INVESTIGATION OF PERTURBATION APPROACHES IN THE SOLUTION OF ILL-CONDITIONED LARGE-SCALE POWER FLOW PROBLEMS

LAICE NEVES DE OLIVEIRA

PHD THESIS
IN ELECTRICAL ENGINEERING

ELECTRICAL ENGINEERING DEPARTMENT

TECHNOLOGY FACULTY

UNIVERSITY OF BRASÍLIA

Universidade de Brasília
Faculdade de Tecnologia
Departamento de Engenharia Elétrica

Investigation of Perturbation Approaches in the Solution of Ill-conditioned Large-scale Power Flow Problems

LAICE NEVES DE OLIVEIRA

TESE DE DOUTORADO SUBMETIDA AO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM ENGENHARIA ELÉTRICA.

Orientador: Prof. Francisco Damasceno Freitas, D.Sc. (UnB)

APROVADA POR:

Prof. Kleber Melo e Silva, D.Sc. (UnB)
(Presidente)

Prof. Rodrigo Andrade Ramos, D. Sc. (USP/São Carlos)
(Examinador Externo)

Prof. Robson Celso Pires, D. Sc. (UNIFEI)
(Examinador Externo)

Prof. Fernando Cardoso Melo, D. Sc. (UnB)
(Examinador Interno)

Prof. Francis Arody Moreno Vásquez, D. Sc. (UnB)
(Suplente)

Brasília - DF, dezembro de 2024.

FICHA CATALOGRÁFICA

DE OLIVEIRA, LAICE NEVES

Investigation of Perturbation Approaches in the Solution of Ill-conditioned Large-scale Power Flow Problems. [Brasília/DF] 2024.

xiv, 78p., 210 x 297 mm (ENE/FT/UnB, Doutor, Tese de Doutorado, 2024).

Universidade de Brasília, Faculdade de Tecnologia, Departamento de Engenharia Elétrica.

1. Power flow
2. Ill-conditioned systems
3. Perturbation methods
4. Large-scale systems

I. ENE/FT/UnB
II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

DE OLIVEIRA, LAICE NEVES (2024). Investigation of Perturbation Approaches in the Solution of Ill-conditioned Large-scale Power Flow Problems. Tese de Doutorado, Publicação PPGEE 211/25, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 136p.

CESSÃO DE DIREITOS

AUTOR: LAICE NEVES DE OLIVEIRA

TÍTULO: Investigation of Perturbation Approaches in the Solution of Ill-conditioned Large-scale Power Flow Problems.

GRAU: Doutor ANO: 2024

LAICE NEVES DE OLIVEIRA

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

Faculdade de Tecnologia - FT

Departamento de Engenharia Elétrica(ENE)

Brasília - DF CEP 70919-970

To the Author of life and the source of all inspiration.

ACKNOWLEDGEMENTS

I express my deep gratitude to the presence of God, whose guidance has been my constant strength throughout this journey.

To my mother, Nazilda, who continues to be present even miles away and whose constant love and support are my source of inspiration and motivation, and to my entire family who supported me.

I am deeply grateful to my advisor, Prof. Dr. Francisco Damasceno Freitas, for his exceptional guidance, academic expertise, and firm commitment, essential pillars for the development of this research.

My sincere thanks to the Postgraduate Program in Electrical Engineering (PPGEE) for the opportunity to develop academic research under the guidance of qualified and supportive professors. Part of this thesis was also financially supported through a scholarship granted by the Brazilian Coordination for Improvement of Higher Education Personnel (CAPES). All financial support provided by CAPES throughout the development of this thesis is greatly acknowledged.

I would also like to thank the National Electric System Operator (ONS) for allowing the development of this research in parallel with the operation of the Brazilian National Interconnected System and all the colleagues who supported this work.

I express my gratitude to professors Dr. Kleber Melo e Silva, Dr. Rodrigo Andrade Ramos, Dr. Robson Celso Pires, and Dr. Fernando Cardoso Melo for kindly accepting the invitation to be part of the defense committee for my doctoral thesis. Your invaluable expertise, constructive feedback, and insightful comments have significantly contributed to the refinement and improvement of this work.

Finally, I thank the University of Brasilia (UnB) for all the support provided and for the commitment to excellence in teaching and research.

ABSTRACT

Title: Investigation Of Perturbation Approaches in the Solution of Ill-conditioned Large-scale Power Flow Problems

This Ph.D. thesis presents approaches to calculate the solution of the power flow problem (PFP) that involves ill-conditioned and large-scale systems. Iterative non-linear methods are employed to solve the problem. The main strategy used in problem formulation is based on applying a conditioning step to the initial of the iterative methods. This step consists of modifying the initial estimate of the iterative method through a process that involves the Jacobian matrix and the *mismatch* of the balance equations. Four approaches were developed. In the first approach, the Jacobian matrix is used to form a linear system whose *perturbed* matrix results in a better condition number. Another perturbation approach proposed in this work based on modal analysis demonstrates that the primary cause of the ill-conditioning problem is associated with the smallest magnitude eigenvalue of the first iteration of the Jacobian matrix. In addition, a procedure is proposed to circumvent this problem by shifting away from zero, the smallest magnitude eigenvalue of the Jacobian matrix. The third approach uses Tikhonov's regularization to initialize the iterative process. Subsequent iterations use the result of a regularized normal equation, where the regularization parameter is selected using the traditional L-curve technique. Finally, the last proposed approach is based on a two-step hybrid method to calculate the PFP solution. The proposed techniques were investigated using the classical NR method, the Heun-King-Werner (HKW) method, and some variants. The performance of the proposed approaches is evaluated for various scenarios and test systems, including a 109,000-bus system. The results demonstrated that the investigated methods significantly improved the convergence process of the iterative techniques used to solve large and ill-conditioned PFPs, including the classical Newton-Raphson method.

Keywords: Power flow problem; Newton-Raphson method; ill-conditioned systems; conditioning step; Heun-King-Werner; MATPOWER.

RESUMO

Título: Investigação de Abordagens de Perturbação na Solução de Problemas de Fluxo de Carga de Grande Porte e Mal-Condicionados

Esta tese de doutorado apresenta abordagens para calcular a solução do Problema de Fluxo de Potência (PFP) envolvendo sistemas mal-condicionados e de grande porte. A estratégia baseia-se em aplicar uma etapa de condicionamento à estimativa inicial usada nos métodos iterativos. Essa etapa consiste em modificar a estimativa inicial do método iterativo através de um processo que envolve a matriz Jacobiana e o *mismatch* das equações de balanço, ambas calculadas para a estimativa inicial. Foram desenvolvidas quatro estratégias. Na primeira, a matriz jacobiana é então usada para formar um sistema linear cuja matriz *perturbada* resulta em um melhor número de condição. Uma segunda abordagem de perturbação proposta baseia-se em análise modal e demonstra que a causa primária do problema de mau condicionamento está associada ao autovalor de menor magnitude da primeira iteração da matriz Jacobiana. Deste modo, é proposto um procedimento para contornar este problema afastando da região próximo de zero o autovalor de menor magnitude da matriz Jacobiana. A terceira abordagem baseia-se na utilização da regularização de Tikhonov para inicializar o processo iterativo. Nessa abordagem, as iterações subsequentes utilizam o resultado de uma equação normal regularizada, onde o parâmetro de regularização é selecionado utilizando a técnica tradicional da curva L. Por fim, a última abordagem proposta baseia-se em um método híbrido para calcular a solução do PFP que é composto por duas etapas. O desempenho das abordagens propostas é avaliado para uma variedade de cenários e de sistemas-teste, incluindo um sistema de 109.000 barras. Os resultados obtidos demonstraram que os métodos investigados conseguiram melhorar significativamente o processo de convergência das técnicas iterativas usadas para resolver PFPs mal-condicionados e de grande porte, incluindo o método clássico de Newton-Raphson.

Palavras-chave: Problema de Fluxo de Potência, Método de Newton-Raphson, Sistemas Mal-Condicionados, Etapa de Condicionamento; Heun-King-Werner; MATPOWER.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Y_{BUS}	Admittance Matrix
Y_{ij}	ij-th element of the Admittance Matrix
B	Susceptance matrix
V_i	Voltage magnitude at bus i
θ_i	Phase angle at bus i
S_i	Complex power at bus i
ΔP_i	Mismatches of active power at bus i
ΔQ_i	Mismatches of reactive power at bus i
P_i^{sp}	Active power specified at bus i
Q_i^{sp}	Reactive power specified at bus i
P_i	Total injection of active power at bus i.
Q_i	Total injection of reactive power at bus i.
Q_i^{cal}	Reactive power calculated at bus i
I_i	Total current at node i
n_b	Number of network buses
n_{PQ}	Total number of PQ buses in the system
n_{PV}	Total number of PV buses in the system
Q_i^{min}	Minimum limit of reactive power injection at bus i
Q_i^{max}	Maximum limit of reactive power injection at bus i
$J^{(k)}$	Jacobian matrix in the k-th iteration

$\Delta\theta_i$	Increment of phase angle at bus i
ΔV_i	Increment of voltage magnitude at bus i
$a_{p,i}$	Elements of the Runge-Kutta matrix
b	Weight vector
h	Integration step
p	Number of stages
$\mathbf{x}$	State vector
$\mathbf{x}^{(0)}$	Initial estimate of the state vector
$\psi^{(k)}$	Correction parameter of the Heun-King-Werner method
$\widetilde{\psi}$	Considered limit for automatic switching from the iterative HKW process to the NR method
h_{min}	Minimum allowed step size h
h_{max}	Maximum allowed step size h
μ	Parameter to initialize the step size h
α	Error limit for updating the step size h
ϵ	Tolerance for convergence
λ	Damping factor
κ	Condition Number
δ	Perturbation parameter
σ_{min}	Minimum singular value
σ_{max}	Maximum singular value
$iter_{max}$	Maximum number of iterations

GLOSSARY

BM	Broyden Method
CN	Continuous Newton's method
CS	Conditioning Step
FDXB	Fast Decoupled method considering the XB version
FNR	Fast Decoupled Newton-Raphson method
GS	Gauss-Seidel method
HELM	Holomorphic Embedding Load-Flow Model
HKW	Heun-King-Werner method
IM	Iterative Method
LM	Levenberg-Marquardt method
MLP	Maximum Load Point
NR	Newton-Raphson method
NR1J	Newton-Raphson method with freezing of the Jacobian matrix in the 1st iteration
ODE	Ordinary Differential Equations
OM	Optimal Multiplier
PF	Power Flow
PFP	Power Flow Problem
RK	Classical Runge-Kutta method
RK4	fourth-order Runge-Kutta method
RK4B	fourth-order Runge-Kutta-Broyden's method

RSM	Robust Simple Method
SEJ	Smallest eigenvalue of the Jacobian matrix

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Power Flow studies consist of an essential tool for determining the steady-state operating point of electrical power systems. Based on the solution obtained for the problem, it is possible to investigate conditions that guarantee the successful operation of the power system under several conditions (GLOVER *et al.*, 2012).

The Power Flow Problem (PFP) is formulated as a set of nonlinear algebraic equations, and due to its complex nature, iterative methods are typically employed to obtain a solution. Among these methods, the classical Newton-Raphson (NR) technique has traditionally been adopted as the standard approach (KUNDUR, 2007). The NR method relies on an initial estimate that is ideally situated near the solution. However, several factors in today's power systems can introduce significant shifts in operating conditions. The widespread integration of renewable energy sources (WENG *et al.*, 2012), frequent reversals of power flows (TARANTO *et al.*, 2022), and load variations are just a few examples of such factors. As a result, the voltage magnitudes and angles within the system can undergo substantial changes.

The dynamic nature of these changes poses a challenge in selecting an appropriate initial estimate for the Power Flow (PF) solver. A common practice is to initialize the PF solver using a flat start estimation, *i.e.*, all bus voltage phase angles equal to zero and voltage magnitudes in load buses equal 1 pu. However, this initial estimate may deviate significantly from the desired solution in the analyzed power system model, and consequently, classical iterative methods may diverge (TOSTADO-VÉLIZ *et al.*, 2020a; TOSTADO-VÉLIZ *et al.*, 2021).

According to Milano (2009), when the solution of the power flow problem cannot be obtained using conventional methods and the standard flat-start initial guess, the problem is considered ill-conditioned, despite the problem having a valid solution. Consequently, there is a need to

develop more robust and accurate techniques for initializing the PF solver in order to enhance its effectiveness in handling diverse operating conditions.

Several researchers have investigated different approaches to address ill-conditioned PFPs. A method with optimal multipliers (OM) was first addressed in rectangular coordinates (IWAMOTO; TAMURA, 1981) and studied in the polar coordinates in (BRAZ *et al.*, 2000). In (TATE; OVERBYE, 2005), the superior performance of the polar coordinate form was proved compared to the rectangular coordinate. In the previous studies involving polar coordinates, only the information of the second-order term from the Taylor's expansion is used in the computation of OM. An NR method with OM using the sum of high-order terms of the Taylor's expansion in polar coordinates was proposed in (PAN *et al.*, 2019). However, the study assumes that the maximum absolute deviation of states approaches 0 or the OM approaches the unitary value. This is a weak numerical property concerning the application of ill-conditioning and large-scale systems.

Alternatively, techniques based on Continuous Newton's (CN) philosophy, in which any numerical integration method can be adapted to solve the PFP, were proposed in (HETZLER, 1997). Following this philosophy, in (MILANO, 2009), the problem was formulated as a set of Ordinary Differential Equations (ODE) in which two new methodologies to solve the PFP based on numerical methods were presented: the Robust Simple Method (RSM), which is based on the methodology of Euler forward and the fourth-order Runge-Kutta method (RK4). Moreover, similar study was proposed in (MILANO, 2019), using an Euler backward (implicit) approach. Along the same lines investigated by (MILANO, 2009), a methodology based on the combination of RK4 and the Broyden Method (BM) was proposed in (TOSTADO-VÉLIZ *et al.*, 2018a). The authors present the fourth-order Runge–Kutta Broyden method (RK4B) to reduce the number of inversions in the Jacobian matrix to just one. In (TOSTADO-VÉLIZ *et al.*, 2020a), a method was developed for well- and ill-conditioned electrical systems models. The resulting method is a combination of the King-Werner and Heun methods, and for this reason, it was called the Heun-King-Werner (HKW) method. This method presented a superior performance in terms of computational burden and iterations for convergence for ill-conditioned systems compared to methods such as Newton-Raphson, Euler, Runge-Kutta, among others. However, it depends on the setting of parameters highly sensitive to changes when applied to

ill-conditioned and large-scale power system models.

Other recent advances for realistic large-scale ill-conditioned systems computational methods for the power flow equations were reported in (TOSTADO-VÉLIZ *et al.*, 2021; TOSTADO-VELIZ *et al.*, 2020; FREITAS; SILVA, 2022; PANDEY *et al.*, 2021; JEREMINOV *et al.*, 2019). In (TOSTADO-VÉLIZ *et al.*, 2021) a robust and efficient solution framework, competitive in both well- and ill-conditioned cases were proposed. The explored technique presents some similitudes with the OM. Therefore, it is also based on the truncation of the Newton's increment vector. However, it introduces paradigms aiming at overcoming some of the difficulties posed by this kind of technique. In (TOSTADO-VELIZ *et al.*, 2020), it was also developed an efficient and robust power flow method but based on a Semi-Implicit approach which incorporates numerical arrangements for enhancing its potentialities. The iterative algorithm reaches features by employing a combination of different numerical arrangements such as Lavrentiev's regularization (ARGYROS; GEORGE, 2014), Chebyshev-like method with cubic convergence (BABAJEE *et al.*, 2010), and Heun's method (BUTCHER, 2008).

Unlike the CN philosophy techniques, which have a dynamic form, there are methods characterized by a parametric static approach, in which several intermediate nonlinear systems are sequentially solved until operating on the system that is really of interest. The methods that explore the homotopy technique (FREITAS; SILVA, 2022) have these characteristics. Several researches have been and continue to be developed in this direction. For example, PFP homotopy-based approaches were investigated in (JEREMINOV *et al.*, 2019; PANDEY *et al.*, 2021; FREITAS; SILVA, 2022). This methodology solves the PFP to satisfy a homotopy path (curve) formed by the states computed for a given homotopy parameter set. Then, the PFP needs to be solved for each path point. However, only the latter one has its result of interest. The way how the homotopy problem is solved distinguishes the specific approach. In (JEREMINOV *et al.*, 2019; PANDEY *et al.*, 2021) an approach based on electric circuit is considered. The authors in (FREITAS; SILVA, 2022) presented a methodology to depart from the flat start estimate, whose solution of the parameterized PFP for this first point coincides with the initial guess. In contrast, the last point of the set gives the effective PFP solution.

All methodologies previously discussed search the solution aiming for an accurate result for the PFP. Depending on the adopted strategy, it can present limitations to convergence or even

inferior performance computationally. The limitation may be, for instance, in some stagnation or a low convergence rate.

The investigations previously mentioned motivated this doctoral thesis to search for new contributions in the field of robust methods for solving the PFP with refined techniques.

1.2 OBJECTIVE

Considering the aspects and contributions pointed out for computing the solution of the PFP in other publications, the main objective of this doctoral thesis is to investigate and improve techniques for solving the PFP in ill-conditioned large-scale power system models. The proposed approaches aim to overcome the limitations of traditional methods, such as the Newton-Raphson solver, and provide accurate solutions, avoiding convergence problems. In this context, the specific objectives include:

- Develop and implement methodologies based on Jacobian matrix perturbation to solve power flow problems in large-scale and ill-conditioned systems, evaluating their effectiveness in enhancing convergence and reducing computational cost;

- Conduct simulations using large-scale system models characterized by ill-conditioning, aiming to evaluate the performance of the proposed methodologies and comparing it with the performance of traditional methods, such as the Newton-Raphson method;

- Demonstrate the sensitivity of the results for an iterative PFP, starting from an initial estimate based on a flat start. The study will emphasize how applying a conditioning step in the first iteration benefits the process and proves to be sufficient for achieving successful convergence of the problem;

- Proposal of different techniques to improve the resolution of the PFP for ill-conditioned large-scale power system models;

- Perform simulations in networks considering base cases and critical loading conditions, including operational power reactive limits in power generators;

- Investigate the performance of direct and iterative linear methods, and reordering techniques to optimize the computational cost of ill-conditioned large-scale PFP.

The proposed techniques investigated in this work basically involve adaptations to traditional methods only in the calculations of the first iteration. As a first idea, the usual Jacobian of the PFP is computed using the initial estimate. Then, it is modified by introducing a perturbed term to improve the condition number of the resulting matrix. Another perturbation technique, based on a modal approach, is also proposed. It consists in modifying the ill-conditioned PFP Jacobian matrix for the initial iterate of the standard Newton's method by moving away just its smallest magnitude eigenvalue from near zero. The state deviations for this modified condition are then efficiently computed. This procedure is performed by adding a 1-rank perturbation matrix to the Jacobian matrix, but just for the first iteration and removing it for all others. Another proposed approach incorporates the well-known Tikhonov's regularization as a conditioning step to initiate the iterative process. Tikhonov's regularization is crucial in reducing the norm deviation of states, facilitating efficient convergence during the iterative process. A hybrid approach to solving the ill-conditioned power flow problem is also presented as a fourth alternative. This approach is based on the same principle of improving the ill-conditioning of the Jacobian matrix for the first iteration. The technique comprises a hybrid method because the first step is characterized by the calculation of an inaccurate solution of the PFP, while in the second, the high-accuracy solution is computed. The low precision solution is determined by a homotopy technique (FREITAS; SILVA, 2022). However, the high-precision solution is determined by another iterative method (IM) defined by the user, for example, the classical NR. Experiments were carried out in five ill-conditioned large-scale power system models, including a 109,000-bus system, revealing that the proposed initial estimate always leads these methods to convergence.

1.3 CONTRIBUTIONS

This PhD Thesis presents some contributions to the field of iterative methods for solving the power flow problem involving ill-conditioned and large systems. The main contributions of this work can be summarized as follows:

- Introduction of a procedure to improve the calculation of the result of the first iteration of an iterative nonlinear problem, such as PFP, in which without this treatment, the convergence fails;

- Developing conditioning techniques that allow the successful application of the classical NR solver in the resolution of large-scale and ill-conditioned power system models, overcoming traditional convergence limitations;

- Proposition of techniques for handling scenarios with limited or unreliable initialization information, critical for contingency analysis and restoration planning;

- Development of appropriate techniques for use under uncertain scenarios, such as load variations and renewable generation fluctuations, providing a reliable solution for planning under probabilistic conditions;

- Exploration of the root causes of ill-conditioning in power flow problems, emphasizing the role of the Jacobian matrix's smallest eigenvalues and proposal of a novel eigenvalue-shifting strategy to mitigate ill-conditioning;

- Presentation a two-step hybrid methodology where approximate states computed in the first step are refined using iterative techniques in the second step, combining the strengths of different solvers;

- Investigation of the computational impact of direct methods, iterative linear methods, and reordering techniques in the solution process of the Power Flow Problem, involving large-scale ill-conditioned systems;

- Presentation of extensive numerical results for ill-conditioned and large-scale systems, validating the effectiveness of the proposed approaches using the MATPOWER tool, while demonstrating significant potential for integration into commercial power system software.

Furthermore, it is worth noting that the studies conducted throughout this doctoral research have yielded the following papers to date:

- FREITAS, Francisco Damasceno; DE OLIVEIRA, Laice Neves. A Fractional Order Derivative Newton-Raphson Method for the Computation of the Power Flow Problem Solution in Energy Systems. Fractional Calculus and Applied Analysis, v. 27, p. 1-32, 2024.

- FREITAS, Francisco Damasceno; DE OLIVEIRA, Laice Neves. Conditioning step on the initial estimate when solving ill-conditioned power flow problems. International Journal of Electrical Power & Energy Systems, v. 146, p. 108772, 2023.

- DE OLIVEIRA, Laice Neves; FREITAS, Francisco Damasceno; MARTINS, Nelson. A Modal-based Initial Estimate for the Newton Solution of Ill-conditioned Large-scale Power Flow Problems. IEEE Transactions on Power Systems, V. 38, N. 5, pp. 4962-4965, Sep. 2023.

- FREITAS, Francisco Damasceno; DE OLIVEIRA, Laice Neves. Two-step hybrid-based technique for solving ill-conditioned power flow problems. Electric Power Systems Research, v. 218, p. 109178, 2023.

- DE OLIVEIRA, Laice Neves; FREITAS, Francisco Damasceno. Estratégia baseada em Perturbação Inicial da Matriz Jacobiana para a Solução de Problemas de Fluxo de Carga em Sistemas de Grande Porte e Mal-condicionados. In: Simpósio Brasileiro de Sistemas Elétricos - SBSE, 2023, Manaus - AM.

- DE OLIVEIRA, Laice Neves; FREITAS, Francisco Damasceno. An Ill-conditioned System Study in a 11-bus Network and Characterization of the Problem considering the AC Power Flow in the MATPOWER Tool. In: Simpósio Brasileiro de Sistemas Elétricos - SBSE, 2023, Manaus - AM.

- DE OLIVEIRA, Laice Neves; FREITAS, Francisco Damasceno. Regularização Parcial de Tikhonov aplicada na Resolução do Problema de Fluxo de Carga em Sistemas Elétricos de Potência de Grande-Porte e Mal-condicionados. In: XXV Congresso Brasileiro de Automática - CBA, 2024, Rio de Janeiro - RJ.

Part of the results of this thesis was highlighted at the XXV Brazilian Congress of Automation, where the paper entitled 'Partial Tikhonov's Regularization Applied to the Solution of the Power Flow Problem in Large-Scale and Ill-Conditioned Power Systems' received an honorable

mention. This distinction recognizes the significant contribution of the work in advancing the robustness and efficiency of solution methods for large-scale and ill-conditioned power flow problems.

1.4 OUTLINE

This thesis includes eight chapters. Besides the introductory chapter, the remaining are:

- Chapter 2 explains the fundamentals of the Power Flow Problem, as well as the issues related to ill-conditioning and the traditional solution methods;
- Chapter 3 presents direct and iterative methods for solving linear systems, as well as reordering techniques used to improve the efficiency of these methods;
- Chapter 4 details the state of the art of the methods used to solve the ill-conditioned power flow problem;
- Chapter 5 describes alternative methodologies to the Newton-Raphson method used to solve the ill-conditioned PFP, such as techniques based on the Newton-Continuous methodology, Levenberg-Marquadt method, and Tikhonov's regularization;
- Chapter 6 presents the main methodology proposed in this work;
- Chapter 7 presents the numerical results obtained based on the methodologies analyzed involving ill-conditioned and large-scale systems;
- Chapter 8 presents the main conclusions, and this chapter duly draws future perspectives.
- Also, Appendix A details the numerical fundamentals for implementing complex expressions in the power flow problem.

CHAPTER 2

FUNDAMENTALS

The Power Flow Problem consists of determining the state of the network, the distribution of flows, and other variables of interest (KUNDUR, 2007; JR; GRAINGER, 1994). It is an essential tool for determining the state of the electrical network in a steady state, for real-time operations, as well as in planning, expansion, and control studies of electrical power systems (TOSTADO-VÉLIZ *et al.*, 2021).

The PFP formulation is based on a set of non-linear algebraic equations that represent the physical characteristics of the electrical systems. Various solution methods have been developed and studied for decades. In this chapter, the basic formulation of power flow is presented, as well as the problem discussion associated with ill-conditioning. Subsequently, more traditional PFP solution methods are described.

2.1 POWER FLOW EQUATIONS

The goal of power flow calculations is to determine the voltages at all nodes and the currents in all branches of the grid. Typically, for a grid with n independent nodes, the equations (2.1)–(2.2) are applicable.

$$\begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{13} \\ Y_{21} & Y_{22} & \cdots & Y_{23} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \cdots & Y_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} \tag{2.1}$$

$$[\mathbf{Y}]\,[\mathbf{V}] = [\mathbf{I}] \tag{2.2}$$

The node voltage vector is denoted as V, and the node current injection vector is denoted as I. The admittance matrix Y or Y_{BUS} contains elements that describe the electrical properties

of the network. The diagonal element Y_{ii} represents the self-admittance at node i, which is the sum of all branch admittances connected to that node. Conversely, the off-diagonal element Y_{ij} is the negative value of the branch admittance between nodes i and j. If no branch exists between nodes i and j, then Y_{ij} is zero.

The complex power at node i is expressed as the product of the voltage and current vectors, represented in (2.3). Consequently, the current flow at each node is represented in (2.4), where the current entering each node from other nodes equals the current drawn by the complex power load at that node.

$$\overline{S}_i = \overline{V}_i \overline{I}_i^* = P_i + jQ_i \tag{2.3}$$

$$\overline{I}_i = \sum_{k=1}^{n_b} \overline{Y}_{ik} \overline{V}_k = \frac{P_i - jQ_i}{\overline{V}_i^*}, i \,\epsilon\, \beta \tag{2.4}$$

in which $\beta = \{1,2,...,n_b\}$, n_b is the number of network buses, $\overline{Y}_{ik}$ is the element (i,k) of the admittance matrix Y, and P_i and Q_i are the total injection of active and reactive power at node i.

Therefore, (2.4) can be rewritten as (2.5):

$$P_i + jQ_i = \overline{V}_i \left[\sum_{k=1}^{n_b} \overline{Y}_{ik} \overline{V}_k \right]^* \tag{2.5}$$

Since $\overline{V}_i = V_i e^{j\theta_i}$ and $\overline{Y}_{ik} = Y_{ik} e^{j\delta_{ik}} = G_{ik} + jB_{ik}$, with $i,k = 1, 2, \cdots, n_b$, , then (2.5) becomes

$$P_i + jQ_i = V_i \sum_{k=1}^{n_b} Y_{ik} V_k e^{j(\theta_i - \theta_k - \delta_{ik})} \tag{2.6}$$

By separating the real and imaginary components of equation (2.6), we can express the power balance equations as follows:

$$P_i = V_i \sum_{k=1}^{n_b} Y_{ik} V_k \cos(\theta_i - \theta_k - \delta_{ik}) \tag{2.7}$$

$$Q_i = V_i \sum_{k=1}^{n_b} Y_{ik} V_k \sin(\theta_i - \theta_k - \delta_{ik}) \tag{2.8}$$

Alternatively, when Y_{ik} is represented in rectangular coordinates, the power balance equations (PBE) can be written as:

$$P_i = V_i \sum_{k=1}^{n_b} V_k[G_{ik}\cos(\theta_i - \theta_k) + B_{ik}\sin(\theta_i - \theta_k)] \tag{2.9}$$

$$Q_i = V_i \sum_{k=1}^{n_b} V_k[G_{ik}\sin(\theta_i - \theta_k) - B_{ik}\cos(\theta_i - \theta_k)] \tag{2.10}$$

or, in terms of active and reactive power mismatches:

$$\Delta P_i = P_i^{sp} - V_i \sum_{k=1}^{n_b} V_k[G_{ik}\cos(\theta_i - \theta_k) + B_{ik}\sin(\theta_i - \theta_k)] \tag{2.11}$$

$$\Delta Q_i = Q_i^{sp} - V_i \sum_{k=1}^{n_b} V_k[G_{ik}\sin(\theta_i - \theta_k) - B_{ik}\cos(\theta_i - \theta_k)] \tag{2.12}$$

in which ΔP_i and ΔQ_i correspond to *mismatches* of active and reactive power at bus #i, respectively, V_i and θ_i are, respectively, the voltage magnitude and phase angle of a nodal voltage phasor at bus #i, $\overline{V}_i = V_i\angle\theta_i$; P_i^{sp} is the net active power injected into bus #i, while Q_i^{sp} is the net reactive power injected into the same bus; G_{ik} and B_{ik} are the real and imaginary parts of the entry $\overline{Y}_{ik} = G_{ik} + jB_{ik}$ of the system bus admittance matrix or simply matrix Y_{BUS}. Additionally, P_i^{sp} can be characterized by a generation P_{Gi}^{sp} and a load contribution P_{Li}^{sp}, in such way that $P_i^{sp} = P_{Gi}^{sp} - P_{Li}^{sp}$. In this sense, depending on the type of bus, the generation and load components can assume a voltage-dependent characteristic. Similarly, $Q_i^{sp} = Q_{Gi}^{sp} - Q_{Li}^{sp}$, where Q_{Gi}^{sp} is the component for the generation and Q_{Gi}^{sp}, for load. A traditional form is to represent the load by a well-known ZIP model, where the power voltage-dependent is composed of constant contributions of power, current, and impedance (KUNDUR, 2007).

In the traditional power flow formulation, there are four variables for each node: the node voltage magnitude V, the voltage angle θ, the node active power injection P, and the node reactive power injection Q. Typically, to solve equations (2.9)-(2.10), two of these variables need to be specified at each node. Based on the known variables, the nodes can be classified into (QIN, 2017; GLOVER *et al.*, 2012):

- **Slack Bus**: It is also known as the reference node because both the voltage magnitude

and voltage angle are specified. The voltage angles at other nodes are measured relative to the voltage angle at the Slack bus. The active and reactive power at the Slack bus are utilized to balance the power of the system. Consequently, this bus should be selected to ensure it has a plant with high power availability.

- **PV bus**: For this type of bus, the active power and voltage magnitude are specified. It typically represents the bus where a voltage source, such as generators with excitation systems, is connected. The voltage angle and reactive power at this node are unknown. Reactive power is usually subjected to constraints, such as those imposed by the generation capability curve, which limits the generator's reactive power output based on its active power production.

- **PQ bus**: At this type of bus, both active and reactive power are specified. It typically represents the busbar where loads are connected, such as points of power consumption or junctions of branches. However, the generator without voltage control abilities is also connected to PQ node, as a negative load to produce the current instead of consuming it. PQ buses are considered passive nodes, with their voltage magnitude and voltage angle determined by the current flow in the grid.

Table 2.1 summarizes the main characteristics of the different type of buses. While the mentioned buses are commonly classified in traditional power flow analysis, additional types have been contemplated for specific scenarios (QIN, 2017).

Table 2.1. Variables and parameters for each bus type in the classical power flow problem formulation.

Bus type	Variables	Data
Slack	P,Q	V, θ
PV	Q,θ	P, V
PQ	V, θ	P, Q

2.1.1 The Ill-Conditioned Power Flow Problem

The power balance equations in polar coordinates will be used in this work. The classical PBEs as described in the previous section are given by (KUNDUR, 2007)

$$0 = P_k^{sp} - V_k \sum_{m=1}^{n_b} V_m \left[G_{km} cos(\theta_k - \theta_m) + B_{km} sin(\theta_k - \theta_m) \right], \tag{2.13}$$

$$0 = Q_k^{sp} - V_k \sum_{m=1}^{n_b} V_m \left[G_{km} sin(\theta_k - \theta_m) - B_{km} cos(\theta_k - \theta_m) \right], \tag{2.14}$$

Assume that the set of PQ buses has n_{PQ} equations, while the set of PVs contributes with n_{PV}. Also, we assume that the AC network is synchronous with only one slack bus in such a way that $n_b = n_{PQ} + n_{PV} + 1$.

Therefore, the number of equations (2.13)-(2.14) is given by $N = 2n_{PQ} + n_{PV}$. Then, this set of nonlinear equations can be represented generically by

$$\mathbf{g}(\mathbf{x}) = \mathbf{0} \tag{2.15}$$

where the state vector is defined as $\mathbf{x} = [\boldsymbol{\theta}^T \ \ \boldsymbol{V}^T]^T \in \mathbb{R}^n$ and $\mathbf{g}(\mathbf{x})$ is a map of the type $\mathbf{g}(\mathbf{x}) : \mathbb{R}^n \mapsto \mathbb{R}^n$.

In instances where (2.15) exhibits well-conditioning, achieving a solution is straightforward. However, conventional solvers may encounter challenges in achieving convergence when confronted with ill-conditioned Power Flow equations. The ill-conditioning of a power system network may manifest as a result of the factors such as heavy loading conditions, large number of radial lines, existence of negative line reactance, lines with high resistance to reactance ratios or initial guess point outside the region of attraction or far from the solution (FAN, 1989; TRIPATHY *et al.*, 1982; TOSTADO-VÉLIZ *et al.*, 2018a).

In (MILANO, 2009), the Power Flow Problem is categorized into four distinct classes, delineated as follows:

- *Well-conditioned*: when the PF solution exists and is reachable using a flat initial guess (*i.e.*, all load voltage magnitudes equal to 1 pu and all bus voltage angles equal to 0) and a standard Newton–Raphson's method;

- *Ill-conditioned*: when the solution of the PFP does exist, but standard solution methods fail to get this solution starting from a flat initial guess. This scenario often occurs when the region of attraction for the power flow solution is either narrow or situated far from the initial guess. In such instances, the ineffectiveness of conventional power flow solution

methods can be attributed to the instability of the numerical method rather than any instability in the PF equations.

- *Bifurcation point*: whose PF solution exists but it is either a saddle-node bifurcation or a limit-induced bifurcation. In a system operating under maximum loading conditions, saddle-node bifurcations may occur. The inadequacy of standard or robust power flow methods in obtaining a solution is attributed to the singularity of the power flow Jacobian matrix at the solution point. On the other hand, limited-induced bifurcations happen when there is a physical limit in the system, such as insufficient reactive power from generators.

- *Unsolvable*: when the PF solution does not exist.

In this PhD thesis, the focus is on exploring the second classification and illustrating instances related to the ill-conditioning of the power flow problem.

If the initial estimation is within the solution point's attraction region, standard numerical methods typically converge. However, divergence may occur in certain instances. In such cases, where the initial guess is considerably distant from the solution, it becomes appropriate to employ robust numerical methods primarily before the Newton-Raphson (NR) method.

In the following subsections, the classic methods of Gauss-Seidel (GS), Newton-Raphson, and their variants are reviewed and described. Additionally, some particularities associated with each of them are discussed.

2.2 GAUSS-SEIDEL METHOD

The Gauss-Seidel method was first applied to solve the Power Flow Problem in 1957 by Glimn and Stagg (GLIMN; STAGG, 1957). This method is iterative and therefore requires an initial estimate of the voltage values to start the process. During the iterations, the initial values are continuously updated and replaced with newly calculated values. The process repeats until the solution converges to a stable value or until a maximum number of iterations is reached.

The convergence of the Gauss-Seidel method is highly sensitive to the initial values chosen. In many cases, an inappropriate choice of initial values can result in convergence problems,

causing the method to fail in finding a satisfactory solution. Additionally, the convergence rate of the method can be relatively slow, especially for large or complex systems. This limits the effectiveness of the method in practical applications where the speed of obtaining solutions is crucial (GLOVER *et al.*, 2012).

However, the Gauss-Seidel method has some advantages. Its implementation is relatively simple, and it requires less memory compared to other methods, such as Newton-Raphson. Furthermore, the method is particularly effective in systems where the admittance matrix is well-conditioned and the initial values are close to the actual solution. The methodology for executing the Gauss-Seidel method in the context of solving the power flow problem is detailed below (MILANO, 2010).

The Gauss-Seidel method is used to solve a system of linear equations represented in the following form:

$$\mathbf{Ay} = \mathbf{b} \tag{2.16}$$

It is important to note that the Gauss-Seidel method has seen widespread adoption in recent decades for solving the power flow problem, primarily due to its avoidance of matrix A factorization. However, with the relaxation of computation constrains, other methods have become more favored (MILANO, 2010).

Decomposing $\mathbf{A}$ in a lower triangular component $\mathbf{L}$ and a strictly upper triangular one $\mathbf{U}$,

$$\mathbf{A} = \mathbf{L} + \mathbf{U} \tag{2.17}$$

in which,

$$\mathbf{L} = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \mathbf{U} = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \tag{2.18}$$

Therefore, Equation 2.16 can be expressed as:

$$\mathbf{Ly} = \mathbf{b} - \mathbf{Uy} \tag{2.19}$$

The Gauss-Seidel method involves iteratively solving the left-hand side of equation using the current values of the elements of the vector on the right-hand side:

$$\mathbf{y}^{(i+1)} = \mathbf{L}^{-1}(\mathbf{b} - \mathbf{U}\mathbf{y}^{(i)}) \tag{2.20}$$

or, in tensorial form, can be represented as:

$$y_h^{(i+1)} = \frac{1}{a_{hh}} \left(b_h - \sum_{k=h+1}^{n} a_{hk} y_k^{(i)} - \sum_{k=1}^{h-1} a_{hk} y_k^{(i+1)} \right), h = 1, 2, \cdots, n \tag{2.21}$$

The iterative process stops if the maximum equation mismatch is less than a given tolerance ϵ:

$$max \left\{ \left| \mathbf{A}\mathbf{y}^{(i+1)} - \mathbf{b} \right| \right\} < \epsilon \tag{2.22}$$

or the maximum variable variation is smaller than the tolerance ϵ:

$$max \left\{ \left| \mathbf{y}^{(i+1)} - \mathbf{y}^{(i)} \right| \right\} < \epsilon \tag{2.23}$$

or even when the number of iterations is greater than a given limit $iter_{max}$.

Equation (2.16) is linear, thus is not directly applicable to power flow equations. However, consider the complex power injection into the buses as (2.24):

$$S_h = V_h I_h^* = V_h \sum_{k \ \epsilon \ \beta} Y_{hk}^* V_k^*, h \ \epsilon \ \beta \tag{2.24}$$

in which $\beta = \{1,2,...,n_b\}$, n_b is the number of network buses, and Y_{hk} is the element (h,k) of the admittance matrix Y_{BUS}.

The equation (2.24) can then be rewritten as (2.25):

$$\sum_{k \ \epsilon \ \beta} Y_{hk} V_k = S_h^* / V_h^* \tag{2.25}$$

Alternatively, in vectorial form

$$\mathbf{YV} = [\tilde{\mathbf{V}}^*]^{-1} \mathbf{S}^* \tag{2.26}$$

in which $\tilde{\mathbf{V}} = diag(V_1, V_2,...,V_{n_b})$. From the equation (2.16), defining $\mathbf{A} = \mathbf{Y}$, $\mathbf{y} = \mathbf{V}$ and $\mathbf{b} = [\tilde{\mathbf{V}}^*]^{-1}\mathbf{S}^*$, the Gauss-Seidel method can be directly implemented.

