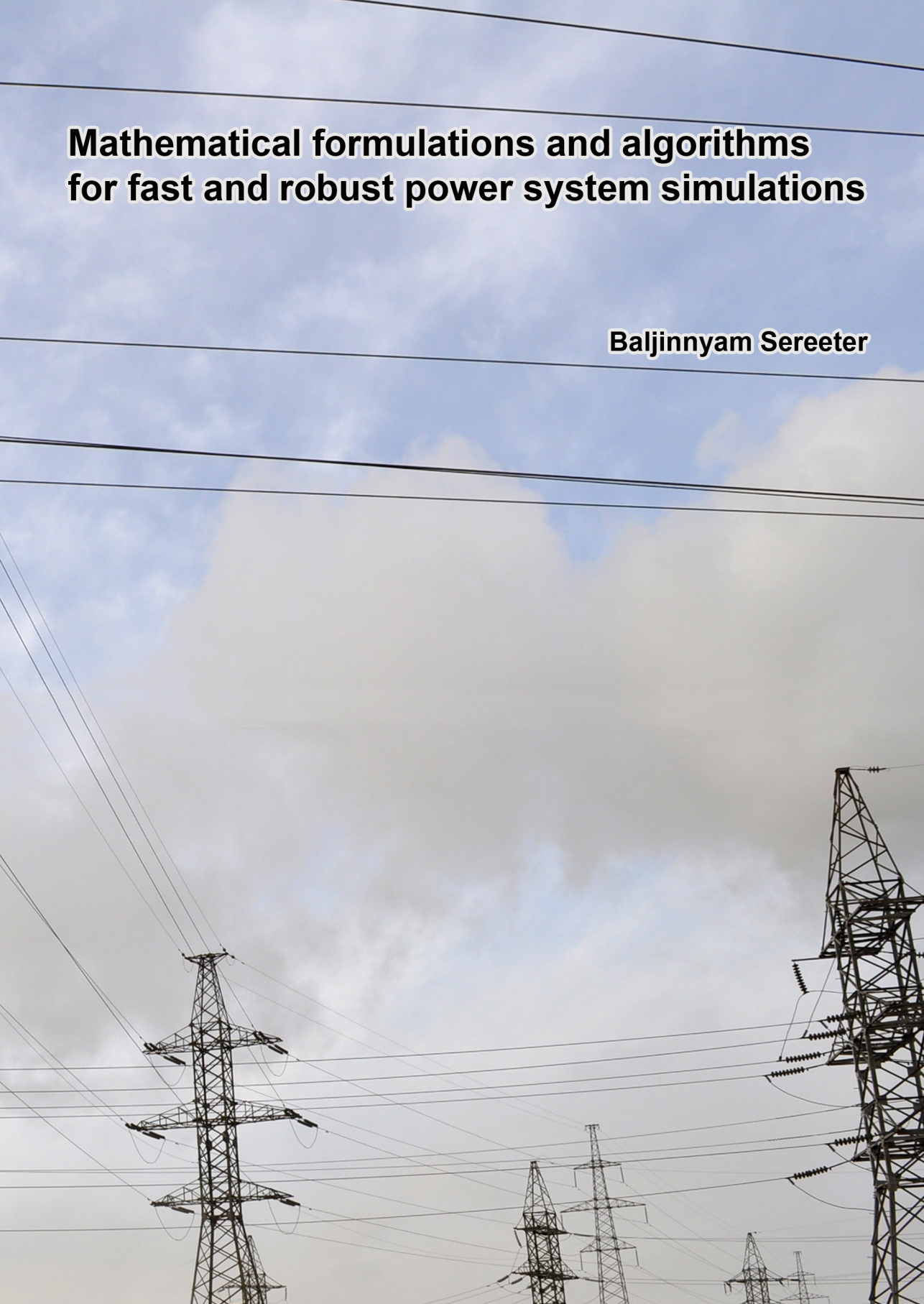
Mathematical formulations and algorithms for fast and robust power system simulations

Baljinnyam Sereeter



Mathematical formulations and algorithms for fast and robust power system simulations

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.dr.ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op
woensdag 18 maart 2020 om 10:00 uur

door

Baljinnyam SEREETER
Master of Science in Applied Mathematics, Technische Universiteit Delft, Nederland
Master of Science in Scientific Computing, Technische Universiteit Berlijn, Duitsland

geboren te Zavkhan, Mongolië

Dit proefschrift is goedgekeurd door de

promotor: Prof.dr.ir. C. Vuik
promotor: Prof.dr. C. Witteveen

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof.dr.ir. C. Vuik	Technische Universiteit Delft, promotor
Prof.dr. C. Witteveen	Technische Universiteit Delft, promotor
Onafhankelijke leden:	
Prof.dr. J.L. Hurink	University of Twente
Prof.dr. P. Palensky	Technische Universiteit Delft
Prof.dr. M. Gibescu	University of Utrecht
Prof.dr.ir. C.W. Oosterlee	Technische Universiteit Delft
Prof.dr.ir. H.X.Lin	Technische Universiteit Delft

Keywords: *Power flow analysis, Nonlinear power flow problem, Newton–Raphson method, Power mismatch formulation, Current mismatch formulation, Optimal Power Flow problem, Interior Point Method, Linear power flow problem, Unbalanced distribution networks, Numerical analysis, Krylov subspace methods*

Mathematical formulations and algorithms for fast and robust power system simulations
Dissertation at Delft University of Technology.
Copyright © 2020 by B. Sereeter (e-mail: baljinnyamss@gmail.com)





ISBN 978-94-6384-119-1

Printed by: ProefschriftMaken (www.proefschriftmaken.nl)
Cover design by: Batdorj Tsedev (www.selba.mn)

To my grandfather Vandan Burne

Contents

Summary ix

Samenvatting xi

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Power system simulations . . . 2
- 1.3 Thesis outline . . . 3

2 Solution methods 5
- 2.1 Linear system of equations . . . 5
  - 2.1.1 Direct solvers . . . 5
  - 2.1.2 Iterative solvers . . . 7
- 2.2 Nonlinear system of equations . . . 9
  - 2.2.1 Newton-Raphson method . . . 9
- 2.3 Optimization problems . . . 10
  - 2.3.1 Primal-Dual Interior Point method . . . 10

3 Power system modeling 13
- 3.1 Power systems . . . 13
- 3.2 Transmission lines . . . 14
- 3.3 Loads . . . 14
  - 3.3.1 Three-phase loads . . . 15
- 3.4 Generators . . . 16
- 3.5 Shunts . . . 16
- 3.6 Transformers . . . 17
  - 3.6.1 Three-phase transformers . . . 17
- 3.7 Network admittance . . . 18

4 Nonlinear Power Flow Computations 21
- 4.1 Introduction . . . 22
- 4.2 Nonlinear Power Flow problem . . . 24
  - 4.2.1 The power-mismatch formulation . . . 25
  - 4.2.2 The current-mismatch formulation . . . 26
- 4.3 Newton Power Flow methods . . . 27
  - 4.3.1 Polar power-mismatch version (NR-p-pol [1]) . . . 28
  - 4.3.2 Cartesian power-mismatch version (NR-p-car) . . . 28

4.3.3 Complex power-mismatch version (NR-p-com) . . . 29
4.3.4 Polar current-mismatch version (NR-c-pol) . . . 29
4.3.5 Cartesian current-mismatch version (NR-c-car) . . . 30
4.3.6 Complex current-mismatch version (NR-c-com) . . . 31
4.3.7 First order partial derivatives of mismatch functions . . . 32
4.3.8 Corrections . . . 32
4.4 Numerical results . . . 34
4.4.1 Comparison between single-phase and three-phases . . . 34
4.4.2 Single-phase nonlinear power flow computations . . . 36
4.4.3 Three-phase nonlinear power flow computations . . . 41
4.5 Conclusion . . . 43

5 Linear Power Flow Computations 45
5.1 Introduction . . . 45
5.2 Linear Power Flow problem . . . 46
5.3 Linear Power Flow methods . . . 48
5.3.1 Direct approach . . . 49
5.3.2 Iterative approach . . . 49
5.4 Numerical results . . . 50
5.4.1 Direct approach . . . 51
5.4.2 Iterative approach . . . 53
5.5 Conclusion . . . 54

6 Application of NA techniques on MV/LV network simulations 57
6.1 Introduction . . . 57
6.2 Case study of large Dutch power grid (LLPF) . . . 58
6.2.1 Data and assumptions . . . 58
6.2.2 Solving in terms of only real numbers . . . 59
6.2.3 Simulation results . . . 61
6.3 Application of NA techniques . . . 62
6.4 Numerical Results . . . 64
6.4.1 LLPF problem with real components . . . 64
6.4.2 LLPF problem with complex components . . . 65
6.5 Conclusion . . . 66

7 Optimal Power Flow Computations 67
7.1 Introduction . . . 67
7.2 Optimal Power Flow problem . . . 69
7.2.1 Variables . . . 69
7.2.2 Objective function . . . 70
7.2.3 Equality constraints . . . 70
7.2.4 Inequality constraints . . . 70
7.2.5 Four equivalent formulations of the OPF problem . . . 71
7.3 Interior Point Method (MIPS) . . . 71
7.3.1 Derivatives of objective function $f(\mathbf{x})$ . . . 72
7.3.2 Derivatives of equality constraints $g(\mathbf{x})$ . . . 72
7.3.3 Derivatives of inequality constraints $h(\mathbf{x})$ . . . 74
7.4 Numerical Results . . . 75

