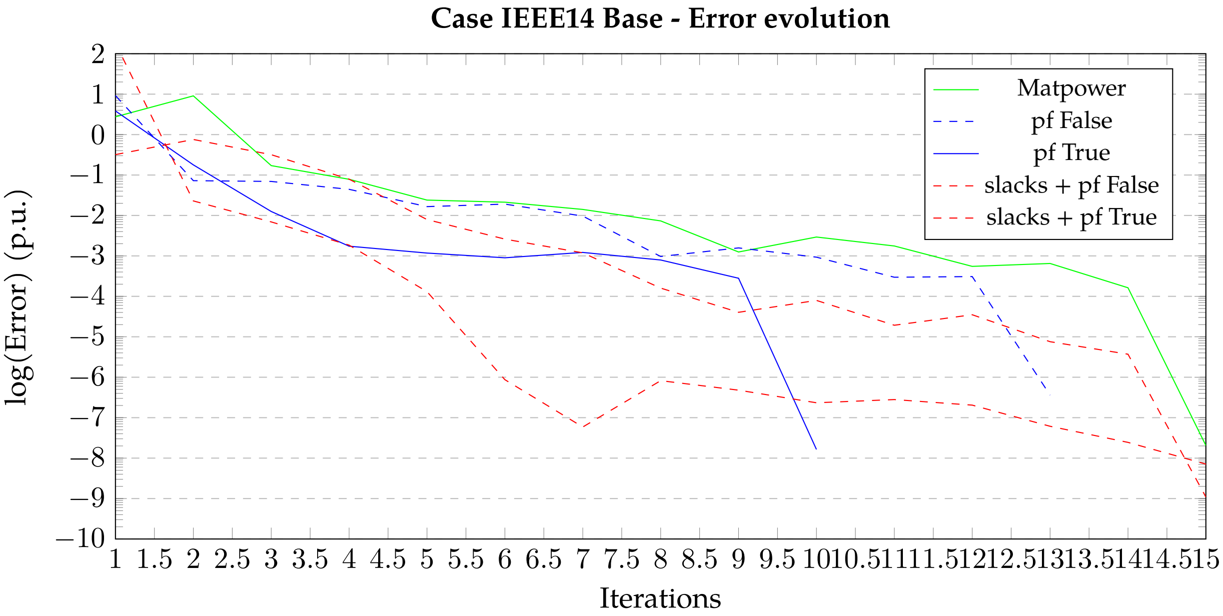


Figure 6.7: Error evolution for the Case IEEE14 for different initialization options.

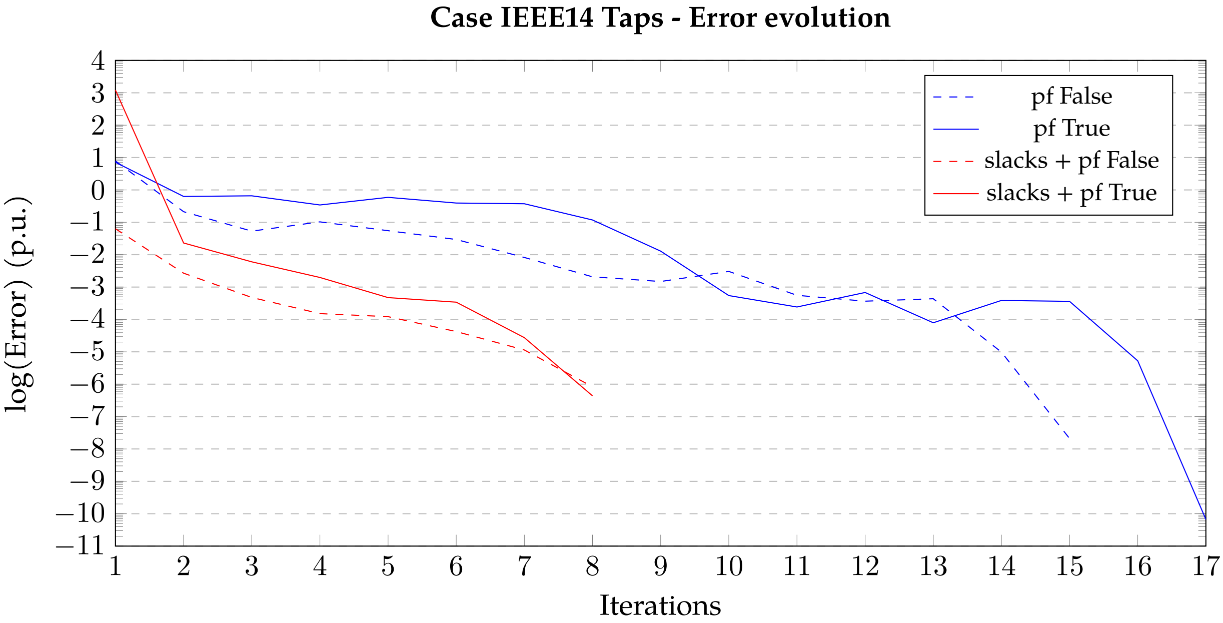


Figure 6.8: Error evolution for the Case IEEE14 with tap variables for different initialization options.