Thus, the voltage calculated in the i^{th} iteration using the Gauss-Seidel method in a bus h of type PQ is given by:

$$V_h^{(i+1)} = \frac{1}{Y_{hh}} \left(\frac{S_h^*}{V_h^{*,(i)}} - \sum_{k>h} Y_{hk} V_k^{(i)} - \sum_{k<h} Y_{hk} V_k^{(i+1)} \right), h \, \epsilon \, \beta, k \, \epsilon \, \beta \tag{2.27}$$

where $S_h^* = P_h^{sp} - jQ_h^{sp}$.

When dealing with a PV bus h, the calculations are performed differently. This is because the reactive power injection is unknown, and the voltage magnitude of the bus must be maintained at a constant value V_h^{sp}. Due to the reactive power limitations of the generator connected to the bus, the reactive power injection Q_h at a PV bus must be kept within the limits Q_h^{min} and Q_h^{max}. In this case, the value of Q_h^{sp} is replaced at each iteration by a value calculated from (2.30):

$$Q_h^{cal,(i)} = -Im \left\{ V_h^{*,(i)} \sum_{k \, \epsilon \, \beta} Y_{hk} V_k^{(i)} \right\}, h \, \epsilon \, \beta \tag{2.28}$$

where $Im\{\circ\}$ corresponds to the imaginary part of $\circ$. Based on the value of Q_h^{cal}, the voltage V_h is calculated using (2.27). Note that equation (2.27) provides both new magnitude and new phase of voltage $V_h^{(i+1)}$. Since the PV generator sets the voltage magnitude, only the voltage angle needs to be updated.

2.3 NEWTON-RAPHSON METHOD

In industrial applications, the conventional approach for determining the solution of Power Flow Equations is through the use of the Newton-Raphson method (TINNEY; HART, 1967). Typically, the method converges in a few iterations for well-conditioned cases. However, for systems characterized as ill-conditioned, NR, like other classical techniques, may fail to converge (MILANO, 2009; TOSTADO-VÉLIZ *et al.*, 2020a; TOSTADO-VÉLIZ *et al.*, 2021).

To demonstrate how the method is applied, consider the equation in a single variable, where $x \in \mathbb{R}$ and $g(x) \in \mathbb{R}$, expressed as:

$$g(x) = c \tag{2.29}$$

in which $c \in \mathbb{R}$ is a constant.

Let $x^{(0)}$ represent an initial guess for the solution of Eq. (2.29), and let $\Delta x^{(0)}$ denote a small disturbance around $x^{(0)}$. In this case, the function $g(x)$ can be approximated as:

$$g(x^{(0)} + \Delta x^{(0)}) = c \tag{2.30}$$

A Taylor series expansion of the left-hand side of Equation (2.30) around $x^{(0)}$ yields the following expression:

$$g(x^{(0)}) + \left(\frac{dg}{dx}\right)^{(0)} \Delta x^{(0)} + \frac{1}{2!}\left(\frac{d^2g}{dx^2}\right)^{(0)} (\Delta x^{(0)})^2 + \cdots \approx c \tag{2.31}$$

Assuming that the root $x^{(0)}$ is close enough to the correct solution such that $\Delta x^{(0)}$ is very small, the higher-order terms can be neglected. Thus, Equation (2.31) can be linearly approximated as shown in Equation (2.32):

$$\Delta c^{(0)} \simeq \left(\frac{dg}{dx}\right)^{(0)} \Delta x^{(0)} \therefore \Delta x^{(0)} \simeq \left[\left(\frac{dg}{dx}\right)^{(0)}\right]^{-1} \Delta c^{(0)} \tag{2.32}$$

where $\Delta c^{(0)} = c - f(x^{(0)})$ represents the mismatch associated with the solution $x^{(0)}$. Therefore, by incorporating the increment $\Delta x^{(0)}$ into the initial estimate, we obtain the updated approximation of the solution, as indicated in Eq. (2.33).

$$x^{(1)} = x^{(0)} + \Delta x^{(0)} \tag{2.33}$$

In this way, Equations (2.34), (2.35), and (2.36) can be used to construct a general structure for the algorithm aimed at implementing the Newton-Raphson method.

$$\Delta c^{(i)} = c - g(x)^{(i)} \tag{2.34}$$

$$\Delta x^{(i)} = \left[\left(\frac{dg}{dx}\right)^{(i)}\right]^{-1} \Delta c^{(i)} \tag{2.35}$$

$$x^{(i+1)} = x^{(i)} + \Delta x^{(i)} \tag{2.36}$$

Alternatively, Equation (2.35) can be described as

$$\Delta c^{(i)} = J^{(i)} \Delta x^{(i)} \tag{2.37}$$

where $J^{(i)} = \left(\frac{dg}{dx}\right)^{(i)}$ is the Jacobian matrix for $g(x)$ evaluated at $x = x^{(i)}$.

Extending the application to a multivariable problem with $x \in \mathbb{R}^n$ and $g(x) \in \mathbb{R}^n$, the linearization around an initial estimate $x^{(0)}$ is computed as shown in (2.38):

$$\begin{cases} g_1^{(0)} + \left(\frac{\partial g_1}{\partial x_1}\right)^{(0)} \Delta x_1^{(0)} + \left(\frac{\partial g_1}{\partial x_2}\right)^{(0)} \Delta x_2^{(0)} + \cdots + \left(\frac{\partial g_1}{\partial x_n}\right)^{(0)} \Delta x_n^{(0)} = c_1 \\ g_2^{(0)} + \left(\frac{\partial g_2}{\partial x_1}\right)^{(0)} \Delta x_1^{(0)} + \left(\frac{\partial g_2}{\partial x_2}\right)^{(0)} \Delta x_2^{(0)} + \cdots + \left(\frac{\partial g_2}{\partial x_n}\right)^{(0)} \Delta x_n^{(0)} = c_2 \\ \vdots \\ g_n^{(0)} + \left(\frac{\partial g_n}{\partial x_1}\right)^{(0)} \Delta x_1^{(0)} + \left(\frac{\partial g_n}{\partial x_2}\right)^{(0)} \Delta x_2^{(0)} + \cdots + \left(\frac{\partial g_n}{\partial x_n}\right)^{(0)} \Delta x_n^{(0)} = c_n \end{cases} \tag{2.38}$$

where c_1, c_2,...,c_n are constant elements.

Equation (2.38) can be reformulated in matrix terms, as shown in (2.39). In this way, it is possible to determine the increment $\Delta x_i^{(0)}$ for $i = 1, 2, ..., n$. Similarly, the increments for the i-th iteration can also be computed.

$$\begin{bmatrix} c_1 - g_1^{(0)} \\ c_2 - g_2^{(0)} \\ \vdots \\ c_n - g_n^{(0)} \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial g_1}{\partial x_1}\right)^{(0)} & \left(\frac{\partial g_1}{\partial x_2}\right)^{(0)} & \cdots & \left(\frac{\partial g_1}{\partial x_n}\right)^{(0)} \\ \left(\frac{\partial g_2}{\partial x_1}\right)^{(0)} & \left(\frac{\partial g_2}{\partial x_2}\right)^{(0)} & \cdots & \left(\frac{\partial g_2}{\partial x_n}\right)^{(0)} \\ \vdots & \vdots & \vdots & \vdots \\ \left(\frac{\partial g_n}{\partial x_1}\right)^{(0)} & \left(\frac{\partial g_n}{\partial x_2}\right)^{(0)} & \cdots & \left(\frac{\partial g_n}{\partial x_n}\right)^{(0)} \end{bmatrix} \begin{bmatrix} \Delta x_1^{(0)} \\ \Delta x_2^{(0)} \\ \vdots \\ \Delta x_n^{(0)} \end{bmatrix} \tag{2.39}$$

A generalized expression for (2.39) is given by:

$$\mathbf{\Delta f} = \mathbf{J} \mathbf{\Delta x} \tag{2.40}$$

where $\mathbf{\Delta x}$, $\mathbf{\Delta f}$ and $\mathbf{J}$ are calculated using (2.41), (2.42), and (2.43), respectively.

$$\mathbf{\Delta x} = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{bmatrix} \tag{2.41}$$

$$\mathbf{\Delta f} = \begin{bmatrix} c_1 - g_1(x_1, x_2, \cdots, x_n) \\ c_2 - g_2(x_1, x_2, \cdots, x_n) \\ \vdots \\ c_n - g_n(x_1, x_2, \cdots, x_n) \end{bmatrix} \tag{2.42}$$

$$\mathbf{J} = \begin{bmatrix} \left(\frac{\partial g_1}{\partial x_1}\right) & \left(\frac{\partial g_1}{\partial x_2}\right) & \cdots & \left(\frac{\partial g_1}{\partial x_n}\right) \\ \left(\frac{\partial g_2}{\partial x_1}\right) & \left(\frac{\partial g_2}{\partial x_2}\right) & \cdots & \left(\frac{\partial g_2}{\partial x_n}\right) \\ \vdots & \vdots & \vdots & \vdots \\ \left(\frac{\partial g_n}{\partial x_1}\right) & \left(\frac{\partial g_n}{\partial x_2}\right) & \cdots & \left(\frac{\partial g_n}{\partial x_n}\right) \end{bmatrix} \tag{2.43}$$

In this context, $\mathbf{J}$ is known as the Jacobian matrix. If the estimates were exact, both $\mathbf{\Delta f}$ and $\mathbf{\Delta x}$ would equal zero. However, since they are only approximations, the errors $\mathbf{\Delta f}$ are non-zero. Eq. (2.40) establishes a linear relationship between the errors $\mathbf{\Delta f}$ and the corrections $\mathbf{\Delta x}$ via the Jacobian matrix of the system of equations. Thus, for the ith iteration of the NR method, the solution of the set of equations is iteratively calculated by:

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \Delta\mathbf{x}^{(i)} \tag{2.44}$$

This iterative process continues until the errors Δf_i are reduced below a specified tolerance and, for each iteration, the Jacobian matrix must be recalculated.

2.3.1 Application of the NR method to power flow solution

In power flow analysis, the Newton-Raphson method stands out as one of the most effective and widely adopted techniques for solving the nonlinear equations that characterize the power flow problem. The power flow equations, as represented by (2.15), are inherently nonlinear due to the complex relationships between bus voltages, angles, and injected powers. Consequently, direct solutions are impractical, necessitating the use of iterative methods.

Among the various iterative techniques available, the NR method is preferred in most applications because of its quadratic convergence properties and robust performance on power systems.

This high convergence rate allows the NR method to achieve accurate solutions within a relatively low number of iterations, provided the initial guess is sufficiently close to the true solution (MILANO, 2009).

In each step of the process defined by Eq. (2.44), applied to PFP, $\Delta\mathbf{x}^{(i)}$ is calculated for a mismatch $\mathbf{g}(\mathbf{x}^{(i)})$ and Jacobian matrix $\mathbf{J}(\mathbf{x}^{(i)}) = \mathbf{g}_x(\mathbf{x}^{(i)}) = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}^{(i)}}$ as

$$\Delta\mathbf{x}^{(i)} = -[\mathbf{J}(\mathbf{x}^{(i)})]^{-1}\mathbf{g}(\mathbf{x}^{(i)}). \tag{2.45}$$

It is important to mention that to obtain the PFP solution it is necessary to perform the LU factorization of the Jacobian matrix, which is considered as the heaviest computational part of NR (TOSTADO-VÉLIZ *et al.*, 2018a). The computation of the Jacobian matrix elements follows this procedure:

$$\mathbf{J}(\mathbf{x}^{(i)}) = \left[\begin{array}{ccc|ccc} \frac{\partial P_1^{(i)}}{\partial \theta_1} & \cdots & \frac{\partial P_1^{(i)}}{\partial \theta_n} & \frac{\partial P_1^{(i)}}{\partial |V_1|} & \cdots & \frac{\partial P_1^{(i)}}{\partial |V_n|} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial P_n^{(i)}}{\partial \theta_1} & \cdots & \frac{\partial P_n^{(i)}}{\partial \theta_n} & \frac{\partial P_n^{(i)}}{\partial |V_1|} & \cdots & \frac{\partial P_n^{(i)}}{\partial |V_n|} \\ \hline \frac{\partial Q_1^{(i)}}{\partial \theta_1} & \cdots & \frac{\partial Q_1^{(i)}}{\partial \theta_n} & \frac{\partial Q_1^{(i)}}{\partial |V_1|} & \cdots & \frac{\partial Q_1^{(i)}}{\partial |V_n|} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial Q_n^{(i)}}{\partial \theta_1} & \cdots & \frac{\partial Q_n^{(i)}}{\partial \theta_n} & \frac{\partial Q_n^{(i)}}{\partial |V_1|} & \cdots & \frac{\partial Q_n^{(i)}}{\partial |V_n|} \end{array}\right] = \begin{bmatrix} \frac{\partial P}{\partial \theta} & \frac{\partial P}{\partial V} \\ \frac{\partial Q}{\partial \theta} & \frac{\partial Q}{\partial V} \end{bmatrix}^{(i)} \tag{2.46}$$

The elements of the Jacobian matrix correspond to the partial derivatives of the equations for PQ and PV buses evaluated at $\Delta\theta_k^{(i)}$ and $\Delta|V_k^{(i)}|$. Another notation for (2.45) is as follows:

$$\begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix} = - \begin{bmatrix} \frac{\partial P}{\partial \theta} & \frac{\partial P}{\partial V} \\ \frac{\partial Q}{\partial \theta} & \frac{\partial Q}{\partial V} \end{bmatrix}^{-1} \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \tag{2.47}$$

or, written in another way

$$\begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix} = - \begin{bmatrix} J_1 & J_3 \\ J_2 & J_4 \end{bmatrix}^{-1} \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \tag{2.48}$$

It is assumed that the value $\mathbf{x}^{(i+1)}$ converges numerically to the root $\mathbf{x}$ when $||\mathbf{g}(\mathbf{x})||_\infty < \epsilon$ for $\epsilon > 0$ is sufficiently small. Typically, the number of iterations in the search for convergence is limited for practical numerical reasons. The procedure is stopped if convergence does not occur before reaching a maximum number of iterations, say $iter_{max}$. Under these conditions,

it is said that the iterative process is divergent and stopped (TOSTADO-VéLIZ *et al.*, 2019). Algorithm 1 briefly describes the steps of the classical Newton-Raphson method for solving the iterative power flow problem for satisfying (2.15) through the iterate process (2.44), given a tolerance for convergence ϵ.

Algorithm 1 Newton-Raphson Method Algorithm

1: Initialize the system state variables $\mathbf{x}^{(0)}$ and set the iteration counter $k = 0$.
2: Set convergence criteria: tolerance ϵ, maximum iterations $k_{max} = iter_{max}$.
3: **while** $||\mathbf{g}(\mathbf{x}^k)||_\infty > \epsilon$ and $k < k_{max}$ **do**
4: Calculate power mismatches $\mathbf{g}(\mathbf{x}^k)$ and construct the Jacobian matrix $\mathbf{J}(\mathbf{x}^k)$
5: Solve the linear system (2.45)
6: Update the state variables (2.44)
7: **if** $||\mathbf{g}(\mathbf{x}^{k+1})||_\infty < \epsilon$ **then** ▷ Check for convergence
8: **break** ▷ The method converged
9: **else**
10: $k = k + 1$ ▷ Increment the iteration counter k and repeat from step 3.
11: **end if**
12: **end while**
13: **return solution**

The NR method has the property of quadratic convergence. Therefore, it is expected to converge when an initial estimate $\mathbf{x}^{(0)}$ is assigned in the solution attraction region. However, even within this region, the convergence is not necessarily assured (MILANO, 2009). One of the reasons is an estimate very far from the solution. In conditions where one does not know the initialization of the iterative procedure, it is usual to start with a flat-start estimate. In this situation, the voltage magnitudes on the load buses are initialized to 1.0 pu, while all phase angles are initialized to zero. However, divergence may occur even for operating voltages close to the nominal. In this case, the system is characterized as ill-conditioned (MILANO, 2009). On the other hand, according to (GOLUB; LOAN, 1996), a problem is *mathematically defined* as ill-conditioned if the calculated values are very sensitive to small changes in the parameters that characterize the problem.

Systems categorized as ill-conditioned can have very large increment in (2.45). As a consequence, very high values are obtained for $\mathbf{x}^{(i+1)}$ in (2.44). A strategy to reduce the impact of the 'explosion' of the step $\Delta\mathbf{x}^{(i)}$ in $\mathbf{x}^{(i+1)}$ is to introduce a single optimal multiplier (OM) h_i

(IWAMOTO; TAMURA, 1981; PAN *et al.*, 2020) in such way that

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + h_i \Delta \mathbf{x}^{(i)}. \tag{2.49}$$

Despite the insertion of an OM to control the step length h_i during the PFP solution, the problem can still diverge. Mainly when large-scale and ill-conditioned models are studied. Because the bottleneck of the problem is related to the ill-conditioning of the Jacobian matrix when it is calculated for a given state at the point $\mathbf{x}^{(i)}$.

2.3.2 Complex-Variable NR Method

Historically, most algorithms for power system applications have been developed in the real domain. However, real-valued models do not accurately represent the complex-valued voltage and current phasors, which can result in solution methods that are prone to high computation times and issues related to ill-conditioning, as discussed in the previous section. To address these limitations, several recent studies have proposed algorithms that solve PFP directly in the complex plane (NGUYEN, 1997; NGUYEN, 2006; PIRES *et al.*, 2019; PIRES *et al.*, 2022; CHAGAS; PIRES, 2023).

As an advantage of implementing the approach in the complex plane, one can highlight that directly handling complex quantities reduces the need for calculation transformations or approximations. A detailed analysis of the formulation confirms a significant reduction in memory access when using the NR formulation with complex variables compared to the traditional formulation based on real variables (NGUYEN, 2006). Additionally, this approach can be competitive in terms of both accuracy and computational efficiency when compared to state estimation using real variables (DŽAFIĆ; JABR, 2020).

Considering the advantages observed in implementing analyses in the complex domain, MATPOWER, one of the most widely used tools for power flow studies, already incorporates power flow equations formulated directly in terms of complex matrices (SEREETER; ZIMMERMAN, 2018). This approach not only simplifies calculations but also enhances the computational performance of simulations (see fundamental details as implemented by MATPOWER's team in Appendix A).

2.4 FAST DECOUPLED LOAD FLOW

The strong interdependence among active powers, bus voltage angles, and reactive powers, along with the magnitudes of voltages, represents an intrinsic characteristic of every practical system of electric power transmission in steady-state operation (FAN, 1989). Based on this strategy, the Fast Decoupled method (or fast decoupled NR method [FNR]) was proposed by Stott & Alsac (1974). The method is a variant of the Newton-Raphson method, however, it offers Jacobian matrix calculation simplifications, overcoming the heavier computational part of NR. In addition to improving computational efficiency, the Fast-Decoupled method provides valuable insights for the analysis of complex steady-state electrical systems, accurate or approximate routine calculations, and offline and online contingency for networks of any size, providing the solution in seconds (STOTT; ALSAC, 1974; GLOVER *et al.*, 2012).

Considering that the systems under analysis exhibit a high X/R ratio, variations in active power ΔP are less influenced by voltage magnitude ΔV changes and more susceptible to phase angle variations $\Delta\theta$. Similarly, reactive power is less sensitive to changes in phase angle and is predominantly associated with variations in voltage magnitude. In this context, it is appropriate to assume that the elements J_2 and J_3 of the Jacobian matrix can be set to zero (SAADAT *et al.*, 1999). Thus, the Jacobian matrix can be represented in a reduced form as Eq. (2.50):

$$\begin{bmatrix}\Delta P\\ \Delta Q\end{bmatrix} = \begin{bmatrix}J_1 & 0\\ 0 & J_4\end{bmatrix}\begin{bmatrix}\Delta\theta\\ \Delta V\end{bmatrix} \tag{2.50}$$

The separation of Equation (2.50) into two independent parts, as expressed by (2.51) and (2.52), is a procedure that not only simplifies the approach but also significantly reduces the time required to solve these decoupled equations compared to solving (2.48). Additionally, it is possible to further optimize the process through a simplification that eliminates the need to recalculate J_1 and J_4 at each iteration. This optimization leads to the development of decoupled power flow equations: (STOTT; ALSAC, 1974)

$$\Delta P = J_1\Delta\theta = \left[\frac{\partial P}{\partial\theta}\right]\Delta\theta \tag{2.51}$$

$$\Delta Q = J_4 \Delta V = \left[\frac{\partial Q}{\partial V}\right] \Delta V \tag{2.52}$$

Equations (2.51) and (2.52) can also be simplified and represented in a format called fast decoupled, which has the following reduced form:

$$\frac{\Delta P_i}{V_i} = -B_i{}' \Delta\theta, \;\; i = 1,2,\ldots,(N-1) \tag{2.53}$$

$$\frac{\Delta Q_i}{V_i} = -B_i{}'' \Delta V, \;\; i = 1,2,\ldots,N_{PQ} \tag{2.54}$$

where $B_i{}'$ denotes a row of the matrix B', while $B_i{}''$ represents a row of the matrix B''. Both of these matrices derive from the imaginary part of the admittance matrix Y_{BUS}, equivalent to the susceptance matrix B. The fundamental distinction between B' and B'' is the fact that B'' is directly equivalent to the imaginary part of Y_{BUS}. On the other hand, B' is simplified even further, assuming the resistance values in the interconnections as negligible (MONTICELLI; GARCIA, 1999).

Therefore, within the Fast Decoupled method, successive changes in voltage magnitude and phase angle can be computed according to equations (2.55) and (2.56):

$$\Delta\theta = -[B']^{-1} \frac{\Delta P}{V} \tag{2.55}$$

$$\Delta V = -[B'']^{-1} \frac{\Delta Q}{V} \tag{2.56}$$

where $\frac{\Delta P}{V}$ and $\frac{\Delta Q}{V}$ form vectors whose elements are, respectively, $\frac{\Delta P_i}{V_i}$ and $\frac{\Delta Q_i}{V_i}$, as calculated in (2.53) and (2.54). Note that (2.55) and (2.56) only require one LU factorization for both B' and B''.

2.5 THE DC POWER FLOW

The DC Power Flow method stands as a pivotal simplification in power system analysis, offering a streamlined approach to solving complex electrical network problems. By focusing solely on

Algorithm 2 Fast Decoupled Algorithm

Initialize the system state variables $\mathbf{x}^{(0)}$ and set the iteration counter $k = 0$.
Set convergence criteria: tolerance ϵ, maximum iterations $k_{max} = iter_{max}$.
Compute the admittance matrix Y_{BUS} using network parameters
Build matrices B' and B'' from Y_{BUS}
while $||\mathbf{g}(\mathbf{x}^k)||_\infty > \epsilon$ and $k < k_{max}$ **do**
    Solve equations (2.55) and (2.56)
    Update the state variables (2.44)
    if $||\mathbf{g}(\mathbf{x}^{k+1})||_\infty < \epsilon$ **then** ▷ Check for convergence
        break ▷ The method converged
    else
        $k = k + 1$
    end if
end while
return solution

active power and voltage phase angles while neglecting reactive power and voltage magnitudes, this method significantly reduces computational complexity. Originally developed as a means of addressing the challenges posed by nonlinear power flow equations, the DC Power Flow method has significant importance in applications involving power system analysis, planning, and optimization (GLOVER *et al.*, 2012).

The fundamental premise of the DC power flow model revolves around prioritizing the (P, θ) correlation while entirely disregarding the (Q, V) correlation. This approach is substantiated by the following hypotheses (MILANO, 2010):

- All voltage magnitudes are assumed constant and equal to 1.0 pu and reactive powers are neglected;
- The computation of the simplified admittance matrix B excludes considerations of line resistance and charging;
- Bus voltage phases are assumed to be small, allowing for simplifications such as $\sin\theta_{ij} \approx \theta_{ij}$ and $\cos\theta_{ij} \approx 1.0$.

As a result, the system equations are simplified to:

$$P = B\theta \tag{2.57}$$

where P represents the injected active powers, B denotes the simplified admittance matrix, θ represents the vector of voltage angles.

Note that equation (3.22) resembles a linear equation commonly found in the solution of DC resistive circuits, which explains the term "DC power flow" for this technique. However, unlike previous power flow algorithms, the DC power flow method only provides an approximate solution, with the level of approximation varying depending on the system characteristics.

In certain simulations, initiating the nonlinear PFP iterative resolution by using the result obtained from the DC power flow as an initial estimate is considered. This approach provides a starting point for the nonlinear iterative process, potentially improving its efficiency and accuracy.

2.6 FINAL CONSIDERATIONS

In this chapter, fundamental concepts related to the ill-conditioned power flow problem were presented, in addition to the basic formulation of PFP. Due to the nonlinear nature of the system of equations involved in the power flow problem, iterative methods are commonly used to solve it. The main classical techniques used to solve the PFP were discussed, including Gauss-Seidel, Newton-Raphson and its Fast Decoupled variants, in addition to the DC Power Flow method. Table 2.2 summarizes the main advantages and disadvantages of the discussed methods.

The understanding of these classical techniques is crucial, as it establishes a solid foundation for the analysis of advanced iterative methods. Although traditional methods such as Gauss-Seidel and Newton-Raphson are effective in many situations, they present significant limitations when applied to ill-conditioned systems, especially in cases where the power flow solution's region of attraction is narrow or far from the initial guess. Given the complexity and importance of power systems, it is essential to develop and implement robust and efficient methods to overcome these challenges.

The next chapter presents a literature review of some PFP methods since their origin, and ad-

dresses a review class of methods that discuss and solve the ill-conditioned power flow problem.

Table 2.2. Comparison of classical power flow methods.

Method	Main Advantages	Main Disadvantages
Gauss-Seidel	• Simple implementation • Low memory requirements	• Slow convergence • Sensitive to initial guess
Newton-Raphson	• Fast convergence • Robust and reliable • Quadratic convergence order	• Requires good initial guess • Fail in ill-conditioned systems
Fast Decoupled	• Faster than Newton-Raphson • Less computational burden • Easier implementation compared to Newton-Raphson	• Convergence issues in ill-conditioned systems • Less accurate for systems with strong coupling
DC Power Flow	• Very fast and simple • Good for initial analysis	• Reactive problem is neglected • Voltage magnitudes are not calculated • Less accurate for detailed analysis

CHAPTER 3

DIRECT AND ITERATIVE METHODS

This chapter provides the theoretical foundation for solving linear systems using both direct and iterative methods, critical for handling large-scale problems in power systems. Large-scale ill-conditioned power flow problems often involve linear systems with sparse matrices of significant size, where efficient and stable solutions are essential due to computational constraints. For these cases, iterative linear methods leveraging Krylov subspaces, such as GMRES (Generalized Minimal Residual) and BiCGStab (Bi-Conjugate Gradient Stabilized), have shown effective performance and are the primary focus of this study.

In power flow analysis, iterative linear methods are characterized by their memory efficiency and ability to handle the large dimensions and sparsity of matrices in high-dimensional systems. For ill-conditioned problems, direct solvers can suffer from numerical instability, which has led to the development and improvement of iterative approaches.

To improve the performance and convergence of these iterative methods, particularly in ill-conditioned and large-scale scenarios, this work examines several complementary techniques. These include preconditioners which transform the original system into a form that improves the efficiency of iterative methods. The use of complete and incomplete LU factorizations is discussed, with incomplete LU offering a more memory-efficient alternative suited to iterative methods while still preserving much of the matrix structure. Matrix reordering techniques are also essential, as they can reduce fill-in during factorization and improve the sparsity pattern of matrices, directly impacting the efficiency of solvers.

3.1 DIRECT METHODS FOR SOLVING LINEAR SYSTEMS

Direct methods are widely used to solve linear systems, especially in problems where it is necessary to obtain all variables simultaneously. These methods are capable of finding the

exact solution, but are subject to rounding errors. Furthermore, their application in large systems, especially with sparse matrices, presents some limitations that can compromise their efficiency and computational feasibility (SAAD, 2003; GOLUB; LOAN, 1996).

In large power systems, consisting of thousands of buses, the representation of the relationships between variables can result in high-dimensional and highly sparse matrices.

3.2 DESCRIPTION AND FORMULATION OF DIRECT METHODS

Consider the following linear system represented by Eq. (3.1):

$$Ax = b \tag{3.1}$$

in which $A \in \mathbb{R}^{n \times n}$ is a coefficient nonsingular matrix of the variables, $x \in \mathbb{R}^n$ represents the unknown vector, and $b \in \mathbb{R}^n$ is the independent vector.

The objective of direct methods is to solve x through factorizations that facilitate the calculation of the variables simultaneously. This section presents some of main direct methods used for solving linear systems.

3.2.1 Gauss-Jordan

The Gauss-Jordan method is named in honor of the scientists Carl Friedrich Gauss (1777-1855) and Wilhelm Jordan (1842-1899). This method extends the concept of Gaussian elimination and works by transforming the expanded matrix (composed of the coefficient matrix and the vector of independent terms) until it reaches the reduced row-echelon form. The main idea is to simplify the system until it becomes equivalent to a system where each variable appears separately in each row, making it easier to read the solutions.

Thus, the method begins by applying elementary row operations to transform the elements below the main diagonal into zeros, creating an upper triangular matrix. The process continues so that the elements above the main diagonal also become zeros. In this way, each row of the matrix will have only one element, 1, in the position corresponding to a variable, with zeros in the other elements of the column.

The ultimate goal is to obtain an identity matrix in the coefficient part. When this identity matrix is achieved, the solutions to the system of equations can be read directly from the vector of independent terms.

In this way, the original system $Ax = b$ is transformed into:

$$Dx = c \tag{3.2}$$

in which D is a diagonal matrix and c is a vector, both with dimensions A and b, respectively. Thus, the linear system (3.2) is then solved directly.

3.2.2 LU factorization

The LU decomposition in linear algebra consists of expressing a non-singular matrix as the product of two triangular matrices, one lower and one upper, both with the same dimension as the original matrix. The lower matrix is triangular below the main diagonal, while the upper matrix is triangular above the diagonal.

In some cases, it is necessary to pre-multiply the matrix to be decomposed by a permutation matrix. This decomposition is widely used in numerical analysis, particularly for solving systems of equations more efficiently or calculating inverse matrices.

In general, the LU decomposition process is performed using Gaussian elimination, where, during the transformation, matrix L stores the multiplier coefficients used in the elimination operations, and matrix U contains the resulting elements of matrix A after these transformations, such that

$$A = LU. \tag{3.3}$$

After performing the decomposition, solving the linear system $Ax = b$ can be done more efficiently. Instead of solving the equation $Ax = b$ directly, two simpler systems of equations are solved:

$$Ly = b \tag{3.4}$$

$$Ux = y \tag{3.5}$$

where L is the lower triangular matrix and U is the upper triangular matrix.

The value of y, obtained from the first system, is then substituted into the second system to find the solution x.

3.2.3 QR factorization

The QR decomposition, also known as QR factorization, is a mathematical technique used to express a matrix as the product of two specific types of matrices: an orthogonal matrix and a triangular matrix. For a real square matrix A, this decomposition is represented as:

$$A = QR, \tag{3.6}$$

where Q is an orthogonal matrix (i.e. $Q^TQ = I$) and R is an upper triangular matrix. If A is nonsingular, then this factorization is unique.

Numerous techniques can be employed to compute the QR decomposition, with the Gram-Schmidt process being one of the commonly used methods (GOLUB; LOAN, 1996). Therefore, consider the Gram-Schmidt procedure below, with the vectors to be considered in the process as columns of the matrix A. That is,

$$A = [\mathbf{a}_1|\mathbf{a}_2|\cdots|\mathbf{a}_n]. \tag{3.7}$$

Then,

$$\mathbf{u}_1 = \mathbf{a}_1, \quad \mathbf{e}_1 = \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|}, \tag{3.8}$$

$$\mathbf{u}_2 = \mathbf{a}_2 - (\mathbf{a}_2 \cdot \mathbf{e}_1)\mathbf{e}_1, \quad \mathbf{e}_2 = \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|}. \tag{3.9}$$

$$\mathbf{u}_{k+1} = \mathbf{a}_{k+1} - (\mathbf{a}_{k+1} \cdot \mathbf{e}_1)\mathbf{e}_1 - \cdots - (\mathbf{a}_{k+1} \cdot \mathbf{e}_k)\mathbf{e}_k, \quad \mathbf{e}_{k+1} = \frac{\mathbf{u}_{k+1}}{\|\mathbf{u}_{k+1}\|}. \tag{3.10}$$

Note that $\|\cdot\|$ is the Euclidean norm.

Thus, the resulting QR factorization is given by Eq. (3.11).

$$A = [\mathbf{a}_1|\mathbf{a}_2|\cdots|\mathbf{a}_n] = [\mathbf{e}_1|\mathbf{e}_2|\cdots|\mathbf{e}_n] \begin{bmatrix} \mathbf{a}_1 \cdot \mathbf{e}_1 & \mathbf{a}_2 \cdot \mathbf{e}_1 & \cdots & \mathbf{a}_n \cdot \mathbf{e}_1 \\ 0 & \mathbf{a}_2 \cdot \mathbf{e}_2 & \cdots & \mathbf{a}_n \cdot \mathbf{e}_2 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{a}_n \cdot \mathbf{e}_n \end{bmatrix} = QR. \tag{3.11}$$

QR factorization also provides a reliable method for solving underdetermined systems $Ax = b$, where A is m-by-n with $m < n$ (BUKHSH, 2018). Ignoring the fill-reducing ordering, consider the QR factorization $QR = A^T$. The system becomes $R^T Q^T x = b$. If the upper triangular system $R^T y = b$ is solved for y, the solution to $Ax = b$ is $x = Qy$.

3.2.4 Cholesky factorization

When the matrix of a linear system is symmetric, the calculations for LU decomposition can be significantly simplified by taking advantage of this symmetry. This simplification is utilized by the Cholesky method, also known as Cholesky factorization or decomposition. The method is named after André-Louis Cholesky(1875-1918), who showed that a symmetric, positive-definite matrix can be factored as the product of a lower triangular matrix and its transpose. This approach allows the matrix A to be represented in the form:

$$A = LL^T \tag{3.12}$$

where L is a lower triangular matrix with strictly positive elements on the main diagonal. For this, it is required that the matrix be symmetric and positive-definite. Thus, substituting (3.12) into $Ax = b$ yields

$$(LL^T)x = b \tag{3.13}$$

which can be decomposed into the following two triangular systems:

$$Ly = b \tag{3.14}$$

$$L^T x = y \tag{3.15}$$

In this way, first, $Ly = b$ is solved for y, and then $L^T x = y$ for x, using successive substitutions.

Therefore, the Cholesky decomposition is especially useful because it allows fast and stable resolution of linear systems, provided the matrix has the necessary properties. It has an advantage over other decompositions, such as LU, because it requires half the operations on a symmetric matrix, resulting in lower computational cost (SÜLI; MAYERS, 2003).

3.2.5 Comparison and Limitations of Direct Methods for Large Systems

Each direct method presents advantages and disadvantages that must be considered when choosing an approach for large-scale linear systems, especially in contexts where the matrices are sparse. The primary limitation of direct methods is the occurrence of fill-in in sparse matrices, a phenomenon where non-zero elements are introduced in previously zero positions during factorization, drastically increasing memory usage and processing time.

3.3 ITERATIVE METHODS

The iterative methods require an initial estimate, $x(0)$, from which successive approximations $x_i^{(k)}$ are generated at each step k, converging to an acceptable solution within a previously established tolerance after a finite number of iterations. However, the effectiveness of iterative methods can be compromised by the conditioning of the matrix A, which directly influences the convergence and accuracy of the solution. Therefore, techniques such as reordering and preconditioning are often applied to improve the efficiency and robustness of these methods, especially in ill-conditioned systems.

Iterative methods can be classified into two main categories: stationary and non-stationary. Stationary methods, such as Jacobi, Gauss-Seidel, and Successive Over-Relaxation (SOR), are characterized by performing similar calculations with vectors at each iteration. On the other hand, non-stationary methods aim to find a value for the variable by using search directions derived from the theory of Krylov subspaces (SAAD; SCHULTZ, 1986). In this approach, the coefficients are updated at each iteration, and the final result depends on a sequence of vectors generated by the product of a power of the matrix with the initial residue. These vectors, which form an orthogonal basis for a subspace, justify the term Krylov subspace.

Thus, a solution to Eq. (3.1) can be obtained with the specified numerical precision. Starting from an initial solution $x^{(0)}$, a sequence of approximations $x^{(k)}$ is generated through the iterations k, so that, after a finite number of iterations and reaching the defined tolerance, the solution converges to a value close to x:

$$x \approx A^{-1}b \tag{3.16}$$

Although Krylov subspace-based methods theoretically converge to the solution with the number of iterations determined by the dimension of the matrix, in practice, rounding errors can hinder this convergence, especially if the matrix is ill-conditioned. Therefore, a rational criterion for deciding when to stop a method's iterative calculation process should consider parameters for solution evaluation, such as those based on the residual of the solution and the monitoring of the convergence process (RECKTENWALD, 2012).

Based on equation Eq (3.1), the residue is defined as follows:

$$r = b - Ax \tag{3.17}$$

Therefore, if x is the solution of $Ax = b$, r will be zero. For an iterative method the residue in an iteration k is given by:

$$r^{(k)} = b - Ax^{(k)}, \quad k = 0,1,2,\cdots \tag{3.18}$$

If a method converges, the residual $r^{(k)}$ approaches zero as k tends to infinity.

3.3.1 Krylov subspace Methods

In 1931, Krylov introduced a technique for constructing the characteristic polynomials of matrices (KRYLOV, 1931). This approach is based on the construction of a regular (non-singular) matrix K and a Hessenberg matrix H such that their product $K^{-1}AK = H$ is verified. Due to this similarity relationship, both matrices share the same characteristic polynomial. In the initial stage of this technique, the columns of matrix K are generated by multiplying a vector r by A; specifically, the j -th column of K is given by $A^{j-1}r$, resulting in $K = [r\ Ar\ A^2r\ \ldots\ A^{k-1}r]$.

Thus, the Krylov subspace can be defined by a sequence of vectors whose basis consists of the following:

$$K_k(A,r^{(0)}) := span(r^{(0)}, Ar^{(0)}, A^2r^{(0)}, \cdots, A^{k-1}r^{(0)}) \tag{3.19}$$

$$r^{(0)} = b - Ax^{(0)} \tag{3.20}$$

where A corresponds to the coefficient matrix, $r^{(0)}$ the initial residual and $x^{(0)}$ the initial estimate.

The Krylov subspace method for solving Eq. (3.1) is based on an approximate solution $x^{(k)}$ that belongs to a subspace K_k. Under these conditions, a minimum error norm condition for the exact solution x^* of Eq. (3.1) is established:

$$\left\|x^* - x^{(k)}\right\|_2 = min\left\{\left\|x^* - x^{(k)}\right\|_2 : x \in x_0 + K_k\right\} \tag{3.21}$$

In the subsection below, several of the primary methods based on the Krylov subspace are presented and discussed, including GMRES, BiCG, BiCGStab, and CGS(VORST, 2003).

3.3.2 GMRES

The Generalized Minimum Residual (GMRES) method aims to compute an approximation of the solution $x^{(k)}$, that is, the value of x at the k-th iteration, such that it satisfies the minimum residual norm condition. To achieve this, an orthogonal basis of vectors is constructed from the Krylov subspace (3.19) using the orthogonalization process known as the Arnoldi method (SAAD, 2003). With this basis, $x^{(k)}$ is determined such that the resulting Euclidean norm of the residual is minimized.