7.4.1 Number of nonzero elements 76
7.4.2 Number of iterations 77
7.4.3 CPU time on each iteration 79
7.5 Conclusion 79

8 Conclusions and Recommendations **83**
8.1 Conclusions 83
8.2 Recommendations 86

A Numerical results for the LPF method handling PV buses **89**

B Derivatives $f_{\mathbf{xx}}$, $g_{\mathbf{xx}}$ and $h_{\mathbf{xx}}$ **91**
B.1 Derivatives of objective function $f(\mathbf{x})$ 91
B.2 Derivatives of equality constraints $g(\mathbf{x})$ 91
B.3 Derivatives of inequality constraints $h(\mathbf{x})$ 95

Bibliography **107**

Curriculum Vitae **109**

List of Scientific Activities **111**

Acknowledgements **113**

Summary

Mathematical formulations and algorithms for fast and robust power system simulations

Baljinnyam Sereeter

During the normal operation, control and planning of the power system, grid operators employ numerous tools including the Power Flow (PF) and the Optimal Power Flow (OPF) computations to keep the balance in the power system. The solution of the PF computation is used to assess whether the power system can function properly for the given generation and consumption, whereas the OPF problem provides the optimal operational state of the electrical power system, while satisfying system constraints and control limits.

In this thesis, we study advanced models of the power system that transform the physical properties of the network into mathematical equations. Furthermore, we develop new mathematical formulations and algorithms for fast and robust power system simulations, such as PF and OPF computations, that can be applied to any balanced single-phase or unbalanced three-phase network.

The power flow, or load flow, problem is the problem of computing the voltages in each bus of a power system where the power generation and consumption are given. Mathematically, the power flow problem comes down to solving a nonlinear system of equations where all variables are given in complex numbers. In practice, the Newton power flow method using the power balance equations in polar coordinates is preferred in terms of quadratic convergence. In order to obtain the required fast and robust PF solution method for an changing electrical power system, we examine all six formulations of the PF problem using two different mismatch formulations: the current and power balance equations, and three different coordinate systems: Cartesian, polar, and complex form. Moreover, we develop new versions of the Newton power flow method based on all six formulations of the PF problem. Our newly developed versions are compared with the existing variants of the Newton power flow method for both balanced single-phase and unbalanced three-phase networks in terms of the computational speed and robustness. Two Newton power flow variants developed in this thesis are proven to be faster and more robust than the existing Newton power flow methods.

We introduce the new approach to linearize the original nonlinear PF problem using the connection between actual buses in the network to artificial ground buses. Direct and iterative methods are developed in this research work to solve the resulting Linear Power Flow (LPF) problem. Accuracy and efficiency of both direct and itera-

tive linear approaches are validated by comparing them with the conventional Newton power flow algorithm on various transmission and distribution networks. The direct LPF method is further improved with Numerical Analysis (NA) techniques to solve very large LPF problems with 27 million buses simulating both the entire LV and MV networks in a single simulation. Reordering technique (RCM), a couple of direct solvers (Cholesky, IC, LU, and ILU), and various Krylov subspace methods (CG, PCG, GMRES, and BiCGSTAB) are used to improve the computational time of the direct LPF method. We confirm that our LPF algorithms are very fast and user friendly for power flow computations on a large distribution network.

The OPF problem is an optimization problem that has an objective function, equality, and inequality constraints. There is no method that is the best for all OPF problems, because each OPF problem results in an optimization problem with different properties depending on the choice of objective functions, control variables and system constraints. In this thesis, we consider the OPF problem with minimization of active power generation costs as an objective function, nonlinear power flow equations as equality constraints, and squared apparent power limits as inequality constraints. Furthermore, we study four equivalent mathematical formulations of the OPF problem and their computational impacts on the performance of the OPF solution methods. In order to identify the formulation that results in the best convergence characteristics for the solution method, we apply MIPS (Matpower's Interior Point Method), KNITRO (Commercial software package for solving large scale nonlinear optimization problems), and FMINCON (Matlab's optimization solver) on various test cases. We compare all four formulations in terms of impact factors on the solution method such as a number of nonzero elements in the Jacobian and Hessian matrices, the number of iterations and computational time on each iteration. Our numerical results verify that the performance of any OPF solution method can be improved by changing the mathematical formulation used to specify the OPF problem while keeping the same algorithm.

Mathematical formulations and computational methods based on this thesis are implemented in *Matpower 7.0* for future research and practical use.

Samenvatting

Wiskundige formuleringen and algoritmes voor snelle en robuuste simulaties van energievoorzieningssystemen

Baljinnyam Sereeter

Tijdens standaard beheer, controle en planning van het energievoorzieningssysteem passen netbeheerders verscheidene technieken toe, waaronder de Power Flow (PF) en de Optimal Power Flow (OPF) berekeningen, om het energievoorzieningssysteem in balans te houden. De oplossing van de PF berekening wordt gebruikt om vast te stellen of het energievoorzieningssysteem naar behoren kan functioneren voor het gegeven verbruik en de gegeven opwekking. Het OPF probleem geeft de optimale operationele staat van het elektrische energievoorzieningssysteem, zodanig dat aan systeemeisen en controle limieten voldaan wordt.

In dit proefschrift bestuderen we geavanceerde modellen van het energievoorzieningssysteem, die de fysische eigenschappen van het netwerk vertalen naar wiskundige vergelijkingen. Verder ontwikkelen we nieuwe wiskundige formuleringen en algoritmes voor snelle en robuuste simulaties van het energievoorzieningssysteem, zoals PF en OP berekeningen, die toegepast kunnen worden op ieder gebalanceerd eenfase of ongebalanceerd driefase netwerk.

Het power flow, of load flow, probleem beschrijft het probleem om de spanningen in iedere bus van elektriciteitsnet te berekenen, waarbij het energieverbruik en de energieopwekking gegeven zijn. Wiskundig gezien komt het power flow probleem neer op het oplossen van een stelsel van niet-lineaire vergelijkingen, waarbij alle variabelen complexe getallen zijn. In de praktijk wordt de voorkeur gegeven aan de Newton power flow methode met vermogensbalansvergelijkingen in poolcoördinaten vanwege de kwadratische convergentie. Om tot de benodigde snelle en robuuste PF oplossing voor een veranderd elektriciteitsnet te komen, onderzoeken we alle zes formuleringen van het PF probleem, gebruikmakend van twee formuleringen van de residuvergelijking: the stroom- en vermogensbalansvergelijking, en gebruikmakend van drie verschillende coördinatenstelsels: Cartesisch, pool, en complexe vorm. Bovendien ontwikkelen we nieuwe versies van de Newton power flow methode gebaseerd op alle zes formuleringen van het PF probleem. Onze nieuw ontwikkelde versies worden vergeleken met de bestaande varianten van de Newton power flow methode, voor zowel gebalanceerde eenfase als ongebalanceerde driefase netwerken, in termen van rekensnelheid en robuustheid. Twee van de Newton power flow methoden ontwikkeld in dit proefschrift zijn sneller en robuuster dan de bestaande Newton power flow methoden.

We introduceren een nieuwe manier om het oorspronkelijk niet-lineaire PF prob-

leem te lineariseren, waarbij we gebruik maken van de verbindingen tussen echte netwerkbussen en artificiële grondbussen. Directe en iteratieve methoden worden ontwikkeld in dit proefschrift voor het oplossen van het resulterende Linear Power Flow (LPF) probleem. Nauwkeurigheid en efficiëntie van zowel directe als iteratieve lineaire benaderingen worden gevalideerd door de benaderingen te vergelijken met het conventionele Newton power flow algoritme voor verschillende transmissie- en distributienetwerken. De directe LPF methode wordt verder verbeterd met Numerieke Analyse (NA) technieken, om een erg groot LPF probleem, bestaande uit 27 miljoen bussen die zowel een volledig laagspannings- als middenspanningsnet simuleren, op te lossen in een enkele simulatie. Reordering technieken (RCM), een aantal directe solvers (Cholesky, IC, LU en ILU) en verschillende Krylov subspace methoden (CG, PCG, GMRES en BiCGSTAB) worden gebruikt om de rekentijd van de directe LPF methode te verbeteren. We bevestigen dat onze LPF algoritmes zeer snel en gebruiksvriendelijk zijn voor power flow berekeningen van een groot distributienet.

Het OPF probleem is een optimalisatieprobleem met een kostfunctie, gelijkheids- en ongelijkheidsrandvoorwaarden. Er bestaat geen methode die het beste is voor alle OPF problemen, omdat ieder OPF probleem leidt tot een optimalisatieprobleem met verschillende eigenschappen, afhankelijk van de keuze voor de kostfunctie, controlevariabelen en systeemeisen. In dit proefschrift beschouwen we het OPF probleem met de minimalisatie van de kosten van de opwekking van actief vermogen als kostfunctie, de niet-lineaire power flow vergelijkingen als gelijkheidsrandvoorwaarden, en de limieten voor gekwadrateerd schijnvermogen als ongelijkheidsrandvoorwaarden. We bestuderen bovendien vier equivalente wiskundige formuleringen van het OPF probleem en hun computationele impact op de prestatie van de OPF oplossingsmethoden. Om de formulering te identificeren die leidt tot de beste convergentiekarakteristieken van de oplossingsmethoden, passen we MIPS (Matpowers Interior Point Method), KNITRO (Commercieel software pakket voor het oplossen van grote niet-lineaire optimalisatieproblemen) en FMINCON (Matlabs optimalisatie solver) toe op verschillende testproblemen. We vergelijken alle vier de formuleringen in termen van impactfactoren op de oplossingsmethode, zoals het aantal niet-nul elementen in de Jacobiaan en Hessiaan matrices, het aantal iteraties en rekentijd van iedere iteratie. Onze numerieke resultaten bevestigen dat de prestatie van iedere OPF oplossingsmethode verbeterd kan worden door de wiskundige formulering, gebruikt om het OPF probleem te specificeren, te veranderen terwijl hetzelfde algoritme gebruikt wordt.