There are two important remarks obtained from these error evolutions. Firstly, the case is solved in fewer iterations in the version of the solver developed in this project. Secondly, the usage of bound slacks improves convergence for the case with tap variables. A more detailed analysis is done for the solution values for this case to analyze the impact of adding new optimization variables. For simplicity, the comparison will be made only for the cases with power flow initialization and without bound slacks, as they add to the cost function and the interest of this analysis is just testing the effect of using tap optimization.

Figures 6.9 and 6.10 show the values of the solution for the bus voltages and generators. Note that there are no differences between the case solved with Matpower and the developed software. The case with controlled tap variable has some slight differences, specially regarding the voltage magnitude. The impact in the generation is also noticeable for the reactive power generation, which has shifted mostly from generator 1 to generator 3.

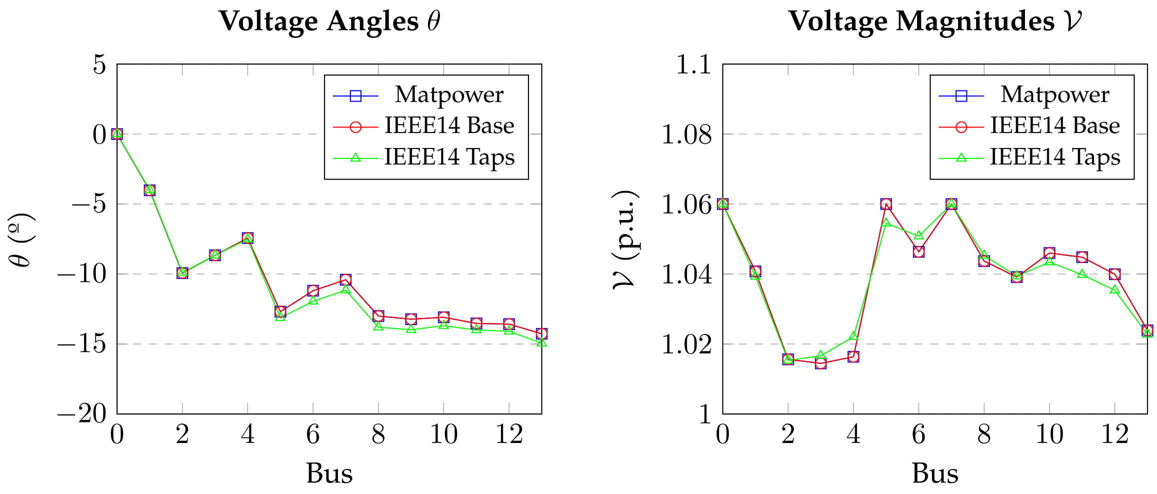


Figure 6.9: Voltage comparison for the Case IEEE14 considering tap variables.

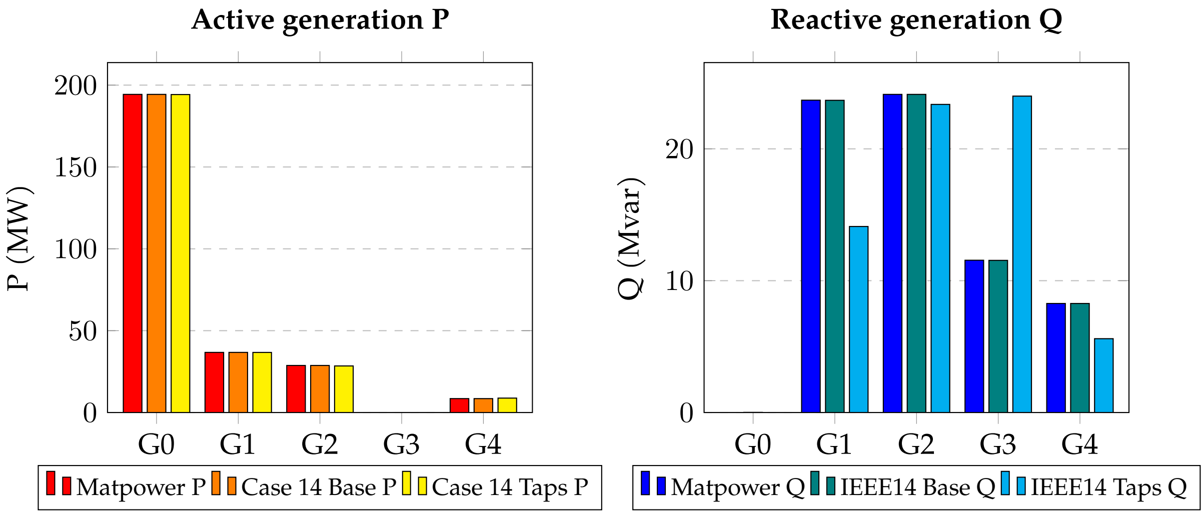


Figure 6.10: Generation comparison for Case IEEE14.