The approximate solution $x^{(k)}$ at the k-th iteration can be represented as:

$$x^{(k)} = x^{(0)} + Vy^{(k)} \tag{3.22}$$

where $x^{(0)}$ corresponds to the initial estimate, V represents a matrix whose columns are the vectors $v^{(k)}$ generated by the Arnoldi method and $y^{(k)}$ corresponds to the solution of the minimum

residual norm problem, as expressed by:

$$H_k y^{(k)} = \left\| r^{(0)} \right\|_2 \hat{e}_1 \tag{3.23}$$

where H_k is the upper Hessenberg matrix of size $(k+1) \times k$, according to Eq. (3.24), $r^{(0)}$ is the initial residual, defined as $r^{(0)} = b - Ax^{(0)}$, and $\hat{e}_1$ is the canonical unit vector of dimension k, given by $\hat{e}_1 = [1\ 0\ 0\ \ldots\ 0]^T$.

$$H_k = \begin{bmatrix} h_{11} & h_{12} & \cdots & \cdots & h_{1k} \\ h_{21} & h_{22} & \cdots & \cdots & h_{2k} \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & h_{k,k-1} & h_{kk} \\ 0 & \cdots & \cdots & 0 & h_{k+1,k} \end{bmatrix} \tag{3.24}$$

After determining the matrix V and the vector $y^{(k)}$, the approximation $x^{(k)}$ is computed as given by Eq. (3.22), and the residual $r^{(k)}$ is calculated as described by Eq. (3.18). This iterative process is repeated for each iteration k until the convergence condition is satisfied. A description of the operation of the GMRES method is presented in Algorithm 3.

Algorithm 3 GMRES

1: Compute $r_0 = b - Ax_0, \beta := \|r_0\|_2$, and $v_1 := r_0/\beta$.
2: **for** $j = 1, 2, \cdots, m$ **do**
3: Compute $w_j := Av_j$
4: **for** $i = 1, \cdot, j$ **do**
5: $h_{ij} := (w_j, v_i)$
6: $w_j := w_j - h_{ij} v_i$
7: **end for**
8: $h_{j+1,j} = \|w_j\|_2$ .If $h_{j+1,j} = 0$ set $m := j$ and go to 11
9: $v_{j+1} = w_j / h_{j+1,j}$
10: **end for**
11: Define the $(m+1) \times m$ Hessenberg matrix $\bar{H}_m = \{h_{ij}\}_{1 \le i \le m+1, 1 \le j \le m.}$ and V_m composed of the vectors $v_i = 1,2, \cdots, m$
12: Compute y_m the minimizer of $\left\| \beta e_1 - \bar{H}_m y \right\|_2$ and $x_m = x_0 + V_m y_m$

The main drawback of the GMRES method is that the computational effort and storage requirements per iteration increase linearly with the number of iterations, leading to a cost that quickly becomes high. The most common approach to mitigate this problem is to restart

the iteration process. After setting a number m of iterations, the accumulated information is discarded, and the most recent results serve as initial data for the next m iterations. This procedure is repeated until convergence is achieved. However, determining a suitable value for m is challenging, because if m is too small, the GMRES(m) method may converge slowly or even fail. On the other hand, an excessively large value of m requires a large amount of memory space. Since there are no specific rules for choosing m, the decision about when to restart is a practical matter, making the restart timing a decisive factor for the effective application of GMRES(m) (SAAD, 2003).

3.3.3 The Conjugate Gradients Method

The Conjugate Gradient (CG) algorithm is one of the best known iterative techniques for solving sparse Symmetric Positive Definite linear systems (VORST, 2003). Through this technique, sequences of vectors with successive approximations of the solution and iteration residuals are generated. Updates are performed by aiming to find the minimum residual. In each iteration, two inner products are computed to satisfy orthogonality conditions.

The Conjugate Gradient algorithm can be derived from the Lanczos algorithm, as presented in Algorithm 4 (SAAD, 2003).

Algorithm 4 CG

1: Compute $r_0 := b - Ax_0$ and $p_0 := r_0$.
2: **for** $j = 0, 1, 2, \cdots,$ until convergence **do**
3: $\quad a_j := (r_j,r_j)/(Ap_j,p_j)$
4: $\quad x_{j+1} := x_j + a_j p_j$
5: $\quad r_{j+1} := r_j - a_j A p_j$. If $\|r_{j+1}\| < \epsilon$ (where ϵ is the tolerance), stop the iteration.
6: $\quad \beta_j := (r_{j+1},r_{j+1})/(r_j,r_j)$
7: $\quad p_{j+1} := r_{j+1} + \beta_j p_j$
8: **end for**

The Conjugate Gradient method has several advantages, making it an efficient choice for solving large linear systems. One of its main advantages is efficiency, as the method only requires the calculation of matrix-vector products, which makes it significantly faster than direct methods for large systems. Additionally, for symmetric and positive-definite matrices, the Conjugate

Gradient method typically converges quickly, often reaching a satisfactory solution in just a few iterations. However, the method also has some limitations. It is not applicable to nonsymmetric systems or indefinite matrices, which limits its use to matrices with these properties.

3.3.4 The Bi-Conjugate Gradient Method

The Bi-Conjugate Gradient (BiCG) method exhibits significant differences compared to GMRES, particularly because it does not aim to minimize the residual directly at each iteration. This method has a high computational cost, primarily due to requiring two matrix-vector products per iterative step, which can negatively affect overall performance (BARRETT *et al.*, 1994). Additionally, in each iteration, the BiCG performs separate matrix-vector multiplications for two associated matrices, further increasing the computational cost.

A distinctive feature of the BiCG method is that orthogonalizations are not performed in a conventional sequence but instead are mutually orthogonal, or bi-orthogonal. In the context of Algorithm 5, steps 6 and 7 correspond to the computation of residuals, while steps 9 and 10 deal with the update of search directions, which are essential components for the progression of the method.

Algorithm 5 BiCG

1: Compute $r_0 := b - Ax_0$. Choose r_0^* such that $(r_0,r_0^*) \neq 0$.
2: Set $p_0 := r_0, p_0^* := r_0^*$
3: **for** $j = 0, 1, 2, \cdots,$ until convergence **do**
4: $\quad a_j := (r_j,r_j^*)/(Ap_j,p_j^*)$
5: $\quad x_{j+1} := x_j + a_j p_j$
6: $\quad r_{j+1} := r_j - a_j A p_j.$
7: $\quad r_{j+1}^* := r_j^* - a_j A^T p_j^*$
8: $\quad \beta_j := (r_{j+1},r_{j+1}^*)/(r_j,r_j^*)$
9: $\quad p_{j+1} := r_{j+1} + \beta_j p_j$
10: $\quad p_{j+1}^* := r_{j+1}^* + \beta_j p_j^*$
11: **end for**

3.3.5 The Bi-conjugate Gradient Stabilized Method

The Bi-Conjugate Gradient Stabilized method (BiCGStab) was developed to solve non-symmetric linear systems, avoiding the irregular convergence patterns of the conjugate gradient method. BiCGStab can be interpreted as the outcome of the BiCG method combined with the repetitive application of the GMRES method (BARRETT *et al.*, 1994). It performs a localized minimization of a residual vector, resulting in a smoother convergence behavior. However, if the local GMRES step stagnates during its application, the Krylov space is not expanded, and the BiCGStab method fails to converge.

Algorithm 6, described below, provides an approach for solving linear systems using the BiCGStab method(SAAD, 2003).

Algorithm 6 BiCGStab

Compute $r_0 := b - Ax_0$.
Set $p_0 := r_0$
for $j = 0, 1, 2, \cdots,$ until convergence **do**
    $a_j := (r_j, r_0)/(Ap_j, r_0)$
    $s_j := r_j + a_j Ap_j$
    $w_j := (As_j, s_j)/(As_j, As_j)$
    $x_{j+1} := x_j + a_j p_j + w_j s_j$
    $r_{j+1} := s_j - w_j As_j$
    $\beta_j := \frac{(r_{j+1}, r_0)}{(r_j, r_0)} \times \frac{a_j}{w_j}$
    $p_{j+1} := r_{j+1} + \beta_j (p_j - w_j Ap_j)$
end for

Iterative methods applied directly to solve linear systems with asymmetric and indefinite matrices often face challenges related to convergence. To address these limitations, system preconditioning becomes an essential step. Furthermore, the reordering of equations and the arrangement of variable elements within the system are fundamental aspects that can positively influence the efficiency and success of the iterative methods' convergence process (BENZI, 2002; GALIANA *et al.*, 1994).

3.4 PRECONDITIONERS

The convergence rate of iterative methods for solving linear systems is significantly influenced by the spectral properties of the coefficient matrix. Poor spectral properties can lead to slow convergence, making the solution process computationally expensive. To address this, preconditioning techniques are employed. A *preconditioner* is a matrix designed to transform the original system into an equivalent one, maintaining the same solution while improving its spectral properties. For example, given a coefficient matrix A, a preconditioner M that approximates A can be used to form the system

$$M^{-1}Ax = M^{-1}b \tag{3.25}$$

This transformed system (3.25) has the same solution as the original $Ax = b$, but the spectral properties of its coefficient matrix $M^{-1}A$ may be more favorable (BARRETT *et al.*, 1994).

The fundamental concepts related to preconditioners are presented below, highlighting their importance in improving the efficiency and convergence of iterative methods applied to the solution of linear systems.

For a matrix A , its eigenvalues and eigenvectors are obtained through the following equation:

$$(A - \lambda I)x = 0 \tag{3.26}$$

where λ represents the eigenvalue of A, x the corresponding right eigenvector, and I is the identity matrix. The left eigenvector x^L of A associated with λ can be determined using the equation given in Eq. (3.27).

$$x^L(A - \lambda I) = 0. \tag{3.27}$$

By calculating all the eigenvalues of A and constructing a matrix M whose columns are formed by the corresponding right eigenvectors, it is possible to determine the matrix D using the expression (3.28).

$$D = M^{-1}AM \tag{3.28}$$

where D is a diagonal matrix whose elements are the eigenvalues of A. The matrix M^{-1}, required for the diagonalization of A, can be obtained from the left eigenvectors of A.

Thus, as shown in Eq. (3.28), any square matrix can be decomposed into a product of its eigenvalues and eigenvectors. In iterative methods, the convergence rate of the eigenvectors is directly related to the magnitude of their respective eigenvalues. Specifically, the convergence of the eigenvector occurs faster when the magnitude of the eigenvalue is larger. On the other hand, the convergence of an iterative method can be slow when there is a large difference between the magnitudes of the eigenvalues of the system, which results in a higher number of iterations and greater propagation of rounding errors.

To mitigate this issue, preconditioners are used to modify the matrix A in such a way that its eigenvalues have more *balanced* magnitudes, thereby accelerating the convergence of the iterative process. Preconditioning, therefore, aims to transform the system in a way that reduces the number of iterations required to reach the solution, without altering the final eigenvalues of the system.

There are two main types of preconditioning: the sparse approximation of the inverse matrix, which aims to approximate $M \approx A^{-1}$, and the decomposed matrix, which seeks a matrix M such that $M \approx A$. For preconditioners approximating the inverse of the matrix ($M \approx A^{-1}$), the system can be preconditioned in the following ways:

- **Left Preconditioning**: $MAx = Mb$, where the preconditioner M is multiplied on the left side of the original system $Ax = b$. This approach modifies the matrix A and the right-hand side b, preserving the solution x.

- **Right Preconditioning**: $AMy = b$, where $x = My$. Here, the preconditioner modifies the solution vector, leaving the right-hand side b unchanged.

- **Two-Sided Preconditioning**: In this case, the preconditioner is applied on both sides, resulting in the transformed system $M_2AM_1y = M_2b$, where $x = M_1y$.

For preconditioners based on a decomposition of the matrix ($M \approx A$), the preconditioned system can be expressed as:

- **Left Preconditioning**: $M^{-1}Ax = M^{-1}b$, where the preconditioner M^{-1} is multiplied on the left. This directly modifies both A and b to improve the conditioning of the system.
- **Right Preconditioning**: $AM^{-1}y = b$, with the solution obtained as $x = M^{-1}y$.
- **Two-Sided Preconditioning**: When the preconditioner is applied on both sides, the system becomes $M_2^{-1}AM_1^{-1}y = M_2^{-1}b$, with $x = M_1^{-1}y$.

Both approaches, sparse approximation of the inverse and decomposed matrix preconditioning, offer flexibility in improving the convergence properties of iterative solvers by altering the original structure of the system. The choice between left, right, or two-sided preconditioning depends on the specific problem characteristics and the trade-offs in computational cost and implementation complexity. In this sense, decomposed matrix preconditioning, particularly through incomplete LU (ILU) factorization, gained significant attention for its ability to approximate the matrix A efficiently while preserving sparsity. The following subsection delves into ILU factorization, exploring its principles and applications in preconditioning for large-scale linear systems.

3.4.1 Incomplete LU Factorization (ILU)

A broad class of preconditioners is based on incomplete factorizations of the coefficient matrix. A factorization is called incomplete if, during the factorization process, certain fill elements, which are nonzero elements in the factorization at positions where the original matrix had a zero, are ignored. Such a preconditioner is then given in factored form $M = LU$, where L is lower triangular and U is upper triangular. The efficacy of the preconditioner depends on how well M^{-1} approximates A^{-1} (BARRETT *et al.*, 1994).

In the context of incomplete factorizations, managing fill elements often introduces challenges. The ILU method may encounter issues with pivot elements, such as zero or negative values, which can compromise the stability of the factorization. To address these issues, it becomes necessary to apply strategies like replacing problematic pivots with arbitrary positive values or reordering the matrix to ensure numerical robustness (MANTEUFFEL, 1980; BARRETT *et al.*, 1994).

There are several variations of the incomplete LU factorization methods. The ILU(0) variant is a structure that provides a simpler and computationally inexpensive factorization. The method involves decomposing the matrix A as $A = LU - R$, where L and U are the lower and upper triangular matrices, respectively, and, when added, they share the same distribution of non-zero elements as the original matrix A. The matrix R represents the residual or error resulting from the approximation by the product LU (SAAD, 2003).

Unlike ILU(0), which retains only the nonzero elements of the original matrix A , ILU(m) allows controlled fill-ins during factorization. With ILU(m), all fill-ins with a level greater than m are discarded. Note that for $m = 0$, the no-fill ILU(0) preconditioner is recovered. In many cases, ILU(1) already provides a significant improvement over ILU(0), and the cost of constructing and using the preconditioner remains acceptable. It is rarely advantageous to consider higher values of m, except in very challenging problems, due to the rapidly increasing cost of computing and applying the preconditioner as m increases (BENZI, 2002).

However, the level-of-fill approach may not be robust enough for certain classes of problems. For matrices that are far from being diagonally dominant, ILU(m) may require storing many fill-ins that are small in absolute value and, as a result, contribute little to the quality of the preconditioner while significantly increasing the computational and storage costs. In many cases, an efficient preconditioner can be achieved using an incomplete factorization where new fill-ins are accepted or discarded based on their magnitude. This approach ensures that only fill-ins that significantly enhance the quality of the preconditioner are retained and utilized.

A drop tolerance is a positive value used as a criterion for discarding elements. An absolute dropping strategy may be employed, where new fill-ins are accepted only if their absolute value exceeds a predefined threshold. However, this criterion may perform poorly if the matrix is poorly scaled, in which case it is more appropriate to use a relative drop tolerance. For instance, when eliminating row i, a new fill-in is accepted only if its absolute value is greater than $\tau\|a_i\|_2$, where a_i denotes the i-th row of matrix A.

One drawback of this approach is the difficulty in selecting an appropriate drop tolerance value: typically, this is done by trial and error using a few sample matrices from a given application, until a satisfactory value of τ is found. In many cases, good results are achieved with values of τ in the range of 10^{-4} to 10^{-2}, but the optimal value is highly problem-dependent (BENZI,

2002).

As a result, the preconditioner obtained, referred to as $ILU(\tau, p)$, proves to be highly effective. Here, p limits the maximum number of non-zero elements allowed in each row of the L and U factors, while τ represents the drop tolerance used to discard elements considered too small. In this way, p controls memory usage, and τ helps reduce computational time.

3.5 REORDERING TECHNIQUES

Matrix reordering techniques are widely used in numerical methods to improve efficiency and stability in several applications, such as matrix factorizations and iterative methods. The main idea is to permute rows and/or columns of the original matrix to change its structure, aiming to meet specific criteria that facilitate the resolution of linear systems.

By optimizing the matrix structure, reordering methods can significantly improve the convergence of iterative algorithms, often reducing both the number of iterations and the computational time required (BENZI *et al.*, 1999). As a consequence, obtaining preconditioning matrices based on this reordered matrix also tends to improve the performance of algorithms for iterative resolution of linear systems.

The reordering of a matrix is performed through permutations of its rows, columns, or both. These permutations are represented as the product of matrix A with permutation matrices R_L and R_R, which are derived from rearranging the rows and columns, respectively, of the identity matrix. When R_L multiplies A on the left, it rearranges the rows of A. Similarly, the multiplication of A by R_R on the right results in the rearrangement of the columns. If both multiplications are applied, the rows and columns are simultaneously reordered (SAAD, 2003; BENZI, 2002). Thus, the original Eq. (3.1) can be replaced by the equivalent reordered system (3.29).

$$R_L A R_R y = R_L b \quad x = R_R y \tag{3.29}$$

Importantly, the structural modifications introduced through reordering do not alter the spectral properties of the coefficient matrix, such as its eigenvalues or condition number, ensuring

that the reordered system remains mathematically equivalent to the original. Despite this invariance, the improvements in computational efficiency and convergence are substantial.

The literature presents various matrix reordering methods, including the Reverse Cuthill-McKee (RCM) (CUTHILL; MCKEE, 1969) and the Approximate Minimum Degree (AMD) (AMESTOY *et al.*, 1996). Figure 3.1 shows an initial sparse matrix with $n = 60$ and its reordered versions using the RCM and AMD methods. In both cases, the number of nonzero elements remains unchanged compared to the original matrix. However, the reordering strategies differ in their objectives and outcomes.

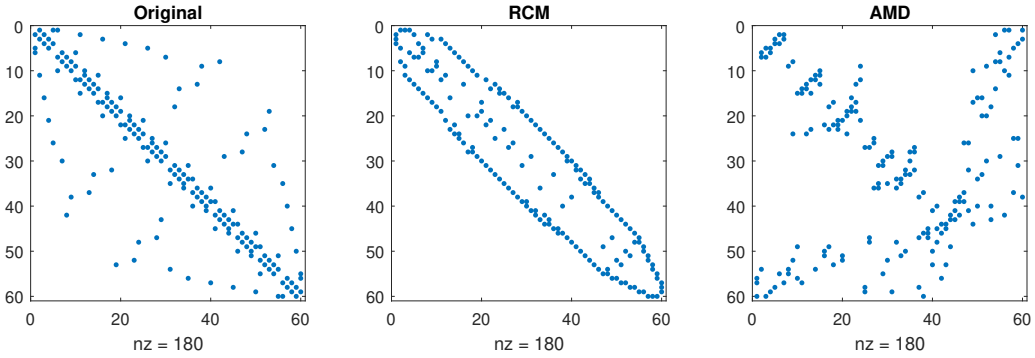


Figure 3.1. Representation of an original sparse matrix and its reordered versions using the Reverse Cuthill-McKee (RCM) and Approximate Minimum Degree (AMD) methods.

While the RCM method focuses on reducing the matrix bandwidth by concentrating the nonzero elements into a narrow region around the main diagonal, the AMD method prioritizes minimizing fill-in during matrix factorization, reducing the number of nonzero elements generated during the process. Thus, in some cases it can be seen that reordering the coefficient matrix before performing the incomplete factorization can have the effect of producing stable triangular factors, and hence more effective preconditions (BENZI *et al.*, 1999).

Several other ordering methods are also discussed in the literature, including Cuthill-McKee (CM) (CUTHILL; MCKEE, 1969), Nested Dissection (GEORGE, 1973), One-Way Dissection (GEORGE; LIU, 1981), and Spiral Ordering (DUFF *et al.*, 1976). For instance, in (DUFF; MEURANT, 1989), the impact of a wide range of orderings on the convergence of the preconditioned conjugate gradient method is analyzed. Among these, the spiral ordering demonstrated superior performance compared to the Minimum Degree ordering in the scenarios considered, highlighting its effectiveness in improving convergence.

3.6 FINAL CONSIDERATIONS

This chapter provided an overview of direct and iterative methods for addressing ill-conditioned problems. To enhance the performance and convergence of iterative methods, particularly in large-scale and ill-conditioned scenarios, various complementary techniques were examined. These included the use of preconditioners and LU factorizations, both complete and incomplete. Incomplete LU factorization offers a memory-efficient alternative, well-suited for iterative methods, while preserving much of the matrix structure. Additionally, matrix reordering techniques were evaluated, and the results regarding their impact on the efficiency and convergence of the methods are detailed in Section 7. In the next section, a comprehensive literature review on the power flow problem is presented, emphasizing the challenges related to ill-conditioning and the proposed methods to address them.

CHAPTER 4

STATE OF THE ART

This chapter provides a simplified bibliographic review concerning the power flow problem theme addressed in this thesis proposal. More specific details are found inside each reference, which will be highlighted in the present work.

The main focus is discussing ill-conditioned problems and how they were historically identified and treated along a timeline.

Before the digital computations introduction, the power flow studies were assessed using network analyzers (BROWN; CLOUES, 1956).

It provided quick and economical results of good accuracy. However, in some studies, the network analyzer did not provide sufficient precision and in such cases, the digital computer solution gained a considerable advantage. Among the various limitations of this type of analysis tool was the detailed representation of the network. Therefore, developing equivalents that were not always highly accurate for certain network portions was necessary. Interesting to note that modification of the analyzer setup had been necessary whenever it was desired to change from a system representation suitable for a load-flow study to a system representation suitable for a stability study.

Around 1956, increasing attention was given to the capabilities of automatic digital computers in application to power system problems (WARD; HALE, 1956). Digital solutions to this type of problem could provide a valuable tool to supplement the AC network analyzer. So, in (WARD; HALE, 1956), an iterative technique known as the "node method" was proposed. The publication (MCGILLIS, 1957) illustrates the type of difficulty found in that date for a nodal iterative technique implemented for the computation of power flow problem solution using an IBM 604 Digital Computer.

Despite the limitations posed by the hardware, significant advancements were made in the research of computational methods applied to the PFP. This progress was highlighted by the

investigations utilizing the iterative Newton-Raphson method, which gained attention towards the end of 1959 (NESS, 1959). After this, notable improvements were observed in the 1960s (TINNEY; HART, 1967). Notably, methods aimed at accelerating computations were introduced, exploiting simplifications in the Jacobian matrix (STOTT; ALSAC, 1974), and were widely accepted by the power community.

Many solvers for digital simulations, including developments based on the NR method, were developed until the mid-1970s (STOTT, 1974). The review paper (STOTT, 1974) presented several methods that have received widespread practical application, recent attractive developments for the date, and other interesting or useful techniques.

In the 1960s, despite hardware limitations, the Newton-Raphson method effectively addressed power flow issues in most power systems (TINNEY; HART, 1967). The use of sparsity techniques (OGBUOBIRI *et al.*, 1970; OGBUOBIRI, 1970) boosted the use of the NR method, making it feasible to apply to larger networks than those usual at the time.

However, some systems are difficult to solve using NR. These systems are typically called ill-conditioned systems, and their frequency has increased with network expansion since the 1970s. As a result, while the solution to the power flow has been regarded as a classic and well-studied topic, finding a solution to this problem in ill-conditioned systems has sparked great interest among researchers.

Cases of difficult convergence were already investigated in the early 1970s (TREVINO, 1970; SASSON *et al.*, 1971). For example, (SASSON *et al.*, 1971) demonstrated that a simple modification to Newton's method, using nonlinear programming, made the iterative process more powerful than the classical Newton-Raphson method. Moreover, the slight change could be easily incorporated into the existing Newton's solver. The modifications included a more efficient convergence process and a non-divergent characteristic for the problem.

From what we know, the ill-conditioned problem investigation for the PFP was introduced in (IWAMOTO; TAMURA, 1981) in 1981. Notably, we follow Milano's definition of an ill-conditioned system in (MILANO, 2009). According to this definition, *a power system is considered ill-conditioned if, although a solution exists, it cannot be achieved using the Newton-Raphson method with flat start estimation.* The strategy used in (IWAMOTO; TAMURA, 1981) as well as in (SASSON *et al.*, 1971) is based on the *optimal multiplier* (OM) approach.

This type of technique is known for its robustness and low sensitivity to the initial estimate, providing a solution to the power flow problem even in situations where traditional methods fail. However, these methods still do not compete with traditional solvers in some cases, as they tend to converge slowly or even stagnate numerically (TOSTADO *et al.*, 2019a; TOSTADO-VÉLIZ *et al.*, 2020a). In recent literature, optimal multiplier-based power flow solution methods have been discussed in (PAN *et al.*, 2019). These methods are also utilized to identify low-voltage solutions of power flow equations (OVERBYE; KLUMP, 1996) and to tackle diverse challenges in power flow computation, including automatic controls, heavy loads, and contingency issues (KLUMP; OVERBYE, 2000).

In the 1990's, the well-known method of Continuation Power Flow was proposed (AJJARAPU; CHRISTY, 1992) to overcome the difficulties experienced by methods near the Maximum Load Point (MLP). This type of technique can calculate power flow solutions for different loading conditions and estimate a network's maximum loading point. However, this technique aims to calculate multiple power flow solutions. While this is interesting for other related tools, such as voltage stability analysis, it has a disadvantage compared to conventional techniques, which aim to calculate only a snapshot of the system for a specific load profile. Thus, the method can be computationally intensive, requiring significant computational resources and time, especially for large and complex power systems.

Alternatively, regularization techniques or Levenberg-type techniques (DEUFLHARD, 1974) can also be employed to *circumvent* the singularity of the Jacobian matrix (POURBAGHER; DERAKHSHANDEH, 2016; POURBAGHER; DERAKHSHANDEH, 2018). By enhancing the condition number of the Jacobian matrix through the addition of diagonal elements, these methods offer a viable solution. However, although regularization techniques tend to be more efficient than continuation power flow methods, they incur a higher computational expense compared to the Newton-Raphson method (MILANO, 2015). This is largely due to the requirement for extra matrices product computations and a dense LU factorizations. Furthermore, these methods often demonstrate slower convergence rates, especially when the initial guess is very far from the solution. Additionally, the implementation of regularization might involve tuning parameters, which can add to the complexity of the method.

In (TOSTADO *et al.*, 2019b), the well-known Gauss-Newton minimization method is also stu-

died for PF analysis. The proposed approach was evaluated in a wide variety of ill-conditioned systems, ranging from 1888- to 70000-bus systems, under different scenarios. However, one disadvantage of the method is associated with the high computational cost of factoring the product of the Jacobian matrix and its adjoint matrix at each iteration.

Other strategies trying to circumvent the problem of ill-conditioning in the PFP were proposed considering a non-iterative numerical strategy. The investigation was called the holomorphic embedding load-flow model (HELM), a technique based on the complex analysis technique. The algorithm was first applied to the power flow problem by Trias (TRIAS, 2012). Several works have been proposed following this framework. The method starts from a flat start operational condition and is guided to a final state based on setting a parameter for unity. In (RAO *et al.*, 2015), various techniques based on the holomorphic embedding method were presented. These non-iterative techniques result in infinite formulations, each with different numerical properties. However, the authors identified issues with numerical precision, particularly in scenarios with high system loading levels. In (FREITAS *et al.*, 2019), a restarted HELM (RHELM) was introduced. The method works with a much reduced number of bus voltage Taylor's series coefficients, different from HELM. Despite being non-iterative, the framework based on the holomorphic embedding method is sensitive to Taylor's series coefficients, requiring an elevated precision for such coefficients. In some cases, the highlighted limitation can lead to a stagnation of the result of the PFP.

In the context of solving the PFP involving ill-conditioned systems, it is also worth highlighting the techniques based on the Continuous Newton's Method (CN) (NEUBERGER; NEUBERGER, 2010) in which, strictly speaking, any numerical integration method can be adapted to solve the PFP. The NC methodology was initially proposed by Hetzler in 1997 (HETZLER, 1997) and later adapted for the PFP by Milano (MILANO, 2009), through the reformulation of a set of Ordinary Differential Equations (ODEs).

In (MILANO, 2009), two methodologies were proposed for solving the PFP using explicit numerical integration methods: the Simple Robust Method (SRM), based on the forward Euler's method, and the fourth-order Runge-Kutta method (RK4). The results obtained with these techniques were compared to those from the NR and FNR methods and the method proposed by Iwamoto (IWAMOTO; TAMURA, 1981). Milano's study demonstrated that both metho-

dologies were more robust than standard load flow solvers when initialized with a flat start and significantly more efficient than Iwamoto's method for a 1254-bus system. However, a drawback noted in the study was the high computational time and number of iterations required. Additionally, the number of iterations increased logarithmically with the convergence limit.

Following the research path explored by Milano (MILANO, 2009), Tostado-Véliz *et al.* (TOSTADO-VÉLIZ *et al.*, 2018b) developed a methodology combining RK4 with the Broyden Method (BM). They introduced the fourth-order Runge-Kutta Broyden method (RK4B) with the aim of reducing the number of Jacobian matrix inversions from four (as in the original method) to just one. The results indicated that these approaches exhibited superior computational performance compared to traditional multi-stage numerical methods, all while preserving the robustness characteristics of the fourth-order Runge-Kutta method.

In (TOSTADO-VÉLIZ *et al.*, 2020a), an approach for well-conditioned and ill-conditioned power systems was also proposed. This approach combines the King-Werner and Heun's methods, resulting in the Heun-King-Werner (HKW) method. The developed approach involves LU factorization of two Jacobian matrices per iteration and naturally operates as a robust method in ill-conditioned systems and as a high-order Newton-like method in well-conditioned cases. Notably, the HKW method demonstrated superior efficiency and convergence performance compared to conventional PFP solution methods. Other methods based on the HKW were also proposed (OLIVEIRA; FREITAS, 2021). However, it's worth highlighting that the parameters of the HKW methods are subject to change when applied to ill-conditioned and large-scale power system models.

In (TOSTADO-VELIZ *et al.*, 2020), a three-stage algorithm called 3S-SIA was introduced for addressing power-flow problems in large-scale ill-conditioned systems. It incorporates Lavrentiev's regularization, a Chebyshev-like method, and Heun's method to obtain a balance between robustness and efficiency. The validation of this algorithm involves simulations on realistic systems across various demand conditions, alongside comparisons with standard and robust power-flow solution methods. Results indicate that the proposed algorithm offers a robust methodology that effectively resolves all examined scenarios. However, 3S-SIA involves managing two degrees of freedom that require updates in each iteration through adaptive mechanisms, increasing the computational complexity involved in obtaining the solution compared

to other methods. Other approaches to solving PFP in ill-conditioned systems were also investigated in (TOSTADO-VéLIZ *et al.*, 2019; TOSTADO-VÉLIZ *et al.*, 2020b; TOSTADO-VÉLIZ *et al.*, 2021).

Recent research has employed the homotopy technique to solve the ill-conditioned power flow problem. The homotopy technique is widely applicable in various fields of knowledge and is especially used for solving nonlinear ordinary and partial differential equations (LIAO, 2012). In (FREITAS; LIMA-SILVA, 2023), the methodology structures the PFP equations to include a fictitious network with shunt admittances, dimensioned so the initial solution coincides with the initial estimate. These admittances are gradually removed as the homotopy process advances, controlled by a parameter ranging from zero to one. A second homotopy parameter and a scaling factor are used to reinforce interconnections near the slack bus, but only during the intermediate stages. Once the first parameter reaches one, the shunt admittance network and the scaling factor are fully removed, returning to the original network model used to determine the final PFP solution. This strategy allows for the use of the traditional NR method at each step of the first homotopy parameter. However, due to the multiple steps required to transition from the initial solution to the final one, the homotopy-based method presents a computational load that can reach approximately eight times higher than the cost of the standard Newton-Raphson solver (FREITAS; SILVA, 2022), despite being able to obtain solutions in cases where standard methods fail to converge. Additionally, in (LIMA-SILVA *et al.*, 2023), a homotopy-based approach was presented to solve the power flow for isolated microgrids, considering droop-controlled DG units and different load characteristics and loading factors.

4.1 FINAL CONSIDERATIONS

As discussed in this chapter, most of the currently available robust power flow solvers are inefficient in terms of computational performance when applied to large-scale, ill-conditioned systems. This limitation restricts the widespread use of these solvers in industrial applications.

The next chapters are dedicated to introducing efficient and robust power flow solution approaches developed in this work. These approaches aim to be comparable to the Newton-Raphson method in terms of computational efficiency while surpassing this traditional method in solving

ill-conditioned cases.

To achieve these objectives, the next chapter will present various robust numerical methodologies, such as methods based on the Continuous Newton's philosophy and the Levenberg-Marquardt method. These methodologies form the basis for the development of the proposed new approaches.

CHAPTER 5

NON-CONVENTIONAL METHODS

Research on robust methods to solve the Power Flow Problem has recently intensified. This increasing pursuit is driven by various factors, such as the need to compute the operating point in networks with diverse configurations, integrate new energy generation sources, and manage heavily loaded and significantly sized networks. Particularly in cases involving large-scale networks, obtaining an initial estimate for solving the power flow problem can become a substantial challenge.

In certain situations, the classical Newton-Raphson (NR) method only presents convergence when the solution is within the convergence region and the initial estimate is very close to the solution (MILANO, 2009; TOSTADO-VÉLIZ *et al.*, 2021; MILANO, 2019). However, in complex scenarios, especially in large systems, this estimate close to the solution is often only known after substantial modifications to the network for already known states and small changes in the system structure. Consequently, considerable modifications to the network configuration can represent a significant obstacle to establishing an effective initial estimate for resolving the PFP using the NR method.

When dealing with the Power Flow Problem, a common practice is to initiate with a flat start estimate. However, in large-scale systems, this strategy may fail when employing the traditional NR method to calculate solutions close to the nominal values of the buses. This chapter, therefore, primarily focuses on exploring alternative methods to the traditional NR, considering the mentioned situations.

5.1 CONTINUOUS NEWTON'S METHOD

The Newton-Continuous (NC) approach is based on the premise that it is possible to adapt any numerical integration method to solve the load flow problem, as discussed by (HETZLER,

1997). The techniques derived from the Newton-Continuous philosophy are recognized for their efficiency, robustness and high reliability, as highlighted by (TOSTADO-VÉLIZ *et al.*, 2020a).

Assume that (5.1) represents the standard form for the notation of a set of ordinary differential equations of an autonomous system, which states $x(t)$ are also a function of an implicit variable t.

$$\dot{x}(t) = f(x(t)) \tag{5.1}$$

Solving (5.1) in the discrete-time domain can be performed using a numerical integration method. In the literature, the most elementary known method is the explicit Euler's method (FAIRES; BURDEN, 2012), where each k-th step is performed as follows:

$$\begin{aligned} \Delta\mathbf{x}^{(k)} &= \Delta t f(\mathbf{x}^{(k)}) \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \Delta\mathbf{x}^{(k)} \end{aligned} \tag{5.2}$$

where Δt denotes the integration interval, considering that the variable $\mathbf{x}$ is a function of the parameter t; $\mathbf{x}^{(k)}$ represents the vector containing the values of $\mathbf{x}$ in step k; and $\Delta\mathbf{x}^{(k)}$ is the increment calculated in step k to obtain the result in step $k+1$ of the variable $\mathbf{x}$.

A direct analogy between (2.45) and (5.2) can be made, assuming the Jacobian matrix $\mathbf{J} = \frac{dg}{dx}$ non-singular:

$$f(\mathbf{x}^{(k)}) = -\mathbf{J}(\mathbf{x}^{(k)})^{-1}\mathbf{g}(\mathbf{x}^{(k)}) \tag{5.3}$$

Based on this analogy, we can interpret the traditional Newton-Raphson (NR) method for solving non-linear equations as applying the explicit Euler's method, where $\Delta t = 1$ is the integration step for solving an ordinary differential equation (ODE).

Therefore, other numerical integration methods, as will be presented in the sequence, can also be applied to solve the power flow problem, considering the appropriate adaptation of the problem as treatment by ODEs.

5.1.1 Runge-Kutta Method

In this subsection, the classical Runge-Kutta (RK) method is discussed as a tool for solving the PFP as a NC solver.

The suggested RK method for the NC is the same technique developed by the German mathematician Carl Runge in 1895 and extended by Martin Wilhelm Kutta in 1901. It is a powerful numerical technique widely employed for solving ordinary differential equations (FAIRES; BURDEN, 2012). Its popularity stems from the ability to approximate solutions with high precision and stability while maintaining computational efficiency (OKEKE *et al.*, 2019; RAHMAN *et al.*, 2022).

The most comprehensive formulation of a Runge-Kutta technique to numerically approach a problem involving a differential equation, such as that expressed by (5.1), is characterized by the following expression:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + h \sum_{i=1}^{p} b_i v_i^{(k)} \tag{5.4}$$

where, $h \in \mathbb{R}^+$ is the integration step size, $b_i \in \mathbb{R}$ are the weights, $p \in \mathbb{N}$ is the number of stages defined for the method and $v_i \in \mathbb{R}^n$ are calculated as follows:

$$\begin{cases} \mathbf{v}_1^{(k)} = f(\mathbf{x}^{(k)}) \\ \mathbf{v}_2^{(k)} = f(\mathbf{x}^{(k)} + h a_{21} \mathbf{v}_1^{(k)}) \\ \mathbf{v}_3^{(k)} = f(\mathbf{x}^{(k)} + h(a_{31}\mathbf{v}_1^{(k)} + a_{32}\mathbf{v}_2^{(k)})) \\ \vdots \\ \mathbf{v}_p^{(k)} = f(\mathbf{x}^{(k)} + h(a_{p,1}\mathbf{v}_1^{(k)} + a_{p,2}\mathbf{v}_2^{(k)} + \cdots + a_{p,p-1}\mathbf{v}_{p-1}^{(k)})) \end{cases} \tag{5.5}$$

in which, $a_{p,i} \in \mathbb{R}$, $i = 1,2,\ldots,p-1$ are the elements of the Runge-Kutta matrix.

The main computational effort in applying Runge-Kutta method is the evaluation of f. The Runge-Kutta technique requires p Jacobian matrix calculations and respective LU factorization at each iteration. Therefore, the more p stages, the greater the number of Jacobian matrix calculations per iteration.

The order of convergence of RK method is limited by its number of stages. Table 5.1 defines the highest order of convergence that can be obtained using RK methods with different numbers of stages (FAIRES; BURDEN, 2012).

Table 5.1. Maximum convergence order for an RK method with p stages.

Number of stages (p)	1	2	3	4	$5 \leq p \leq 7$	$8 \leq p \leq 9$	$10 \leq p$
Best possible order	1	2	3	4	$p-1$	$p-2$	$p-3$

5.1.1.1 Characterization of the Parameters in an Explicit Runge-Kutta Method

The parameters $a_{i,j}$, b_i, and c_i of an RK method of p stages can be organized as shown in Table 5.2. This table type is a Butcher table (BUTCHER, 1964).

Table 5.2. Butcher table for an explicit RK method.

0				
c_2	a_{21}			
c_3	a_{31}	a_{32}		
$\vdots$	$\vdots$	$\vdots$	$\ddots$	
c_p	a_{p1}	a_{p2}	$\cdots$	a_{pp}
	b_1	b_2	$\cdots$	b_p

The coefficients for RK methods of orders 1, 2, 3, and 4 are given, respectively, by Tables 5.3, 5.4, 5.5 and 5.6.

Table 5.3. Runge-Kutta method of order 1.

0	
	1

Table 5.4. Runge-Kutta method of order 2.

0		
1	1	
	$\frac{1}{2}$	$\frac{1}{2}$

Table 5.5. Runge-Kutta method of order 3.

0			
1	1		
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	
	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{2}{3}$

Table 5.6. Runge-Kutta method of order 4.

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

5.1.2 Heun-King-Werner Methods

The Heun-King-Werner (HKW) method, presented by Tostado-Véliz *et al.* em (TOSTADO-VÉLIZ *et al.*, 2020a), is based on the Newton-Continuous approach. It stands out for its effectiveness in dealing with ill-conditioned power systems, while also exhibiting comparable performance to higher-order Newton's methods in well-conditioned scenarios.