De wiskundige formuleringen en computationele methoden gebaseerd op dit proefschrift zijn geïmplementeerd in *Matpower 7.0* voor toekomstig onderzoek en praktisch gebruik.

Chapter 1

Introduction

1.1 Background

Electrical power systems are one of the most complex system types built by engineers [2]. They continuously provide us the electricity that is used in every second of our modern life. A power grid is a network of electrical components such as generators, transformers, transmission lines, distribution feeders, substations, loads and so on as shown in Figure 1.1. Traditionally, electricity was generated by a small number of large bulk power plants that used coal, oil, or nuclear fission and was delivered to the consumers through the power system in one-way traffic (vertical). In practice, this whole process is called **centralized generation**. Furthermore, it has been known for a long time that these conventional approaches of the power generation are not sustainable and environmental friendly.

Due to the modernization of the existing electrical grid, a large number of newly developed grid elements including smart meters, smart appliances, renewable energy resources, and storage devices are being integrated into the power system. Thus, the existing electrical grid is changing rapidly and becoming more and more complex to control for grid operators. A Smart Grid (SG) is offered as the solution to this problem [4–6]. One can find many concepts and descriptions of SG, but we prefer the following definition:
"Smart Grid is a modernized electrical grid that uses analog or digital information and communications technology to gather and act on information, such as information about the behaviors of suppliers and consumers, in an automated fashion to improve the efficiency, reliability, economics, and sustainability of the production and distribution of electricity" [7].

In SG, most of the new grid elements are directly connected to the distribution network. Conventionally, the distribution network has been considered as a passive network that depends on the transmission network for control and regulation of system parameters. However, due to the utilization of renewable energy resources, the distribution network becomes an active network in a way that electrical power can be consumed and generated in the distribution network. For example, the distribution network is now capable of providing energy supply, frequency control and voltage

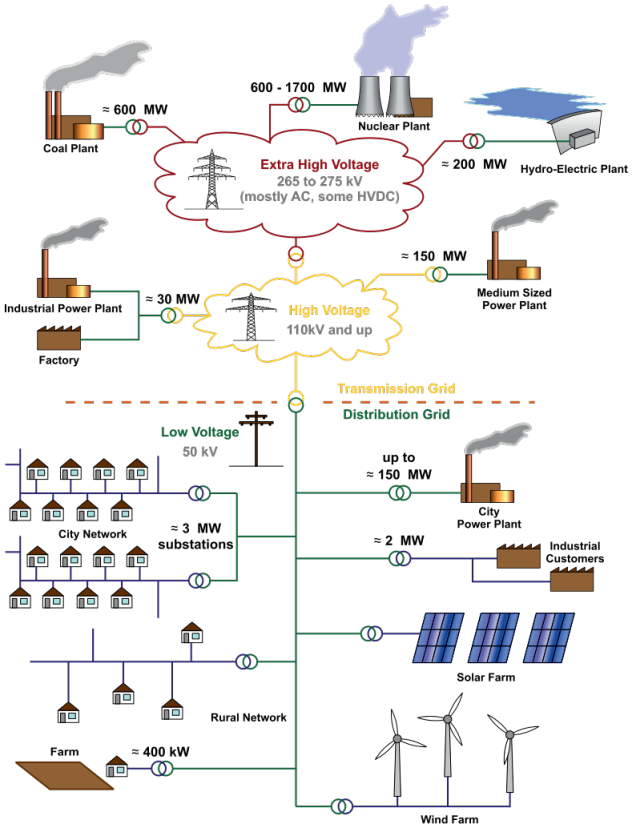


Figure 1.1: *Power system schematic [3]*

regulation without any interaction from the transmission network. Besides, if there is more power generation than consumption, then extra power can be delivered to other networks through the transmission network. This changes the conventional structure of the power system and makes the direction of the power flow in the network into two-way traffic (horizontal). Furthermore, this new process is called **decentralized or distributed generation**. Central grid operators or transmission system operators (TSOs) of the power system must have different approaches for maintaining and operating the electrical power grid because in this case, the primary purpose of the operator has been adjusted to interconnect the various active distribution networks.

1.2 Power system simulations

The power system is designed and organized in a way that the amount of power consumptions and generations must be in balance at all time. However, consumer demand changes continuously over time, and therefore the balance between the amount of generated and consumed powers has to be maintained by control actions. During the normal operation of the power system, numerous procedures are employed by

grid operators for the different time frame to keep the balance in the power system. Depending on the time scale, grid operators perform the *steady-state* (seconds to years) or *dynamic* (milliseconds) or *transient* (microseconds) analysis. In this thesis, we focus more on *steady-state analysis*, namely the *Power Flow* (PF) and *Optimal Power Flow* (OPF) computations. The solution of the PF computation is used to assess whether the power system can function properly for the given generation and consumption whereas the OPF problem provides the optimal operational state of the electrical power system while satisfying system constraints and control limits. Therefore, PF and OPF computations are performed in power system operation, control and planning.

Traditionally, PF computations were calculated only in the transmission network, and the distribution networks were aggregated as buses in the power system model. However, in the new operation and maintenance of the distribution network, PF computations must be done on the distribution network as well. If the LV distribution network is included in the power system simulations, the size of the problem is measured in the order of million buses. Thus, very fast computational methods are required for real-time monitoring and control of the smart grid infrastructure. The conventional PF solution methods do not always converge when they are applied to the distribution power flow problem due to some special features of the distribution network. In addition, many sub-classes of the OPF problem have been developed over the years on both transmission and distribution networks. But there is no method that is the best for all OPF problems because each OPF problem results in an optimization problem with different properties depending on the choice of objective functions, control variables and system constraints. Therefore, current simple mathematical models and computational methods are considered insufficient in a complex electrical network.

1.3 Thesis outline

In this thesis, we aim to develop new advanced mathematical formulations and algorithms for fast and robust power system simulations such as PF and OPF computations that are required for changing electrical power system. The main objectives are defined as follows:

1. Consider both balanced single-phase transmission and unbalanced three-phase distribution networks.
2. Develop a robust PF solution method that can be applied to distribution networks with any special characteristics.
3. Implement a fast PF algorithm for large hybrid power system simulations.
4. Formulate new mathematical equations for fast and robust OPF computations.

This thesis is based on the author's publications [8–15] and structured as follows:

- Chapter 2 explores the general theories and solution techniques for mathematical equations that are determined in this thesis, such as linear systems of equations, nonlinear systems of equations, and constrained optimization problems.

- In Chapter 3, we introduce the power system modeling that transforms the physical properties of the network into mathematical equations. We describe mathematical modeling of electrical power systems and its components such as transmission lines, loads, generators, and transformers. Both balanced single-phase and unbalanced three-phase networks are considered.

- In Chapter 4, we address research objectives 1 and 2 based on author's publications [8, 9]. Various mathematical formulations of the nonlinear power flow problem are studied in details. A general framework is given for applying the Newton-Raphson method to solve nonlinear power flow problems, using power and current-mismatch functions in polar, Cartesian coordinates and complex form. We develop new versions of the Newton power flow method that are faster and more robust than existing versions. Each new Newton power flow variant is theoretically explained and compared with current versions in terms of the difference on balanced transmission and unbalanced distribution networks.

- We propose a linear formulation of the original nonlinear power flow problem in Chapter 5. This chapter explores research objective 2 using author's publication [10]. The theoretical background is given for the linearization of the nonlinear power flow problem. We develop the direct and iterative algorithms for the linear power flow problem. Accuracy and efficiency of both direct and iterative linear approaches are validated by comparing it with the conventional Newton power flow algorithm on various transmission and distribution networks.

- Chapter 6 proposes a fast linear power flow algorithm improved with numerical analysis techniques based on author's publications [11, 12]. The research objective 3 is studied in this chapter. We solve the Large Linear Power Flow (LLPF) problem with 27 million buses simulating both the entire LV and MV networks of Alliander DNO in a single simulation. In the numerical analysis, different reordering techniques, numerous direct solvers and various Krylov subspace methods are chosen and applied to the LLPF problem with both real and complex components.

- In Chapter 7 considering research objective 4, we study four equivalent mathematical formulations of the OPF problem and their impacts on the performance of solution methods. We show how four mathematical formulations of the OPF problem can be obtained by rewriting equality constraints while keeping the same physical formulation. In order to identify the formulation that results in the best convergence characteristics for the solution method, we apply three different solution methods on various test cases using multiple initial conditions. This chapter is based on author's publications [13–15].