Table 6.1 shows the setpoints for the tap variables of the transformers in the grid as obtained with the solver developed. Note that the transformers that only have one variable controlled have a N/C (Non-Controlled) as a solution.

Table 6.1: Tap variable setpoints for the transformers of the corresponding branches in IEEE14.

Controlled trafos	**m_p (p.u)**	τ (º)
Branch 17	0.965876	0.754872
Branch 18	N/C	1.365530
Branch 19	0.974728	N/C

The important result is shown in Table 6.2, which shows the impact of being able to control the tap variables setpoints in the cost of operation. Of course, this impact may not seem big in this case, representing a reduction of cost of only 0.03%. However, this result has only considered a small grid, without using properly modelled generation costs, and only controlling three transformers. And still, considering the total expenses of a country-sized grid as well as this result being an instant photo of a given grid state, it can translate in substantial savings in the long-term.

Table 6.2: Comparison of costs between different cases.

	Matpower	**Case 14 Base**	**Case 14 Taps**
Cost (€/MWh)	8081.53	8081.53	8078.85

6.3 Effect of reactive control

Using the same base case, now the test is performed over the reactive power control. The limit is set as a maximum of a power factor of 0.8, which is translated to a maximum reactive power generation of 75% of the active power generation. The constraint is applied with respect to the absolute value of Q, meaning this constraint applies to both the injection and consumption of reactive power. Figure 6.11 shows the effect in the voltage values. There is a noticeable drop in magnitude in some of the buses.

In Figure 6.12, it can be seen that generator 2, which previously had a reactive generation of 24.12 Mvar, now has a generation of 20.96 Mvar, which corresponds exactly to the 75% of the active power generation. This generator being at its limit for reactive power generation means the other generator have to contribute to reactive generation, impacting the voltage levels in the grid.

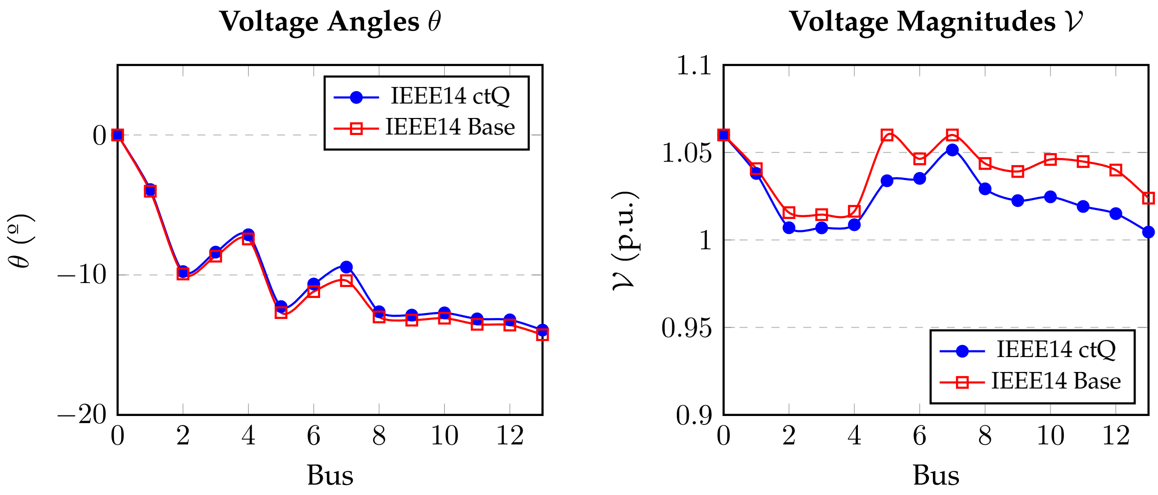


Figure 6.11: Voltage comparison for the Case IEEE14 with and without reactive control.