The adopted HKW strategy combines the Heun's method, also known as the trapezoidal rule, employed as the main loop and corrector, and the King-Werner method, mentioned in (ARGYROS; MAGREÑÁN, 2017). The King-Werner solver family is recognized for achieving a convergence order of $1+\sqrt{2}$, thus surpassing the NR method. Furthermore, the HKW method incorporates a prediction step in which a root $\mathbf{y}^{(k)}$ is calculated based on an estimate $\mathbf{x}^{(k)}$. In this step, the Euler's method with a step h is used, and the predicted value is determined by $\mathbf{y}^{(k)} = \mathbf{x}^{(k)} + hf(\mathbf{x}^{(k)})$, where $f(\mathbf{x}^{(k)}) = -\mathbf{J}(\mathbf{x}^{(k)})^{-1}\mathbf{g}(\mathbf{x}^{(k)})$. The contribution of the King-Werner method is incorporated by calculating an intermediate point between $\mathbf{x}^{(k)}$ and the prediction $\mathbf{y}^{(k)}$, resulting in $\widetilde{\mathbf{x}}^{(k)} = \frac{1}{2}(\mathbf{x}^{(k)} + \mathbf{y}^{(k)})$. This value is then used to calculate the update increment $f(\widetilde{\mathbf{x}}^{(k)}) = -\mathbf{J}(\widetilde{\mathbf{x}}^{(k)})^{-1}\mathbf{g}(\widetilde{\mathbf{x}}^{(k)})$.

It is well-known that the form assumed by the second-order Heun's method is as follows:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \frac{h}{2}[f(\widetilde{\mathbf{x}}^{(k)}) + f(\mathbf{x}^{(k)})]. \tag{5.6}$$

In (TOSTADO-VÉLIZ *et al.*, 2020a), the expression (5.7) is modified, being changed to

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \frac{h}{2}[(2-\psi^{(k)})\Delta\widetilde{\mathbf{x}}^{(k)} + \psi^{(k)}\Delta\mathbf{x}^{(k)}] \tag{5.7}$$

where $\psi^{(k)} = 2|SSR^{(k)} - SSR^{(0)}|/SSR^{(0)}$ represents a weighting factor calculated at each iteration, with initialization recommended as $\psi^{(0)} = 1$; and $SSR^{(k)} = \frac{1}{2}\left[\mathbf{g}(\mathbf{x}^{(k)})\right]^T \mathbf{g}(\mathbf{x}^{(k)})$.

In (5.7), it is observed that as $x^{(k)}$ approaches the solution of the equation, $\psi^{(k)}$ tends to 2. Under these circumstances, the iterations resemble the Euler's or NR methods, especially when $h = 1$. Consequently, the authors propose restricting ψ to a maximum value of 2. Following this restriction, whenever $\psi^{(k)} > \psi_{max}$, it is recommended to use the NR method directly. Furthermore, the step h can be updated at each iteration.

The application of the Heun-King-Werner method to solve the PFP is outlined in the following steps. The Jacobian matrix is assumed to be nonsingular at every evaluation of the state $\mathbf{x}^{(k)}$.

Step 0: This step involves reading input data and determining the system admittance matrix. Subsequently, the iteration counter, denoted as k, is set to zero. Parameters are initialized, such as tolerance ε, maximum number of iterations, k_{max}, step h, limiting α, and a starting point $\mathbf{x}(0)$ is selected to initiate the iterative process.

Step 1: This step involves determining and evaluating the Jacobian matrix at the current iteration point, denoted as $\mathbf{x}^{(k)}$. Subsequently, (5.8) is calculated. This step is similar to the process performed using the NR method.

$$\Delta\mathbf{x}^{(k)} = -\mathbf{J}(\mathbf{x}^{(k)})^{-1}\mathbf{g}(\mathbf{x}^{(k)}). \tag{5.8}$$

Step 2: Euler's iteration update - as part of the Heun's method, the state vector is initially updated using the Euler's method according to (5.9):

$$\mathbf{y}^{(k)} = \mathbf{x}^{(k)} + h^{(k)}\Delta\mathbf{x}^{(k)} \tag{5.9}$$

In this step, the increment vector in Newton's method is adjusted to incorporate the effect of the integration step h. The initialization of the step h follows the recommendations established according to (TOSTADO-VÉLIZ *et al.*, 2020a):

$$h^{(0)} = max\left\{h_{min}, min\left\{h_{max}, \frac{1}{(SSR^{(0)})^{\mu}}\right\}\right\} \tag{5.10}$$

where SSR is defined according to Equation (5.11). A value of $\mu = 0.06$ is adopted, based on experiments reported in (TOSTADO-VÉLIZ *et al.*, 2020a), considered suitable for computing the solution of the PFP.

$$SSR^{(k)} = \frac{1}{2}\left[\mathbf{g}(\mathbf{x}^{(k)})\right]^{T}\mathbf{g}(\mathbf{x}^{(k)}) \tag{5.11}$$

Step 3: King-Werner method update - the state vector is adjusted by calculating the midpoint of the interval bounded by $\mathbf{x}^{(k)}$ and $\mathbf{y}^{(k)}$, as expressed in Equation (5.12).

$$\widetilde{\mathbf{x}}^{(k)} = \frac{1}{2}\left(\mathbf{x}^{(k)} + \mathbf{y}^{(k)}\right) \tag{5.12}$$

Step 4: Evaluation of state increments at the midpoint - in this step, the nonlinear equations are reevaluated at the midpoint determined in Step 3. Consequently, an intermediate increment vector is calculated using Equation (5.13).

$$\Delta\widetilde{\mathbf{x}}^{(k)} = -\mathbf{J}(\widetilde{\mathbf{x}}^{(k)})^{-1}\mathbf{g}(\widetilde{\mathbf{x}}^{(k)}) \tag{5.13}$$

Step 5: Updating the solution using the Heun's method - the state vector is updated using Heun's method as described in Equation (5.14).

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \frac{h^{(k)}}{2}(\psi^{(k)}\Delta\mathbf{x}^{(k)} + (2-\psi^{(k)})\Delta\widetilde{\mathbf{x}}^{(k)}) \tag{5.14}$$

where $\psi^{(k)}$ is a correction parameter updated at each iteration.

The authors in (TOSTADO-VÉLIZ *et al.*, 2020a) suggest setting the initial value of the parameter $\psi^{(0)} = 1$. Consequently, (5.14) represents the conventional configuration of the Heun's method with $k = 0$, as both $\Delta\mathbf{x}$ and $\Delta\widetilde{\mathbf{x}}$ have the same weight. As the iterative process progresses, adjustments in the parameter $\psi^{(k)}$ become essential in subsequent iterations (see Step 7).

Step 6: Convergence evaluation - if $k > k_{max}$ or the *mismatch* in (5.15) is satisfied for $\mathbf{x}^{(k+1)}$, the iterative process is stopped. Otherwise, the k counter is incremented, and the algorithm proceeds to Step 7.

$$\|\mathbf{g}(\mathbf{x})\|_\infty \leq \varepsilon \tag{5.15}$$

Step 7: Parameters update - at each iteration, the parameters $\rho^{(k)}$, $h^{(k)}$ and $\psi^{(k)}$ are updated according to (5.16), (5.17) and (5.18), respectively.

$$\rho^{(k+1)} = \left\|\mathbf{x}^{(k+1)} - \mathbf{y}^{(k)}\right\|_\infty \tag{5.16}$$

$$h^{(k+1)} = \begin{cases} max\left\{0.9h^{(k)}, h_{min}\right\} & \text{if } \rho^{(k+1)} > \alpha \\ min\left\{1.1h^{(k)}, h_{max}\right\} & \text{otherwise} \end{cases} \tag{5.17}$$

$$\psi^{(k+1)} = 2 \left| \frac{SSR^{(k)} - SSR^{(0)}}{SSR^{(0)}} \right| \tag{5.18}$$

After correctly updating the parameters, the process returns to Step 1.

As can be seen, the Heun-King-Werner method involves evaluating and factoring two Jacobian matrices at each iteration to achieve the solution of the nonlinear problem, as described in (5.8) and (5.13). In (TOSTADO-VÉLIZ *et al.*, 2020a), an automatic transition to the NR method was suggested when ψ exceeds a predefined limit $\widetilde{\psi}$, typically set as $\widetilde{\psi} = 2$. However, the need to assemble and factorize the Jacobian matrix of the HKW method at each iteration continues to represent a significant portion of computational time, especially in large-scale systems. In this regard, two new approaches to the Heun-King-Werner method were proposed in (OLIVEIRA; FREITAS, 2021) to achieve better computational performance of the HKW method.

5.1.3 Improved Heun-King-Werner Method Approaches

In (OLIVEIRA; FREITAS, 2021), two variants were presented to optimize the Heun-King-Werner method, aiming to enhance its computational efficiency by reducing the number of calculation operations and factorizations of Jacobian matrices. The first of these approaches, HKW-1, focuses on avoiding one of the Jacobian matrix calculation steps during the iteration of the HKW method and its subsequent factorization. Essentially, this entails computing and maintaining only the expression (5.8) frozen during the iteration in (5.13) instead of performing a new calculation as in Step 4 of the HKW method. Thus, the proposal involves replacing (5.13) with (5.19).

$$\Delta\widetilde{\mathbf{x}}^{(k)} = -\mathbf{J}(\mathbf{x}^{(k)})^{-1}\mathbf{g}(\widetilde{\mathbf{x}}^{(k)}) \tag{5.19}$$

The second proposed strategy, called HKW-2, introduces an approach where the assembly and factorization of Jacobian matrices from (5.8) and (5.13) are performed only in the first iteration or in another iteration indicated by the user. After this, the last Jacobian matrix calculated in (5.13) is fixed and kept constant for all subsequent iterations, thereby eliminating the need for new LU factorizations.

Even though both approaches have greatly reduced the computational cost compared to the original method and have successfully solved the power flow problem, they still rely on a set of parameters that are highly sensitive to changes when dealing with complex and large-scale power system models. This sensitivity makes it challenging to implement these methodologies for this type of application.

5.2 LEVENBERG–MARQUARDT METHOD

The power flow problem is typically solved as a system of nonlinear equations using iterative methods. However, it can also be formulated as an optimization problem, allowing it to be solved using minimization techniques (TOSTADO-VELIZ *et al.*, 2020; BUKHSH, 2018). Levenberg's method, renowned for its versatility in nonlinear programming, has been widely utilized in various optimization problems due to its effectiveness (MILANO, 2015).

Consider $\mathbf{x}_i$ as the vector $\mathbf{x}$ value at the $i-th$ iteration of an iterative method applied to equation (2.15). The corresponding error vector associated to $\mathbf{x}_i$ is:

$$\epsilon_i = \mathbf{g}(\mathbf{x}_i) \tag{5.20}$$

When $\mathbf{x}_i$ is modified to $\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta\mathbf{x}_i$, it results in a new error vector ϵ_{i+1}, which can be estimated through a first-order Taylor's expansion:

$$\epsilon_{i+1} = \mathbf{g}(\mathbf{x}_i + \Delta\mathbf{x}_i) \approx \epsilon_i + \mathbf{J}_i\Delta\mathbf{x}_i, \tag{5.21}$$

where $\mathbf{J}_i = \nabla^T\mathbf{g}(\mathbf{x}_i)$ represents the Jacobian matrix of $\mathbf{g}$ evaluated at $\mathbf{x}_i$. In Levenberg's method, the objective is to determine the variation $\Delta\mathbf{x}_i$ that minimizes the sum of the squares of errors:

$$\eta_i = \epsilon_{i+1}^T\epsilon_{i+1} = \Delta\mathbf{x}_i^T\mathbf{J}_i^T\mathbf{J}_i\Delta\mathbf{x}_i + 2\Delta\mathbf{x}_i^T\mathbf{J}_i^T\epsilon_i + \epsilon_i^T\epsilon_i \tag{5.22}$$

The minimum of (5.22) is obtained at:

$$\bigtriangledown\eta_i(\Delta\mathbf{x}_i) = 0 \tag{5.23}$$

Therefore, by combining equations (5.22) and (5.23), we obtain:

$$2\mathbf{J}_i^T\mathbf{J}_i\Delta\mathbf{x}_i + 2\mathbf{J}_i^T\epsilon_i = 0 \tag{5.24}$$

Next, (5.24) is solved for $\Delta\mathbf{x}_i$:

$$\Delta\mathbf{x}_i = -[\mathbf{J}_i^T\mathbf{J}_i]^{-1}\mathbf{J}_i^T\epsilon_i \tag{5.25}$$

and the vector $\mathbf{x}$ is updated as follows:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta\mathbf{x}_i \tag{5.26}$$

Note that, based on the property of the inverse of the product of two invertible square matrices, we have:

$$[\mathbf{J}_i^T\mathbf{J}_i]^{-1}\mathbf{J}_i^T = \mathbf{J}_i^{-1}[\mathbf{J}_i^T]^{-1}J_i^T = \mathbf{J}_i^{-1}\mathbf{I} = \mathbf{J}_i^{-1} \tag{5.27}$$

Therefore, equation (5.25) represents the increment obtained using the familiar Newton-Raphson method. Additionally, it is recognized that the Newton-Raphson method, and consequently equation (5.25), may diverge if the initial guess x_0 is significantly distant from solution of problem. Levenberg's main contribution lies in modifying the objective function η_i to improve numerical convergence. This improvement involves incorporating the distance from the current point $\mathbf{x}_i$ into the objective function (MILANO, 2015), which can be expressed as follows:

$$\eta_{\lambda,i} = \eta_i + \lambda\Delta\mathbf{x}_i^T\Delta\mathbf{x}_i \tag{5.28}$$

where λ is known as the *damping factor* in Levenberg's method. Consequently, equation (5.28) modifies equation (5.25) as follows:

$$\Delta\mathbf{x}_i = -[\mathbf{J}_i^T\mathbf{J}_i + \lambda\mathbf{I}]^{-1}\mathbf{J}_i^T\epsilon_i. \tag{5.29}$$

Note that the value of λ significantly impacts the number of iterations needed to achieve convergence. A larger value of λ reduces the number of iterations, but as the relative weight

of $\mathbf{J}_i^T\mathbf{J}_i$ decreases, convergence is not guaranteed. In most cases, a variant of equation (5.29) proposed by Marquardt is preferred:

$$\Delta\mathbf{x}_i = -[\mathbf{J}_i^T\mathbf{J}_i + \lambda diag(\mathbf{J}_i^T\mathbf{J}_i)]^{-1}\mathbf{J}_i^T\epsilon_i, \tag{5.30}$$

where the identity matrix $\mathbf{I}$ is substituted with a diagonal matrix composed of the diagonal elements of $\mathbf{J}_i^T\mathbf{J}_i$. This adjustment scales the impact of the damping factor and decreases the number of iterations needed to achieve convergence. This modification leads to the well-known Levenberg-Marquardt method (LM) described in equation (5.30). Aiming to improve the convergence and numerical stability of the LM method, several variants have been proposed in the literature such as (FAIRES; BURDEN, 2012).

5.3 TIKHONOV'S REGULARIZATION

The situation characterized by the Jacobian matrix severely ill-conditioned and/or the data (for example, the mismatches) with very high deviations is challenging for the PFP convergence. This occurs because the numerical solution of the problem can differ significantly from the expected one. Therefore, it is necessary to investigate a way of circumventing such restrictions. An alternative is to generate an adequate approximate solution for the state deviations. This process generally involves transforming a nearby system, which is more robust concerning perturbations. The process done this way is commonly known as regularization (GOLUB; LOAN, 1996). One well-known regularization method is Tikhonov's regularization (HANSEN, 1992; LIU, 2013; GOLUB; LOAN, 1996).

In its simplest form, the Tikhonov regularization problem for the ith iteration replaces the solution of the linear system in (2.45) by the minimization problem

$$\underset{\Delta\mathbf{x}^{(i)}}{min}\left\{\left\|\mathbf{J}(\mathbf{x}^{(i)})\Delta\mathbf{x}^{(i)} - (-\mathbf{g}(\mathbf{x}^{(i)}))\right\|^2 + \mu\left\|\Delta\mathbf{x}^{(i)}\right\|^2\right\}, \tag{5.31}$$

in which μ is an appropriate positive value known as the regularization parameter.

We can obtain the Tikhonov's solution $\Delta\mathbf{x}_\mu^{(i)}$ by solving the "regularized normal equation" (GOLUB *et al.*, 1979; HANSEN, 1992; GOLUB; LOAN, 1996)

$$([\mathbf{J}(\mathbf{x}^{(i)})]^T\mathbf{J}(\mathbf{x}^{(i)}) + \mu\mathbf{I})\Delta\mathbf{x}_\mu^{(i)} = -[\mathbf{J}(\mathbf{x}^{(i)})]^T\mathbf{g}(\mathbf{x}^{(i)}) \tag{5.32}$$

It is important to note that the minimization problem (5.31) has a unique solution for any $\mu > 0$, $\Delta\mathbf{x}_\mu^{(i)}$, given by:

$$\Delta\mathbf{x}_\mu^{(i)} = -[\mathbf{J}(\mathbf{x}^{(i)})^T\mathbf{J}(\mathbf{x}^{(i)}) + \mu\mathbf{I}]^{-1}\mathbf{J}(x^{(i)})^T\mathbf{g}(\mathbf{x}^{(i)}) \tag{5.33}$$

Therefore, comparing (5.31) and (5.29), it is verified that the results of Levenberg's method and Thikhonov's regularization are equivalent.

Given the parameter regularization μ, our objective is to compute the solution (5.33) by solving the linear system (5.32). However, directly solving the large-scale linear system (5.32) is not recommended due to the presence of the matrix product $\mathbf{J}^T\mathbf{J}$ in the process. Evidently, straightforward use of the resultant matrix compromises the sparsity of the conventional power flow problem, even the Jacobian matrix being highly sparse. Then, in the case of large-scale systems, an option is to investigate a modification of the original regularized normal equation with a form in which the sparsity pattern is preserved.

5.3.1 Efficient Resolution of the Minimization Problem

Given that a straightforward resolution of (5.32) is carried out, the linear coefficient matrix is composed of a product of matrices involving $\mathbf{J}(\mathbf{x}^{(i)})$. This operation leads to a sparsity reduction in relation to the original power flow problem. Then, aiming to reduce the loss of sparsity in the linear system, it is proposed to solve (5.32) by using an equivalent augmented linear system.

A generic linear system representing the normal equation (GOLUB; LOAN, 1996) $(\mathbf{A}^T\mathbf{A} + \mu\mathbf{I})\Delta\mathbf{x} = \mathbf{A}^T\mathbf{b}$ for the regularized problem is equivalent to the form:

$$\mathbf{A}^T(\mathbf{A}\Delta\mathbf{x} - \mathbf{b}) + \mu\Delta\mathbf{x} = \mathbf{0}. \tag{5.34}$$

Then, we propose the following modifications. Let $-\mathbf{z} = \mathbf{A}\Delta\mathbf{x} - \mathbf{b}$. Then (5.34) is modified to $-\mathbf{A}^T\mathbf{z} + \mu\Delta\mathbf{x} = 0$. Rewriting the system with $\mathbf{z} = \sqrt{\mu}\mathbf{y}$, we can obtain $\mathbf{A}^T\mathbf{y} - \sqrt{\mu}\Delta\mathbf{x} = \mathbf{0}$. Finally, an augmented linear system is determined:

$$\begin{bmatrix} \mathbf{A} & \sqrt{\mu}\cdot\mathbf{I} \\ -\sqrt{\mu}\cdot\mathbf{I} & \mathbf{A}^T \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \tag{5.35}$$

Therefore, the equivalent augmented linear system applied to the PFP corresponds to the following equation:

$$\begin{bmatrix} \mathbf{J}(\mathbf{x}^{(i)}) & \sqrt{\mu^{(i)}}\cdot\mathbf{I} \\ -\sqrt{\mu^{(i)}}\cdot\mathbf{I} & [\mathbf{J}(\mathbf{x}^{(i)})]^T \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}^{(i)} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} -\mathbf{g}(\mathbf{x}^{(i)}) \\ \mathbf{0} \end{bmatrix}, \tag{5.36}$$

in which $\mathbf{I}$ is the identity matrix, $\Delta\mathbf{x}^{(i)}$ is the solution of interest and $\mathbf{y}$ is another variable, but without interest for the PFP.

In (5.36), there is no product of matrices in the coefficient linear resulting matrix. On the other hand, the linear system presents twice as many equations and unknowns as compared to the conventional load flow problem. Despite this, the formulation implementation uses only the original Jacobian matrix and PFP mismatch. Therefore, the gain in sparsity is expected to compensate for the increase in dimension in the linear system.

Similarly, a second approach is proposed using an equivalent augmented linear system. The normal equation can still be decomposed as

$$[\mathbf{A}^T \quad \sqrt{\mu}\mathbf{I}] \begin{bmatrix} \mathbf{A} \\ \sqrt{\mu}\mathbf{I} \end{bmatrix} \Delta\mathbf{x} = \mathbf{A}^T\mathbf{b}. \tag{5.37}$$

Then, by establishing

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \sqrt{\mu}\mathbf{I} \end{bmatrix} \Delta\mathbf{x} \quad \longrightarrow \quad \mathbf{A}\mathbf{x} = \mathbf{z}_1, \quad \sqrt{\mu}\mathbf{I}\Delta\mathbf{x} = \mathbf{z}_2. \tag{5.38}$$

Therefore, (5.37) can be solved through the augmented linear system

$$\begin{bmatrix} \mathbf{A}^T & \sqrt{\mu}\mathbf{I} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \mathbf{A} \\ \mathbf{0} & -\mathbf{I} & \sqrt{\mu}\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \Delta\mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^T\mathbf{b} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{5.39}$$

In this case, the resulting linear system has three times as many variables as the original system. For this situation, as in the augmented system defined in (5.36), one must evaluate whether its implementation is the most appropriate.

5.3.2 Selection of the Regularization Parameter

The solution of (5.33) is intrinsically related to selecting the parameter μ. Then, its choice aroused the interest of several kinds of investigations (GOLUB *et al.*, 1979; SCHERZER, 1993;

CALVETTI *et al.*, 2000; HANSEN, 1992). Some different techniques were proposed in this direction.

Let $\mathbf{Ax} = b$, a generic linear system, where $\mathbf{A}$ is an ill-conditioned matrix; μ, the regularization parameter; and the linear system associated solution $\mathbf{x}_\mu$.

A technique known as generalized cross-validation (GCV) was defined in (GOLUB *et al.*, 1979) and consists of finding a parameter μ that minimizes the GCV function (LIU, 2013). The discrepancy principle (SCHERZER, 1993) is another technique used to determine the regularization parameter. For this strategy, the selection is based on choosing the parameter μ for satisfying the norm of the residual vector for the regularized solution $\mathbf{x}_\mu$, *i.e.*, $||\mathbf{Ax}_\mu - \mathbf{b}|| = \tau\delta$, where $\tau > 1$ and δ is a small positive number.

The method introduced in (HANSEN, 1992) known as L-curve proposes to find the optimized regularization parameter by determining the value lying on the corner of a curve in the form of "L". The curve is parameterized as a function of μ and is composed of the points defined by ($\rho(\mu)$,$\eta(\mu)$), where $\eta(\mu) = \Phi(||\mathbf{x}_\mu||)$, $\rho(\mu) = \Phi(||\mathbf{Ax} - \mathbf{b}||)$, and in general the function $\Phi(t)$ is chosen as $\Phi(t) = log_{10}(t)$. Other L-curve versions are also found in (CALVETTI *et al.*, 2000).

As described previously, all techniques to determine an optimized regularization parameter are associated with computations involving a determined optimization problem in the regularization parameter.

5.4 FINAL CONSIDERATIONS

In this chapter, methods based on the Newton-Continuous methodology, the Levenberg-Marquardt method, and Tikhonov's regularization were presented as robust alternatives for addressing ill-conditioned PFP scenarios. These methodologies were designed to enhance convergence reliability, making them suitable for solving PFP in large-scale networks with varying configurations and significant loading conditions. However, despite their robustness, these methods exhibit some disadvantages, particularly in terms of computational performance compared to the traditional NR method. Additionally, they often require extensive adjustments to accurately obtain the PFP solution.

To address these drawbacks, the following chapter will propose methods aimed at improving both the computational performance and robustness of existing techniques. These approaches

focus on enhancing the conditioning of the Jacobian matrix, thereby facilitating more efficient and reliable solutions to the Power Flow Problem.

CHAPTER 6

THESIS PROPOSED METHODS

In this chapter, new approaches are proposed to resolve the issue of ill-conditioning in the Jacobian matrix within the context of the power flow problem. The ill-conditioning of the Jacobian matrix poses a significant challenge, impacting the accuracy and efficiency of classical methods like Newton-Raphson, particularly in cases where the power flow solution's region of attraction is narrow or far from the initial guess. This challenging condition can result in convergence problems, making the solution of the power flow problem a complex and computationally intensive task.

To overcome these challenges, several approaches were proposed, including a conditioning step to improve the initialization of the iterative Power Flow Problem. Among these approaches, the application of Tikhonov's regularization was also considered. Furthermore, a hybrid method was developed, combining different techniques to address the issue of ill-conditioned power flow in large-scale systems starting from a flat estimate. These approaches aim to overcome the limitations of traditional methods and offer accurate solutions while avoiding convergence problems.

6.1 THE TWO-STEP HYBRID METHOD

This section describes the two-step hybrid method to efficiently compute the PFP solution. The approach was presented in (FREITAS; OLIVEIRA, 2023b). The first step uses a relaxed homotopy approach. In contrast, the second stage employs the result of the relaxed part as an estimate for a user-defined iterative solver, which computes the PFP-accurate solution. For this reason, we call this approach a two-step hybrid method. The primary purpose is to deal with ill-conditioned and large-scale power system models.

In (FREITAS; SILVA, 2022), a homotopy-based technique was presented to solve the PFP

emanating from the flat start estimate and for ill-conditioned large-scale problems. However, it is time-consuming because it is based on computations of iterations of the classical NR method and requires many iterations to achieve the final desired solution. On the other hand, several researches have demonstrated that NR-based solvers may be inefficient in solving ill-conditioned power flow problems at all (TOSTADO-VÉLIZ *et al.*, 2020a). Therefore, the study performed in the sequence aims to exploit the weak points of methods such as homotopy and NR-based and combine them into a hybrid and efficient manner to solve ill-conditioned and large-scale PFP.

The principle of the method consists in determining an accurate solution of the PFP, $\mathbf{x}_*$, exploring a partially inaccurate solution, $\mathbf{x}_h$, which in turn is obtained from a guess $\mathbf{x}^{(0)}$.

The computation process can be interpreted as a hybrid approach, where the relaxed intermediate solution, $\mathbf{x}_h$, is determined by the homotopy method as proposed in (FREITAS; SILVA, 2022). In that reference, the iterative PFP solution is determined exactly from a flat start estimation. This partial result is used as an initial estimate for a generic Iterative Method (IM) defined by the user. For example, the scheme in Fig. 6.1 illustrates a schematic for the generic iterative method to obtain the final (accurate) PFP solution $\mathbf{x}_*$. Therefore, the proposal in this paper does not employ the homotopy technique in the second step to obtain an accurate PFP solution. At the same time, in (FREITAS; SILVA, 2022), the high-precision result is already determined directly from the homotopy technique.

Aiming to deal with ill-conditioned power system models, we set $\mathbf{x}^{(0)}$ as a flat start type. As stated before, this initialization was explored in (FREITAS; SILVA, 2022) for determining a high-precision solution for the PFP. However, considering that this procedure is time-consuming since it needs to use the NR solver several times to reach an accurate solution, our objective is to relax the computations. *i.e.*, a *very low accuracy for the PFP result* is set in this first step. This way, the computational cost for the NR solver is significantly reduced, despite obtaining an inaccurate solution. In contrast, it is expected that i) the weak solution be efficient in initializing the IM and speed up the calculation of the accurate solution; and ii) the attraction region of convergence in relation to the initial estimate assigning for the IM be more constrained and have inside the PFP solution.

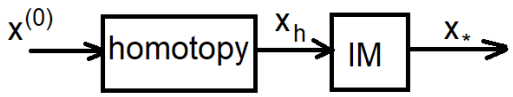


Figure 6.1. Scheme illustrating a hybrid procedure composed of an intermediate relaxed state obtained by a homotopy process and used as an initial estimate to obtain the solution of the PFP.

6.1.1 The first stage: partial solution

This subsection briefly describes the methodology presented in (FREITAS; SILVA, 2022) to justify the employment of the homotopy stage and the computational burden reduction to achieve the partial solution from the flat start guess.

The homotopy method's partial solution $\mathbf{x}_h$ is computed using a given tolerance for a relaxed convergence for the power mismatch. For this situation, a tolerance value $\epsilon_h = 2.0$ pu of power mismatch is suggested, which is very high compared to a desired high-precision tolerance adopted for the PFP, assigned as $\epsilon = 10^{-8}$ pu.

The PFP relaxed solution determined following the homotopy technique according (FREITAS; SILVA, 2022) requires the computation of roots at each point of the homotopy path. Therefore, solving a nonlinear equation system should find each path point. Considering that the tolerance for the relaxed homotopy method is $\epsilon_h >> \epsilon$, the procedure reduces the demanded iterations per homotopy point. This occurs because the NR solver is used for each point with a low accuracy for convergence. Then, minimizing the iterations along the homotopy process can be interpreted as another target to be reached in the proposed approach. A synthetic explanation is highlighted in the sequence to understand better the path points composing the homotopy path.

The homotopy process is formulated by incorporating parameters to incrementally alter a fictitious compensated network, referred to as the "easy" problem, which has a trivial solution. This network is gradually adjusted until the original network is restored, known as the "difficult" problem, which involves finding the zeros of (2.15), *i.e.*, without the fictitious compensated network. It is also assumed that the simple solution of the "easy" problem is precisely equal to the initial estimate used for solving the NR problem (FREITAS; LIMA-SILVA, 2023). The homotopy path in (FREITAS; SILVA, 2022) is guided by two-homotopy parameters h_1 and h_2 featuring a two-dimensional problem. These parameters, in turn, are used to modify:

- The diagonal admittance matrix of the system, which starts with a fictitious value and returns to its original value at the end of the homotopy process;
- The admittance values connecting the slack bus to adjacent buses, forming a region around the slack bus.

The nonlinear equation system resolution for each point i in the discrete parameters (h_{1i},h_{2i}) of the homotopy path is *performed by the classical NR* solver. The number of points for the path depends on how the parameters h_1 and h_2 are discretized. The step-length for these parameters are chosen differently. This process is carried out considering that parameter h_2 has the binary type, assuming only the on/off status (0 or 1 value). Then, first, for the value $h_2 = 0$, only the parameter h_1 is changed. It is modified by step-length $\Delta h_1 = 1/N$, where N is the number of step-length assigned in the interval [0,1]. In this case, each discrete value h_{1i} is updated as $h_{1i} = i\Delta h_1$, $i = 0,1,\ldots,(N-1)$, and $h_{1N} = 1$.

The homotopy problem, when executed with $h_2 = 0$ works with modified impedances connecting the slack bus, as shown in Eq. (6.1).

$$\hat{z}_{km} = r_{km} + j(1 - h_2)\delta x_{km} + jh_2 x_{km} \tag{6.1}$$

where $z_{km} = r_{km} + jx_{km}$ is the effective impedance connecting the slack bus, for $h_2 = 0$ and $\delta \in (0,1]$ is a scaling factor assigned by the user. In most cases, it is enough to set δ for the unitary value, *i.e.*, without changing in z_{km}. It is important to note that this strategy aims to turn the region near the slack bus stronger and offer a "higher transmission capacity" for the respective links along with the homotopy process. This way, it is expected that the transmission links that connect the slack bus have a wide range of power variation flexibility (FREITAS; SILVA, 2022). And we assume that the critical iterate oscillations can be verified near the slack bus.

The parameter h_1 ($0 \leq h_1 \leq 1$) enables the connection of a fictitious shunt admittance at a given bus k, y_k (FREITAS; SILVA, 2022). Thus, just the diagonal of the admittance matrix needs to be changed. Consider $Y_{BUS}(k,k)$ to be a diagonal entry of the original admittance matrix Y_{BUS}. The homotopy parameter h_1 is utilized to calculate a dependent diagonal element as follows:

$$Y_{BUS}^{(i)}(k,k) = \hat{Y}_{BUS}(k,k) + (1 - h_{1i})y_k, \tag{6.2}$$

where $y_k = g_k + jb_k$ is the fictitious admittance fitted for $i = 0$, the initial path point; $\hat{Y}_{BUS}$ is the network admittance matrix computed considering $\hat{z}_{km}$. For the initial path point, the initial estimate is $\mathbf{x}^{(0)}$ and the PFP solution is adjusted to coincide with $\mathbf{x}^{(0)}$. So, y_k must be imposed to reach the initial solution without needing additional iteration.

Finally, for $h_2 = 1$, only the point for h_{1N} need to be calculated, and the updated matrix is $Y_{BUS}^{(N)}(k,k) = Y_{BUS}$, with Y_{BUS} being the original admittance matrix computed with the value z_{km} instead of $\hat{z}_{km}$. Therefore, the original network is entirely recovered, and the PFP solution $\mathbf{x}_h$ is obtained.

Given that initially, only a low-precision solution is sufficient, it imposes a low number of step-length N on the homotopy curve, because this factor reduces the computational impact to arrive at the solution $\mathbf{x}_h$. Conversely, when N is too much reduced, a large distance between the points on the homotopy curve is verified. The result can be detrimental to the initial estimates used on the homotopy curve to solve the partial PFP. Assuming that at $h_1 = 0$ there is a known ("easy") solution, the subsequent are calculated using the previous result as an initial estimate. Therefore, considering this compromise, a relaxed solution with high mismatches, as well as the objective of reducing the computational load to determine the solution in $h_1 = 1$, N, must be chosen in order to meet these requirements. In general, choosing at most $N = 10$ produces adequate results. Also, no more than 2 iterations are enough to reach the prescribed mismatch ϵ_h at a given point h_{1i}. Therefore, the total number of iterations to calculate $\mathbf{x}_h$ is in the range N to $2N$. But, in most cases, only N iterations are enough at all.

6.1.2 The second stage: general iterative method

The IM part of the hybrid method investigated in this thesis proposal uses the partial solution $\mathbf{x}_h$ determined in the first stage using the homotopy technique. In this second stage, it is where the accurate solution is determined. The IM is implemented by employing the classical NR solver and its fast decoupled version FDXB (ZIMMERMAN *et al.*, 2011). Besides, the HKW method (TOSTADO-VÉLIZ *et al.*, 2020a) and variants (OLIVEIRA; FREITAS, 2021) are also

used. However, in principle, any other iterative techniques could also be evaluated.

The algorithm HKW (TOSTADO-VÉLIZ *et al.*, 2020a) is a powerful method to solve (2.15) for electrical networks considered ill-conditioned. In (OLIVEIRA; FREITAS, 2021), two variants of this method, HKW1, and HKW2, are also presented, which allows a reduction in the computational burden of the original approach.

Although the methods HKW, HKW1, and HKW2 are robust and have obtained the solution of the PFP for all systems presented in (OLIVEIRA; FREITAS, 2021), adjustments of the parameters for large ill-conditioned systems need to be done by trial and error. This artifice is laborious and is sometimes unsuccessful due to the number of involved parameters. Therefore, by employing the hybrid strategy, it is expected to use a set of standard parameters (TOSTADO-VÉLIZ *et al.*, 2020a) for any ill-conditioned system involving the HKW methods, without the need for specific adjustments, due to the improvement in the intermediate estimate to be utilized.

6.2 CONDITIONING STEP-BASED METHOD

This section presents a technique based on an initialization step to allow convergence in the iterative process to solve the PFP. The approach can be applied to any iterative method to solve the PFP and can be interpreted as a modified first iteration of the global iterative process to solve a nonlinear equation system. This work uses this adaptation to the classical NR, and the HKW method (TOSTADO-VÉLIZ *et al.*, 2020a) for ill-conditioned and large-scale test systems. The former was reported in several works to fail for most of the experiments (TOSTADO-VÉLIZ *et al.*, 2020a). The latter was reported to be appropriate for dealing with both well- and ill-conditioned systems.

In this approach for the PFP, the flat start is always assumed as the initial estimate. However, the 'conditioning step' procedure on the initial estimate is proposed because of the weakness in obtaining an adequate convergence of the iterating process considering this initialization. Note that the process differs from the pre-conditioning of matrices discussed in Chapter (3). While the former is a proposition involving the sum of a perturbation matrix, the latter is related to the products of matrices.

Initially, given an initial guess $\mathbf{x}^{(0)}$, the straightforward purpose would be to compute the increment $\Delta\mathbf{x}^{(0)}$ by solving the linear system $\mathbf{J}(\mathbf{x}^{(0)})\Delta\mathbf{x}^{(0)} = -\mathbf{g}(\mathbf{x}^{(0)})$ and so to obtain $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \Delta\mathbf{x}^{(0)}$ at the end of the first iteration. However, numerically, the result $\Delta\mathbf{x}^{(0)}$ can be equivalent to solving $[\mathbf{J}(\mathbf{x}^{(0)})]^T\mathbf{J}(\mathbf{x}^{(0)})\Delta\mathbf{x}^{(0)} = -[\mathbf{J}(\mathbf{x}^{(0)})]^T\mathbf{g}(\mathbf{x}^{(0)})$.

Therefore, the proposed procedure involves:

1. Adding a perturbation diagonal matrix $\sqrt{\delta}\mathbf{I}$ to $\mathbf{J}(\mathbf{x}^{(0)})$ or alternatively, $\delta\mathbf{I}$ to $[\mathbf{J}(\mathbf{x}^{(0)})]^T\mathbf{J}(\mathbf{x}^{(0)})$, with a scalar $\delta > 0$ and δ small, in such way that $\left\{\mathbf{J}(\mathbf{x}^{(0)}) + \sqrt{\delta}\mathbf{I}\right\}\Delta\bar{\mathbf{x}}_\delta^{(0)} = -\mathbf{g}(\mathbf{x}^{(0)})$ or $\left\{[\mathbf{J}(\mathbf{x}^{(0)})]^T\mathbf{J}(\mathbf{x}^{(0)}) + \delta\mathbf{I}\right\}\Delta\mathbf{x}_\delta^{(0)} = -[\mathbf{J}(\mathbf{x}^{(0)})]^T\mathbf{g}(\mathbf{x}^{(0)})$ must be solved to compute $\Delta\bar{\mathbf{x}}_\delta^{(0)}$ or $\Delta\mathbf{x}_\delta^{(0)}$, instead to determine simply $\Delta\mathbf{x}^{(0)}$;

2. considering that, $\Delta\bar{\mathbf{x}}^{(0)} \approx \Delta\mathbf{x}_\delta^{(0)}$, the initial estimate $\mathbf{x}^{(0)}$ can be 'conditioned' to $\mathbf{x}_\delta^{(0)} = \mathbf{x}^{(0)} + \Delta\mathbf{x}_\delta^{(0)}$, and the idea is then to set $\mathbf{x}^{(0)} := \mathbf{x}_\delta^{(0)}$ to be used by an iterative method in the subsequent iterations.

Then it is proposed to *initialize the iterative process for the solution of the PFP* by using the conditioned estimate $\mathbf{x}_\delta^{(0)}$, as calculated previously in step 4, instead of initializing directly with the original $\mathbf{x}^{(0)}$. It is important to note that the proposed method differs from Levenberg's method in that the latter incorporates a perturbation parameter—referred to as the damping factor in Levenberg's method—at each iteration. In contrast, the approach outlined above only introduces the perturbation matrix during the first iteration.