- Finally, we give the general conclusions of this thesis and some recommendations for the application of our methods and future research in Chapter 8.

Chapter 2

Solution methods

Throughout the thesis, we compute the linear power flow problems, nonlinear power flow problems, and optimal power flow problems on various levels and types of electrical networks. Mathematically, these power-related problems are given as linear systems of equations, nonlinear systems of equations, and constrained optimization problems. In this chapter, we introduce the general theories and solution techniques for these mathematical equations. This chapter starts exploring a linear system of equations and their solution techniques such as direct solvers (LU and Cholesky decompositions) and iterative solvers (Conjugate Gradient, Bi-Conjugate Gradient Stabilized, and Generalized Minimal Residual). Later, a nonlinear system of equations is considered with its well-known solution technique the Newton-Raphson method. Finally, the Primal-Dual Interior-Point approach is explained for the optimization problem with equality and inequality constraints.

2.1 Linear system of equations

Let us consider a linear system of equations with a square coefficient matrix $A \in \mathbb{R}^{n \times n}$:

$$A\mathbf{x} = \mathbf{b}, \tag{2.1}$$

where $\mathbf{b} \in \mathbb{R}^n$ is the right-hand side vector and $\mathbf{x} \in \mathbb{R}^n$ is the vector of unknowns. A linear system (2.1) is called consistent if there is at least one solution, otherwise, it is called inconsistent. The solution method for a linear system (2.1) is often chosen depending on some properties of the coefficient matrix A such as size, eigenvalues, sparseness, symmetry and positive definiteness. There are two types of solvers for linear systems, namely direct solvers and iterative solvers.

2.1.1 Direct solvers

Direct methods solve the linear system in one attempt by efficiently computing the inverse of the coefficient matrix A as $\mathbf{x} = A^{-1}\mathbf{b}$. In practice, direct solvers use Gaussian elimination techniques to factorize the coefficient matrix A.

LU Decomposition

An invertible coefficient matrix A can be decomposed into a lower triangular matrix L and an upper triangular matrix U as:

$$A = LU, \tag{2.2}$$

where the diagonal elements of L are all set equal to one. There exists an LU decomposition for every invertible matrix A if partial pivoting is allowed which are the row permutations of A. The linear system (2.1) can be rewritten by substituting the decomposed form (2.2):

$$A\mathbf{x} = LU\mathbf{x} = \mathbf{b} \tag{2.3}$$

and the unknown vector $\mathbf{x}$ is obtained by using forward and backward substitutions:

$$L\mathbf{y} = \mathbf{b}, \tag{2.4}$$

$$U\mathbf{x} = \mathbf{y}. \tag{2.5}$$

Incomplete LU decomposition

Incomplete LU (ILU) decomposition factorizes the coefficient matrix A approximation into a lower triangular matrix L and an upper triangular matrix U as:

$$A \approx LU, \tag{2.6}$$

where entries of L and U that are below a certain tolerance, are all made equal to zero. Generally, ILU results in less computational work for forward and backward substitutions compared to the complete LU decomposition due to the smaller number of entries in the L and U matrices. ILU(p) denotes the ILU with fill-in where p is the level of fill-in. In practice, an ILU decomposition is more used as a preconditioner for iterative solvers than as a direct solver.

Cholesky decomposition

In case of Symmetric and Positive Definite (SPD) coefficient matrix A, the linear system (2.1) can be factorized using Cholesky decomposition instead of LU as:

$$A\mathbf{x} = LL^T\mathbf{x} = \mathbf{b}. \tag{2.7}$$

where L is a lower triangular matrix. Again the decomposed linear system (2.7) can be solved with forward and backward substitutions.

Incomplete Cholesky decomposition

The sparse approximation of the Cholesky decomposition is called an Incomplete Cholesky (IChol) factorization where we obtain a product of a lower triangular matrix and its transpose that approximates the SPD coefficient matrix A as:

$$A \approx LL^T. \tag{2.8}$$

The decomposition (2.8) is often used as a preconditioner for the Conjugate Gradient method than as a direct solver.

2.1.2 Iterative solvers

Iterative methods solve the linear system until the best approximation of the exact solution $\mathbf{x}$ is obtained after some number of iterations. Normally, iterative methods start with an initial vector $\mathbf{x}^0$ and update the vector in every iteration k and stop the iterative process when the iterate $\mathbf{x}^k$ is close enough to the exact solution $\mathbf{x}$. Since the exact solution $\mathbf{x}$ is unknown, the residual vector $\mathbf{r}^k = \mathbf{b} - A\mathbf{x}^k$ is used instead of the error vector $\mathbf{e}^k = \mathbf{x} - \mathbf{x}^k$ to measure the accuracy of iterate $\mathbf{x}^k$. The most common stopping criterion or measurement for the error in $\mathbf{x}^k$ is the relative residual error given as:

$$r^k = \frac{||\mathbf{r}^k||}{||\mathbf{b}||}, \tag{2.9}$$

where $||\cdot||$ is the 2-norm of the vector. There are various basic iterative methods such as Jacobi, Gauss-Seidel, Richardson and the Successive Overrelaxation which can be found in [16, 17].

Krylov subspace methods

The Krylov subspace of dimension k corresponding to matrix A and initial residual $\mathbf{r}^0$ is defined as:

$$\mathcal{K}_k(A, \mathbf{r}^0) := span\{\mathbf{r}^0, A\mathbf{r}^0, A^2\mathbf{r}^0, ..., A^{k-1}\mathbf{r}^0\}, \tag{2.10}$$

and the iterate $\mathbf{x}^k$ of the Krylov subspace method is computed as:

$$\mathbf{x}^k \in \mathbf{x}^0 + \mathcal{K}_k(A, \mathbf{r}^0). \tag{2.11}$$

In the following sections, we examine briefly three Krylov subspace methods that are used in our experiments.

Conjugate Gradient method

The Conjugate Gradient (CG [18]) method is used to solve the linear system of equations with a SPD matrix such that the error $e_k = ||\mathbf{x} - \mathbf{x}^k||_{\mathbf{A}}$ in the A norm is minimal where the A norm of vector $\mathbf{x}$ is defined as:

$$||\mathbf{x}||_A = \sqrt{\mathbf{x}^T A\mathbf{x}}. \tag{2.12}$$

Algorithm 1 shows the iteration process of the CG method. The converge of CG is defined by the following inequality:

$$||\mathbf{x} - \mathbf{x}^{k+1}||_A \leq 2||\mathbf{x} - \mathbf{x}^0||_A \left(\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1} \right)^{k+1}, \tag{2.13}$$

where $\kappa_2(A)$ is the condition number of the matrix A. Since A is a SPD matrix, the condition number is computed as $\kappa_2(A) = \frac{\lambda_{\max}}{\lambda_{\min}}$ where λ is the eigenvalues of the matrix A. Equation (2.13) informs that the convergence of the CG method highly depends on the condition number of $\kappa_2(A)$ which can be improved by clustering the spectrum.

Algorithm 1 Conjugate Gradient method

Set $k := 0$ and give initial iterate $\mathbf{x}^0$
Compute $\mathbf{r}^0 = \mathbf{b} - A\mathbf{x}^0$, and set $\mathbf{p}^0 = \mathbf{r}^0$
while not converged
    $\mathbf{w}^k = A\mathbf{p}^k$
    $\alpha^k = \frac{\langle \mathbf{r}^k, \mathbf{r}^k \rangle}{\langle \mathbf{w}^k, \mathbf{p}^k \rangle}$
    $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{p}^k$
    $\mathbf{r}^{k+1} = \mathbf{r}^k - \alpha^k \mathbf{w}^k$
    $\beta^k = \frac{\langle \mathbf{r}^{k+1}, \mathbf{r}^{k+1} \rangle}{\langle \mathbf{r}^k, \mathbf{r}^k \rangle}$
    $\mathbf{p}^{k+1} = \mathbf{r}^{k+1} + \beta^k \mathbf{p}^k$
    $k := k + 1$
end while

Bi-Conjugate Gradient Stabilized method

The Bi-Conjugate Gradient Stabilized (Bi-CGSTAB [19]) is a variant of Conjugate Gradient Squared (CGS [20]) method and one of the most widely used Krylov subspace methods for general matrices. In Bi-CGSTAB, the residual vector is given as:

$$\mathbf{r}^k = Q_k(A)P_k(A)\mathbf{r}^0 \tag{2.14}$$

where P_k is a polynomial of degree k such that $P_k(0) = 1$, and Q_k is a polynomial of degree k defined recursively at each iteration. A simple recurrence for Q_k can be defined as:

$$Q_{k+1}(x) = (1 - \omega_k x)Q_k(x). \tag{2.15}$$

where $\omega_k \in \mathbb{R}$ is a constant to be determined. Bi-CGSTAB stabilizes the irregular convergence behavior of the CGS method by correctly determining ω_k for the recurrence that is minimizing the residual $\mathbf{r}^k$. An implementation of Bi-CGSTAG is presented in Algorithm 2.