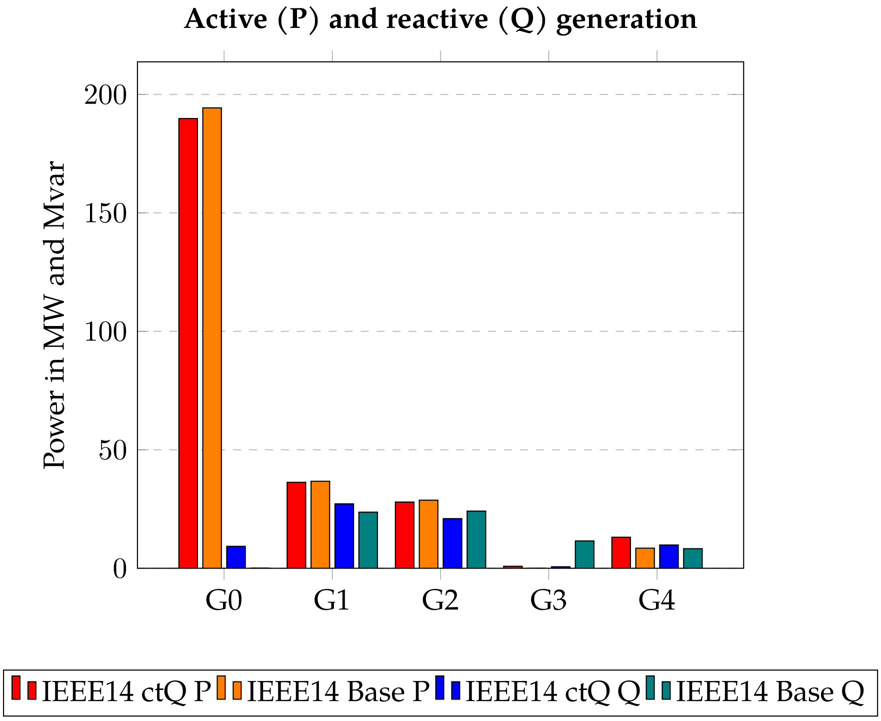


Figure 6.12: Generation comparison for the Case IEEE14 with and without reactive control.

The effect of this constraint can be seen in the cost value of the simulation. Table 6.3 shows an increase of around 0.075% in the cost of operation, which again can be significant in a bigger grid.

Table 6.3: Cost for Case IEEE14 with Q control.

	IEEE14 Base	**IEEE14 ctQ**
Cost (€/MWh)	8081.53	8087.82

6.4 DC link and dual price study

The following study is performed over a small grid with two islands of three buses each. Both of them have the same configuration and elements, but their generators will have different costs profiles as follows:

$$\begin{aligned} c_{g1.0}(P_{g1.0}) &= 1 \cdot P_{g1.0} + 2 \cdot P_{g1.0}^2 \\ c_{g1.1}(P_{g1.1}) &= 1 \cdot P_{g1.1} + 3 \cdot P_{g1.1}^2 \\ c_{g2.0}(P_{g2.0}) &= 1 \cdot P_{g2.0} + 1.5 \cdot P_{g2.0}^2 \\ c_{g2.1}(P_{g2.1}) &= 1 \cdot P_{g2.1} + 1 \cdot P_{g2.1}^2 \end{aligned} \tag{6.1}$$

The grid is firstly tested without interconnection, obtaining the results for each island isolated. Then, a lossless DC link is added between the two islands. In Figure 6.13, the comparison between the generation profiles can be seen.

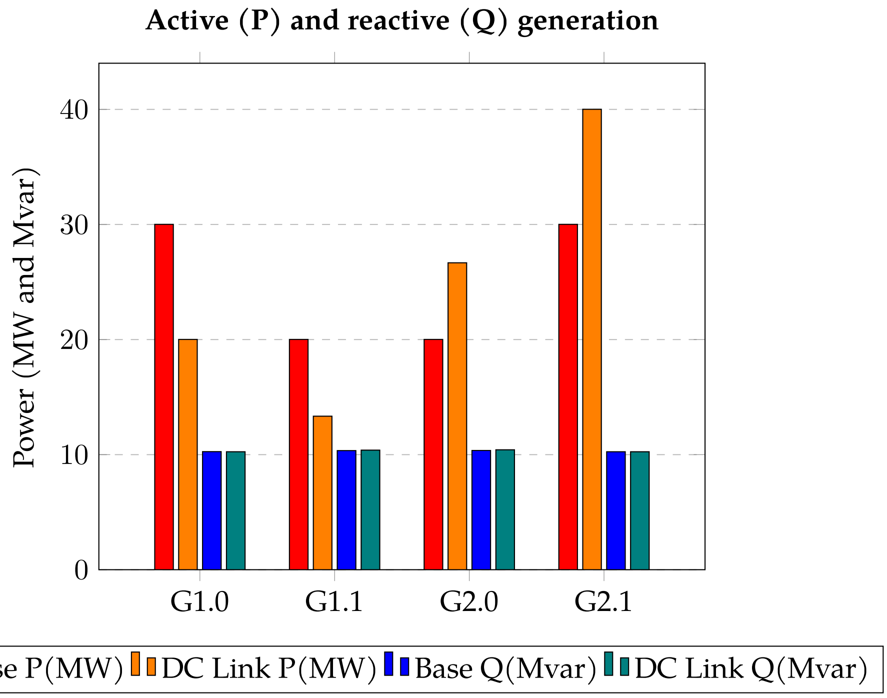


Figure 6.13: Generation comparison for the island case with and without DC link.