The perturbed linear systems in step 3 provide different solutions because they have different forms of perturbations. So we can first classify them as two options:

- Option I: linear system $\left\{\mathbf{J}(\mathbf{x}^{(0)}) + \sqrt{\delta}\mathbf{I}\right\}\Delta\bar{\mathbf{x}}_\delta^{(0)} = -\mathbf{g}(\mathbf{x}^{(0)})$ with solution $\Delta\bar{\mathbf{x}}_\delta^{(0)}$; or
- Option II: $\left\{[\mathbf{J}(\mathbf{x}^{(0)})]^T\mathbf{J}(\mathbf{x}^{(0)}) + \delta\mathbf{I}\right\}\Delta\mathbf{x}_\delta^{(0)} = -[\mathbf{J}(\mathbf{x}^{(0)})]^T\mathbf{g}(\mathbf{x}^{(0)})$, with result $\Delta\mathbf{x}_\delta^{(0)}$.

Assuming that the Jacobian matrix is ill-conditioned, the result $\Delta\mathbf{x}^{(0)}$ in steps 1 and 2 may have a significant norm. Hence, when added to $\mathbf{x}^{(0)}$, it causes large variations at the end of the first iteration. This is one of the possible causes of the divergence of iterative methods. The insertion of the perturbation matrices aims to reduce the condition number of the perturbed matrices. Therefore, this procedure also contributes to reducing the high variations that occur

due to the ill-conditioning of the non-perturbed matrices. However, the perturbation must be introduced considering a limitation in the value of δ related to the condition number of the coefficient matrices of the linear systems and matrix norms. Therefore, considering these aspects for adequate proceeding implementation, this parameter should be chosen.

To select δ, we can use the norm relation between a result of a perturbed and a non-perturbed linear system as presented in (GOLUB; LOAN, 1996). For example, consider for an estimate $\mathbf{x}^{(0)}$ of the PFP the following linear systems:

1. non-perturbed $\mathbf{J}\Delta\mathbf{x}^{(0)} = \mathbf{b}$; and
2. Jacobian perturbed by $\sqrt{\delta}\Delta\mathbf{J}$, resulting in $(\mathbf{J} + \sqrt{\delta}\Delta\mathbf{J})\Delta\mathbf{x}_{\delta}^{(0)} = \mathbf{b}$, $\delta > 0$.

According to (GOLUB; LOAN, 1996), for a small value δ, by using norm analysis, the following result arises involving the deviation between the perturbed and non-perturbed solution to the absolute non-perturbed solution.

$$\frac{||\Delta\mathbf{x}_{\delta}^{(0)} - \Delta\mathbf{x}^{(0)}||}{||\Delta\mathbf{x}^{(0)}||} \leq \sqrt{\delta}||\mathbf{J}^{-1}|| \cdot ||\Delta\mathbf{J}|| = \sqrt{\delta}\frac{\kappa(\mathbf{J})}{||\mathbf{J}||}||\Delta\mathbf{J}|| \tag{6.3}$$

where $\kappa(\mathbf{J})$ is the condition number of $\mathbf{J}$, and $||\cdot||$ is a norm of $\cdot$.

By assigning $\Delta\mathbf{J} = \mathbf{I}$, we conclude that the relations must satisfy a threshold that is associated with δ, $\kappa(\mathbf{J})$ and $||\mathbf{J}||$.

$$\frac{||\Delta\mathbf{x}_{\delta}^{(0)} - \Delta\mathbf{x}^{(0)}||}{||\Delta\mathbf{x}^{(0)}||} \leq \sqrt{\delta}\frac{\kappa(J)}{||\mathbf{J}||}$$

Let ρ be the threshold of the user assignments, assuming a high deviation. Then,

$$\frac{||\Delta\mathbf{x}_{\delta}^{(0)} - \Delta\mathbf{x}^{(0)}||}{||\Delta\mathbf{x}^{(0)}||} \leq \sqrt{\delta}\frac{\kappa(J)}{||\mathbf{J}||} \leq \rho.$$

By setting ρ for a given relative error between the state of the perturbed and non-perturbed linear system, the parameter δ can be limited by

$$\sqrt{\delta} \leq \rho\frac{||\mathbf{J}||}{\kappa(J)}. \tag{6.4}$$

Finally, from (6.4), it is possible to estimate the parameter δ for setting a given small perturbation to the Jacobian matrix. This procedure can also be used when the normal matrix $[\mathbf{J}]^T\mathbf{J}$ is used as an augmented matrix as presented in (6.5).

In relation to using Option I, the perturbed term is added to the Jacobian matrix, which is an indefinite matrix, *i.e.*, its eigenvalues have negative and positive real parts. In this case, the perturbation matrix may cause some eigenvalues that are near zero to move even closer to zero. This would be bad for the conditioning of the linear system matrix, as it would have worse conditioning than the undisturbed matrix. If this is not the case, the perturbation in question is quite beneficial to significantly improve the conditioning of the coefficient matrix of the linear system. Option II does not suffer from the described problem because the perturbed term is added to a normal matrix (GOLUB; LOAN, 1996), *i.e.*, a matrix which eigenvalues are real and positive (positive-definite). Therefore, the perturbed term moves the eigenvalues in the opposite direction to the null eigenvalue, a cause of singularity.

In Option II, an equivalent form for the perturbed linear system is proposed to be solved as

$$\begin{bmatrix} \mathbf{J}(\mathbf{x}^{(0)}) & \sqrt{\delta}\cdot\mathbf{I} \\ -\sqrt{\delta}\cdot\mathbf{I} & [\mathbf{J}(\mathbf{x}^{(0)})]^T \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}_{\delta}^{(0)} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} -\mathbf{g}(\mathbf{x}^{(0)}) \\ \mathbf{0} \end{bmatrix} \tag{6.5}$$

where $\mathbf{I}$ stands for the identity matrix of appropriate order and $\mathbf{z}$ is an auxiliary variable.

The computation of the increment $\Delta\mathbf{x}_{\delta}^{(0)}$ through (6.5) avoids the operation involving the product of the Jacobian matrix required in step 3. However, the straightforward computation of the product is unfavorable concerning sparsity compared with the proposed efficient computation process. Note that the sparsity of matrix $\mathbf{J}$ is preserved. Also, the condition number of the augmented non-perturbed matrix, *i.e.*, with $\delta = 0$ in (6.5) is precisely equal to the one of the Jacobian matrix.

A third option can be proposed as an alternative to the perturbed linear system in (6.5).

- Option III: two-parameter perturbed augmented form:

$$\begin{bmatrix} \mathbf{J}(\mathbf{x}^{(0)}) & (1+d)\sqrt{\delta}\cdot\mathbf{I} \\ -d\sqrt{\delta}\cdot\mathbf{I} & [\mathbf{J}(\mathbf{x}^{(0)})]^T \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}_{\delta}^{(0)} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} -\mathbf{g}(\mathbf{x}^{(0)}) \\ \mathbf{0} \end{bmatrix} \tag{6.6}$$

where d is an additional parameter assigned by the user.

The terms $(1+d)$ and d in (6.6) are introduced empirically, working as an additional freedom degree among an infinite of possibilities to implement the perturbation on the linear system derived from the normal matrix. The parameter d modifies the asymmetry of the perturbed matrix, but its value should also be small as δ.

The flowchart in Fig. 6.2 summarizes the implementation of the conditioning step (CS) and, as a result, is used by an iterative method. Note that the CS procedure works as a modified first iteration. Its implementation requires a little more computational cost than an iteration without CS (bypassing CS), but the overall convergence benefit for an iterative method is tremendously high. Because considering our numerous experiments with ill-conditioned systems and as will be shown in the results later, it provides adequate conditioning to the global convergence of an iterative method.

The following example illustrates the improvement introduced by a conditioning step on the initial estimate for solving a generic nonlinear system which initial iteration consists of an ill-conditioned linear system. Let $g_1(x_1,x_2) = x_1^2 + x_2^2 - 2x_1x_2 - 1 = 0$, $g_2(x_1,x_2) = x_1 + x_2 - 2 = 0$, and $\mathbf{g}(x_1,x_2) = [g_1(x_1,x_2) \quad g_2(x_1,x_2)]^T$ a set of nonlinear algebraic equations. Suppose that the initial estimate for solving the problem consists of the point $x_1^{(0)} = 1$ and $x_2^{(0)} = 1$. In this example, the ill-conditioning (singularity of the Jacobian) is verified exactly at the initial estimate point. However, the illustration is sufficient to highlight the importance of perturbation for improving the performance of the convergence process.

The system's roots can be found iteratively by applying the NR method. However, for the given guess, the Jacobian matrix is singular. On the other hand, for a small increment (say ϵ) only in $x_2^{(0)}$, now produces a nonsingular Jacobian matrix; still, the resulting linear system is ill-conditioned.

The iterative process behavior for the NR method for some values of ϵ is investigated. The behavior is evaluated considering an initial estimate for the NR method used directly and by considering its conditioned step (CS-NR). The value $\delta = 0.005$ was adopted for the perturbation in (6.5). Considering that the initial estimate produces a singular Jacobian matrix, the variable $\mathbf{x}_2^{(0)}$ was modified slightly by a value $\epsilon \neq 0$ to remove the singularity of $\mathbf{J}$, *i.e.*, $\mathbf{x}_2^{(0)} = 1 + \epsilon$. The following values for ϵ were tested: 0.01, 0.001, and 0.0001. The roots converge to $x_{1*} = 0.5$ and $x_{2*} = 1.5$.

According to (GOLUB; LOAN, 1996), a problem is *mathematically defined* as ill-conditioned if the calculated values are very sensitive to small changes in the parameters which characterize the problem. Therefore, given a linear system $\mathbf{Ax} = \mathbf{b}$, a 'condition number' provides an overall assessment of the rate of change of the solution $\mathbf{x}$ concerning modifications in the parameters

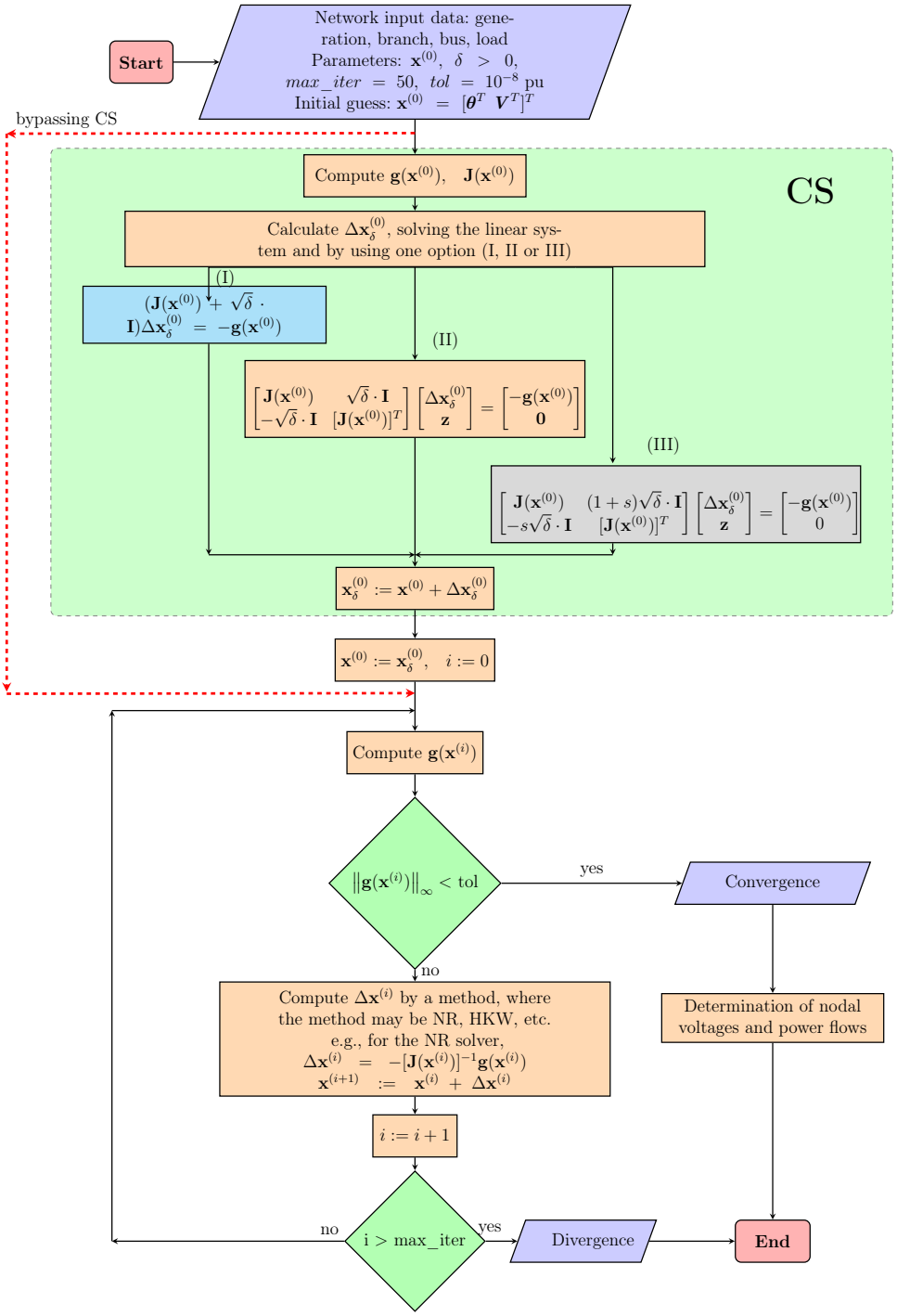


Figure 6.2. Flowchart illustrating the main procedure aspects of the CS and the iterations for the PFP.

in $\mathbf{A}$ or(and) $\mathbf{b}$. A spectral definition (norm-2 based) for the condition number of the matrix $\mathbf{A}$ is given by the index $\kappa(\mathbf{A}) = \frac{\sigma_{max}(\mathbf{A})}{\sigma_{min}(\mathbf{A})}$ (GOLUB; LOAN, 1996), where $\sigma_{max}(\mathbf{A})$ and $\sigma_{min}(\mathbf{A})$ are maximum and minimum singular values of $\mathbf{A}$, respectively. As the condition number increases, the degree of the ill-conditioned system grows as well (GOLUB; LOAN, 1996).

Table 6.1 depicts the condition number κ for four Jacobian matrices computed of different forms to the three values of ϵ. The second column shows κ for the matrix $\mathbf{J}_{CS}$, which is the

Jacobian matrix calculated for $\mathbf{x}_{\delta}^{(0)}$. The third column shows the κ for the Jacobian matrix's initial iteration of the traditional NR solver. The fourth column shows the condition number for the perturbed matrix $\mathbf{J}^T\mathbf{J} + \delta\mathbf{I}$. Finally, the fifth column shows κ calculated when the augmented matrix $\mathbf{A}_a$, the implicit form for $\mathbf{J}^T\mathbf{J} + \delta\mathbf{I}$, is used as proposed in (6.5). We can observe a substantial variation in the condition number associated with the Jacobian matrix of the traditional NR solver.

Table 6.1. Condition number for the ε calculated for four different matrices.

ε	$\mathbf{J}_{CS}$	$\mathbf{J}_{NR}$	$\mathbf{J}^T\mathbf{J} + \delta\mathbf{I}$	$\mathbf{A}_a$
10^{-4}	6.2	10^4	401.0	20.0
10^{-3}	1.6	10^3	400.4	20.0
10^{-2}	13.6	10^2	345.7	18.6

Fig. 6.3 shows plots for the mismatch $||\mathbf{g}(x_1,x_2)||_\infty$. The experiments are carried out for the values of ϵ considering the NR (dashed curve) and CS-NR (solid curve) solvers. Higher values of ϵ give better conditioning for the Jacobian matrix of the initial solution. In this case, the tendency is for the two curves to be close. However, the more ill-conditioned the system, the further apart they are, indicating the better performance of the CS-NR solver. This is proven by observing the number of iterations required for convergence for NR and CS-NR. Note that the latter always requires fewer iterations for convergence. Fig. 6.4 shows plots for the mismatch for the three options of implementation of the CS-NR, considering the NR solver without CS, $\epsilon = 10^{-4}$, $\delta = 0.0036$ and $d = 0.1$. The number of iterations reduces drastically from 17 (NR without CS) to just three iterations when implementing the perturbation with Option III. The other options also significantly reduce the number of iterations but are well above Option III's. This result demonstrates the high impact caused by an adequate selection of perturbation on the global convergence of an iterative process.

Similar results are obtained if the perturbation on the initial estimate of the variable $\mathbf{x}_2$ is taken as $\mathbf{x}_2^{(0)} = 1 - \epsilon$. However, in this case, the convergence occurs for the point $\mathbf{x}_{1*} = 1.5$ and $\mathbf{x}_{2*} = 0.5$.

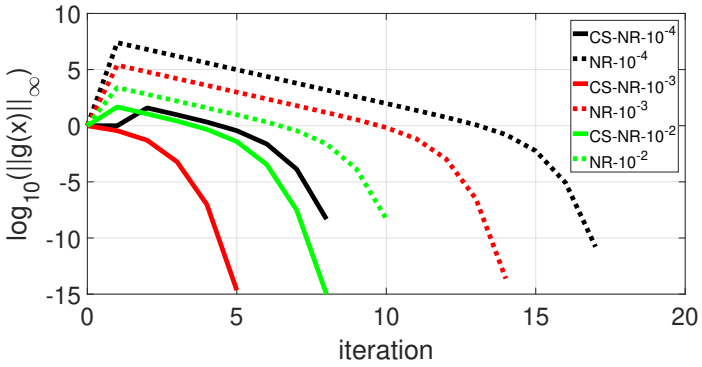


Figure 6.3. Plots illustrating the mismatches for an ill-conditioned initialization case for a generic nonlinear system considering the NR and CS-NR approaches for three values of ε.

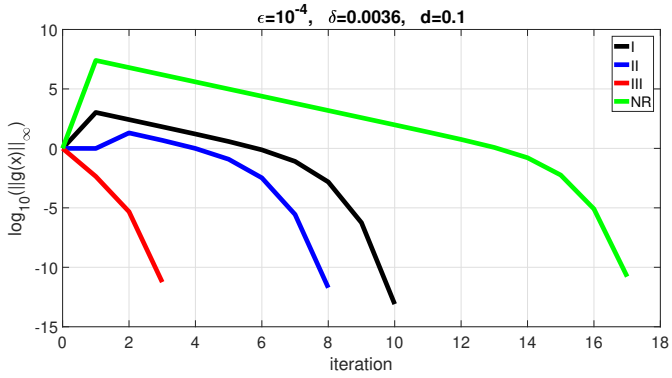


Figure 6.4. Plots illustrating the mismatches for an ill-conditioned initialization case for a generic nonlinear system considering different options to implement CS-NR approaches.

6.3 THE PSEUDO-REGULARIZED LOAD FLOW METHOD

In this section, an alternative approach to solving the power flow problem is proposed, exploring the concept of regularization methodology presented in Section 5.3. However, the idea is to use the regularization parameter only for the first iteration of the iterative Newton's method applied to the PFP, since the ill-conditioning of the problem is generally identified in this iteration. Due to the application of this strategy, the method is referred to as the *pseudo-regularized load flow method*.

The iterative solution of the PFP using this method is carried out in two stages:

- The first stage is equivalent to the first iteration of the load flow. However, the iteration is solved assuming a regularized linear system constructed from the Jacobian matrix of the first conventional PFP, $\mathbf{A} = \mathbf{J}(\mathbf{x}^{(0)})$, and the associated mismatch vector, $\mathbf{b} = -\mathbf{g}(\mathbf{x}^{(0)})$. The user-defined regularization parameter is μ. Therefore, the state-vector in this stage

is the result $\mathbf{x}^{(1)}$, computed from $(\mathbf{A}^T\mathbf{A} + \mu\mathbf{I})\Delta\mathbf{x}^{(0)} = b$, and $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \Delta\mathbf{x}^{(0)}$. The parameter μ can be assigned following one option as presented in Section 5.3.2.

- From the second iteration onward, the regularization parameter μ is removed, and the calculations are performed as in the traditional NR method or another chosen one. For NR, we proposed two alternatives: one uses the full Jacobian, labeled T-NR, while the second uses only the second Jacobian's iteration (frozen for subsequent iterations), called T-NR1J.

The Alg. 7 highlights the main stages of the proposed method. Replace T-NR (T-NR1J) by the desired approach when implementing other techniques. In Alg. 7, the tolerance corresponds to the accuracy desired for the power mismatch in pu. The index max_{iter} is the maximum number of iterations allowed for the load flow convergence. In step 2 of the algorithm, the L-curve can be computed for a given number of points.

Algorithm 7 Pseudo-regularized load flow solver for the classical NR method.

1: **Require:** Flat start estimate $\mathbf{x}^{(0)}$; tolerance ϵ; max_{iter}; grid data.
2: **Ensure:** Solution of the PFP, $\mathbf{x}_*$, *i.e.*, state vector of the system, $\mathbf{x}_* = \left[\boldsymbol{\theta}^T \;\; \boldsymbol{V}^T\right]^T$.
3: Do $k = 0$ and estimate μ_* for the initial iteration, for example, using the L-curve;
4: Assign $\mathbf{A} = \mathbf{J}(\mathbf{x}^{(0)})$, $\mathbf{b} = -\mathbf{g}(\mathbf{x}^{(0)})$, compute $(\mathbf{A}^T\mathbf{A} + \mu_*\mathbf{I})\Delta\mathbf{x}^{(0)} = b$, $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \Delta\mathbf{x}^{(0)}$, compute $\mathbf{g}(\mathbf{x}^{(1)})$ and go on with $k = 1$.
5: **while** $(||\mathbf{g}(\mathbf{x}^{(k)})||_\infty > \epsilon$ and $k < max_{iter})$ **do**
6: If $k = 1$, compute $\mathbf{J}(\mathbf{x}^{(k)})$ and do $\mathbf{J}_1 = \mathbf{J}(\mathbf{x}^{(1)})$.
7: If $k > 1$ and the selected method is T-NR1J, do $\mathbf{J}(\mathbf{x}^{(k)}) = \mathbf{J}_1$; or case the method is T-NR, calculate $\mathbf{J}(\mathbf{x}^{(k)})$.
8: update the deviation $\Delta\mathbf{x}^{(k)} = -[\mathbf{J}(\mathbf{x}^{(k)})]^{-1}\mathbf{g}(\mathbf{x}^{(k)})$;
9: update the state-vector $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta\mathbf{x}^{(k)}$ and proceed with $k := k + 1$;
10: compute the mismatch $\mathbf{g}(\mathbf{x}^{(k)})$.
11: **end while**

6.4 MODAL-BASED JACOBIAN MATRIX PERTURBATION

Aiming to circumvent (attenuate) the ill-conditioning effect of the PFP in the NR solver first iteration, a modal-based technique is also proposed to move away the smallest eigenvalue of the Jacobian (SEJ) matrix. The idea is to compute the SEJ and its respective right- and

left-eigenvectors to move out the problematic mode from near to zero.

Let λ_1 be the SEJ and $\mathbf{u}_1$ and $\mathbf{v}_1$ its respective right- and left-eigenvectors. Then a perturbed Jacobian matrix $\hat{\mathbf{J}} = \mathbf{J} + \alpha\mathbf{v}_1\lambda_1\mathbf{u}_1^T$ has its SEJ changed to $\hat{\lambda}_1 = \lambda_1 + \alpha\lambda_1$ for a factor α. When $\alpha > 1$, the magnitude of $\hat{\lambda}_1$ grows. Then, α can be selected to yield $|\lambda_2| < (1+\alpha)|\lambda_1|$, where λ_2 is the second SEJ. In general, λ_2 is relatively away from zero and is less dominant than λ_1. Therefore, the ill-conditioning related to the initial iterate for determining the approximate increment $\Delta\mathbf{x}$ for (2.44) is expected to be greatly reduced.

For the initial iteration, given $\mathbf{x}^{(0)}$, the straightforward form to compute the increment $\Delta\mathbf{x}^{(0)}$ is by solving the linear system $\mathbf{J}(\mathbf{x}^{(0)})\Delta\mathbf{x}^{(0)} = -\mathbf{g}(\mathbf{x}^{(0)})$ and so to obtain $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \Delta\mathbf{x}^{(0)}$. However, assuming that $\mathbf{J}(\mathbf{x}^{(0)})$ is ill-posed, *we propose to introduce a perturbation term* on the Jacobian matrix in such a way to move just the smallest eigenvalue of $\mathbf{J}(\mathbf{x}^{(0)})$.

Using the 1-rank perturbation matrix for $\mathbf{J}(\mathbf{x}^{(0)})$, we obtain an approximate deviation for the initial iterate $\Delta\hat{\mathbf{x}}^{(0)}$ as

$$[\mathbf{J}(\mathbf{x}^{(0)}) + \mathbf{v}_1\Lambda\mathbf{u}_1^T]\Delta\hat{\mathbf{x}}^{(0)} = -\mathbf{g}(\mathbf{x}^{(0)}) \tag{6.7}$$

where $\Lambda = \alpha\lambda_1$; $\alpha > 1$ must be fixed in order to move away from near to zero the SEJ; and both $\mathbf{v}_1$ and $\mathbf{u}_1$ are dense vectors.

An efficient way for solving (6.7) is applying the 1-rank Sherman-Morrison inverse technique (HAGER, 1989) for the solution

$$\Delta\hat{\mathbf{x}}^{(0)} = -[\mathbf{J}(\mathbf{x}^{(0)}) + \mathbf{v}_1\Lambda\mathbf{u}_1^T]^{-1}\mathbf{g}(\mathbf{x}^{(0)}) = \Delta\mathbf{x}^{(0)} - \Delta\mathbf{v}_1[\Lambda^{-1} + \mathbf{u}_1^T\Delta\mathbf{v}_1]^{-1}\mathbf{u}_1^T\Delta\mathbf{x}^{(0)}, \tag{6.8}$$

and $\Delta\mathbf{x}^{(0)} = -[\mathbf{J}(\mathbf{x}^{(0)})]^{-1}\mathbf{g}(\mathbf{x}^{(0)})$, $\Delta\mathbf{v}_1 = [\mathbf{J}(\mathbf{x}^{(0)})]^{-1}\mathbf{v}_1$.

The expression (6.8) gives the approximate increment result for the states in the first iterate, allowing the computation of $\mathbf{x}^{(1)} \approx \mathbf{x}^{(0)} + \Delta\hat{\mathbf{x}}^{(0)}$. Note that just one LU factorization is needed for the initial iterate, as done for the classical NR solver, and the sparsity of the standard PFP is preserved. Besides, only one additional forward and backward substitution operation is performed to determine $\Delta\mathbf{v}_1$. Finally, the remainder iterates should be determined according to (2.44) as for the standard NR solver, without the 1-rank perturbation matrix.

6.5 FINAL CONSIDERATIONS

One of the main objectives of this work is to develop power flow solvers that are robust and efficient, making them suitable for wide-scale use in solving large-scale and ill-conditioned systems. To address these challenges, solution approaches were proposed that include a conditioning step procedure to initialize the iterative process of the power flow problem, as well as a hybrid method that combines the homotopy technique with different methods to resolve the ill-conditioned power flow problem starting from a flat start estimation.

The introduced solvers exhibit competitive computational load, comparable to the NR method, as the proposed modifications only affect the first iteration. This computational efficiency is a crucial feature for any industrial or commercial solver. In Chapter 7, several numerical results are presented to evaluate the performance of the proposed approaches. These results are validated on ill-conditioned systems of up to 70,000 buses and compared with standard PFP solution methods, demonstrating the effectiveness and competitiveness of the new solvers.

In summary, the proposed conditioning step aims to improve the initial guess for the iterative process, improving the conditioning of the Jacobian matrix, as well as reducing the number of iterations required for convergence and enhancing the robustness of the solver. The hybrid method takes advantage of the strengths of different techniques, such as the combination of the homotopy technique with the Newton-Raphson method, Fast Decoupled Load Flow and techniques based on the Continuous Newton's methodology. This combination ensures that the solver can effectively obtain the solution for large systems, even in ill-conditioned conditions, maintaining stability and accuracy in the solutions.

CHAPTER 7

TESTS AND RESULTS

7.1 INTRODUCTION

In this chapter, the methods introduced in Chapter 6 are assessed regarding efficiency and impact on power flow solutions. These approaches are employed to compute the power flow solutions for large-scale realistic systems, with a particular focus on initializing with a flat start estimate to evaluate the robustness of the techniques. The objective is to conduct simulations and compare the results obtained from various methodologies, including the standard Newton-Raphson (NR) method, an optimized NR variant with improved step calculation (OM-NR), and the Heun King-Werner method (HKW) (TOSTADO-VÉLIZ *et al.*, 2020a) along with its two variants (HKW1 and HKW2) (OLIVEIRA; FREITAS, 2021).

All the analyzed techniques were implemented using the MATPOWER v6.0 application (ZIMMERMAN *et al.*, 2011). This program uses the NR method to solve the PFP based on network databases encoded in '.m' files, known as *cases*. These files contain data of buses, branches, and generation, as required by professional computational programs used by electric utility companies and research institutions. The user can also provide an initial estimate for solving the PFP in the bus data. All the cases presented in the MATPOWER database are convergent. However, this convergence is only assured because the initial estimate for the NR is assigned (data of native files) very close to the solution, thus explaining convergence in a few iterations for MATPOWER's data cases. Therefore, this is the standard result used for comparison when other techniques and different operating points are adopted in this work.

All simulations were conducted on a computer with an Intel® Core™ i5-9300H processor, 2.40 GHz (64-bit), and 8 GB of RAM. To estimate CPU time for evaluating the performance of the methods and minimize the influence of other computational activities, each test on each system was repeated 1,000 times, and the average solution times were calculated. A single convergence

tolerance of $\epsilon = 10^{-8}$ pu was used for active and reactive power mismatches. A maximum of 50 iterations ($iter_{max}$) was set for the convergence of the methods. All initial estimates were initialized using a flat start guess, except the reference cases from the native MATPOWER's databank.

7.2 SYSTEMS ANALYZED

In order to validate the proposed methodologies, the following large realistic systems were considered:

- *case3012wp:* 3012-bus of the Polish Transmission System during winter 2007-2008 evening peak conditions (ZIMMERMAN; MURILLO-SÁNCHEZ, 2020);
- *case3375wp*: 3374-bus of the Polish Transmission System during winter 2007-2008 evening peak conditions (ZIMMERMAN; MURILLO-SÁNCHEZ, 2020);
- *case13659pegase:* 13659-bus portion of the European high voltage transmission network (JOSZ *et al.*, 2016; FLISCOUNAKIS *et al.*, 2013; ZIMMERMAN; MURILLO-SÁNCHEZ, 2020);
- *case_ACTIVSg70k:* Synthetic Eastern US 70000-bus power system model(BIRCHFIELD *et al.*, 2016; BIRCHFIELD *et al.*, 2018; ZIMMERMAN; MURILLO-SÁNCHEZ, 2020); and
- *case109k:* 109,272-bus system synthetic model (VéLIZ; JURADO, 2019).

Several experiments with the different systems were performed. In this proposal, priority was given to presenting experiments considering the base cases of the test systems.

The experiments focus on tests considering the flat start guess for iterative PFP. Therefore, it is important to show how the simulations are performed in practical situations, mainly in relation to the starting point of iterations. Our simulations are carried out by using the MATPOWER tool. For the analyses, test systems with Brazilian networks were not included, as information from the MATPOWER database was used, which does not yet contain data on Brazilian systems. The database of this framework, including the cases for the five test systems

used in simulations in this work, depart from an initial guess $\mathbf{x}^{(0)}$. This information is native to the cases and can be modified by the user. We use the two forms.

When using MATPOWER's 'native' initialization, we consider it the reference because it always presents convergent results that are considered correct. Then, the results obtained this way are used for comparison purposes with the results we have computed. However, for better evaluation, aiming at the robustness of the method, we always assume a flat start type of initialization.

7.2.1 Characteristics of the initial estimate for MATPOWER

Before performing the simulations on the test systems, we check how close the initial estimate given in MATPOWER is to the final computed states of the system.

Figs. 7.1 and 7.2 show the initial estimates and the PFP solution in plots, respectively, for the bus nodal voltage magnitudes and angles for the 70,000-bus model. Without such an initial estimate, divergence from the iterative process occurs when using the classical NR method to determine the PFP solution.

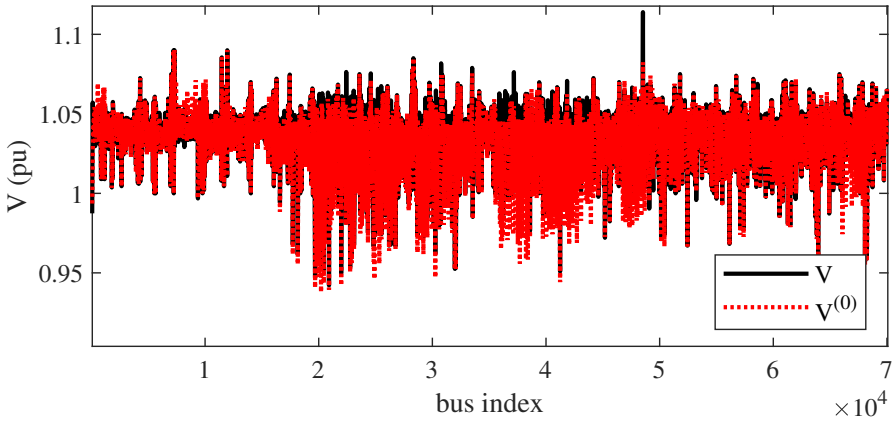


Figure 7.1. Plots illustrating the Initial estimation voltage magnitude, $\mathbf{V}^{(0)}$, used by MATPOWER and final states, $\mathbf{V}$, for the 70,000-bus system.

The plots exhibit the closeness between the initially assigned values (red curve) and the final results of the states (black curve). Conversely, one can observe a significant disparity when a flat initial guess is used, especially with regard to the phase angles. In such a scenario, all phase angles would be zero, resulting in the red curve becoming a straight line with a zero value.

Besides the remarked characteristics on the initial estimations used in the MATPOWER, in

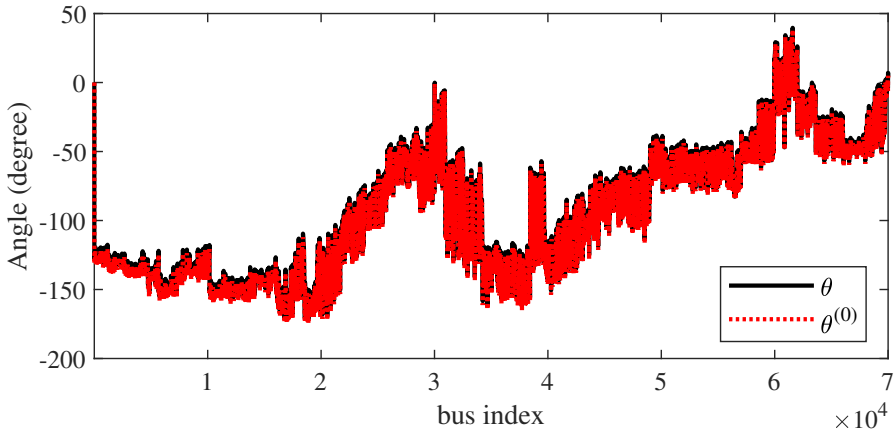


Figure 7.2. Plots illustrating the Initial estimation voltage phase angle, $\theta^{(0)}$, used by MATPOWER and final states, θ, for the 70,000-bus system.

the subsequent subsection, a study is assessed to highlight the problem of ill-conditioning when treated for the Jacobian matrix for the initial iterate of the power flow.

7.2.2 Condition number and changing with shift-δ

This subsection presents a study concerning the condition number of the Jacobian matrix of the initial iteration of the power flow, defined by $\mathbf{J}_0 = \mathbf{J}(\mathbf{x}^{(0)})$, when it is calculated by the initial estimate $\mathbf{x}^{(0)}$. Simulations are performed to verify the changes of the condition number of this matrix, κ, for various conditions imposed by a shift-δ term introduced on its diagonal. This way, for a given $\delta \geq 0$, the matrix after a shift in the initial iteration was handled for simulations as $\mathbf{J}_\delta = \mathbf{J}_0 + \delta\mathbf{I}$. A first experiment was carried out varying δ in the range 0 to 0.1 in steps of 0.005 and calculated the condition number $\kappa(\mathbf{J}_\delta)$ of $\mathbf{J}_\delta$. The spectral condition number (see (GOLUB; LOAN, 1996) for details) $\kappa = \frac{\sigma_{max}}{\sigma_{min}}$, where σ_{max} and σ_{min} are, respectively, the maximum and minimum singular values of $\mathbf{J}_\delta$ are calculated. The computations were performed for the *case13659pegase* (13k) and *case_ACTIVSg70k* (70k). Figure 7.3 shows plots of κ for these systems.

We can observe a similarity between the plots of the condition number of the matrix $\mathbf{J}_\delta$ with the L-curve in the Tikhonov's regularization, which will be seen later. As in the L-curve, there is a region where small variations of κ are detected. This region can be identified where adequate values for δ can be chosen as a perturbation parameter to improve the condition number of the original matrix, and so to solve the first iteration of the PFP. This first iteration can be crucial

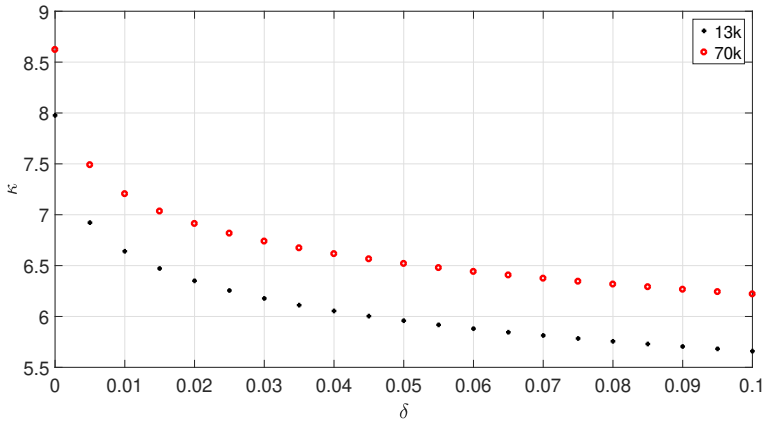


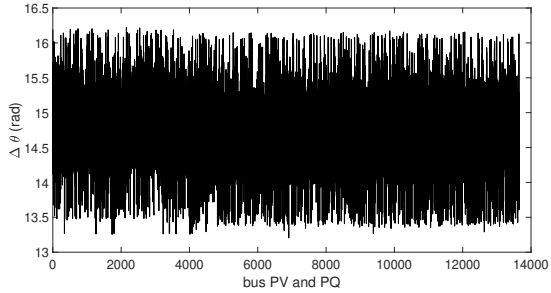
Figure 7.3. Plots illustrating the condition number (ordinate axis in logarithmic scale) of the modified matrix $\mathbf{J}_\delta$ as a function of the perturbation parameter δ.

for resolving the PFP, and the experiments will later demonstrate this. Then, considering these remarks, other experiments were conducted to verify the computations of the states just for the initial interaction. However, considering only one selected value of δ from the plots. The criterion for the selection was the identification of small changes in κ in the range of δ, as explained before.

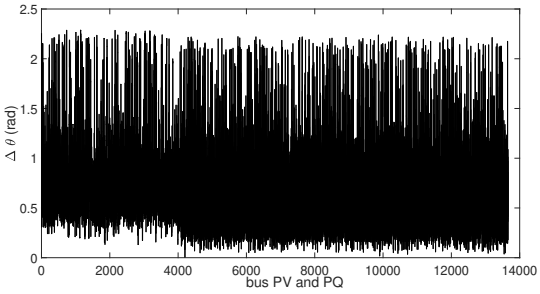
Again, studies were assessed for the *case13659pegase* and *case_ACTIVSg70k*. The results were obtained for the case $\delta = 0$, *i.e.*, the original Jacobian without perturbation; and for the case $\delta = 0.01$, from the plot visualization, $\kappa \approx 10^{6.6} = 4 \times 10^6$ in the *case13659pegase*; and $\kappa \approx 10^{7.2} = 1.6 \times 10^7$ for *case_ACTIVSg70k*. Despite the elevated values for κ, it is emphasized that without perturbation, these values achieve 9.6×10^7 and 4×10^8 for the *case13659pegase* and *case_ACTIVSg70k*, respectively. Therefore, considering the simulations, the interpretation of the condition number results indicates that we need to relatively see the condition number for these large-scale system matrices. Only the absolute value of κ is not sufficient to conclude on the ill-conditioning of a system matrix.