Algorithm 2 Bi-CGSTAB method

Give initial iterate $\mathbf{x}^0$
Compute $\mathbf{r}^0 = \mathbf{b} - A\mathbf{x}^0$
Choose arbitrary $\bar{\mathbf{r}}^0$ s.t. $(\bar{\mathbf{r}}^0)^T \mathbf{r}^0 \neq 0$
Set $\mathbf{p}^0 = \mathbf{r}^0$
for $k = 0, 1, \ldots$ **Do**:
    $\alpha_k = \frac{\langle \mathbf{r}^k, \bar{\mathbf{r}}^0 \rangle}{\langle A\mathbf{p}^k, \bar{\mathbf{r}}^0 \rangle}$
    $\mathbf{s}^k = \mathbf{r}^k - \alpha_k A\mathbf{p}^k$
    $\omega_k = \frac{\langle \mathbf{s}^k, A\mathbf{s}^k \rangle}{\langle A\mathbf{s}^k, A\mathbf{s}^k \rangle}$
    $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{p}^k + \omega_k \mathbf{s}^k$
    $\mathbf{r}^{k+1} = \mathbf{s}^k - \alpha^k A\mathbf{s}^k$
    $\beta = \frac{\langle \mathbf{r}^{k+1}, \bar{\mathbf{r}}^0 \rangle}{\langle \mathbf{r}^k, \bar{\mathbf{r}}^0 \rangle} \frac{\alpha_k}{\omega_k}$
    $\mathbf{p}^{k+1} = \mathbf{r}^{k+1} + \beta(\mathbf{p}^k - \omega_k A\mathbf{p}^k)$
end for

Generalized Minimal Residual method

The Generalized Minimal Residual (GMRES [21]) method is applied for linear systems of equations with non-symmetric matrices. The method computes the approximation solution of the k-th iterate as:

$$\mathbf{x}^k = \mathbf{x}^0 + \mathbf{z}^k, \tag{2.16}$$

where $\mathbf{z}^k \in \mathcal{K}_k(A, \mathbf{r}^0)$ is optimized to obtain the minimum residual norm as:

$$||\mathbf{r}^k||_2 = \min_{\mathbf{z} \in \mathcal{K}_k(A, \mathbf{r}^0)} ||\mathbf{r}^0 - A\mathbf{z}||_2. \tag{2.17}$$

Equation (2.17) states that the residual $\mathbf{r}^k$ is orthogonal to $A\mathcal{K}_k(A, \mathbf{r}^0)$ and orthonormal basis of the Krylov subspace $\mathcal{K}_k(A, \mathbf{r}^0)$ is computed normally using Arnoldi's [22] method in GMRES. Algorithm 3 gives an implementation of the GMRES method.

Algorithm 3 GMRES method

1: Give initial iterate $\mathbf{x}^0$
2: Compute $\mathbf{r}^0 = \mathbf{b} - A\mathbf{x}^0$, and $\mathbf{v}^1 = \frac{\mathbf{r}^0}{||\mathbf{r}^0||_2}$
3: **for** $j = 1, ..., k$ **Do**:
4: $\quad \mathbf{v}^{j+1} = A\mathbf{v}^j$
5: $\quad$ **for** $i = 1, ..., j$ Do:
6: $\quad\quad h_{ij} = \langle \mathbf{v}^{j+1}, \mathbf{v}^i \rangle$
7: $\quad\quad \mathbf{v}^{j+1} = \mathbf{v}^{j+1} - h_{ij}\mathbf{v}^i$
8: $\quad$ **end for**
9: $\quad h_{j+1,j} = ||\mathbf{v}^{j+1}||_2$
10: $\quad \mathbf{v}^{j+1} = \frac{\mathbf{v}^{j+1}}{h_{j+1,j}}$
11: **end for**
12: The entries of upper $k+1 \times k$ Hessenberg matrix $\bar{H}_k$ are the scalars h_{ij}.

2.2 Nonlinear system of equations

Any nonlinear system of equations can be written as:

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}, \tag{2.18}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of unknowns, $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^n$ is the vector function of $\mathbf{x}$, and $\mathbf{0}$ is the vector of zeros. Due to the nonlinearity, it is impossible to directly calculate the analytic solution of a nonlinear system (2.18). Therefore, iterative methods are used to find an approximation of the solution of nonlinear systems of equations. The Newton-Raphson method is the most well-known solution technique for nonlinear systems of equations.

2.2.1 Newton-Raphson method

The Newton-Raphson method linearizes the nonlinear system of equations (2.18) and computes the linearized system of equations in every iteration. The method first

computes the Jacobian $\mathbf{J}$ of $\mathbf{F}$ as:

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial F_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial F_1(\mathbf{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial F_n(\mathbf{x})}{\partial x_n} \end{bmatrix}, \tag{2.19}$$

then construct the linear system of equations as follows:

$$-\mathbf{J}(\mathbf{x}^k)\Delta\mathbf{x}^k = \mathbf{F}(\mathbf{x}^k) \tag{2.20}$$

where $\Delta\mathbf{x}$ is the correction vector. The iteration process of the Newton-Raphson method is shown in Algorithm 4. The stopping criterion of the method is measured

Algorithm 4 Newton-Raphson method

1: set $k := 0$
2: give initial iterate $\mathbf{x}^0$
3: **while** not converged
4: solve the correction $-\mathbf{J}(\mathbf{x}^k)\Delta\mathbf{x}^k = \mathbf{F}(\mathbf{x}^k)$
5: update iterate $\mathbf{x}^{k+1} := \mathbf{x}^k + \Delta\mathbf{x}^k$
6: $k := k + 1$
7: **end while**

in the residual norm $||\mathbf{F}(\mathbf{x}^k)||$, or relative residual norm $\frac{||\mathbf{F}(\mathbf{x}^k)||}{||\mathbf{F}(\mathbf{x}^0)||}$.

2.3 Optimization problems

The general optimization problem can be stated as follows:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g(\mathbf{x}) = \mathbf{0}, \\ & h(\mathbf{x}) \leq \mathbf{0} \end{aligned} \tag{2.21}$$

where $\mathbf{x}$ is the optimization vector and $f(\mathbf{x})$ is the objective function to be minimized (maximized). The vector functions $g(\mathbf{x})$ and $h(\mathbf{x})$ represent equality and inequality constraints respectively. There are many types of optimization problems and solution techniques specially developed for the type of problem. In this work, we use the Interior Point Method used to solve the nonlinear and non-convex optimization problems.

2.3.1 Primal-Dual Interior Point method

The Primal-Dual Interior Point algorithm [23] is used to solve the optimization problem of the form in (2.21). The method first transforms the inequality constraints into equality constraints by applying a logarithmic barrier function to obtain the following

equivalent OPF problem [24]:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) - \gamma \sum_{k=1}^{n_i} \ln(\mathbf{z}_k) \\ \text{subject to} \quad & g(\mathbf{x}) = \mathbf{0}, \\ & h(\mathbf{x}) + \mathbf{z} = \mathbf{0}, \end{aligned} \tag{2.22}$$

where $\mathbf{z} > 0$ is a vector of positive slack variables, n_i is the number of inequality constraints and γ is the barrier parameter. The solution of the OPF problem (2.22) converges to the solution of the original problem (2.21) when γ goes to zero [23]. For the new equality constrained problem (2.22), the Lagrangian function is given by:

$$\mathcal{L}^{\gamma}(\mathbf{x}, \mathbf{z}, \lambda, \mu) = f(\mathbf{x}) + \lambda^T g(\mathbf{x}) + \mu^T (h(\mathbf{x}) + \mathbf{z}) - \gamma \sum_{k=1}^{n_i} \ln(\mathbf{z}_k) \tag{2.23}$$

where λ and μ are Lagrange multipliers for the equality constraints $g(\mathbf{x})$ and $h(\mathbf{x}) + \mathbf{z}$ respectively. The first-order optimality conditions known as Karush-Kuhn-Tucker (KKT) conditions are obtained when the partial derivatives of the Lagrangian function (2.23) with respect to all unknowns are set to zero:

$$\mathcal{L}^{\gamma}_{\mathbf{x}} = f_{\mathbf{x}} + \lambda^T g_{\mathbf{x}} + \mu^T h_{\mathbf{x}} = \mathbf{0}, \tag{2.24}$$

$$\mathcal{L}^{\gamma}_{\mathbf{z}} = \mu^T - \gamma \mathbf{e}^T [\mathbf{z}]^{-1} = \mathbf{0}, \tag{2.25}$$

$$\mathcal{L}^{\gamma}_{\lambda} = g^T(\mathbf{x}) = \mathbf{0}, \tag{2.26}$$

$$\mathcal{L}^{\gamma}_{\mu} = h^T(\mathbf{x}) + \mathbf{z}^T = \mathbf{0}, \tag{2.27}$$

where $\mu \geq \mathbf{0}$, $\mathbf{e} = [1, \cdots, 1]^T$ and $[\mathbf{z}]$ is a diagonal matrix with vector $\mathbf{z}$ on the main diagonal. In general, the first-order optimality conditions (2.24)-(2.27) constitute a nonlinear system of equations $\mathbf{F}(\mathbf{x}, \mathbf{z}, \lambda, \mu) = \mathbf{0}$ where

$$\mathbf{F}(\mathbf{x}, \mathbf{z}, \lambda, \mu) = \begin{bmatrix} f_{\mathbf{x}}^T + g_{\mathbf{x}}^T \lambda + h_{\mathbf{x}}^T \mu \\ [\mu]\mathbf{z} - \gamma \mathbf{e} \\ g(\mathbf{x}) \\ h(\mathbf{x}) + \mathbf{z} \end{bmatrix} = \mathbf{0}. \tag{2.28}$$