As expected, after both grids are connected, the generators of the second island become prevalent as their prices are lower than the first island, and the power is transferred through this DC link with a total power of $P_{DC} = 16.67\text{MW}$. The interconnection has a great impact in the dual price as seen in Figure 6.14.

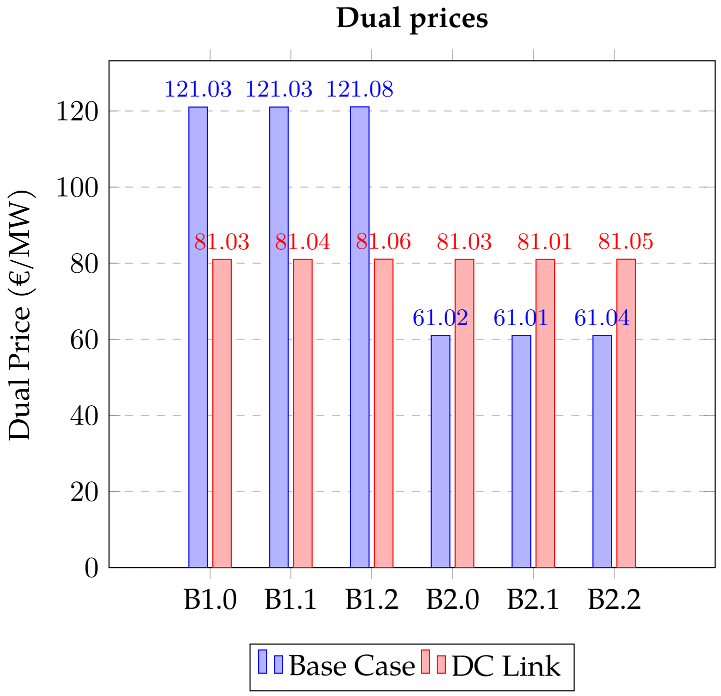


Figure 6.14: Dual price comparison for the case with and without DC link.

These results are really important to understand the meaning of the dual price, which for the optimization problem are equal to the value of the equality multipliers associated to the nodal power balance of each bus. These values indicate the rate of increase in cost that would be caused by demanding an additional infinitesimal amount of power at that bus. The following calculations can be done to check which is the cost of generating an additional MW of power per generation, using their cost function and the obtained setpoint from the optimization solution:

$$\begin{aligned}
\frac{\partial c_{g1.0}}{\partial P_{g1.0}}_{Base} &= 1 + 4 \cdot P_{g1.0} = 1 + 4 \cdot 30.0067 = 121.027 \\
\frac{\partial c_{g1.1}}{\partial P_{g1.1}}_{Base} &= 1 + 6 \cdot P_{g1.1} = 1 + 6 \cdot 20.0056 = 121.034 \\
\frac{\partial c_{g2.0}}{\partial P_{g2.0}}_{Base} &= 1 + 3 \cdot P_{g2.0} = 1 + 3 \cdot 20.0056 = 61.017 \\
\frac{\partial c_{g2.1}}{\partial P_{g2.1}}_{Base} &= 1 + 2 \cdot P_{g2.1} = 1 + 2 \cdot 30.0067 = 61.013
\end{aligned} \tag{6.2}$$

$$
\begin{aligned}
\frac{\partial c_{g1.0}}{\partial P_{g1.0}}_{DC} &= 1 + 4 \cdot P_{g1.0} = 1 + 4 \cdot 20.0062 = 81.025 \\
\frac{\partial c_{g1.1}}{\partial P_{g1.1}}_{DC} &= 1 + 6 \cdot P_{g1.1} = 1 + 6 \cdot 13.3392 = 81.035 \\
\frac{\partial c_{g2.0}}{\partial P_{g2.0}}_{DC} &= 1 + 3 \cdot P_{g2.0} = 1 + 3 \cdot 26.6750 = 81.025 \\
\frac{\partial c_{g2.1}}{\partial P_{g2.1}}_{DC} &= 1 + 2 \cdot P_{g2.1} = 1 + 2 \cdot 40.0057 = 81.012
\end{aligned}
\tag{6.3}
$$

As it can be seen, the values of dual prices for buses with generation are equal to the derivative of those generators connected. For buses with no generation, the price is slightly above since it has to account for the additional power injected due to the losses during the power transfer.

Finally, the cost of operation has obviously been reduced after connecting both islands as seen in Table 6.4. The difference is significant, representing a reduction of 11.5% in the cost of operation. This example shows the economical benefit of having an interconnection to a neighboring grid as the INELFE interconnection between Spain and France, apart from the technical benefits in terms of frequency stability, power support during faults, etc.

Table 6.4: Costs for Case 2 islands.