Therefore, considering the *case13659pegase*, from the plots in Figure 7.4, significative values for the angle deviations are verified when $\delta = 0$ (no perturbation) and Figure 7.4a and a deviation that it is close to the final states as shown in Figure 7.4b. Concerning to the voltage magnitude, significative deviations in relation to the perturbed and undisturbed cases are not observed.

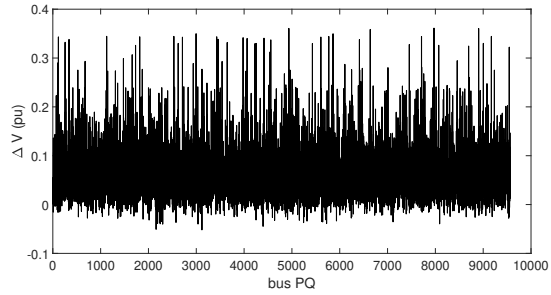
Similar simulations were performed for the *case_ACTIVSg70k*, assuming the same value δ for the perturbed condition. Figure 7.5 depicts the plots for this system.



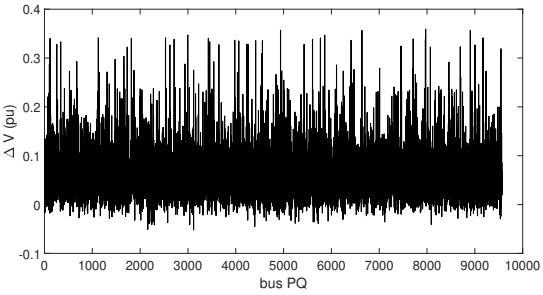
(a) Deviations in θ for $\delta = 0$



(b) Deviations in θ for $\delta = 0.01$

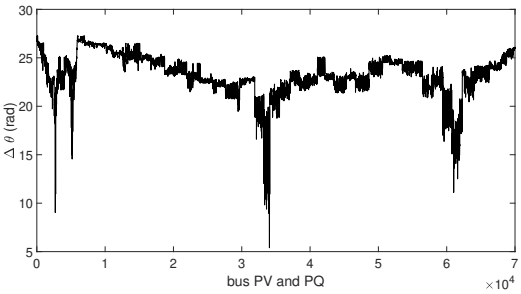


(c) Deviations in V for $\delta = 0$

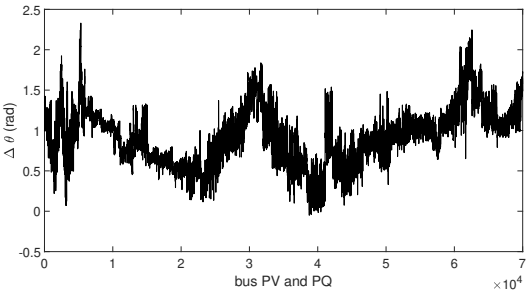


(d) Deviations in V for $\delta = 0.01$

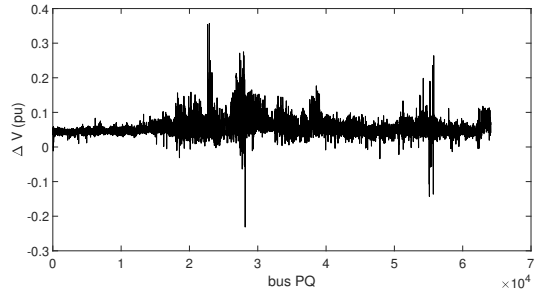
Figure 7.4. Deviations in the states (voltage angle, θ, and magnitude, V) computed in the *case13659pegase* for the initial iteration for θ (a) and (b); and Voltages (c) and (d).



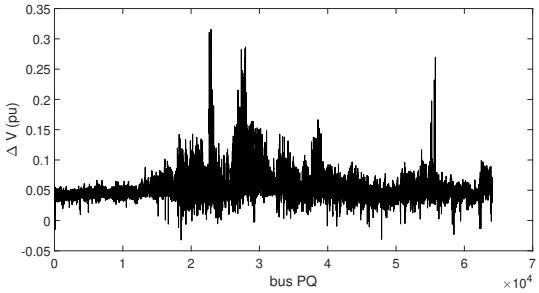
(a) Deviations in θ for $\delta = 0$



(b) Deviations in θ for $\delta = 0.01$



(c) Deviations in V for $\delta = 0$



(d) Deviations in V for $\delta = 0.01$

Figure 7.5. Deviations in the states computed in the *case_ACTIVSg70k* for the initial iteration for θ (a) and (b); and Voltages (c) and (d).

Again, similar behavior of the results observed for *case13659pegase* was also confirmed for the *case_ACTIVSg70k*. An elevated change in the deviations of voltage angles is verified in the

case without perturbation and small differences when the analysis is for voltage magnitude.

7.2.3 Tests considering the classical NR with initialization by the Gauss-Seidel method or with an optimal multiplier

This subsection discusses simulations and results assessed with the classical NR solver. The four test systems were used for experiments with the NR method for different initializations. Tests were performed considering the estimate presented in MATPOWER (ZIMMERMAN; MURILLO-SÁNCHEZ, 2020) (MAT-NR), besides flat start (flat-NR) and Gauss-Seidel (GS-NR). In the latter experiment, the Gauss–Seidel's method was investigated similarly as the homotopy approach for the first step in the scheme of Fig. 6.1 to generate a partial solution $\mathbf{x}_h$ and providing this result as the initial guess for the standard NR, used as IM. Two iterations of the GS method were calculated for the PFP departing from the flat start estimation. Additional experiments with more than two iterates for the GS were also investigated. Finally, the last approach, considering an estimate with a flat start but evaluated for the Optimal Multiplier (OM-NR) solver (see [2.49]), was studied. Tab. 7.1 depicts a summary iterate number for convergence demanded for each technique, except for the cases using flat-NR because the NR solver does not converge for these experiments. The divergent PFP results for the flat-NR cases are in accordance with those reported in (TOSTADO-VÉLIZ *et al.*, 2020a) and references therein.

Table 7.1. Iterations for the solver GS-NR, OM-NR and MAT-NR.

	Solver		
System	**MAT-NR**	**OM-NR**	**GS-NR**
case3012wp	3	20	3
case3375wp	4	X	4
case13659	5	16*	6*
case_ACTIVSg70k	6	X	X

* other point; X: divergent.

In Tab. 7.1, the results in the second column show that despite the initialization provided in MATPOWER, at least 3 iterates are required for the smallest system and 6 for the largest. In conjunction with an OM adjusted as suggested in (PAN *et al.*, 2019), the NR solver presents convergence for the correct operation point for the *case3012wp*, as exhibited in the third column. Also, it presents convergence for the *case13659pegase*. However, the operation point has a

voltage angular difference greater than 90^o for some interconnections. Tests for the *case3375wp* and *case_ACTIVSg70k* diverge. We verify from the NR-OM solver experiments that when the PFP converges, the OM is approximately constant. Conversely, when divergence occurs, the OM is reduced to values close to zero. Even in the event of convergence, it can be verified for an unstable operation point with an angular difference above 90 degrees in interconnections.

The Gauss–Seidel solver (GS-NR) presents convergence for the cases *case3012wp*, *case3375wp*, and *case13659pegase*. Even so, the latter converges for an unstable operation point (voltage angular difference greater than 90^o). The *case_ACTIVSg70k* diverges, and additional iterations for the GS as the first step in Fig. 6.1 cause a detrimental impact on the results to initialize the NR method. Therefore, considering the four test systems investigated, using the GS technique to provide a partial solution $\mathbf{x}_h$ is not recommended for the investigated test systems.

The following simulations were assessed considering the perturbation approaches and the two-step homotopy-based method.

7.2.4 Simulations considering the HKW method with flat start estimation

This subsection considers experiments involving the HKW's method and its variants.

The methods HKW, HKW1 and HKW2 involve several parameters that in (OLIVEIRA; FREITAS, 2021) were configured according to the complexity of the analyzed system. The initial parameters of the HKW method (universal parameters) and its approaches were defined as $h_{min} = 0.4$, $h_{max} = 1$, $\mu = 0.06$, $\alpha = 500$, $\psi^{(0)} = 1$ and $\psi_{max} = 1.9$ as suggested in (TOSTADO-VÉLIZ *et al.*, 2020a).

Tests were performed for the four test systems assuming the universal parameter values. However, this setting was not able to obtain the PFP solution for all systems, particularly for *case13659pegase* and *case_ACTIVSg70k*. Then, empirical adjustments of the terms needed to be performed until convergence was reached. This task is not trivial, because it can demand hours to obtain a successful parameter setting. Tab. 7.2 shows the results reproduced for the experiments developed in (OLIVEIRA; FREITAS, 2021) for the two largest power system models.

From Tab. 7.2, it is possible to verify the considerable difference values between the universal

Table 7.2. Parameters used in the simulations for the HKW method and variants as adjusted in (OLIVEIRA; FREITAS, 2021).

	case13659			*case_ACTIVSg70k*		
Parameters	**HKW**	**HKW1**	**HKW2**	**HKW**	**HKW1**	**HKW2**
h_{min}	0.3	0.3	0.3	0.1	0.3	0.3
$\psi^{(0)}$	1.0	1.0	1.0	1.22	1.20	1.15
ψ_{max}	1.6	1.6	2.0	1.6	2.0	1.5

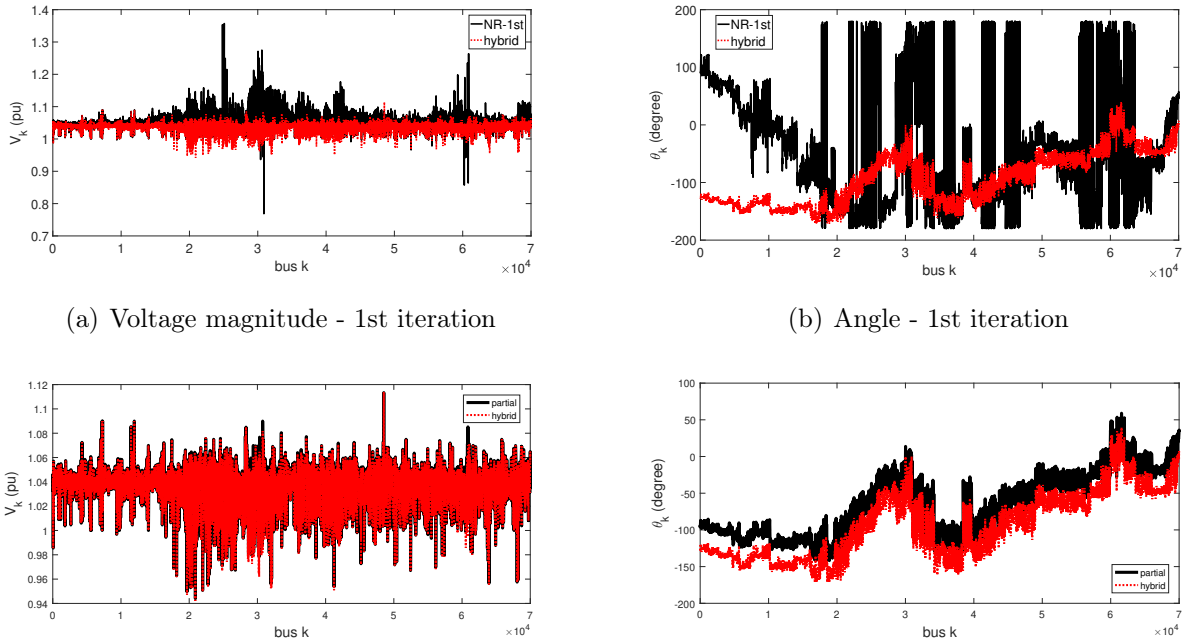
parameters and those adjusted for the most significant models. The gap is most sensitive for the *case_ACTIVSg70k*. The solvers demonstrated limitations for large-scale systems, as verified in the latter two cases. Also, any significant deviation (e.g., 5%) on one of the parameters causes divergence of the iterative process. These aspects have motivated the experiments with the proposed technique as performed in the following simulations.

7.3 SIMULATIONS CONSIDERING THE TWO-STEP HYBRID METHOD

The two-step hybrid-based method was applied to solve the PFP for the four ill-conditioned test systems, according to the *case3012wp*, *case3375wp*, *case13659pegase*, and *case_ACTIVSg70k*. Initially, the problem was partially solved through the homotopy method (first stage) as proposed in (FREITAS; SILVA, 2022), considering a flat start estimate, to determine the partial solution, $\mathbf{x}_h$, and assuming convergence tolerance $\epsilon_h = 2.0$ pu for the power mismatch. In the second stage, the value $\mathbf{x}_h$ was used as an initial estimate to solve the PFP through an iterative method (IM stage). However, using tolerance $\epsilon = 10^{-8}$ pu. The iterative methods used in the second stage were the classical NR (H-NR), the fast decoupled NR method considering the XB version (FDXB) (H-FNR) (ZIMMERMAN *et al.*, 2011), the HKW (H-HKW), and two versions HKW1 (H-HKW1) and HKW2 (H-HKW2). Following (FREITAS; SILVA, 2022), the full homotopy method for convergence tolerance 10^{-8} pu for each point of the homotopy curve was also run. This procedure was labeled full-H. The relaxed homotopy-based procedure for mismatch 2.0 pu was called partial-H. These two latter simulations were performed to also demonstrate the contribution to the partial-H for an estimate and low computational cost in the two-step homotopy-based method compared to the full-H.

A first set of experiments was performed to illustrate the high discrepancies in using the NR

solver with flat start initialization (flat-NR) and the result (partial) provided by the first step in the two-step homotopy-based solver. The accurate solution obtained by the two-step homotopy-based method (hybrid) was computed to serve as a reference and compare partial results. The simulations were assessed just to the largest model. The ones with the homotopy method were performed with $N = 8$ and $\delta = 1$.



(a) Voltage magnitude - 1st iteration

(b) Angle - 1st iteration

(c) Voltage magnitude - partial-H

(d) Angle - partial-H

Figure 7.6. Voltages for the 70k-bus system at the end of the first iteration for the traditional NR method using flat start and for partial-H (state $\mathbf{x}_h$): magnitude for 1st iteration (a), angle for 1st iteration (b), magnitude of $\mathbf{x}_h$ (c) and angle of $\mathbf{x}_h$ (d).

Fig. 7.6 exhibits the results of voltages for the 70k-bus for different situations. Figs. 7.6.a and 7.6.b refer to simulations with the flat-NR solver (black) and the accurate solution obtained by the two-step homotopy-based solver (red). Just plots for the voltage magnitudes and phase angles of the first iteration for the flat-NR solver are shown in comparison with the same quantities of the accurate result. Because, already for the first iteration of the flat-NR solver, an elevated difference occurs among the respective values for the accurate result. Furthermore, the ill-conditioning of the system generated very high deviations for the states to be used in the second iteration. Remark that there is even a growth in the opposite direction in the case of the phase angles in relation to the accurate result. The result confirms the previously mentioned inefficiency of the flat-NR solver.

Meanwhile, Figs. 7.6.c and 7.6.d refer to simulations with the partial (black) and two-step

homotopy-based accurate result (red). Although it has used a poor tolerance mismatch for convergence, $\epsilon_h = 2.0$ pu, the partial solution $\mathbf{x}_h$ (black) presents visual effects closer to the accurate solution (red). Therefore, this characteristic qualifies as favorable conditions for starting an IM and having convergence for the precise solution. This way, the hybrid technique's first-step can be viewed as a virtual enabler for improving the original attraction convergence region for the flat start estimate. However, this facilitator does not exist for the flat-NR solver since it starts directly from the flat start guess.

Table 7.3. Iteration number required for the base loading case of the systems.

Method	*3012wp*	*3375wp*	*13659*	*70k*
HKW	8	8	8	15
HKW1	8	8	9	15
HKW2	22	24	38	32**
H-NR	3	3	3	4
H-FNR	9P/9Q	9P/9Q	14P/13Q	16P/16Q
H-HKW	6	6	6	7
H-HKW1	6	6	6	7
H-HKW2	15	15	17	7
full-H	32	32	33	41
partial-H	8	8	9	8

** Jacobian matrix was intentionally frozen from the 7th iteration.

The set of experiments in the sequence has considered the method performance concerning the algorithm convergence merit. Another objective is to use this merit index and show the computational burden of the PFP with the homotopy method for their accurate and inaccurate applications. Tab. 7.3 exhibits convergence iterations verified for some techniques and the four system models. Second to fourth rows in the table shows results of the HKW's approaches without a first stage of initialization as proposed for the hybrid method (*i.e.*, the computations for solving the PFP are performed directly from a flat start estimation). Fifth to ninth rows present results obtained by using the second stage methods of the two-step homotopy-based method. For the solver H-FNR, the inclusion of the notation P/Q represents the iterations needed for convergence for the loops P and Q of the method FDXB. The tenth row depicts the total amount of iterations required to determine the complete curve of homotopy and consequently the converged PFP results (computation of the accurate result $\mathbf{x}_*$). All the iterations using homotopy are calculated through the classical NR solver. Finally, the eleventh row exhibits only results of the first stage of two-step hybrid method (computation of the inaccurate result

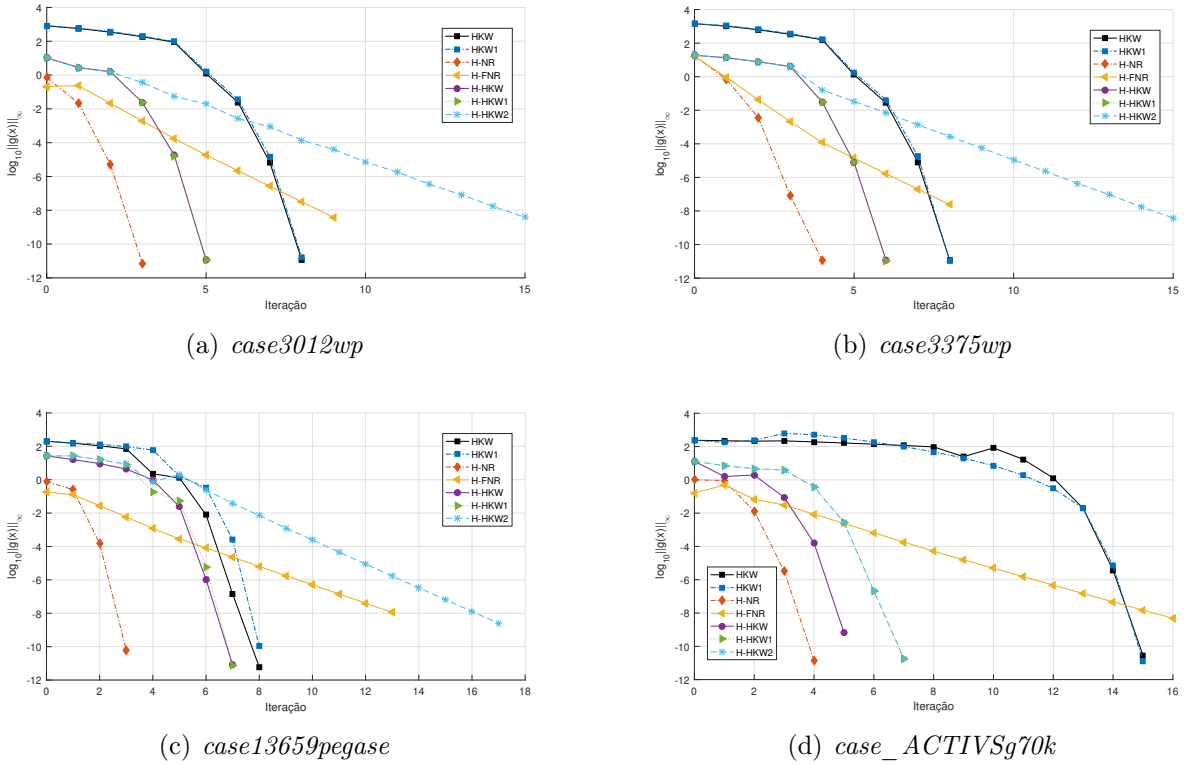
$\mathbf{x}_h$ by the homotopy method). The simulations for the homotopy approaches were assessed with $N = 8$. For all systems, $\delta = 1$ was adopted, except for the system *case13659pegase*, whose setting was $\delta = 1/8$.

The approach HKW and its version HKW1 presented similar number of iterations for the three smaller systems, even when the hybrid method was activated. However, for the largest system the introduction of the first stage of the hybrid method contributed to significant reduction from 15 to 7 iterations for the hybrid method. The HKW2 approach presented the highest iterations for all systems. Conversely, for this solver the highest computational cost occurs for the first iteration for the three smaller systems and the seven initial iterations for the largest model. However, when an initial estimate obtained by the hybrid method is used, the number of iterations of the HKW2 solver are drastically reduced. Extraordinary results are obtained using the NR and the FDXB method with initial guess using the result of the first stage of the hybrid method. These solvers present divergent results for all systems when flat start initialization is directly used. Not only the two techniques are convergent with the hybrid approach, but the NR method has the lowest number of iterations among all the tested solvers.

From Tab. 7.3, we can conclude that the accurate resolution of the PFP through the homotopy method (full-H) requires an elevated number of iterations. However, when the method is used to determine a partial result (partial-H), the number of iterations is significantly reduced (8 to 9 iterations of type NR for any size of tested system).

Fig. 7.7 shows plots of power mismatch norm for several solvers for the four test systems. The curves reinforce the high-performance of the standard NR solver when it is used in the two-step hybrid method, already detected in the results of simulations exhibited in Tab. 7.3.

The computational burden was also measured assuming the simulations verified in Tab. 7.3. The CPU time was measured by performing 1000 repetitions for the same calculation in order to avoid the influence of parallel tasks on the results. In this way, an average execution time value was considered. Based on this information, the average CPU times were taken for the four test systems and a specific method studied. Tab. 7.4 exhibits the average CPU time demanded for each solver. In the table's average CPU time exhibited for the two-step hybrid methods, the execution time required for the assessment of the first step (partial-H) is excluded (*i.e.*, the measure is only for the second step in the solver).



(a) *case3012wp* (b) *case3375wp*

(c) *case13659pegase* (d) *case_ACTIVSg70k*

Figure 7.7. Characteristic of convergence as a function of the number of iterations for *case3012wp* (a), *case3375wp*(b), *case13659pegase* (c) and *case_ACTIVSg70k* (d).

Table 7.4. Mean execution time in seconds for different approaches and systems.

Method	*3012wp*	*3375wp*	*13659pegase*	*70k*
HKW	0.3160	0.3640	1.5780	19.7300
HKW1	0.2297	0.2625	1.0250	13.2900
HKW2	0.0750	0.0830	0.6220	6.9900
H-NR	0.0724	0.0827	0.4087	3.5450
H-FNR	0.0321	0.0455	0.2577	1.2960
H-HKW	0.2456	0.2770	1.0760	9.3700
H-HKW1	0.1860	0.2359	0.7781	6.0650
H-HKW2	0.0609	0.0657	0.2380	1.5000
full-H	0.8590	0.9300	3.9400	32.9700
partial-H	0.2700	0.3300	1.2200	6.6500

From Tab. 7.4, it is highlighted the best performance of the solvers H-FNR (FDXB method with initialization by the innacurate homotopy solver) and H-HKW2 (HKW2 used in the hybrid approach). The result shows the importance of the initial estimate provided to the methods, exemplifying that even methods neglected due to their inherent simplifications are capable of converging. The classical NR (H-NR) solver also presented a low computational time requirement. However, even presenting a much smaller number of iterations than the H-FNR, it needed

a higher computational cost. The highest computational cost was required from the full homotopy solver (full-H), demonstrating that its straightforward application is not recommended in the PFP, particularly for applications that involve repetitive power flow calculations, such as contingency analysis and stability studies. However, when combined with an IM it is powerful to improve the calculation performance even for a very simple solver as the FDXB. Conversely, the partial homotopy-based solution for the largest model is determined with approximately 20% of the computational cost of the full-H solver.

7.3.1 Influence of the initial estimate on the loading level

In this section, the load level of the studied systems was incremented to evaluate the behavior of the proposed technique concerning the critical system loading. In the considered test scenario, a scaling factor λ was used to modify uniformly the active and reactive powers injected into the bus i through $P_i^{sp} = \lambda P_{i0}^{sp}$ to PQ and PV buses and $Q_i^{sp} = \lambda Q_{i0}^{sp}$ for PQ buses, where P_{i0}^{sp} and Q_{i0}^{sp} are the net and constant contribution of active and reactive power studied in the load base case, respectively. The loading factor was also applied for the active and reactive parts of constant impedance loads in the same bus i. The critical level λ was considered the maximum value at which the H-NR solver diverges. *i.e.*, the system loading λ was progressively incremented until this limit was reached. Then, the immediately lower value of λ was considered for the simulations. The same estimates used for the base case were assigned for the loading level cases as well. The results in terms of iteration number for each method are reported in Table 7.5. This table presents similar information to that of Tab. 7.3 aside from the last two rows (homotopy iterations) that were removed.

From Tab. 7.5, we conclude that just the approaches with a two-step homotopy-based solver presented convergence for the critical loading level considering a studied system. However, two of them had failed to converge. The fast-decoupled version of the NR had convergence only for the most significant model, while the simplest version of the HKW, the H-HKW2, diverged for all systems. Furthermore, the HKW method departed from a flat start estimate, and its versions could not achieve convergence in all tests. However, when using the result of the first stage of the two-step hybrid method, convergence occurs for the HKW and HKW1 approaches. Another observation regarding these convergent approaches is that they converge with a higher

Table 7.5. Total iterations for the critical loading level considering the four test systems.

Method	*3012wp* $\lambda = 2.358$	*3375wp* $\lambda = 2.462$	*13659* $\lambda = 1.061$	*70k* $\lambda = 1.011$
HKW	11	10	12	X
HKW1	11	10	X	15
HKW2	X	X	X	34**
H-NR	7	7	9	4
H-FNR	X	X	X	17P/17Q
H-HKW	9	9	11	6
H-HKW1	9	9	11	7
H-HKW2	X	X	X	19**

** Jacobian matrix frozen from the 7th iteration X - divergence.

number of iterations than the NR method. One justification for this difference in favor of NR is that the HKW and HKW1 methods depend on the parameter adjustment, which in the simulation in question was kept as those from the base load case. Therefore, the NR method kept the best performance in the critical load situation, as was also the verification in the base case.

7.3.2 Influence of the reactive power operational limits in generators

In this section, studies considering the reactive operational limits of generators were assessed. The load base case was the one investigated considering the NR solver.

The operation point concerning the limits were evaluated interactively based on the classical methodology as adopted in (ZIMMERMAN *et al.*, 2011), as follows. Initially, the base case solution, without constrains, was taken and the generators that had their limits violated were identified. Once these generators were known, their associated PV buses were converted to PQ and then the PFP was re-run. From the results, the situations of operational limits of the generators were again evaluated. Continuing in this way, we reached the end of the process with the final diagnosis of violations and the new states to deal with this situation. It should be emphasized that each PFP with operational limits was executed always taking the result of the previous PFP for the initial estimate of the next one.

Only test results with the largest systems (*i.e.*, *case13659pegase* and *case_ACTIVSg70k*) were considered, as these are those that present greater complexity in obtaining the states in the

base case. The reactive operational limits assigned at the MATPOWER's data bank were preserved in the simulations. Regarding the 13659-bus system, it had only 1 limit violated when verifying the result of the base case. Therefore, running a new simulation with the respective bus converted to PQ, success was achieved in the convergence of the PFP and in the control of all generator reactive limits. The treatment dedicated to the 70k-bus system was more complex, because it presented many more PV buses with limit violations. The search for the solution at each iteration was conducted as follows: 4,588 violated limits (2,190 upper and 2,398 lower) were identified for the first iteration. In the second iteration, a total of 700 violated limits (422 upper and 278 lower) were detected. For the third iteration, the violations were reduced, but 200 limits (143 upper and 57 lower) were still out of allowable ranges. The violations were reduced to 11 in the fourth iteration, and finally, in the fifth iteration no more limits were exceeded.

As a consequence of the rearrangement of the operational limits, the generation in the slack bus was increased to accommodate the new operational situation. In addition, the NR method was sufficient to compute all solutions, demanding at most three iterations per limit iterate calculation.

7.4 TESTS CONSIDERING THE CONDITIONING STEP APPROACHES

This section presents the results obtained from applying the conditioning step (CS) approach to solve the PFP for the ill-conditioned systems *case3012wp*, *case3375wp*, *case13659pegase*, and *case_ACTIVSg70k*. To utilize the conditioning step procedure, restricted perturbations were chosen according to (6.4), and simulations were conducted considering three values for δ: 0.001, 0.01, and 0.05. Note that these values are limited by ρ, a user-defined parameter, which is a measure of the state deviation on using CS; and by the condition number of $\mathbf{J}(\mathbf{x}^{(0)})$ and the norm of this matrix, characterizing the dependence of a system. Then, executing the iteration from the guess $\mathbf{x}_{\delta}^{(0)}$ obtained from the CS, simulations considering the NR and its version NR1J based on the freezing of the Jacobian matrix in the 1st iteration, HKW, HKW1, and HKW2 were carried out with the universal parameters suggested in (TOSTADO-VÉLIZ *et al.*, 2020a). They were called CS-NR, CS-NR1J, CS-HKW, CS-HKW1, and CS-HKW2, respectively.

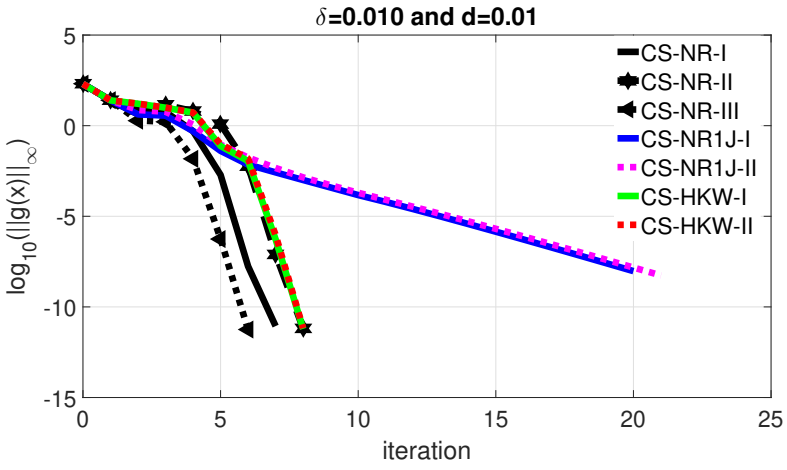
The results for Option II are shown in Tab. 7.6 in terms of iterations for convergence for each system. But, similar results for the Options I and III are also obtained. Interesting to note that almost all simulations presented convergence in the range of values of δ, including the classical NR solver. The NR method with the conditioning step in the initial estimate presented the best performance for all systems. Even the Jacobian frozen version of NR presented convergence for higher values of the parameter δ. The CS-HKW and its variant also had convergence, even using the universal parameters suggested in (TOSTADO-VÉLIZ *et al.*, 2020a). This result demonstrates the robustness of the CS-HKW method when the conditioned initial estimate is used. Additionally, a smaller number of iterations was required for simulation with every test system, mainly concerning the biggest system because the NR solver required only 6 iterations and the CS-HKW solver requested only 7. It should be noted that very small values for δ tend to produce a higher number of iterations for convergence at all. This is justified because in the limit when $\delta \to 0$ the results can be interpreted as $\mathbf{J}(\mathbf{x}^{(0)})\Delta\mathbf{x}^{(0)} = -\mathbf{g}(\mathbf{x}^{(0)})$, which coincides with that one for the first iteration for the traditional NR solver 'bypassing CS'. On the other hand, higher values for δ provide a stabilized number of iterations, demonstrating the effectiveness of the CS strategy, even for ill-conditioned and large-scale systems.

Table 7.6. Iteration number required for the base loading case of the systems.

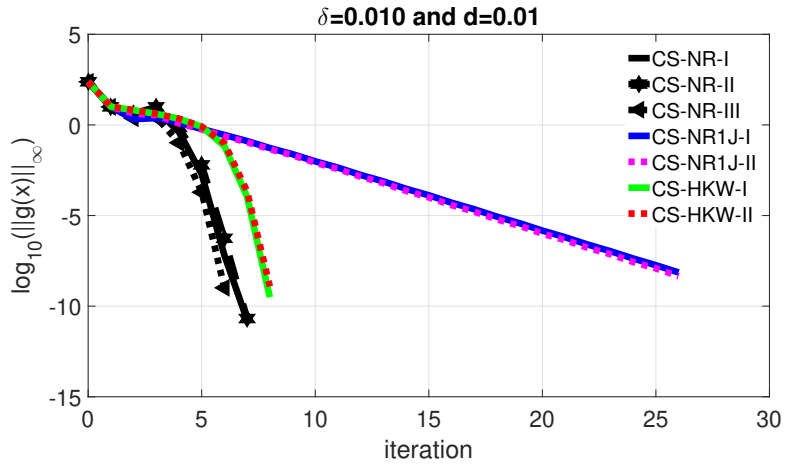
Method	*case3012wp*			*case3375wp*			*case13659pegase*			*case_ACTIVSg70k*		
CS-NR	5	5	5	5	5	4	5	7	7	6	6	6
CS-NR1J	401	22	20	59	21	19	19	20	22	30	25	25
CS-HKW	7	7	7	7	7	7	7	7	8	7	7	7
CS-HKW1	8	7	7	7	7	7	7	7	8	7	7	7
CS-HKW2	326	23	21	51	22	20	22	23	25	28	28	28

Figure 7.8 exhibits plots for simulations considering the three options to implement the perturbation details for the two biggest power system models, according to the options proposed in Section 6.2. The plot for option III was shown just to the NR solver since each other solver had similar results considering this option. The results indicate agreement among the possibilities to implement the amount and form of small perturbation on the ill-conditioned Jacobian matrix. However, when option III is chosen, a slight improvement can be verified.

Figure 7.9 shows the plots of mismatches for simulations involving all test systems and the methods investigated based on the conditioning step of the initial estimate. The value $\delta = 0.01$ was used, and again, Option II was employed. We can observe that the NR method has the best



(a) *case13659pegase*



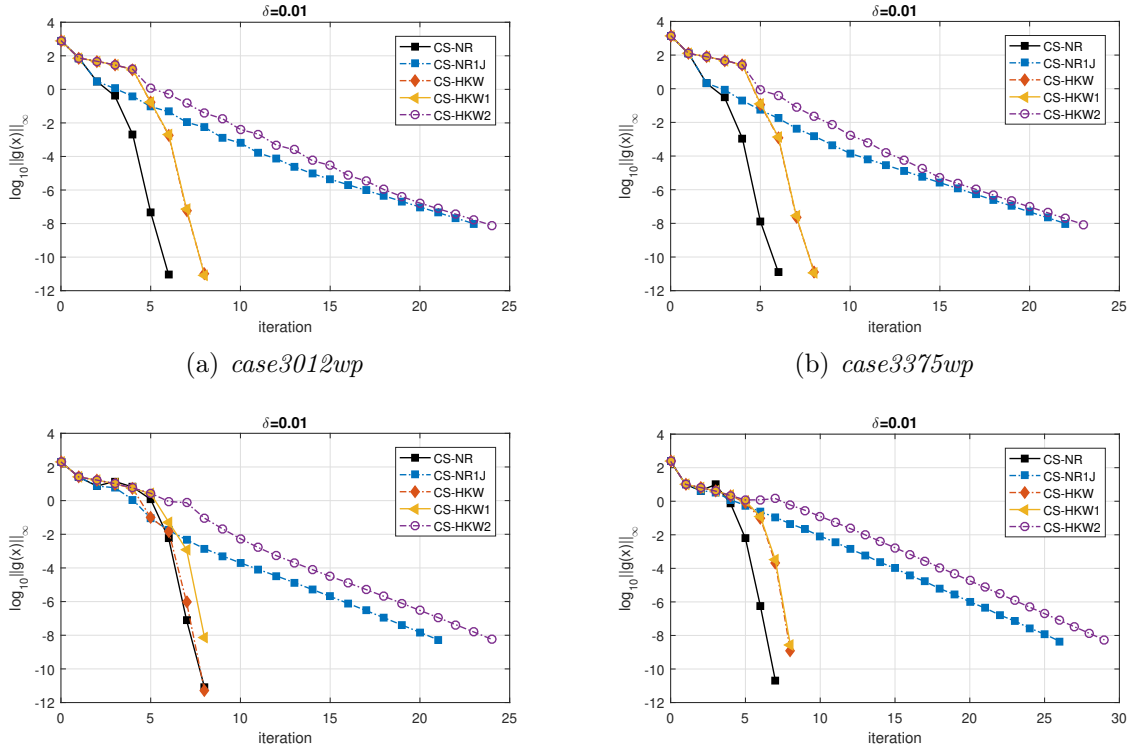
(b) *case_ACTIVSg70k*

Figure 7.8. Convergence characteristics of methods using Options I, II, and III for $\delta = 0.01$ and $d = 0.01$ for Option III.

performance, confirming the results in Tab. 7.6 and demonstrating a high level of convergence quality when compared with the other solvers. Even the frozen version of the NR solver is convergent, despite demanding a higher number of iterations and presenting similar behavior to the HKW2 solver. This behavior would be unthinkable without the use of the CS strategy. Also, the good performance of the HKW and its variant HKW1 is confirmed. However, as they depend on the NR solver and other supplementary computations, the sufficiency of using only the NR when CS is activated can be verified.

Besides their convergence profile, better information about the performance of the methods can be evaluated through the execution time demanded in each case. Then the simulation CPU time was assessed, and the results are exhibited in Tab. 7.7. The computational times for

running the methods were measured by performing 1000 repetitions of the same simulation. Therefore, the results in the table expose the mean CPU time. All simulations assume that the reactive power limits of generators are disregarded. Therefore, the values presented in the fifth line of the table (CS-1^{st}) represent the mean CPU time required only by the conditioning step. This burden given for CS-NR1J is the same time spent in CS-NR, CS-HKW, CS-HKW1, and CS-HKW2 methods. All other computational times refer to the global time to perform CS plus the remaining iterations to reach convergence.



(a) *case3012wp* (b) *case3375wp*

(c) *case13659pegase* (d) *case_ACTIVSg70k*

Figure 7.9. Convergence characteristics of methods using the Option II for implementing the perturbation and $\delta = 0.01$.

From Tab. 7.7, we conclude that the CS procedure provides a condition for using a frozen version of the NR solver, although requiring a higher number of iterations to converge. This conclusion is reinforced considering the frozen version of the CS-HKW solver, CS-HKW2, that, although demanding a high number of iterations, also presents low-cost computational compared to the CS-NR1J. The full methods needed fewer iterations for convergence but at the expense of higher computational consumption. The result is an impressive improvement in investigating iterative methods for computing the PFP solution. Mainly because the simulations involve ill-conditioned and large-scale systems with flat-start initialization. Several works reported failing

Table 7.7. Average computational execution time in seconds of the convergent methods for the PFP.

Method	*case3012wp*	*case3375wp*	*case13659pegase*	*case_ACTIVSg70k*
HKW	0.2308	0.2514	0.9582	12.5188
HKW1	0.1543	0.1831	0.7394	8.2111
HKW2	0.0466	0.0483	0.3497	4.3396
CS-1^{st}	0.0411	0.0471	0.2079	1.4244
CS-NR1J	0.0775	0.0845	0.3444	2.4382
CS-NR	0.1182	0.1225	0.7640	5.1955
CS-HKW	0.2255	0.2547	1.0418	6.7826
CS-HKW1	0.1765	0.1992	0.7990	5.1605
CS-HKW2	0.0867	0.0962	0.5162	2.7471

the classical NR for this type of simulation (TOSTADO-VÉLIZ *et al.*, 2020a), without the conditioning step proposed in this work.