Applying the Newton-Raphson method to (2.28), the following linearized KKT conditions are obtained:

$$\begin{bmatrix} \mathcal{L}^{\gamma}_{\mathbf{xx}} & 0 & g_{\mathbf{x}}^T & h_{\mathbf{x}}^T \\ 0 & [\mu] & 0 & [\mathbf{z}] \\ g_{\mathbf{x}} & 0 & 0 & 0 \\ h_{\mathbf{x}} & I & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{z} \\ \Delta\lambda \\ \Delta\mu \end{bmatrix} = - \begin{bmatrix} f_{\mathbf{x}}^T + g_{\mathbf{x}}^T \lambda + h_{\mathbf{x}}^T \mu \\ [\mu]\mathbf{z} - \gamma \mathbf{e} \\ g(\mathbf{x}) \\ h(\mathbf{x}) + \mathbf{z} \end{bmatrix} \tag{2.29}$$

where

$$\mathcal{L}^{\gamma}_{\mathbf{xx}} = f_{\mathbf{xx}} + g_{\mathbf{xx}}(\lambda) + h_{\mathbf{xx}}(\mu). \tag{2.30}$$

The Primal-Dual Interior Point algorithm assembles the object function, equality, and inequality constraints into the linearized Karush-Kuhn-Tucker (KKT) conditions (2.29) and solves it in each iteration of the solution process. For more detailed information of the method, we refer to [23].

Chapter 3

Power system modeling

In order to perform any power flow simulation on electrical power grids, we must have an adequate mathematical model of the power system. In power system modeling, we transform the physical properties of the network into mathematical equations that can be solved analytically or numerically. This chapter delivers mathematical modeling of electrical power systems and its components such as transmission lines, loads, generators, and transformers. Both balanced single-phase and unbalanced three-phase networks are considered for power systems modeling.

3.1 Power systems

Power systems are modeled as a network of nodes (buses) and branches (transmission lines), whereas a network bus represents a system component such as a generator, load, and transmission substation, etc. There are three types of network buses, namely a slack bus, a generator (PV) bus, and a load (PQ) bus. Each bus in the power network is fully described by the following four electrical quantities:

$|V_i|$: the voltage magnitude
δ_i : the voltage phase angle
P_i : the active power
Q_i : the reactive power

Depending on the type of network bus, two of the four electrical quantities are specified as shown in Table 3.1:

Table 3.1: *Network bus type. i: index of the bus; N_g: number of generator buses; N_b: total number of buses in the network.*

Bus type	**Number of buses**	**Knowns**	**Unknowns**
slack or swing bus	1	$\|V_i\|, \delta_i$	P_i, Q_i
generator or PV bus	N_g	$P_i, \|V_i\|$	Q_i, δ_i
load or PQ bus	$N_b - N_g - 1$	P_i, Q_i	$\|V_i\|, \delta_i$

3.2 Transmission lines

Every branch that is connecting bus i and bus j, has an impedance $z_{ij} = r_{ij} + \imath x_{ij}$ with resistance r_{ij} and reactance x_{ij} as shown in Figure 3.1. The admittance of the transmission line is computed as $y_{ij} = \frac{1}{z_{ij}}$.

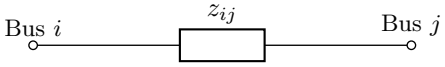


Figure 3.1: *Transmission line model.*

3.3 Loads

For all load buses in the network, active P and reactive Q powers must be known in advance. In steady-state power system analysis, these loads are modeled as a static load such that active P and reactive Q powers are expressed as a function of the voltages. The following are commonly used load models [25]:

- Constant power (PQ):
  The powers (P and Q) are independent of variations in the voltage magnitude $|V|$:
  $$\frac{P}{P_0} = 1, \qquad \frac{Q}{Q_0} = 1$$

- Constant current (I):
  The powers (P and Q) vary directly with the voltage magnitude $|V|$:
  $$\frac{P}{P_0} = \frac{|V|}{|V_0|}, \qquad \frac{Q}{Q_0} = \frac{|V|}{|V_0|}$$

- Constant impedance (Z):
  The powers (P and Q) vary with the square of the voltage magnitude $|V|$:
  $$\frac{P}{P_0} = \Big(\frac{|V|}{|V_0|}\Big)^2, \qquad \frac{Q}{Q_0} = \Big(\frac{|V|}{|V_0|}\Big)^2$$

- Polynomial (Po):
  The relation between powers (P and Q) and voltage magnitudes $|V|$ is described by a polynomial equation:
  $$\frac{P}{P_0} = a_0 + a_1\frac{|V|}{|V_0|} + a_2\Big(\frac{|V|}{|V_0|}\Big)^2, \qquad \frac{Q}{Q_0} = b_0 + b_1\frac{|V|}{|V_0|} + b_2\Big(\frac{|V|}{|V_0|}\Big)^2$$
  where a_0, a_1, a_2 and b_0, b_1, b_2 are constant parameters of the model and satisfy the following equations:
  $$a_0 + a_1 + a_2 = 1, \qquad b_0 + b_1 + b_2 = 1$$

- Exponential:
  The relation between powers (P and Q) and voltage magnitudes $|V|$ is described by an exponential equation:

$$\frac{P}{P_0} = \left(\frac{|V|}{|V_0|}\right)^n, \qquad \frac{Q}{Q_0} = \left(\frac{|V|}{|V_0|}\right)^n$$

  where n is a constant parameter of the model.

Here, P_0, Q_0, and V_0 are the specified parameters of the each bus in the network.

3.3.1 Three-phase loads

Three-phase loads are unequally specified for each phase of the unbalanced distribution networks. Three-phase loads are physically connected to the electrical grid in a grounded Wye (Y) configuration or an ungrounded delta (Δ) configuration, as shown in Figure 3.2. In a four-wire Wye configuration, loads can be connected phase-to-neutral or phase-to-phase whereas a three-wire delta configuration allows loads to connect phase-to-phase. In this work, the grounded Wye configuration connecting loads phase-to-neutral is chosen for all three-phase power flow computations.

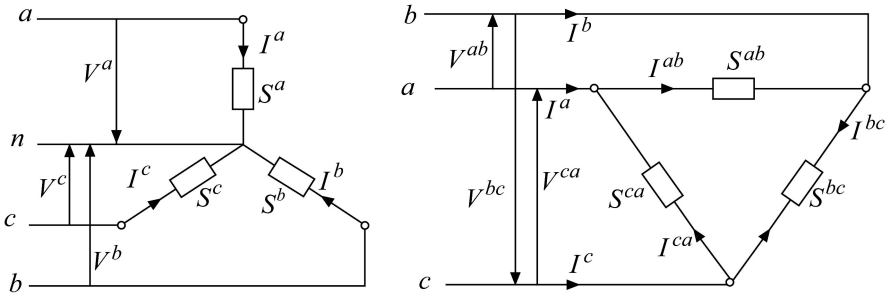


Figure 3.2: *Wye and Delta connections for three-phase loads [26].*

Let all three-phase loads are connected using a grounded Wye configuration and modeled as exponential loads as described in Section 3.3. Then, three-phase nodal loads S_i^p and nodal currents I_i^p at bus i for given phase p are given as follows:

$$\begin{bmatrix} S_i^a \\ S_i^b \\ S_i^c \end{bmatrix} = \begin{bmatrix} P_i^a + \imath Q_i^a \\ P_i^b + \imath Q_i^b \\ P_i^c + \imath Q_i^c \end{bmatrix} = \begin{bmatrix} \left(P_{i0}^a + \imath Q_{i0}^a\right)\left(\frac{|V_i^a|}{|V_0^a|}\right)^n \\ \left(P_{i0}^b + \imath Q_{i0}^b\right)\left(\frac{|V_i^b|}{|V_0^b|}\right)^n \\ \left(P_{i0}^c + \imath Q_{i0}^c\right)\left(\frac{|V_i^c|}{|V_0^c|}\right)^n \end{bmatrix}, \quad \begin{bmatrix} I_i^a \\ I_i^b \\ I_i^c \end{bmatrix} = \begin{bmatrix} \left(\frac{S_i^a}{V_i^a}\right)^* \\ \left(\frac{S_i^b}{V_i^b}\right)^* \\ \left(\frac{S_i^c}{V_i^c}\right)^* \end{bmatrix}. \tag{3.1}$$

In the case that a ungrounded Δ connection is considered, three-phase loads and currents are specified as follows:

$$\begin{bmatrix} S_i^{ab} \\ S_i^{bc} \\ S_i^{ca} \end{bmatrix} = \begin{bmatrix} \left(P_{i0}^{ab} + \imath Q_{i0}^{ab}\right)\left(\frac{|V_i^{ab}|}{|V_0^{ab}|}\right)^n \\ \left(P_{i0}^{bc} + \imath Q_{i0}^{bc}\right)\left(\frac{|V_i^{bc}|}{|V_0^{bc}|}\right)^n \\ \left(P_{i0}^{ca} + \imath Q_{i0}^{ca}\right)\left(\frac{|V_i^{ca}|}{|V_0^{ca}|}\right)^n \end{bmatrix}, \quad \begin{bmatrix} I_i^a \\ I_i^b \\ I_i^c \end{bmatrix} = \begin{bmatrix} \left(\frac{S_i^{ab}}{V_i^{ab}}\right)^* - \left(\frac{S_i^{ca}}{V_i^{ca}}\right)^* \\ \left(\frac{S_i^{bc}}{V_i^{bc}}\right)^* - \left(\frac{S_i^{ab}}{V_i^{ab}}\right)^* \\ \left(\frac{S_i^{ca}}{V_i^{ca}}\right)^* - \left(\frac{S_i^{bc}}{V_i^{bc}}\right)^* \end{bmatrix}. \tag{3.2}$$

3.4 Generators

Since conventional power plants generally have a control for the active power and the voltage magnitude, generators are modeled as a PV bus such that the active power P and the voltage magnitude $|V|$ are specified for the power flow computations. However, most of the small Distribution Generators (DGs) do not have both P and $|V|$ controls. Therefore, DGs are often modeled as a load bus (PQ) with a positive injected power. Figure 3.3 shows which type of power converter is employed to what types of renewable energy sources.