	Base Case 2 islands	**2 islands + DC link**
Cost (€/MWh)	4602.23	4102.12

6.5 Runtime comparison

Lastly, the execution times of all the cases with each different initialization combination have been measured. All measurements have been reproduced in the same conditions, repeating 5 times each execution to have a larger sample size, although all the values had low deviation. Results are shown in Figures 6.15 and 6.16, separating the larger case for better visualization.

Overall, the solver seems to be on the same level of performance as Matpower in the cases where a comparison could be made. For smaller cases, Matpower has a slight advantage, while for bigger cases the initialization with a power flow solution greatly improves the performance of the solver. The slack initialization is slower in most of the cases as expected, but in some cases it performs at the same level as the power flow initialization with no slacks. Bear in mind that this option is not expected to be used for improved speed, rather for improved convergence.

An important remark to be made is that the power flow execution time is not included in the runtime of the solver, as it is assumed that the user would be able to calculate it once with the built-in power flow solver, and then use it directly for all the desired studies. Another remark is that this comparison is made with the Python adaptation of Matpower, while the original implementation runs in Matlab/Octave and should be quicker.

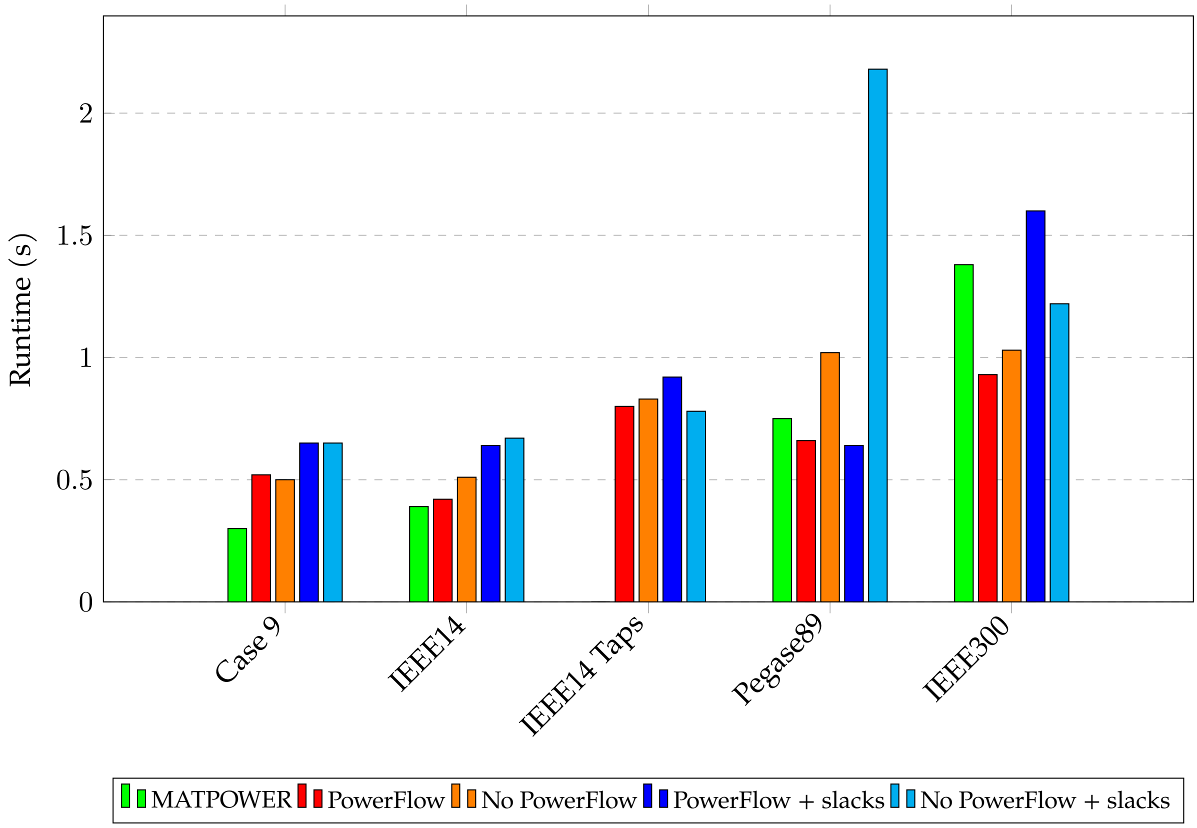


Figure 6.15: Runtime comparison for different cases.