7.5 TESTS CONSIDERING MODAL-BASED JACOBIAN MATRIX PERTURBATION

In this section, experiments were carried out for two case studies in order to demonstrate the performance of the proposed modal-based perturbation technique for PFP. The models include a 13659- and a 70k-bus systems.

The simulations consist in computing the smallest eigenvalues of the Jacobian matrix $\mathbf{J}(\mathbf{x}^{(0)})$ in (6.7) to identify the problematic mode affecting the ill-conditioning and from this result, introducing a perturbation on the Jacobian matrix. The smallest eigenvalue λ_1 of $\mathbf{J}(\mathbf{x}^{(0)})$ and its associated right- and left-eigenvectors were calculated using the `eigs` code available in Matlab's kernel. Similar computation was performed for two other smallest eigenvalues greater than λ_1. Then, the participation factors (KUNDUR, 2007) associated with the modes were determined. Fig. 7.10 illustrates the participation factors for the three modes for the 70k-bus system load-base case, $\lambda_1 = 3.98 \times 10^{-4}$, $\lambda_2 = 2.72 \times 10^{-3}$ and $\lambda_3 = 6.69 \times 10^{-3}$. The zoom region in the plot comprises states ranging from 70k to 130k. This range is formed by the voltage magnitudes states.

The PFs for λ_1 presents high participation of a large number of states. However, the contribution profile is more dominant for some states than for the profile for the modes λ_2 and λ_3. In fact, when we quantify the participation with more than 90%, the mode λ_1 has 7.2% of states from a total of 134,104, while λ_2 and λ_3 have the participation of only 0.089% and 0.44% states,

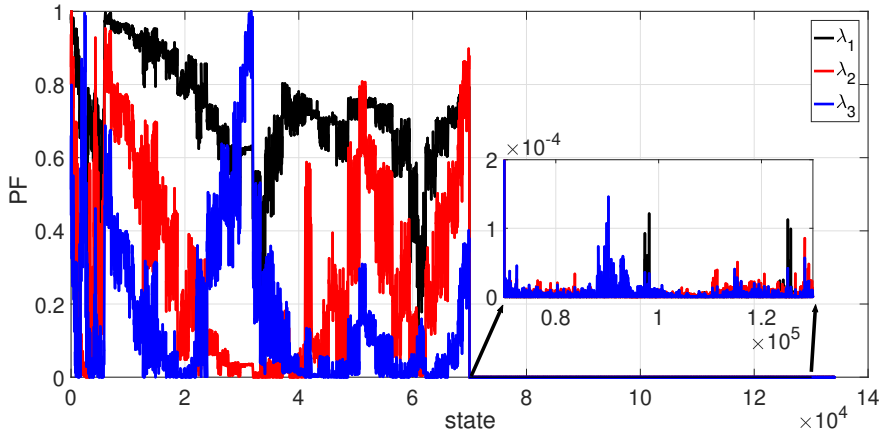


Figure 7.10. Plots illustrating the participation factors of the three smallest modes.

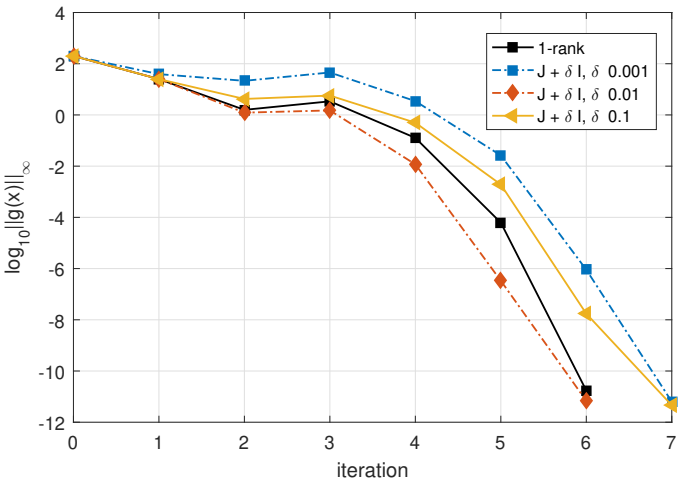
respectively. Also, the contributions are dominant for states characterized by phase angles. The participation of states of type voltage magnitude is neglected for all three modes (see range above the 69,999th index for the state scale in the zooming plots). Therefore, the critical mode for the ill-conditioning is λ_1.

In the sequence, the modal information for the critical mode was explored to solve the PFP. Experiments were performed considering the classical NR (CS-NR) and the fast decoupled NR (CS-FDXB) solvers using the 1-rank perturbation matrix for $\mathbf{J}(\mathbf{x}^{(0)})$. Also, a uniform perturbation at the diagonal of the Jacobian matrix was added by introducing a scaling factor $\delta > 0$. This approach was called here as *shift-*δ. In this case, the perturbed Jacobian matrix was replaced in (6.7) by $\mathbf{J}(\mathbf{x}^{(0)}) + \delta I$, as proposed in (FREITAS; OLIVEIRA, 2023a), where δ acts shifting uniformly all eigenvalues of $\mathbf{J}(\mathbf{x}^{(0)})$. The value δ was assigned for $\delta > \lambda_1$, but in (FREITAS; OLIVEIRA, 2023a) this factor is assigned empirically.

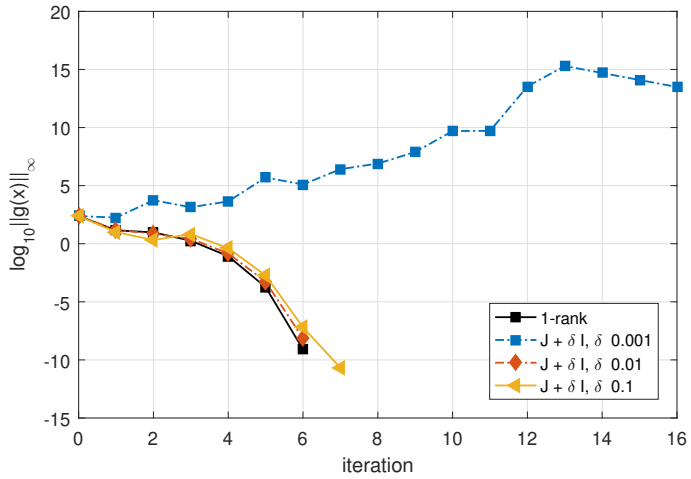
The simulations with the 1-rank modal-based approach were carried out for $\alpha = 100$, *i.e.*, λ_1 was moved to $\lambda_1 \approx 3.98 \times 10^{-2}$. Three values for δ, 0.001, 0.01 and 0.1 were adopted for the shift-δ. Therefore, the computation of λ_1 is applied for both 1-rank modal-based and shift δ methodologies. A tolerance of 10^{-8} was adopted for convergence of power mismatches, while a limit of 20 iterations was established for convergence.

All simulations are performed with flat start estimate and the perturbation term is introduced just for the initial iteration of the NR solver. Both *case13659pegase* and *case_ACTIVSg70k* diverge when they are solved with the classical NR method and its decoupled version without the perturbation term added to the Jacobian matrix. Table 7.8 shows, in 2nd and 3rd columns,

the iterates required for convergence of the simulated cases to the perturbed situations. For the CS-FDXB, the table indications refer to the P- and Q-iteration loops, respectively. When using the 1-rank perturbation matrix for $\mathbf{J}(\mathbf{x}^{(0)})$, the CS-NR as well as the CS-FDXB obtained the PFP solution for all systems and with less iterations. The shift-δ approach demonstrated to be equivalent to the 1-rank perturbation for $\delta = 0.01$. However, for other simulated values it has presented worst performance. A similar conclusion was verified for the CS-FDXB approach. The results indicate the importance of knowing λ_1 for all methodologies investigated, because it is a reference for the set α and δ. Figs. 7.11 show graphs that illustrate convergence characteristics for the CS-NR solver and the perturbation profile in the Tab. 7.8. Again, similar results are verified for the 1-rank and the shift-δ for $\delta = 0.01$.



(a) *case13659pegase*



(b) *case_ACTIVSg70k*

Figure 7.11. Convergence characteristics of the CS-NR methods.

Table 7.9 shows the computational costs demanded by each method with its respective pertur-

Table 7.8. Iteration required for the base loading case of the systems.

Method	*case13659pegase*	*case_ACTIVSg70k*
CS-NR (1-rank)	5	5
CS-NR $(J + \delta I, \delta = 0.001)$	6*	-
CS-NR $(J + \delta I, \delta = 0.01)$	5	5
CS-NR $(J + \delta I, \delta = 0.1)$	6	6
CS-FDXB (1-rank)	17P/16Q	17P/17Q
CS-FDXB $(J + \delta I, \delta = 0.001)$	17P/16Q	18P/18Q
CS-FDXB $(J + \delta I, \delta = 0.01)$	17P/16Q	17P/17Q
CS-FDXB $(J + \delta I, \delta = 0.1)$	17P/16Q	18P/18Q

* other operating point.

bation approach in $\mathbf{J}(\mathbf{x}^{(0)})$, according to simulations for Tab. 7.8. The mean execution time for calculating only one eigenvalue and its right- and left-eigenvectors is 0.2877 s and 2.1545 s for the *case13659pegase* and *case_ACTIVSg70k*, respectively, and they were excluded from measured times in the table. It is observed that the CS-NR method (1-rank) needed smaller computational time for the NR method than for the shift-δ approach. A similar conclusion is verified for the CS-FDXB approach. Furthermore, the CS-FDXB has presented the best performance for the solver requiring approximately only 1/3 of the computational cost of the CS-NR. The results are amazing, not only because of the convergence of the NR method for ill-conditioned and large-scale systems but also because of the excellent performance in terms of computational time and accuracy achieved by the FDXB solver.

Table 7.9. Mean execution time in seconds for the cases in Tab. 7.8.

Method	*case13659pegase*	*case_ACTIVSg70k*
CS-NR (1-rank)	0.4622	3.0370
CS-NR $(J + \delta I, \delta = 0.001)$	0.5594	-
CS-NR $(J + \delta I, \delta = 0.01)$	0.5543	3.3694
CS-NR $(J + \delta I, \delta = 0.1)$	0.5552	3.3751
CS-FDXB (1-rank)	0.1857	1.0756
CS-FDXB $(J + \delta I, \delta = 0.001)$	0.2003	1.1076
CS-FDXB $(J + \delta I, \delta = 0.01)$	0.1952	1.0992
CS-FDXB $(J + \delta I, \delta = 0.1)$	0.1961	1.1049

7.6 TESTS CONSIDERING THE PROPOSED TIKHONOV'S REGULARIZATION APPROACH

In this section, simulations and results are presented considering the regularization parameter method. Firstly, experiments were assessed to determine the test systems' regularization parameters. The parameters were estimated through the traditional L-curve method. Therefore, only the Jacobian matrix and mismatch vector of the first iteration (*i.e.*, calculations assuming flat start estimation) were used in the process.

Fig. 7.12 highlights the L-curve for the largest model system, (*i.e.*, *case109k*). In general, twenty points are calculated to get the plot. The plot shows the L-curve represented by $\eta \times \rho$ (in black) and the regularization parameter curve $\mu \times \rho$ (in red). It can be verified that the corner of L-curve is well characterized. A value range can be identified for the regularized parameter satisfying the corner curve. Therefore, results evidence that several values of μ near a selected μ_0 are still acceptable. Considering this assumption, we estimated the approximate value $\mu = 2 \times 10^{-4}$ for the 109k-bus system.

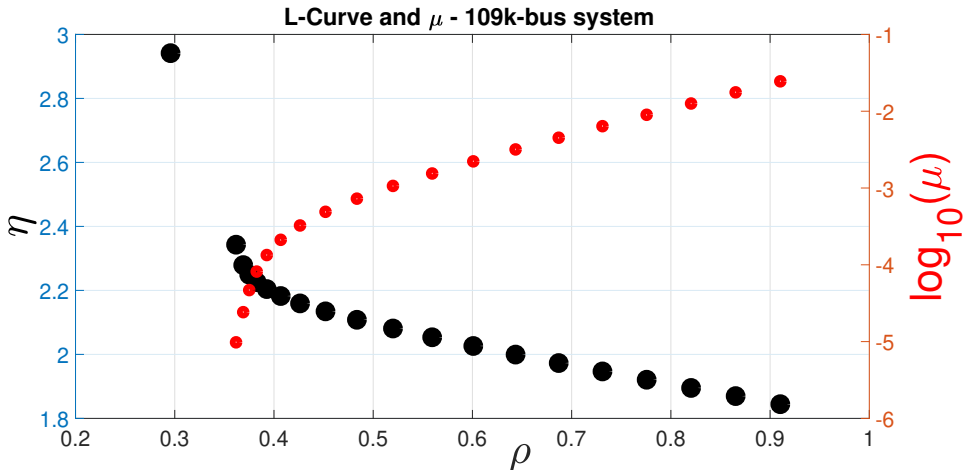


Figure 7.12. L-Curve for the case109k.

The performance of the simulations with the regularized normal equation was evaluated for the computation of the L-curves, because it involves many calculations of linear systems of type (5.32). The computation performance in CPU time just for one specific μ in the L-curve for a given system is exhibited in Tab. 7.10. The three forms for calculations studied in this paper are evaluated. The standard form is for the linear system (5.32). While Implicit I and II are those proposed in (5.36) and (5.39), respectively.

Table 7.10. Average CPU time in seconds for three different forms of computations involving the regularized normal equation.

Method	*3012wp*	*3375wp*	*13659pegase*	*70k*	*109k*
standard	0.079	0.102	0.405	3.65	2.64
Implicit I	0.093	0.109	0.403	2.83	2.78
Implicit II	0.143	0.191	0.648	4.36	4.44

From Tab. 7.8, there is a tendency for the best performance of the implicit I form in relation to the standard. On the other hand, in our experiments, this fact was contradicted by the results presented in tests with the largest system. In this situation, the standard form performed the best. We justify this result considering that the largest system is generated artificially by the interconnection of eight original *case13659pegase* (VéLIZ; JURADO, 2019).

For the remaining systems, the approximated regularization parameters identified from the respective L-curves were 2×10^{-3}, 5×10^{-3}, 1×10^{-4}, and 5×10^{-4} respectively, for the case3012wp, case3375wp, case13659pegase, and case_ACTIVSg70k. It was verified that variations around the selected value of μ do not significantly affect the PFP convergence process in the simulations. Therefore, Tikhonov's method can achieve good performance with an approximate estimation of the regularization parameter derived from the L-curve. Another contribution of the computation of this parameter is that it can be used to estimate the shift-δ parameter adopted in the perturbation method presented in Section 7.4. In the previous section, the parameter δ is chosen empirically. Then, it is proposed to estimate it from Tikhonov's regularization parameter, for example, as $\delta \approx \sqrt{\mu}$.

After determining the regularization parameter, simulations were performed to compare the proposed performance of the method, considering the numerical errors along the iterations with other perturbation approaches. In these situations, the first iteration is executed with the regularization parameter, which is removed from the second onward. To compare with similar methods that adopt a perturbation in the Jacobian matrix only in the first iteration, two techniques were used: the shift-δ technique (FREITAS; OLIVEIRA, 2023a), and a modal approach (OLIVEIRA *et al.*, 2023).

Simulations considering the Tikhonov's regularized PFP, the shift-δ, and modal approaches were performed assuming classical NR solver from the second iteration. All systems were studied using the previously standardized parameters estimated and the shift-δ approach using

$\delta = \sqrt{\mu}$. For the modal approach, it was suggested to use a scaling factor $\alpha = 1/\sqrt{\mu}$ and just one eigenvalue for all systems unless the case109k. For this latter case, moving only one eigenvalue is insufficient to eliminate the cause of ill-conditioning. Then, for this specific experiment, it was proposed to move sixteen eigenvalues. Fig. 7.13 shows the mismatches along the iterations for the 109k-bus system, considering Tikhonov's regularization, shift-δ and modal approaches. The results indicate the approximate equivalence of the three perturbation techniques.

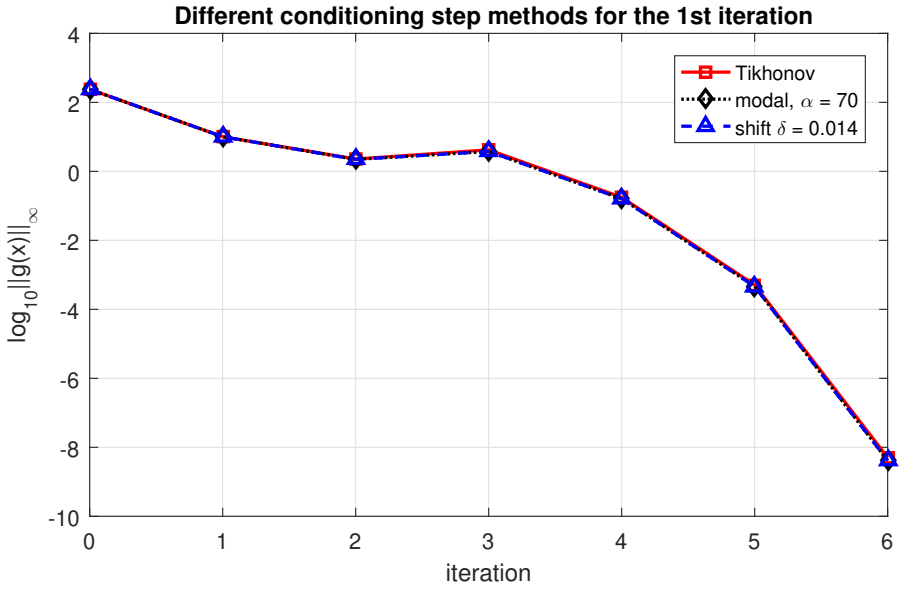


Figure 7.13. Norm for Tikhonov's regularization, shift-δ and 1-rank modal perturbation approaches.

In addition to the classical NR solver, other simulations considered from the second iteration were the methods HKW, HKW1, HKW2, and an approximation for the NR method in which the Jacobian matrix is frozen from the second iteration. So, the methods were labeled T-NR, T-HKW, T-HKW1, T-HKW2, and T-NR1J, respectively. The HKW method and its modified solver used universal parameters (TOSTADO-VÉLIZ *et al.*, 2020a) $h_{min} = 0.4$; $h_{max} = 1$; $\mu = 0.06$; $\psi(1) = 1$; $\psi_l = 1.9$; $\alpha = 500$. All simulations use the first iteration result obtained from Tikhonov's regularized NR method. The test results for the iterations required for convergence, assuming that the regularization parameters determined previously according to the L-curve are shown in the Tab. 7.11. It is possible to verify that all cases are convergent, even for the stringent situation for the solvers T-NR1J and T-HKW2. Tab. 7.12 exhibits the CPU time for the computation of the power flow solutions considering the cases in Tab. 7.11.

From Tab. 7.12, the solvers that use frozen Jacobian reach correct solutions of the PFP and require the smallest computational costs. This result demonstrates the efficiency of using a

Table 7.11. Iteration number required for the base loading case of the systems, considering the respective μ determined for each case.

Method	*3012wp*	*3375wp*	*13659pegase*	*70k*	*109k*
T-NR	6	6	6	6	6
T-NR1J	24	22	22	28	22
T-HKW	8	8	8	8	8
T-HKW1	8	8	8	8	8
T-HKW2	24	25	24	30	24

Table 7.12. CPU time in seconds for executing the power flows in Tab. 7.11.

Method	*3012wp*	*3375wp*	*13659pegase*	*70k*	*109k*
T-NR	0.327	0.371	1.431	9.645	11.74
T-NR1J	0.203	0.232	1.037	6.054	7.64
T-HKW	0.567	0.630	2.642	16.495	21.47
T-HKW1	0.451	0.495	1.860	11.626	14.49
T-HKW2	0.236	0.252	1.090	6.530	7.81

Tikhonov's regularized parameter to compute the problem solution. The calculation of the regularized parameter requires the estimation of the L-curve. Despite this computational cost, how the PFP solution result is obtained is highly beneficial, considering that it is possible to solve ill-conditioned large-scale problems. In this sense, when estimating the L-curve, we can generally obtain it with around twenty points. This is equivalent to solving the generalized normal equation twenty times. Then, the CPU time can be estimated from the results presented in Tab. 7.10.

7.6.1 Impact of the reactive power limits in generators

Simulations considering operational power reactive limits of generators (MAMANDUR; BERG, 1982) for the base were performed for all test systems. The reactive power operational generator limits from the MATPOWER's databank cases were preserved in the simulations. For simplicity purposes, only the case considering Tikhonov's initialization approach, with the NR solver (T-NR), was employed for the simulations. The power reactive limit method used in the MATPOWER was applied for simulations. The method considers an external loop (labeled here as external iteration). It is activated when power reactive limits are violated after the convergence of the NR solver T-NR (the internal loop and where it is dependent on Tikhonov's initialization iteration). The external iteration is applied until a convergence/divergence status is verified. Those generators with violated limits are selected and converted to load (PQ

bus) in this external iteration. The procedure is repeated until all generators meet the limit requirements or divergence occurs.

Table 7.13 depicts the information on the convergence process for the studied five test systems. For each system, the number of violated limits in the PV buses is monitored and inserted from the second to ninth columns. For the external ith iteration ($iter_i$), it is verified if the upper (U) or/and the lower(L) limits of the PV buses are violated. In the positive case, the number of violations is counted, and the total is included in the table, depending on whether the situation is upper or lower.

Table 7.13. Performance of the Newton-Raphson solver with Tikhonov's initialization procedure (T-NR) for the base cases assuming reactive power limits in PV buses.

	$iter_1$		$iter_2$		$iter_3$		$iter_4$	
CASE	U	L	U	L	U	L	U	L
case3012	228	9	4	-	-	-	-	-
case3375	123	76	11	2	1	-	-	-
case13659	1	-	-	-	-	-	-	-
case70k	2190	2398	422	278	143	57	8	3
case109k	8	-	-	-	-	-	-	-

All base cases in Tab. 7.13 present convergence using the approach T-NR and have reactive power limits in PV buses with violations. The most critical situation occurs for the case70k, which converges only in the fifth iteration. Note that, for this situation, the fourth column in the table indicates that eight upper and three lower limits were still violated, respectively. Therefore, the fifth iteration was necessary to check whether the PV to PQ conversion for the buses resulted in successful convergence. The results in the table demonstrate that despite starting from a flat start estimate and hard limits in the case70k, Tikhonov's initialization is efficient for providing convergence in the simulation. However, this initialization is used only for determining the $iter_1$, because the further external iterates use the results of previous external iterations as the initial estimate.

7.6.2 Experiments for an operational point near the maximum loading

This subsection demonstrates the ability of the proposed methodology based on Tikhonov's initialization used for the NR solver (T-NR) again. It evaluates the performance of the stringent problem considering the loading until near its maximum value, λ_*, including reactive power

limits in generators. The experiments were carried out for the three largest model systems.

The previous section showed the influence of the reactive power limits in PV buses when the base loading cases were studied. For the situation where the systems are submitted to additional loads and supplied generation, we have assumed that the loads P_L, Q_L (for the base case load P_{L0}, Q_{L0}) maintain the same power factor. *i.e.*, $P_L = P_{L0}(1+\lambda)$, $Q_L = Q_{L0}(1+\lambda)$, where $\lambda \geq 0$; and the active power in generator, P_g, are incremented by the same factor λ in such way that, for the base case generation P_{g0}, $P_g = P_{g0}(1+\lambda)$. We are distinguishing in the simulations loads of type constant power and constant impedance Z_{shunt} as explained further. We are assuming that the formers always change with λ, while the latter are simulated changing or not with the loading factor.

To determine the maximum load factor, we developed an iterative strategy starting from a supposed $\lambda_0 < \lambda_* \approx \lambda_{max}$, where λ_{max} is assumed as the maximum loading factor. In the first stage (PART I), the empirical scheme consists in incrementing λ geometrically from an initial point, defined by λ_0, in such a way that for an iteration k, $\lambda_k = \lambda_0 2^k$ up to divergence is detected. When this divergence is reached, say for a given k_d, the procedure switches to a second computation stage (PART II) to determine a smaller and refined value $\lambda_k < \lambda_{k_d}$, now for iterations $k > k_d$. However, in this second stage, the well-known bisection method is applied, taking as extreme values the points λ for the iterates $(k_d - 1)$ (convergent case) and k_d (divergent case). The iterative process stops when a tolerance error δ is reached. In this case, an approximation for the maximum loading $\lambda_* \approx \lambda_{aprox}$ is obtained. The Flowchart in Fig. 7.14 illustrates the two parts of the numerical procedure. We emphasize that the scheme is applied similarly for simulations taking into account (or not) power reactive limits in generators.

To illustrate the determination of the effectiveness of the proposed technique using Tikhonov's initialization approach to the NR solver (T-NR version), simulations were assessed in the largest model systems (case 13659, case70k, and case 109k). This means that for each point λ_k, the NR solver uses the result of Tikhonov's initialization approach, which in turn departs from a flat start estimation of the states. Tests were done to check the effect of power reactive limits on generators. Therefore, situations with or without limits on PV buses. For all simulations, the iterate starts with $\lambda_0 = 0.2$ and an absolute error $\delta = 5 \times 10^{-4}$ is required for global convergence.

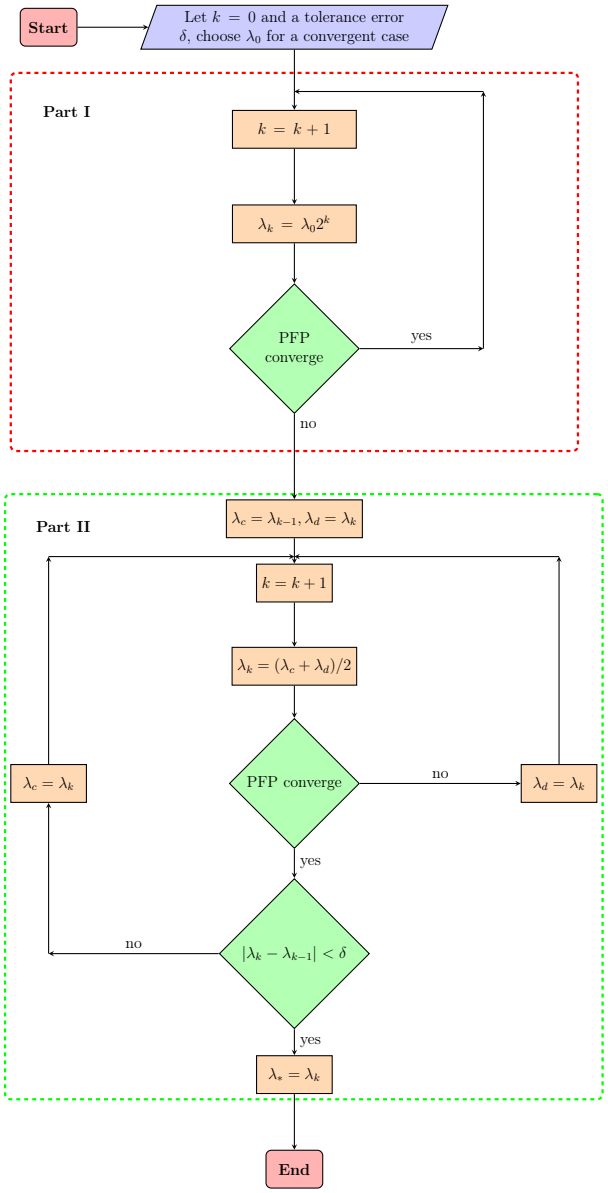


Figure 7.14. Flowchart with the procedure to determine the maximum loading factor with or without power limits in PV buses.

In the first set of simulations, with respect to the loads, only those ones with active and reactive constant power declared in MATPOWER were submitted to change in the loading factor λ. Therefore, loads declared as constant impedance, Z_{shunt}, had preserved their values for any λ. This is justified since many shunt impedances characterized this way are sources of reactive power. Anyway, the loading for them can also be implemented in the iterative process, as will be done in the second set of simulations ahead. Fig. 7.15 shows plots for iterations relating to the computations of the loading factor until near the maximum value when the generators are free of limits in PV buses. For case 70k, the loading factor converges to $\lambda_* = 0.09594$, while for case109k and case13659, the convergence is verified for $\lambda_* = 0.09594$. This latter situation,

whose values coincide, occurs because case109k is a synthetic system generated considering the composition of eight case13659 systems (VéLIZ; JURADO, 2019). Then, when the PV buses are limit-free, the loading factor for both systems is the same.

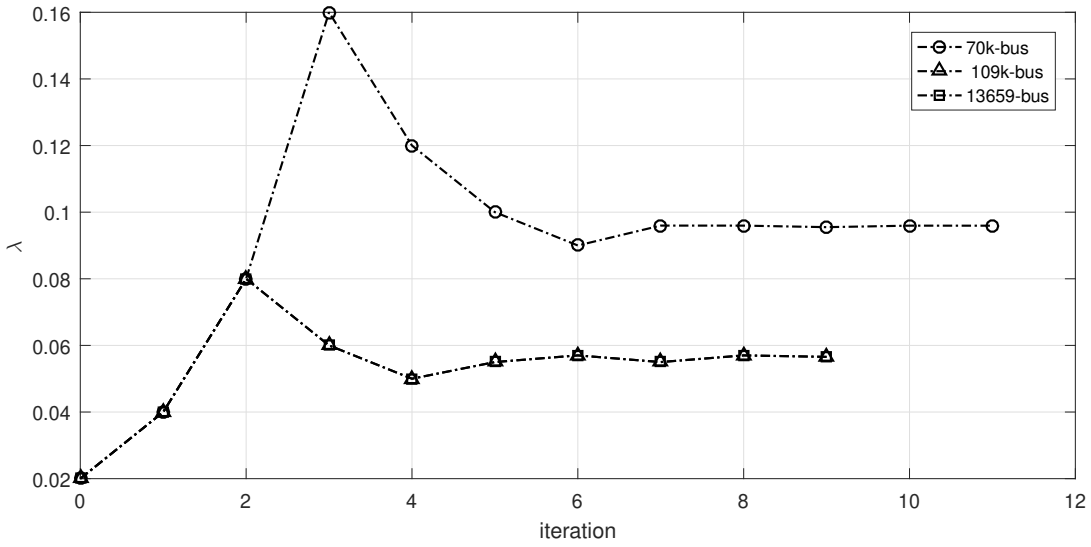


Figure 7.15. Results for the loading factor convergence for maximum loading point when the reactive power limits in PV buses are off.

Still considering constant impedance with preserved values for any λ, Fig. 7.16 shows the situation when all generators' power reactive limits are considered. The convergence related to the loading factors near its maximum value λ_* are verified for the values 0.02151, 0.02625, and 0.03042, respectively, for the case70k, case109k, and case13659. Note that all these limit values are much smaller than the respective values for the cases when the reactive power limits in generators are free. Therefore, this is an important result since it demonstrates that the effectiveness of the initialization procedure based on Tikhonov's approach also works properly even for hard situations near the maximum loading factor. Furthermore, it proves what is already well known in the technical literature that the *results of the load flow problem only make sense if the operational limits of equipment are considered.* In the particular case studied in this paper, the limit evaluated was only for generators (reactive power in PV buses).

Now, a second set of experiments considering changes with the loading factor of loads of type constant impedance, Z_{shunt}, such as they are declared in MATPOWER was performed. Table 7.14 summarizes the limit factor λ_* for this load type with and without modification using λ. Results of the first experiment set, where Z_{shunt} is insensitive to λ, are informed again in columns four and five of the table.

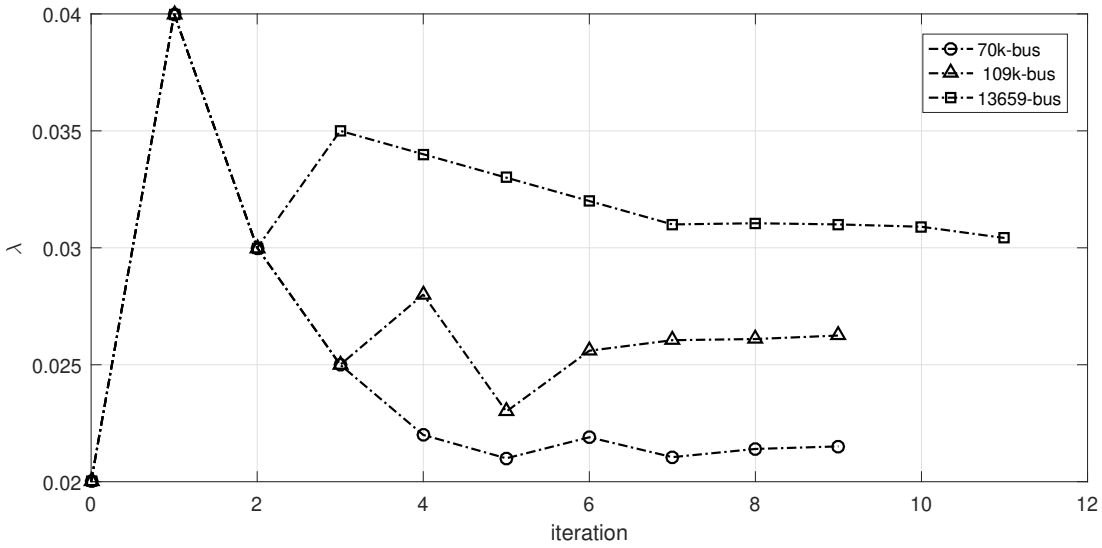


Figure 7.16. Results for the loading factor convergence for maximum loading when the reactive power limits in PV buses are activated.

Table 7.14. Performance of the Newton-Raphson solver with Tikhonov's initialization procedure (T-NR) for the base cases assuming reactive power limits and impedance load type Z_{shunt} changing with the loading.

	Z_{shunt} changing		Z_{shunt} fixed	
CASE	limit free	limit in PV	limit free	limit in PV
case13659	0.061015	0.032539	0.056563	0.03042
case70k	0.099297	0.023825	0.09594	0.02151
case109k	0.061015	0.028135	0.056563	0.02625

From Tab. 7.14, imposing Z_{shunt} changing with λ allows a loading factor higher than the case for a fixed value. However, the changes are not significant for both with or without reactive power limits in PV buses. On the other hand, the existence of this type of support of shunt reactive power alleviates the reactive power produced by the generators and is beneficial for incrementing the system transmission capacity, even by a small amount as observed in the table.

We also followed the evolution of the convergence process in detail for the case where the value of λ_* is calculated for case70k. The reactive power limits are assumed in the generators. The step-by-step process is illustrated through the results shown in Tab. 7.15. Again, the iterative process is used to determine the maximum loading factor by adopting the Newton-Raphson solver with Tikhonov's initialization procedure (T-NR). The results in the table correspond to the black curve in Fig. 7.16. However, the table exhibits additional details, such as the number of violated limits for each external PFP iteration. In the table, $iter_i$ stands for an external iteration of the PFP, for verifying whether the upper (U) or/and lower (L) limits are violated.

In the first column of Tab. 7.15, the loading factor λ is monitored at each iteration k. The result in each table row shows the number of limits violated in PV buses according to the iteration $iter_i$ for an external loop of the T-NR solver. Hence, the table illustrates the convergence for λ and the power reactive limits in PV buses.

Table 7.15. Iterative process illustration for determining the maximum loading factor adopting the Newton-Raphson solver with Tikhonov's initialization procedure (T-NR) assuming reactive power limits and a loading factor λ for the case 70k.

	$iter_1$		$iter_2$		$iter_3$		$iter_4$		$iter_5$		$iter_6$		$iter_7$
λ_k	U	L	U	L	U	L	U	L	U	L	U	L	status
0.02	2478	2156	603	198	127	34	26	1	1	-	-	-	conv
0.04	2720	1992	-	-	-	-	-	-	-	-	-	-	div
0.03	2612	2087	651	161	-	-	-	-	-	-	-	-	div
0.025	2542	2126	621	179	173	32	-	-	-	-	-	-	div
0.0225	2511	2140	607	189	153	30	31	1	-	-	-	-	div
0.02125	2504	2152	601	196	125	30	36	3	5	-	-	-	conv
0.0219	2505	2148	606	194	138	29	39	1	17	-	-	-	div
0.0216	2504	2148	605	194	130	29	43	9	-	-	-	-	div
0.0214	2504	2149	604	196	127	28	40	5	7	-	1	-	conv

conv: convergence; div: divergence.

A description of the results can be done as follows. Following the flowchart according to PART I in Fig. 7.14, an initial value assigned for the loading factor was established as $\lambda_0 = 0.02$. Six external iterations (reactive power limit iterations) were necessary to converge (see the row in the table corresponding to λ_0). The second iteration, $k = 1$, calculated the loading factor as $\lambda_1 = 0.02 \times 2^1$. However, the iteration with limits diverges for the second external iteration ($iter_2$). Then, the algorithm switches to the PART II, considering $\lambda_2 = (0.2 + 0.4)/2 = 0.03$. But, this iteration for λ_2 is also divergent. The iterations continue, and for $\lambda_5 = 0.2125$, the iterative process converges again, with an absolute error $\delta = 0.00125$, requiring six iterations to converge to the reactive power limits. Finally, after three more iterations on λ, *i.e.*, for $\lambda_8 = 0.0214$, the convergence is verified with $\delta = 2 \times 10^{-4}$, converging in the seventh external iteration of the PFP.

The plots and table results demonstrate the efficiency of the NR solver when initialized with just one iteration of Tikhonov's approach, even considering flat start estimation for determining this initialization. The problem was investigated even for stringent conditions in this section, such as the loading near its maximum allowable value and constrained limits in PV buses.

7.7 TESTS CONSIDERING THE USE OF DIRECT AND ITERATIVE METHODS

This section details the results of the simulations performed with the direct and iterative methods discussed in Chapter 3. The main objective is to evaluate the performance of these approaches in scenarios that use the Tikhonov and δ perturbation strategies. This perturbation strategy was applied exclusively during the first iteration of the solution process. Therefore, for the subsequent iterations, the traditional Newton-Raphson (NR) method was used. For the solution of the linear system involving the Jacobian matrix and mismatches both the direct and iterative linear solvers were applied and their performance was evaluated. The linear system was solved directly through the command $dx = -J\backslash F$ and also using the LU factors obtaining via LU factorization in MATLAB's *lu* function. The iterative linear methods analyzed included GMRES, BiCG, BiCGStab, and a method called Conjugate Gradient Squared (CGS), all of them using codes available in MATLAB®. Unlike BiCG, which uses the residual and its conjugate, CGS avoids using the transpose of the coefficient matrix by working with a squared residual (BARRETT *et al.*, 1994). These methods were studied for the two reordering schemes AMD and RCM, which have been extensively studied in solving large-scale power flow problems (FERNANDES, 2014).

Table 7.16 presents the results of the direct methods in simulations involving the three largest systems studied. Two main approaches were analyzed: the first, referred to as T-NR, using Tikhonov's regularization for the first iteration as defined in (5.36), followed by the traditional NR method in subsequent iterations. The second approach, CS-NR, incorporates a conditioning step described in Option I of Section 6.2. For all cases, the computational time presented in the table corresponds only to the time required to solve the linear system, directly or through the LU method, without considering the conditioning step. Each simulation was repeated 100 times, and the average computational time was calculated. The standard deviation of the measurements is shown in parentheses. The computational cost for the first iteration (perturbation approaches) was already presented in Tables 7.7 and 7.10. The LU factorizations demonstrated a lower computational cost slight better than the direct computation of $dx = -J\backslash F$ in both approaches. This highlights an efficiency almost equivalent in both cases, even for large-scale systems.

Table 7.17 presents the simulation results for the iterative linear methods GMRES, CGS, BiCG,

Table 7.16. Average CPU time in seconds for solving power flow problems using direct methods, with standard deviation shown in parentheses for 100 iterations.