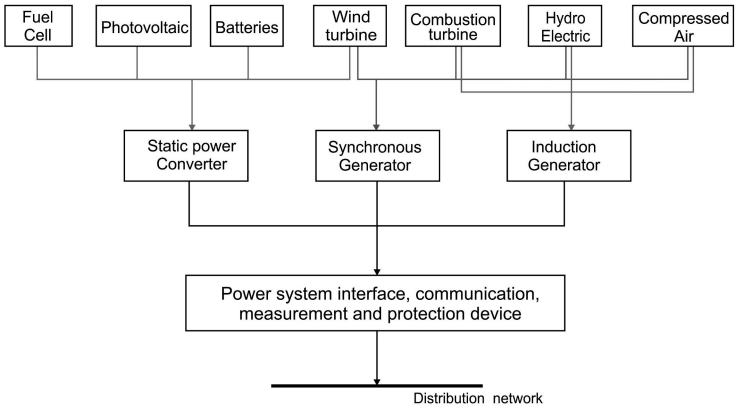


Figure 3.3: *Combination of power converters and energy sources [27].*

Depending on the types of energy sources and energy converters, the DGs are modeled as follows:

- The constant power factor model (PQ bus):
  The active power P output and power factor pf are specified and the reactive power Q is determined by these two variables.
- The variable reactive power model (PQ bus):
  The active power P output is specified and the reactive power Q is determined by applying a predetermined polynomial function.
- The constant voltage model (PV bus):
  The active power P output and voltage magnitude $|V|$ are specified.

In power system analysis, one generator bus is chosen as a slack bus to balance the difference between the total power generation, and the total power consumption plus the power losses. For a slack node, the voltage magnitude $|V|$ and angle δ are specified and in practice, the voltage angle δ is set to zero.

3.5 Shunts

A shunt connecting the bus and the ground is modeled as a reactance $z_s = \imath x_s$. A shunt admittance y_s is obtained as $y_s = \frac{1}{z_s} = -j\frac{1}{x_s} = jb_s$ and is divided equally ($\frac{y_s}{2}$)

for two connected buses, as shown in Figure 3.4. If $x_s > 0$ the shunt is inductive, otherwise the shunt is capacitive. Shunt capacitors inject reactive powers, and thus the node voltages increase whereas shunt inductors consume reactive powers resulting in lower node voltages.

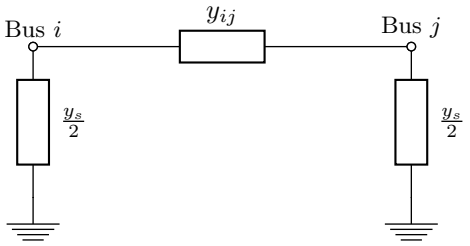


Figure 3.4: *Transmission line model with a shunt.*

3.6 Transformers

Transformers are used to connect two networks with different voltage levels as shown in Figure 3.5. Generally, transformers are added to the admittance matrix Y of the network. For each transformer, there is a turn ratio $T : 1$ used to scale the voltage levels.

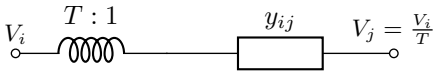


Figure 3.5: *Transmission line model with a transformer.*

After the power system simulation, the voltage and current of the secondary side are calculated as $V_s = T^{-1}V_p$ and $I_s = TI_p$, where p, s denote the primary and secondary side of the transformer respectively.

3.6.1 Three-phase transformers

Three-phase transformers are modeled by an admittance matrix Y_T^{abc} which depends upon the connection of the primary and secondary taps, and the leakage admittance y_t of the transformer. The admittance matrix $Y_T^{abc} \in \mathbb{C}^{6\times 6}$ is given as:

$$Y_T^{abc} = \begin{bmatrix} Y_{pp}^{abc} & Y_{ps}^{abc} \\ Y_{sp}^{abc} & Y_{ss}^{abc} \end{bmatrix} \tag{3.3}$$

where Y_{ps}^{abc}, Y_{sp}^{abc} are mutual admittance matrices, and Y_{pp}^{abc}, Y_{ss}^{abc} are self admittance matrices of the primary and the secondary taps, respectively. These mutual and self admittance matrices are selected from the following three matrices depending on the connection of the primary and secondary taps:

$$Y_1 = \begin{bmatrix} y_t & 0 & 0 \\ 0 & y_t & 0 \\ 0 & 0 & y_t \end{bmatrix},\ Y_2 = \frac{1}{3}\begin{bmatrix} 2y_t & -y_t & -y_t \\ -y_t & 2y_t & -y_t \\ -y_t & -y_t & 2y_t \end{bmatrix},\ Y_3 = \frac{1}{\sqrt{3}}\begin{bmatrix} -y_t & y_t & 0 \\ 0 & -y_t & y_t \\ y_t & 0 & -y_t \end{bmatrix}.$$

Table 3.2 shows the connection of the primary and secondary taps, and their corresponding choices for the admittance matrix Y_T^{abc}.

Table 3.2: *Three-phase transformer connections and their corresponding admittance matrices.*

Transformers		**Self admittance**		**Mutual admittance**	
Primary	Secondary	Y_{pp}^{abc}	Y_{ss}^{abc}	Y_{ps}^{abc}	Y_{sp}^{abc}
Wye-G	Wye-G	Y_1	Y_1	$-Y_1$	$-Y_1$
Wye-G	Wye	Y_2	Y_2	$-Y_2$	$-Y_2$
Wye-G	Delta	Y_1	Y_2	Y_3	Y_3^T
Wye	Wye-G	Y_2	Y_2	$-Y_2$	$-Y_2$
Wye	Wye	Y_2	Y_2	$-Y_2$	$-Y_2$
Wye	Delta	Y_2	Y_2	Y_3	Y_3^T
Delta	Wye-G	Y_2	Y_1	Y_3^T	Y_3
Delta	Wye	Y_2	Y_2	Y_3^T	Y_3
Delta	Delta	Y_2	Y_2	$-Y_2$	$-Y_2$

If the transformer has an off-nominal tap ratio α:β where α and β are tappings on the primary and secondary sides respectively, then the mutual and self admittance matrices must be modified as follows:

- Divide the self admittance matrix of the primary side by α^2: $\frac{Y_{pp}^{abc}}{\alpha^2}$
- Divide the self admittance matrix of the secondary side by β^2: $\frac{Y_{ss}^{abc}}{\beta^2}$
- Divide the mutual admittance matrices by $\alpha\beta$: $\frac{Y_{ps}^{abc}}{\alpha\beta}$ and $\frac{Y_{sp}^{abc}}{\alpha\beta}$

3.7 Network admittance

Since all transmission lines are modeled with admittance, the admittance matrix Y of the network can be built as:

- the element on the main diagonal Y_{ii} equals the sum of all admittance directly connected to node i;
- the off-diagonal element Y_{ij} equals the negative value of the net admittance connected between node i and node j (note that the off-diagonal element Y_{ji} has the same value). Thus, the admittance matrix Y is symmetric.

According to Kirchoff's Current Law (KCL), the relation between the nodal injected currents and the voltages, is described by the admittance matrix Y:

$$I = YV \quad \leftrightarrow \quad \begin{bmatrix} I_1 \\ \vdots \\ I_{N_b} \end{bmatrix} = \begin{bmatrix} Y_{11} & \cdots & Y_{1N_b} \\ \vdots & \ddots & \vdots \\ Y_{N_b1} & \cdots & Y_{N_bN_b} \end{bmatrix} \begin{bmatrix} V_1 \\ \vdots \\ V_{N_b} \end{bmatrix}, \tag{3.4}$$

where I_i is the injected complex current, V_i is the complex voltage at bus i and Y_{ij} is the element of the admittance matrix. For unbalanced three-phase networks, equation (3.4) becomes:

$$\begin{bmatrix} I_1^{abc} \\ \vdots \\ I_{N_b}^{abc} \end{bmatrix} = \begin{bmatrix} Y_{11}^{abc} & \cdots & Y_{1N_b}^{abc} \\ \vdots & \ddots & \vdots \\ Y_{N_b1}^{abc} & \cdots & Y_{N_bN_b}^{abc} \end{bmatrix} \begin{bmatrix} V_1^{abc} \\ \vdots \\ V_{N_b}^{abc} \end{bmatrix}. \tag{3.5}$$

with

$$I_i^{abc} = \begin{bmatrix} I_i^a \\ I_i^b \\ I_i^c \end{bmatrix},\ V_j^{abc} = \begin{bmatrix} V_j^a \\ V_j^b \\ V_j^c \end{bmatrix},\ Y_{ij}^{abc} = \begin{bmatrix} Y_{ij}^{aa} & Y_{ij}^{ab} & Y_{ij}^{ac} \\ Y_{ij}^{ba} & Y_{ij}^{bb} & Y_{ij}^{bc} \\ Y_{ij}^{ca} & Y_{ij}^{cb} & Y_{ij}^{cc} \end{bmatrix} \tag{3.6}$$

where I_i^p is the injected complex current, and V_i^p is the complex voltage at bus i for a given phase p. The injected current I_i^p at bus i for a given phase p can be computed from equation (3.5) as follows:

$$I_i^p = \sum_{k=1}^{N_b} \sum_{q=a,b,c} Y_{ik}^{pq} V_k^q. \tag{3.7}$$

For more detailed information on the power system model, we refer to [2, 26–29].