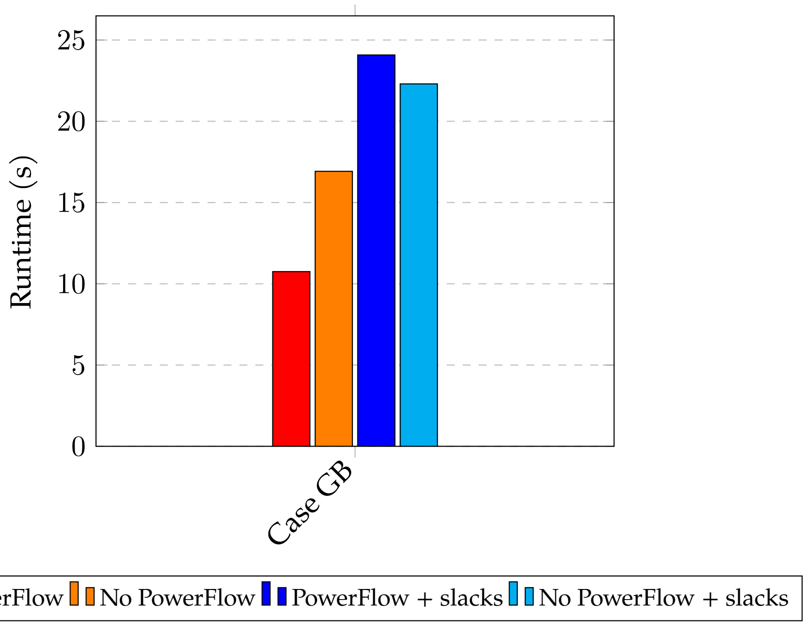


Figure 6.16: Runtime comparison for the GB case.

7 Conclusion

The work developed in this thesis has been focused on the implementation of an AC-OPF solver in GridCal. The results shown in Chapter 6 demonstrate the solver's capabilities and the impact of the features implemented in addition to the preexisting works in the topic.

The standard AC-OPF problem as implemented in Matpower has been successfully replicated, incorporating the option to start from a solution of the power flow with the effect of reducing the necessary iterations to reach convergence in some cases.

The addition of transformer setpoints optimization has also been tested, showing that adding these tap variables in the optimization problem yields a working point that lowers the costs of that found without considering them.

DC links have been added in a simplistic form in the model as a tool to perform quick analysis between islands connected through a DC line.

Reactive power limitations have also been added to the model as a simple relation to also have the possibility to consider such constraints in the optimization problem.

Overall, the implementation of this solver in the GridCal environment can be considered a success. As a tool tailored for Redeia's needs, the whole GridCal package is expected to be used for the future planning of the Spanish electrical grid, which involves the usage of the solver developed in this project, and due to the open-source nature of the package, other TSOs or DSOs could also benefit from it.

7.1 Further Work

The solver holds a lot of potential for enhancement, with many possible improvements of the existing features as well as future additions of new functionalities. Some of the future works that are being discussed are the following:

- Related to raw performance, some intermediate calculations for the gradients and Hessians could be optimized to reduce the time needed to solve the problem. This involves using the sparse structures used in the process of creating the Jacobian and Hessian matrices, although it is not a trivial task and requires a deep understanding of the data structure involved.

- The constraints related to reactive power limitations and DC links have been added in a primitive way, since they were considered as secondary implementations which were nice to have, but still lack some information needed to correctly model them.

- For the reactive power constraints, the grid model will have to include the parametrized equation for their generation curves to be able to obtain the analytic derivatives necessary to include in the gradients and Hessians of the optimization problem.

- For the DC links, the model will need a better implementation of the converters that connect the ends of the line, which will require adequate modelling of the control that they follow, the losses of the link and the power flow equations that are used to calculate the power transfer. It is possible that a full AC/DC solver would be needed to correctly model the power flow of the link.

- The addition of relaxations to the problem is also a possibility if there is a need to solve a great amount of cases for planning purposes, as this approach would increase the speed of the solver. It would come at the cost of precision, but for some cases it could be a good trade-off.

- Time-series studies could be considered by introducing new constraints and elements that have to be modelled considering a time variable, such as batteries and its state of charge, flexibility of loads, generation forecasting for renewable energies...

Acknowledgments

I want to thank Marc Cheah and Josep Fanals, my thesis supervisors from UPC and eRoots respectively, for the opportunity to develop this project in collaboration with Redeia and for their guidance and support through the process. I'd also like to thank Santiago Peñate for his help in developing the tool and integrating it in GridCal in collaboration with Josep and me.

A Environmental, Social, and Gender Impact

The present chapter considers the impact that this project has in the society.

A.1 Environmental Impact

The environmental impact of this project is positive, as it contributes to the development of a tool that will help to optimize the Spanish electrical grid. This optimization will allow the grid to operate more efficiently, reducing the amount of energy that is wasted and minimizing the environmental impact of the grid. By optimizing the grid, it will be possible to reduce the amount of energy that is generated from fossil fuels, which will help to reduce greenhouse gas emissions and combat climate change. In addition, the optimization of the grid will help to reduce the amount of energy that is lost during transmission, which will help to reduce the overall energy consumption of the grid. This will help to reduce the environmental impact of the grid and contribute to the development of a more sustainable energy system.