Method	*13659pegase*	*70k*	*109k*
T-NR (Direct)	0.3228 (0.0325)	2.0842 (0.2135)	2.7053 (0.2712)
T-NR (LU)	0.3141 (0.0317)	2.0531 (0.2066)	2.6175 (0.2611)
CS-NR (Direct)	0.3214 (0.0321)	2.0676 (0.2083)	2.6905 (0.2792)
CS-NR (LU)	0.3134 (0.0314)	2.0656 (0.2109)	2.5985 (0.2602)

and BiCGStab, employing the incomplete LU (ILU) preconditioner generated using MATLAB's *ilu* function. These simulations also consider the application of AMD and RCM reorderings to improve convergence efficiency. The conditioning step used was based on the PFP Tikhonov's regularization approach for the 1st iteration.

Table 7.17. Average CPU time in seconds for solving power flow problems using iterative linear methods with their respective reorderings, and the standard deviation shown in parentheses for 100 iterations.

Reord.	*Method*	*System*		
		13659pegase	*70k*	*109k*
RCM	GMRES	0.4448 (0.0446)	11.8127 (1.6642)	5.4479 (0.5528)
	CGS	0.4892 (0.0502)	13.3543 (1.3485)	5.3444 (0.5477)
	BiCG	0.6771 (0.0708)	15.0637 (1.5137)	8.1833 (0.8655)
	BiCGStab	0.4326 (0.0433)	11.8969 (1.2044)	4.6242 (0.4674)
AMD	GMRES	0.2271 (0.0231)	2.0788 (0.2244)	2.6241 (0.2677)
	CGS	0.2296 (0.0261)	2.1306 (0.2201)	2.4284 (0.2503)
	BiCG	0.3070 (0.0318)	2.8933 (0.2984)	4.1944 (0.4327)
	BiCGStab	**0.2100 (0.0212)**	**1.8781 (0.2072)**	**2.1020 (0.2135)**

From the results obtained, it was observed that the best performance was achieved by combining the BiCGStab method with AMD reordering. For the 70k-bus system, this combination resulted in a computational time reduction of up to 81% when compared to the use of RCM reordering with the same method. Additionally, across all analyzed scenarios, the BiCGStab method converged with the lowest computational cost and the smallest standard deviation, highlighting its consistency and efficiency.

The CGS and GMRES methods, when combined with AMD reordering, also delivered results that were very close in performance to those of BiCGStab. This suggests that when combined with an efficient reordering strategy like AMD, these iterative solvers are also well-suited for solving large-scale ill-conditioned power flow problems.

On the other hand, the use of RCM reordering was proved inefficient in terms of computational cost when addressing ill-conditioned, large-scale systems. This indicates that RCM may offer

benefits in some contexts, but is less effective for the specific challenges posed by this type of power flow problem.

A comparison of the results obtained using direct and iterative linear solvers, as shown in Tables 7.16 and 7.17, respectively, reveals that the iterative linear system based on BiCGStab and AMD reordering scheme performs better. A reduction of about 25 % on the computational burden is verified for the largest system and the T-NR solver (ratio 2.6175/2.102). Although the differences in performance are not significant for the technical evaluations of the proposed techniques in this work, we have retained all previous results obtained using the direct method, as discussed in the earlier sections, for the assessment of the convergence performance of the PFP methods.

7.8 FINAL CONSIDERATIONS

This section presented various numerical results aimed at validating the current PFP solvers. Several large-scale ill-conditioned systems were considered for this purpose, analyzing the performance of the methods in terms of the number of iterations required for convergence, as well as the CPU time, different loading conditions, and the impact of reactive power limits on generators.

Also, a study about the employment of iterative linear systems was conducted revealing the superior performance of techniques such as BICGStab and CGS to solve the PFP in ill-conditioned large-scale power system models.

The main conclusions of this chapter can be summarized as follows:

- A hybrid method approach was investigated, whose states were calculated in the first step using the NR method and utilized as an estimate for an iterative method (IM) in the second step. Solvers such as classical NR and its decoupled version converged quickly after using the first step from the homotopy-based method. However, applying the full homotopy solver (full-H) directly in PFP demands a high computational cost;

- Simulations involving three perturbation options of the Conditioning Step approach were investigated. It was verified that perturbation option III performed slightly better than

the others for the largest power system models. The results indicated that the proposed conditioning strategy significantly improves the convergence process of iterative techniques used to solve large-scale, ill-conditioned PFPs. This includes the classical Newton-Raphson method, which failed to obtain a solution for all systems studied without CS and flat start estimation;

- The proposed modal-based Jacobian matrix perturbation method for detecting the primary cause of ill-conditioning in the PFP also offers an efficient alternative for addressing this issue. Despite requiring the calculation of the smallest eigenvalue and the corresponding right and left eigenvectors of the Jacobian matrix, the results demonstrated the robust performance of the modal-based approach when applied exclusively to the initial iteration of the PFP, surpassing the computational efficiency of the shift -δ method; and

- Finally, the partial regularization method proved to be a robust methodology for solving the ill-conditioned PFP, comparable to the modal approach, achieving convergence in all analyzed scenarios. The main disadvantage observed was the computational cost associated with determining the regularization parameter, μ, which was higher compared to the cost of solving the nonlinear problem. However, given the ill-conditioned nature of the problem, the benefits of using the method outweighs the global computations cost required. It is worth noting that the calculation of the regularization parameter for the Tikhonov's method is also suitable for estimating parameters used in other pertubation techniques.

CHAPTER 8

CONCLUSIONS AND FUTURE WORKS

8.1 GENERAL CONCLUSIONS

In this Ph.D. thesis, several PF solvers based on hybrid strategies and perturbations of the Jacobian matrix applied from the initial iteration were studied and developed. These methodologies aim to efficiently solve ill-conditioned cases where the Newton-Raphson method fails to converge when initialized with a flat start. Extensive numerical results were presented, demonstrating the performance of the proposed methods applied in large-scale ill-conditioned systems under various loading scenarios and assessing the associated computational costs. Furthermore, the impact of reactive power limits on generators was examined, highlighting their influence on system behavior and the robustness of the solutions obtained. Additionally, a study on iterative linear solvers revealed that techniques such as BICGStab and CGS outperform other methods in solving the PFP for large-scale power system models, particularly when combined with AMD reordering. Lastly, the main conclusions of this work are presented in the following.

For a Two-step Hybrid method, it was demonstrated that the first step of the hybrid technique acts as a virtual enabler, effectively expanding the original convergence region of attraction for the flat start estimate. In the first step of the method, a homotopy-based solver was implemented to determine a low-accuracy solution, which was improved by using the second step. Simulations performed on systems ranging from 3,012 to 70,000 buses confirmed that all methods in the second stage employing this approach successfully solved the PFP under base loading conditions. Regarding computational costs, solvers such as the classical NR method and its decoupled variant used in the second step achieved fast convergence after incorporating the initial step of the homotopy-based method. However, it was also shown that directly applying the full homotopy solver (full-H) to calculate a high-precision solution for the PFP involves significantly higher computational costs.

The second proposed approach, the Conditioning Step, consisted of modifying the initial estimate of the iterative method through a process involving the Jacobian matrix calculated for the initial estimate. The Jacobian matrix formed a perturbed linear system whose resultant perturbed matrix had a better condition number. Three options were proposed to implement the perturbed form. Among these, perturbation option III demonstrated slightly better performance for the largest power system models. In general, the results confirmed that the proposed conditioning strategy significantly enhances the convergence process of iterative techniques when applied to large-scale, ill-conditioned PFPs. In particular, the CS-NR method demonstrated exceptional performance, successfully solving the PFP in all scenarios with the fewest number of iterations.

A proposed modal-based Jacobian matrix perturbation method demonstrated that the primary cause of the ill-conditioning problem was associated with the smallest magnitude eigenvalue of the first iteration of the Jacobian PF matrix. Additionally, a procedure was proposed to address this problem by shifting the smallest magnitude eigenvalue of the Jacobian matrix away from zero. For this purpose, the smallest eigenvalue and its respective right- and left-eigenvectors were computed. Then, a 1-rank perturbation matrix formed with this partial eigendata was added to the Jacobian matrix. Finally, an approximation for the deviation of states was efficiently computed, considering the same mismatches determined for the first iteration. The results demonstrated the robust performance of the modal-based approach when applied exclusively to the initial iteration of the PFP, surpassing the computational efficiency of the shift -δ method. Furthermore, the results obtained suggest that the modal study for the first iteration of the PFP and other applications may be promising research in the direction of solving the ill-conditioning also in nonlinear problems at all.

A method based on Partial Tikhonov's Regularization was also employed as a *conditioning step*, applied exclusively to the initial iteration of the PFP. In this approach, the subsequent iterations utilized the result of a regularized normal equation, where the regularization parameter, μ, was selected using the traditional L-curve technique. From the second iteration onward, various iterative methods, including NR, HKW, and their variants, were applied successfully, all of which achieved the PFP solution, even for a 109,000-bus system. However, a disadvantage presented by the method was the computational cost associated with determining the

regularization parameter μ, which was higher than the cost of solving the nonlinear problem. However, given the ill-conditioning characteristics of the problem, the benefits of employing this method outweigh the overall computational costs. Additionally, it is worth highlighting that the process of determining the regularization parameter in the Tikhonov's method is well-suited for estimating parameters used in other perturbation methods, as discussed in Section 7.6. This compatibility further underscores the versatility and potential applications of the Tikhonov-based conditioning approach.

8.2 SUGGESTION FOR FUTURE WORKS

The methodologies presented in this work open several paths for future research and development. These potential directions can be outlined as follows:

- Investigate optimized strategies for determining the regularization parameter μ, aiming to minimize its computational overhead while maintaining accuracy and stability in solving the PFP with partial Tikhonov's Regularization;
- Explore alternative matrix reordering techniques, such as spiral reordering, combined with iterative linear solvers, to assess their impact on the computational cost of solving the PFP for large and ill-conditioned systems;
- Implement other forms of computations for the partial solution in the hybrid method and extend the network modeling to include other devices and controls;
- Investigate the modal-based approach while considering multi-shift strategies to account for a larger number of eigenvalues being shifted from near zero;
- Extension of investigations of the PFP regarding critical loading scenarios based on fractional calculus theory, as preliminarily studied in (FREITAS; OLIVEIRA, 2024).

REFERENCES

AJJARAPU, V.; CHRISTY, C. The continuation power flow: a tool for steady state voltage stability analysis. *IEEE transactions on Power Systems*, Ieee, v. 7, n. 1, p. 416–423, 1992. Cited in page 50.

AMESTOY, P. R.; DAVIS, T. A.; DUFF, I. S. An approximate minimum degree ordering algorithm. *SIAM Journal on Matrix Analysis and Applications*, SIAM, v. 17, n. 4, p. 886–905, 1996. Cited in page 46.

ARGYROS, I. K.; GEORGE, S. Regularization methods for ill-posed problems with monotone nonlinear part. *Punjab University Journal of Mathematics*, v. 46, n. 1, p. 25–38, 2014. Cited in page 3.

ARGYROS, I. K.; MAGREÑÁN, Á. A. *Iterative Methods and Their Dynamics with Applications: A Contemporary Study*. [S.l.]: CRC Press, 2017. Cited in page 59.

BABAJEE, D. K. R.; DAUHOO, M. Z.; DARVISHI, M. T.; KARAMI, A.; BARATI, A. Analysis of two Chebyshev-like third order methods free from second derivatives for solving systems of nonlinear equations. *J. Comput. Appl. Math.*, v. 233, n. 8, p. 2002–2012, 2010. Cited in page 3.

BARRETT, R.; BERRY, M.; CHAN, T. F.; DEMMEL, J.; DONATO, J.; DONGARRA, J.; EIJKHOUT, V.; POZO, R.; ROMINE, C.; VORST, H. Van der. *Templates for the solution of linear systems: building blocks for iterative methods*. [S.l.]: SIAM, 1994. Cited 5 times in pages 39, 40, 41, 43, and 119.

BENZI, M. Preconditioning techniques for large linear systems: a survey. *Journal of computational Physics*, Elsevier, v. 182, n. 2, p. 418–477, 2002. Cited 3 times in pages 40, 44, and 45.

BENZI, M.; SZYLD, D. B.; DUIN, A. V. Orderings for incomplete factorization preconditioning of nonsymmetric problems. *SIAM Journal on Scientific Computing*, SIAM, v. 20, n. 5, p. 1652–1670, 1999. Cited 2 times in pages 45 and 46.

BIRCHFIELD, A. B.; XU, T.; GEGNER, K. M.; SHETYE, K. S.; OVERBYE, T. J. Grid structural characteristics as validation criteria for synthetic networks. *IEEE Transactions on power systems*, IEEE, v. 32, n. 4, p. 3258–3265, 2016. Cited in page 87.

BIRCHFIELD, A. B.; XU, T.; OVERBYE, T. J. Power flow convergence and reactive power planning in the creation of large synthetic grids. *IEEE Transactions on Power Systems*, IEEE, v. 33, n. 6, p. 6667–6674, 2018. Cited in page 87.

BRAZ, L. M.; A, C. C.; MURATI, C. A critical evaluation of step size optimization based load flow methods. *IEEE Transactions on Power Systems*, IEEE, v. 15, n. 1, p. 202–207, 2000. Cited in page 2.

BROWN, W.; CLOUES, W. Power-system representation on network analyzers. *Electrical Engineering*, IEEE, v. 75, n. 1, p. 59–59, 1956. Cited in page 48.

BUKHSH, W. On solving the load flow problem as an optimization problem. University of Strathclyde, 2018. Cited 2 times in pages 33 and 63.

BUTCHER, J. C. On runge-kutta processes of high order. *Journal of the Australian Mathematical Society*, Cambridge University Press, v. 4, n. 2, p. 179–194, 1964. Cited in page 58.

BUTCHER, J. C. *Numerical Methods for Ordinary Differential Equations*. 2nd. ed. England: Wiley, 2008. Cited in page 3.

CALVETTI, D.; MORIGI, S.; REICHEL, L.; SGALLARI, F. Tikhonov regularization and the l-curve for large discrete ill-posed problems. *Journal of computational and applied mathematics*, Elsevier, v. 123, n. 1-2, p. 423–446, 2000. Cited 2 times in pages 67 and 68.

CHAGAS, G.; PIRES, R. The sequential power flow in complex plane for solving the vsc-mtdc hybrid ac/dc transmission grids. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 148, p. 108900, 2023. Cited in page 23.

CUTHILL, E.; MCKEE, J. Reducing the bandwidth of sparse symmetric matrices. In: *Proceedings of the 1969 24th national conference*. [S.l.: s.n.], 1969. p. 157–172. Cited in page 46.

DEUFLHARD, P. A modified newton method for the solution of ill-conditioned systems of nonlinear equations with application to multiple shooting. *Numerische Mathematik*, Springer, v. 22, n. 4, p. 289–315, 1974. Cited in page 50.

DUFF, I. S.; ERISMAN, A. M.; REID, J. K. On george's nested dissection method. *Siam journal on numerical analysis*, SIAM, v. 13, n. 5, p. 686–695, 1976. Cited in page 46.

DUFF, I. S.; MEURANT, G. A. The effect of ordering on preconditioned conjugate gradients. *BIT Numerical Mathematics*, Springer, v. 29, p. 635–657, 1989. Cited in page 46.

DŽAFIĆ, I.; JABR, R. A. Real-time equality-constrained hybrid state estimation in complex variables. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 117, p. 105634, 2020. Cited in page 23.

FAIRES, J. D.; BURDEN, R. L. *Numerical methods, 9th*. [S.l.]: Cengage Learning, 2012. Cited 3 times in pages 56, 57, and 65.

FAN, L. *Solution of the ill-conditioned load flow problem by the tensor method*. 116 p. Dissertação (Mestrado) — Tianjin University, China, 1989. Cited 2 times in pages 13 and 24.

FERNANDES, A. A. Uma estrutura de solucionador iterativo linear com aplicação à solução de equações do problema de fluxo de carga. *Tese de Doutorado*, Universidade de Brasília, 2014. Cited in page 119.

FLISCOUNAKIS, S.; PANCIATICI, P.; CAPITANESCU, F.; WEHENKEL, L. Contingency ranking with respect to overloads in very large power systems taking into account uncertainty, preventive, and corrective actions. *IEEE Transactions on Power Systems*, IEEE, v. 28, n. 4, p. 4909–4917, 2013. Cited in page 87.

FREITAS, F. D.; JR, A. C. S.; FERNANDES, L. F. J.; ACLE, Y. G. Restarted holomorphic embedding load-flow model based on low-order padé approximant and estimated bus power injection. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 112, p. 326–338, 2019. Cited in page 51.

FREITAS, F. D.; LIMA-SILVA, A. A power flow homotopy-based solver emanating from a flat start estimate. In: IEEE. *2023 IEEE Power & Energy Society General Meeting (PESGM)*. [S.l.], 2023. p. 1–5. Cited 2 times in pages 53 and 72.

FREITAS, F. D.; OLIVEIRA, L. N. de. Conditioning step on the initial estimate when solving ill-conditioned power flow problems. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 146, p. 108772, 2023. Cited 2 times in pages 106 and 110.

FREITAS, F. D.; OLIVEIRA, L. N. de. Two-step hybrid-based technique for solving ill-conditioned power flow problems. *Electric Power Systems Research*, v. 218, p. 109178, 2023. ISSN 0378-7796. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0378779623000676>. Cited in page 70.

FREITAS, F. D.; OLIVEIRA, L. N. de. A Fractional Order Derivative Newton-Raphson Method for the Computation of the Power Flow Problem Solution in Energy Systems. *Fractional Calculus and Applied Analysis*, Springer Nature, v. 27, p. 1–32, 2024. Cited in page 125.

FREITAS, F. D.; SILVA, A. L. Flat start guess homotopy-based power flow method guided by fictitious network compensation control. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 142, p. 108311, 2022. Cited 8 times in pages 3, 5, 53, 70, 71, 72, 73, and 94.

GALIANA, F.; JAVIDI, H.; MCFEE, S. On the application of a pre-conditioned conjugate gradient algorithm to power network analysis. *IEEE Transactions on Power Systems*, IEEE, v. 9, n. 2, p. 629–636, 1994. Cited in page 40.

GEORGE, A. Nested dissection of a regular finite element mesh. *SIAM journal on numerical analysis*, SIAM, v. 10, n. 2, p. 345–363, 1973. Cited in page 46.

GEORGE, A.; LIU, J. W. *Computer solution of large sparse positive definite*. [S.l.]: Prentice Hall Professional Technical Reference, 1981. Cited in page 46.

GLIMN, A.; STAGG, G. Automatic calculation of load flows. *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, IEEE, v. 76, n. 3, p. 817–825, 1957. Cited in page 14.

GLOVER, J. D.; SARMA, M. S.; OVERBYE, T. *Power system analysis & design, SI version*. [S.l.]: Cengage Learning, 2012. Cited 5 times in pages 1, 11, 15, 24, and 26.

GOLUB, G. H.; HEATH, M.; WAHBA, G. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, Taylor & Francis, v. 21, n. 2, p. 215–223, 1979. Cited 3 times in pages 65, 67, and 68.

GOLUB, G. H.; LOAN, C. F. V. *Matrix Computations*. [S.l.]: The Johns Hopkins University Press, Baltimore, USA, 1996. Cited 10 times in pages 22, 30, 32, 65, 66, 77, 78, 79, 80, and 89.

HAGER, W. W. Updating the inverse of a matrix. *SIAM review*, SIAM, v. 31, n. 2, p. 221–239, 1989. Cited in page 84.

HANSEN, P. C. Analysis of discrete ill-posed problems by means of the l-curve. *SIAM review*, SIAM, v. 34, n. 4, p. 561–580, 1992. Cited 3 times in pages 65, 67, and 68.

HETZLER, S. M. A Continuous Version of Newton's Method. *The College Mathematics Journal*, Taylor & Francis, v. 28, n. 5, p. 348–351, 1997. Cited 3 times in pages 2, 51, and 56.

IWAMOTO, S.; TAMURA, Y. A load flow calculation method for ill-conditioned power systems. *IEEE transactions on power apparatus and systems*, IEEE, n. 4, p. 1736–1743, 1981. Cited 4 times in pages 2, 23, 49, and 51.

JEREMINOV, M.; TERZAKIS, A.; WAGNER, M.; PANDEY, A.; PILEGGI, L. Robust and Efficient Power Flow Convergence with G-min Stepping Homotopy Method. In: *2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*. IEEE, 2019. p. 1–6. ISBN 978-1-7281-0653-3. Disponível em: <https://ieeexplore.ieee.org/document/8783295/>. Cited in page 3.

JOSZ, C.; FLISCOUNAKIS, S.; MAEGHT, J.; PANCIATICI, P. AC power flow data in MATPOWER and QCQP format: iTesla, RTE snapshots, and PEGASE. *arXiv preprint arXiv:1603.01533*, 2016. Cited in page 87.

JR, W. S.; GRAINGER, J. *Power system analysis*. [S.l.]: McGraw-Hill Education, 1994. Cited in page 9.

KLUMP, R. P.; OVERBYE, T. J. Techniques for improving power flow convergence. In: IEEE. *2000 Power Engineering Society Summer Meeting (Cat. No. 00CH37134)*. [S.l.], 2000. v. 1, p. 598–603. Cited in page 50.

KRYLOV, A. N. On the numerical solution of the equation by which in technical questions frequencies of small oscillations of material systems are determined. *Izvestija AN SSSR (News of Academy of Sciences of the USSR), Otdel. mat. i estest. nauk*, v. 7, n. 4, p. 491–539, 1931. Cited in page 35.

KUNDUR, P. *Power system stability and control*. [S.l.]: CRC Press New York, NY, USA, 2007. Cited 5 times in pages 1, 9, 11, 12, and 105.

LIAO, S. *Homotopy analysis method in nonlinear differential equations*. [S.l.]: Springer, 2012. Cited in page 53.

LIMA-SILVA, A.; FREITAS, F. D.; FERNANDES, L. F. d. J. A homotopy-based approach to solve the power flow problem in islanded microgrid with droop-controlled distributed generation units. *Energies*, MDPI, v. 16, n. 14, p. 5323, 2023. Cited in page 53.

LIU, C.-S. A dynamical tikhonov regularization for solving ill-posed linear algebraic systems. *Acta applicandae mathematicae*, Springer, v. 123, p. 285–307, 2013. Cited 2 times in pages 65 and 68.

MAMANDUR, K.; BERG, G. Automatic adjustment of generator voltages in newton-raphson mehtod of power flow solutions. *IEEE Transactions on Power Apparatus and Systems*, IEEE, n. 6, p. 1400–1409, 1982. Cited in page 112.

MANTEUFFEL, T. A. An incomplete factorization technique for positive definite linear systems. *Mathematics of computation*, v. 34, n. 150, p. 473–497, 1980. Cited in page 43.

MCGILLIS, D. Nodal iterative solution of power-flow problem using ibm 604 digital computer. *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, v. 76, n. 3, p. 803–809, 1957. Cited in page 48.

MILANO, F. Continuous Newton's method for power flow analysis. *IEEE Transactions on Power Systems*, IEEE, v. 24, n. 1, p. 50–57, 2009. Cited 10 times in pages 1, 2, 13, 17, 21, 22, 49, 51, 52, and 55.

MILANO, F. *Power system modelling and scripting*. [S.l.]: Springer Science & Business Media, 2010. Cited 2 times in pages 15 and 26.

MILANO, F. Analogy and convergence of levenberg's and lyapunov-based methods for power flow analysis. *IEEE Transactions on Power Systems*, IEEE, v. 31, n. 2, p. 1663–1664, 2015. Cited 3 times in pages 50, 63, and 64.

MILANO, F. Implicit Continuous Newton Method for Power Flow Analysis. *IEEE Transactions on Power Systems*, v. 34, n. 4, p. 3309–3311, jul 2019. ISSN 0885-8950. Disponível em: <https://ieeexplore.ieee.org/document/8698293/>. Cited 2 times in pages 2 and 55.

MONTICELLI, A. J.; GARCIA, A. *Introdução a sistemas de energia elétrica*. [S.l.]: Ed Unicamp, 1999. Cited in page 25.

NESS, J. E. V. Iteration methods for digital load flow studies. *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, IEEE, v. 78, n. 3, p. 583–586, 1959. Cited in page 49.

NEUBERGER, J.; NEUBERGER, J. Continuous newton's method. *Sobolev gradients and differential equations*, Springer, p. 79–83, 2010. Cited in page 51.

NGUYEN. Complex-variable newton-raphson load-flow analysis with facts devices. In: IEEE. *2005/2006 IEEE/PES Transmission and Distribution Conference and Exhibition*. [S.l.], 2006. p. 183–190. Cited in page 23.

NGUYEN, H. L. Newton-raphson method in complex form. *IEEE Power Engineering Review*, IEEE, v. 17, n. 8, p. 62–62, 1997. Cited in page 23.

OGBUOBIRI, E. Dynamic storage and retrieval in sparsity programming. *IEEE Transactions on Power Apparatus and Systems*, IEEE, n. 1, p. 150–155, 1970. Cited in page 49.

OGBUOBIRI, E.; TINNEY, W. F.; WALKER, J. W. Sparsity-directed decomposition for gaussian elimination on matrices. *IEEE Transactions on Power Apparatus and Systems*, IEEE, n. 1, p. 141–150, 1970. Cited in page 49.

OKEKE, A. A.; TUMBA, P.; ANORUE, O. F.; DAUDA, A. Analysis and comparative study of numerical solutions of initial value problems (ivp) in ordinary differential equations (ode) with euler and runge kutta methods. *American Journal of Engineering Research (AJER)*, v. 8, n. 8, p. 6–15, 2019. Cited in page 57.

OLIVEIRA, L. N. de; FREITAS, F. D. Computational impacts of freezing the jacobian matrix in the hkw method for power flow applied to ill-conditioned systems. In: IEEE. *2021 Workshop on Communication Networks and Power Systems (WCNPS)*. [S.l.], 2021. p. 1–6. Cited 8 times in pages vi, 52, 62, 74, 75, 86, 93, and 94.

OLIVEIRA, L. N. de; FREITAS, F. D.; MARTINS, N. A modal-based initial estimate for the newton solution of ill-conditioned large-scale power flow problems. *IEEE Transactions on Power Systems*, IEEE, 2023. Cited in page 110.

OVERBYE, T. J.; KLUMP, R. P. Effective calculation of power system low-voltage solutions. *IEEE Transactions on Power Systems*, IEEE, v. 11, n. 1, p. 75–82, 1996. Cited in page 50.

PAN, Z.; WU, J.; DING, T.; LIU, J.; WANG, F.; TONG, X. Load flow calculation for droop-controlled islanded microgrids based on direct Newton-Raphson method with step size optimisation. *IET Generation, Transmission & Distribution*, IET, v. 14, n. 21, p. 4775–4787, 2019. Cited 3 times in pages 2, 50, and 92.

PAN, Z.; WU, J.; DING, T.; LIU, J.; WANG, F.; TONG, X. Load flow calculation for droop-controlled islanded microgrids based on direct newton–raphson method with step size optimisation. *IET Generation, Transmission & Distribution*, Wiley Online Library, v. 14, n. 21, p. 4775–4787, 2020. Cited in page 23.

PANDEY, A.; AGARWAL, A.; PILEGGI, L. Incremental Model Building Homotopy Approach for Solving Exact AC-Constrained Optimal Power Flow. In: *Proceedings of the Annual Hawaii International Conference on System Sciences*. Hawaii: [s.n.], 2021. p. 3273–3282. Disponível em: <http://arxiv.org/abs/2011.00587>. Cited in page 3.

PIRES, R.; CHAGAS, G.; MILI, L. Enhanced power flow solution in complex plane. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 135, p. 107501, 2022. Cited in page 23.

PIRES, R.; MILI, L.; CHAGAS, G. Robust complex-valued levenberg-marquardt algorithm as applied to power flow analysis. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 113, p. 383–392, 2019. Cited in page 23.

POURBAGHER, R.; DERAKHSHANDEH, S. Y. Application of high-order levenberg–marquardt method for solving the power flow problem in the ill-conditioned systems. *IET Generation, Transmission & Distribution*, Wiley Online Library, v. 10, n. 12, p. 3017–3022, 2016. Cited in page 50.

POURBAGHER, R.; DERAKHSHANDEH, S. Y. A powerful method for solving the power flow problem in the ill-conditioned systems. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 94, p. 88–96, 2018. Cited in page 50.

QIN, N. *Voltage control in the future power transmission systems*. [S.l.]: Springer, 2017. Cited 2 times in pages 11 and 12.

RAHMAN, H.; KHAIR, A.; SULTANA, N. A competitive study on the euler and different order runge-kutta methods with accuracy and stability. *Int. J. Sci. Res. in Mathematical and Statistical Sciences Vol*, v. 9, n. 1, 2022. Cited in page 57.

RAO, S.; FENG, Y.; TYLAVSKY, D. J.; SUBRAMANIAN, M. K. The holomorphic embedding method applied to the power-flow problem. *IEEE Transactions on Power Systems*, IEEE, v. 31, n. 5, p. 3816–3828, 2015. Cited in page 51.

RECKTENWALD, G. Stopping criteria for iterative solution methods. *Portland State University, Portland, USA*, 2012. Cited in page 35.

SAAD, Y. *Iterative methods for sparse linear systems*. [S.l.]: SIAM, 2003. Cited 6 times in pages 30, 36, 38, 40, 44, and 45.

SAAD, Y.; SCHULTZ, M. H. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, SIAM, v. 7, n. 3, p. 856–869, 1986. Cited in page 34.

SAADAT, H. *et al. Power System Analysis*. [S.l.]: McGraw-hill, 1999. v. 2. Cited in page 24.

SASSON, A. M.; TREVINO, C.; ABOYTES, F. Improved newton's load flow through a minimization technique. *IEEE Transactions on Power Apparatus and Systems*, IEEE, n. 5, p. 1974–1981, 1971. Cited in page 49.

SCHERZER, O. The use of morozov's discrepancy principle for tikhonov regularization for solving nonlinear ill-posed problems. *Computing*, Springer, v. 51, n. 1, p. 45–60, 1993. Cited 2 times in pages 67 and 68.

SEREETER, B.; ZIMMERMAN, R. D. Ac power flows and their derivatives using complex matrix notation and cartesian coordinate voltages. *TN4-OPF-Derivatives-Cartesian. pdf*, 2018. Cited in page 23.

STOTT, B. Review of load-flow calculation methods. *Proceedings of the IEEE*, IEEE, v. 62, n. 7, p. 916–929, 1974. Cited in page 49.

STOTT, B.; ALSAC, O. Fast decoupled load flow. *IEEE Transactions on Power Apparatus and Systems*, PAS-93, n. 3, p. 859–869, 1974. Cited 2 times in pages 24 and 49.

SÜLI, E.; MAYERS, D. F. *An introduction to numerical analysis*. [S.l.]: Cambridge university press, 2003. Cited in page 34.

TARANTO, G. N.; PONTES, C. E. V.; CAMPELLO, T. M.; ALMEIDA, V. A. F.; GRAHAM, J.; ESMERALDO, P. C. V.; SCHICONG, M. Power flow control for an embedded HVDC link to integrate renewable energy in Brazil. *Electric Power Systems Research*, v. 108504, n. 211, p. 1–10, 2022. Cited in page 1.

TATE, J. E.; OVERBYE, T. J. A comparison of the optimal multiplier in polar and rectangular coordinates. *IEEE Transactions on Power Systems*, IEEE, v. 20, n. 4, p. 1667–1674, 2005. Cited in page 2.

TINNEY, W. F.; HART, C. E. Power Flow Solution by Newton's Method. *IEEE Transactions on Power Apparatus and systems*, IEEE, n. 11, p. 1449–1460, 1967. Cited 2 times in pages 17 and 49.

TOSTADO, M.; KAMEL, S.; JURADO, F. Developed newton-raphson based predictor-corrector load flow approach with high convergence rate. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 105, p. 785–792, 2019. Cited in page 50.

TOSTADO, M.; KAMEL, S.; JURADO, F. An effective load-flow approach based on gauss-newton formulation. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 113, p. 573–581, 2019. Cited in page 50.

TOSTADO-VÉLIZ, M.; ALHARBI, T.; ALRUMAYH, O.; KAMEL, S.; JURADO, F. A novel power flow solution paradigm for well and ill-conditioned cases. *IEEE Access*, v. 9, p. 112425–112438, 2021. Cited 5 times in pages 1, 3, 9, 17, and 55.

TOSTADO-VELIZ, M.; KAMEL, S.; ALQUTHAMI, T.; JURADO, F. A three-stage algorithm based on a semi-implicit approach for solving the power-flow in realistic large-scale ill-conditioned systems. *IEEE Access*, IEEE, v. 8, p. 35299–35307, 2020. Cited 3 times in pages 3, 52, and 63.

TOSTADO-VÉLIZ, M.; KAMEL, S.; JURADO, F. Development of combined runge–kutta broyden's load flow approach for well-and ill-conditioned power systems. *IET Generation, Transmission & Distribution*, IET, v. 12, n. 21, p. 5723–5729, 2018. Cited 3 times in pages 2, 13, and 21.

TOSTADO-VÉLIZ, M.; KAMEL, S.; JURADO, F. Development of combined runge–kutta broyden's load flow approach for well-and ill-conditioned power systems. *IET Generation, Transmission & Distribution*, Wiley Online Library, v. 12, n. 21, p. 5723–5729, 2018. Cited in page 52.

TOSTADO-VÉLIZ, M.; KAMEL, S.; JURADO, F. An efficient power-flow approach based on Heun and King-Werner's methods for solving both well and ill-conditioned cases. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 119, p. 105869, 2020. Cited 20 times in pages 1, 2, 17, 50, 52, 56, 59, 60, 61, 62, 71, 74, 75, 86, 92, 93, 101, 102, 105, and 111.

TOSTADO-VÉLIZ, M.; KAMEL, S.; JURADO, F. Power Flow Approach Based on the S-iteration Process. *IEEE Transactions on Power Systems*, IEEE, v. 35, n. 6, p. 4148–4158, 2020. Cited in page 53.

TOSTADO-VÉLIZ, M.; MATOS, M. A.; LOPES, J. A. P.; JURADO, F. An Improved Version of the Continuous Newton's Method for Efficiently Solving the Power-Flow in Ill-conditioned Systems. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 124, p. 106389, 2021. Cited in page 53.

TOSTADO-VéLIZ, M.; KAMEL, S.; JURADO, F. Development of Different Load Flow methods for Solving Large-scale Ill-conditioned Systems. *International Transactions on Electrical Energy Systems*, v. 29, n. 4, p. e2784, 2019. Cited 2 times in pages 22 and 53.

TREVINO, C. Cases of difficult convergence in load-flow problems. In: *IEEE Summer Power Meeting*. [S.l.: s.n.], 1970. Cited in page 49.

TRIAS, A. The holomorphic embedding load flow method. In: IEEE. *2012 IEEE Power and Energy Society General Meeting*. [S.l.], 2012. p. 1–8. Cited in page 51.

TRIPATHY, S.; PRASAD, G. D.; MALIK, O.; HOPE, G. Load-flow solutions for ill-conditioned power systems by a newton-like method. *IEEE Transactions on Power Apparatus and Systems*, PAS-101, n. 10, p. 3648–3657, 1982. Cited in page 13.

VORST, H. A. Van der. *Iterative Krylov methods for large linear systems*. [S.l.]: Cambridge University Press, 2003. Cited 2 times in pages 36 and 38.

VéLIZ, S. K. M. T.; JURADO, F. Matpower ill-conditioned systems v1. 2019. Cited 3 times in pages 87, 110, and 116.

WARD, J.; HALE, H. Digital computer solution of power-flow problems [includes discussion]. *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, IEEE, v. 75, n. 3, p. 398–404, 1956. Cited in page 48.

WENG, Z. X.; SHI, L. B.; XU, Z.; YAO, L. Z.; NI, Y. X. N.; BAZARGAN, M. Effects of wind power variability and intermittency on power flow. In: *IEEE Power and Energy Society General Meeting*. [S.l.: s.n.], 2012. Cited in page 1.

ZIMMERMAN, R. D. Ac power flows, generalized opf costs and their derivatives using complex matrix notation. *MATPOWER Tech. Note*, v. 2, p. 60–62, 2010. Cited in page 135.

ZIMMERMAN, R. D.; MURILLO-SÁNCHEZ. Matpower cases (version 7.1). 2020. Disponível em: <Available:http://matpower.org>. Cited 2 times in pages 87 and 92.

ZIMMERMAN, R. D.; MURILLO-SÁNCHEZ, C. E.; THOMAS, R. J. Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on power systems*, IEEE, v. 26, n. 1, p. 12–19, 2011. Cited 4 times in pages 74, 86, 94, and 100.

APÊNDICE A

HANDLING COMPLEX NUMERICAL EXPRESSION FOR THE RESOLUTION OF THE POWER FLOW PROBLEM

This appendix details the complex expression used by its team to implement the power flow in MATPOWER. The expressions are basics for the computation of both matrix Jacobian and mismatches. However, all results are used in real and imaginary form to perform the power flow problem iteration calculations (ZIMMERMAN, 2010).

A.1 MAKESBUS FUNCTION

The function `makeSbus()` builds the vector of complex bus power injections. The syntax used in the program is presented below.

`SBUS = makeSbus(BASEMVA, BUS, GEN, MPOPT, VM, SG)`: returns the vector of complex bus power injections, that is, generation minus load. Power is expressed in per unit. In which:

`BASEMVA`: Value specifying the system MVA base used for converting power into per unit quantities.

`BUS`: Vector corresponds to the buses of the system.

`GEN`: Vector corresponds to the PV bus types of the system.

`MPOPT`: MATPOWER options struct to override default options can be used to specify the solution algorithm, output options termination tolerances.

`VM`: Vector of voltage magnitudes

`SG`: Complex power of generation buses.

If the MPOPT and VM arguments are present it evaluates any ZIP loads based on the provided voltage magnitude vector. If VM is empty, it assumes nominal voltage. If SG is provided, it is

a complex ng x 1 vector of generator power injections in p.u., and overrides the PG and QG columns in GEN, using GEN only for connectivity information. Another syntax used for the function is presented below.

`[SBUS, DSBUS_DVM] = MAKESBUS(BASEMVA, BUS, GEN, MPOPT, VM)` : With two output arguments, it computes the partial derivative of the bus injections with respect to voltage magnitude, leaving the first return value SBUS empty. If VM is empty, it assumes no voltage dependence and returns a sparse zero matrix.

A.2 DSBUS_DV FUNCTION

The function `dSbus_dV()` is used in the MATPOWER to evaluate the partial derivatives of $\mathbf{S_{bus}}$ with respect to $\mathbf{V}$, where $\mathbf{V}$ represents the complex bus voltages. These derivatives are later used to construct the Jacobian matrix within the `newtonpf.m` code. Below, the main function and variables associated with the `dSbus_dV` code are described:

`[dSbus_dV1, dSbus_dV2] = dSbus_dV(YBUS, V, vcart)`: When `vcart = 1`, the function returns two matrices containing the partial derivatives of the complex bus power injections with respect to the real and imaginary parts of the voltage, respectively, for all buses. In which:

`dSbus_dV1` corresponds partial derivative of the complex bus power injections with respect to the real part of the voltage;

`dSbus_dV2` corresponds partial derivative of the complex bus power injections with respect to the imaginary part of the voltage.

A.3 NEWTONPF FUNCTION

The function `newtonpf` is an implementation of the Newton-method power flow solver. To construct the Jacobian matrix, the following functions are used:

```
[dSbus_dVm, dSbus_dVa] = dSbus_dV(Ybus, V);
```

```
[dummy, neg_dSd_dVm] = Sbus(Vm);
```

```
dSbus_dVm = dSbus_dVm - neg_dSd_dVm;
```

These functions were introduced in Sections A1 and A2 and then the real and imaginary parts of the `dSbus_dVm` function are used to construct the Jacobian matrix.