A.2 Social Impact

The social impact of this project is also positive, as it will help to improve the reliability and stability of the Spanish electrical grid. By optimizing the grid, it will be possible to reduce the risk of blackouts and improve the overall reliability of the grid. This will help to ensure that the grid can meet the energy needs of the population and provide a reliable source of electricity. In addition, the optimization of the grid will help to reduce the cost of electricity for consumers, as it will help to reduce the amount of energy that is wasted and improve the overall efficiency of the grid. This will help to reduce the financial burden on consumers and contribute to the development of a more sustainable energy system.

B Time Planning

Figure B.1 shows the temporal evolution of the various tasks that have constituted the project.

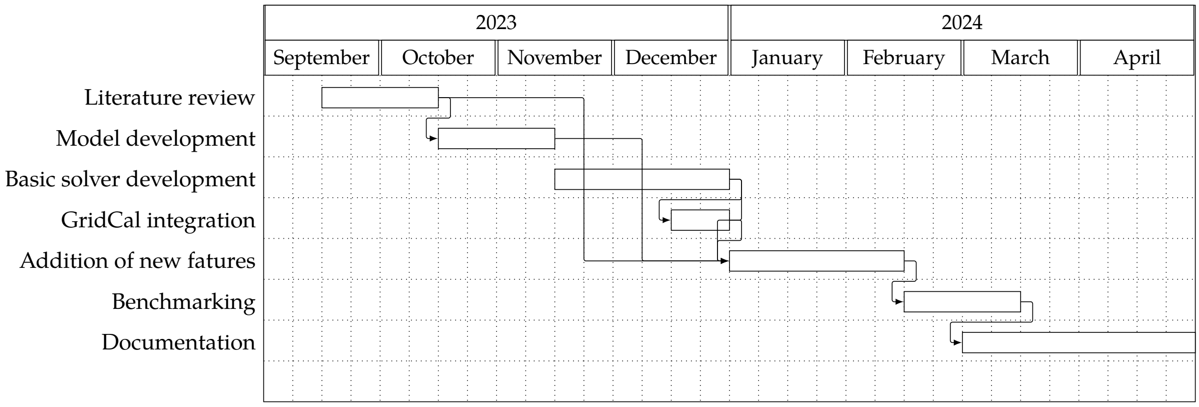


Figure B.1: Gantt Chart of the project.

C Budget

The costs associated to the project are those that have to bee counted for the contract with Redeia, which considering it is an open-source implementation, only consider the consultancy workforce of the employees working on the project. In this case, there has been one full-time development engineer and one part-time supervisor (10-hour week). From Figure B.1 we can see that the project has lasted 32 weeks, although only 26 of them can be considered as working weeks of the project since the documentation work is outside of the contract. The complete costs are detailed in Table C.1.

Table C.1: Total Costs.

Concept	**Unit cost (€/h)**	**Quantity (h)**	**Total (€)**
Development engineer	25.00	1,040	26,000.00
Supervisor	30.00	260	7,800.00
Total			33,800.00

References

[1] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[2] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.

[3] R. D. Zimmerman and H. Wang, “Matpower interior point solver mips 1.4 user’s manual,” 2020.

[4] H. Wang, C. E. Murillo-Sanchez, R. D. Zimmerman, and R. J. Thomas, “On computational issues of market-based optimal power flow,” *IEEE Transactions on power Systems*, vol. 22, no. 3, pp. 1185–1193, 2007.

[5] H. Abaali, T. Talbi, and R. Skouri, “Comparison of newton raphson and gauss seidel methods for power flow analysis,” *International Journal of Energy and Power Engineering*, vol. 12, no. 9, pp. 627–633, 2018.

[6] P. Virtanen et al., “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2

[7] R. D. Zimmerman, “Ac power flows, generalized opf costs and their derivatives using complex matrix notation,” *MATPOWER Tech. Note*, vol. 2, pp. 60–62, 2010.

[8] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, “Matpower: Steady-state operations, planning, and analysis tools for power systems research and education,” *IEEE Transactions on power systems*, vol. 26, no. 1, pp. 12–19, 2010.

[9] R. Lincoln, “Pypower 5.1.16,” *PYPOWER/api*, 2023.

[10] J. H. Chow, *Time-scale modeling of dynamic networks with applications to power systems*. Springer, 1982, p. 70.

[11] C. Josz, S. Fliscounakis, J. Maeght, and P. Panciatici, “Ac power flow data in matpower and qcqp format: Itesla, rte snapshots, and pegase,” *arXiv preprint arXiv:1603.01533*, 2016.

[12] M. Adibi, F. Alvarado, and R. Christie, “Power systems test case archive,” *University of Washington*, 1993.

[13] I. Dabbagchi and R. Christie, “Power systems test case archive,” *University of Washington*, 1993.