Chapter 4

Nonlinear Power Flow Computations

In this chapter, various mathematical formulations of the power flow problem are studied in details. A general framework is given for applying the Newton-Raphson method to solve nonlinear power flow problems, using power and current-mismatch functions in polar, Cartesian coordinates and complex form. These two mismatch functions and three coordinates, result in six possible ways to apply the Newton-Raphson method for the solution of power flow problems. We develop new versions of the Newton power flow method that are faster and more robust than existing versions. Each new Newton power flow variant is theoretically explained and compared with current versions in terms of the difference. We also present a theoretical framework to analyze these variants for load (PQ) buses and generator (PV) buses as it requires to have different PV bus modeling for each variant. Furthermore, we compare computational performances of newly developed versions with existing variants of the Newton power flow method. The convergence behavior of all methods is investigated by numerical experiments on balanced transmission and (un)balanced distribution networks. Moreover, all new variants of the Newton power flow method are extended to three-phase power flow problems. The backward-forward sweep algorithm is implemented for comparison reasons. We further investigate the convergence behavior of all variants for different loading conditions, R/X ratios, and load models to test the robustness of all methods.

This chapter is based on:

B. Sereeter, C. Vuik, and C. Witteveen, "On a comparison of Newton-Raphson solvers for power flow problems," *Journal of Computational and Applied Mathematics*, vol. 360, pp. 157–169, Nov 2019,

B. Sereeter, K. Vuik, and C. Witteveen, "Newton power flow methods for unbalanced three-phase distribution networks," *Energies*, vol. 10, no. 10, p. 1658, 2017.

4.1 Introduction

A power flow computation is used to determine the steady state behavior of the network. The solution of a power flow computation can be used to assess whether the power system can function properly for the given generation and consumption. Therefore, power flow computations are performed in power system operation, control and planning. Traditionally, power flow computations were calculated only in the transmission network and the distribution networks were aggregated as buses in the power system model. However, in the new operation and maintenance of the distribution network, the power flow problem computation must be done on the distribution network as well.

There are conventional power flow solution techniques for transmission networks, such as Gauss–Seidel (GS), Newton power flow (NR), and fast decoupled load flow (FDLF) [1, 30, 31] which are widely used for power system operation, control and planning. In practice, the Newton power flow method is preferred in terms of quadratic convergence and improved robustness [32]. However, these conventional power flow methods do not always converge when they are applied to the distribution power flow problem due to some special features of the distribution network:

- Radial or weakly meshed (radial network with a few simple loops) structure:
  In general, a transmission network is operated in a meshed structure, whereas a distribution network is operated in a radial structure where there are no loops in the network and each bus is connected to the source via exactly one path.

- High R/X ratio:
  Transmission lines of the distribution network have a wide range of resistance R and reactance X values. Therefore, R/X ratios in the distribution network are relatively high compared to the transmission network.

- Multi-phase power flow and unbalanced loads:
  A single-phase representation is used for power flow analysis on transmission network which is assumed to be a balanced network. Unlike the transmission network, a distribution network must use a three-phase power flow analysis due to the unbalanced loads.

- Distributed generations:
  Unlike conventional power plants connected to the transmission network, DGs have fluctuating power output that is difficult to predict and control since it is strongly dependent on weather conditions.

Systems with the above features create ill-conditioned systems of nonlinear algebraic equations that cause numerical problems for the conventional methods [33–35].

Furthermore, many new methods have been developed for distribution power flow problems and generally they are divided into two main categories as:

- Modification of conventional power flow solution methods [28, 36–55]: Methods in this category are generally a proper modification of existing solution techniques such as GS, NR and FDLF.

- Backward–forward sweep (BFS)-based algorithms [56–83]:
  BFS-based algorithms generally take an advantage of the radial network topology. The method is an iterative process in which at each iteration two computational steps are performed, a forward and a backward sweep. The forward sweep is mainly the node voltage calculation and the backward sweep is the branch current or power, or the admittance summation.

Several reviews on distribution power flow solution methods can be found in [32, 84–87].

In this thesis, we focus on the Newton based power flow methods for nonlinear power flow computations on both balanced transmission and (un)balanced distribution networks. Depending on problem formulations (power or current mismatch) and coordinates (polar, Cartesian and complex form), the Newton-Raphson method can be applied in six different ways as a solution method for power flow problems. These six versions of the Newton power flow method are considered as the fundamental Newton power flow methods from which the further modified versions [38, 41, 50, 52, 88–91] are derived. Table 4.1 shows the previously published papers considering each variation of the Newton power flow method. All variations of the Newton power flow

Table 4.1: *The Newton power flow methods using different coordinates.*

Mismatch formulations	**Coordinates**		
	Polar	Cartesian	Complex form
Power	[1]	[92]	[39]
Current	[93]	[40, 93]	

method are developed by various researchers in different ways. The most widely used version is the Newton power flow method using the power-mismatch in polar coordinates which is introduced in [1]. In this method, the reactive power mismatch ΔQ and the voltage magnitude correction ΔV for each generator (PV) bus are eliminated from the Jacobian matrix equations (2.20) and therefore the order of the equation is $(2N - N_g - 2)$.

In the version using the power-mismatch in Cartesian (rectangular) coordinates introduced in [92], the reactive power mismatch ΔQ is not eliminated from the Jacobian matrix equations (2.20) for each PV bus but replaced by a voltage-magnitude-squared mismatch equation:

$$\Delta|V|^2 = (|V|^{sp})^2 - (V^r)^2 + (V^m)^2. \tag{4.1}$$

where $|V|^{sp}$ is specified voltage magnitudes, and V^r and V^m are the real and imaginary parts of the complex voltages respectively. Therefore, the order of the Jacobian matrix equation is $(2N-2)$ and it is concluded in [92] that the method is slightly less reliable and less rapid in convergence than the polar version developed in [1].

Although it is mentioned in [1, 32] that the complex power flow formulation does not mathematically lead to an analytic function of the complex voltage because of conjugate terms, the paper [39] investigated the version of the Newton power flow method using the power-mismatch in complex form. In paper [39], the Jacobian matrix equations are developed in complex form for each load (PQ) bus whereas two separate equations are created for each PV bus. The correction values of complex

voltage for the PQ and PV buses are computed separately using different tolerances at each iteration. However, it is preferred to calculate correction values for both PQ and PV buses using common Jacobian matrix equations and the same tolerance.

The version using the current-mismatch and a mix of Cartesian and polar coordinates is discussed in [93]. In this method, each PQ bus is represented by two equations that are constructed from the real and imaginary parts of the complex current-mismatch function. A PV bus is represented by a single active power-mismatch ΔP and the voltage-magnitude-squared mismatch equation (4.1). The order of the Jacobian matrix equation is $(2N-2)$ and it is concluded in [32] that these versions perform less satisfactorily than the power-mismatch versions.

The version using the current-mismatch in Cartesian coordinates is considered again in [40]. This method introduces a new dependent variable ΔQ for each PV bus and additional equations relating the corrections in polar and Cartesian coordinates:

$$\Delta|V| = \frac{V^r}{|V|}\Delta V^r + \frac{V^m}{|V|}\Delta V^m \tag{4.2}$$

$$\Delta\delta = \frac{V^r}{|V|^2}\Delta V^m - \frac{V^m}{|V|^2}\Delta V^r. \tag{4.3}$$

Using equations (4.2) and (4.3), this method makes the Jacobian matrix equation square in order to have a unique solution. In this method, the real ΔI^r and imaginary ΔI^m current-mismatch functions are expressed in terms of the real ΔP and reactive ΔQ power-mismatch functions. Then the reactive power-mismatch ΔQ is considered as a dependent variable for each PV bus and computed at each Newton iteration. Minor attempts were made to speed up the solution method using a partly constant approximation of the Jacobian during the iterations, but the results were not encouraging [40].

We did not find any discussion covering the Newton power flow method using the current-mismatch in complex form.

This chapter aims to discuss all six versions of the Newton power flow method using a common framework and to introduce new developments to improve the performance of other versions besides the most used version using the power-mismatch and polar coordinates [1]. We did significant improvements in Cartesian power-mismatch, polar current-mismatch and Cartesian current-mismatch versions of the Newton power flow method. Furthermore, we extend the Newton power flow methods developed into three-phase power flow problems. Distribution networks with extreme conditions such as different load models, big loading conditions, and high R/X ratios are considered in order to analyze the convergence ability of all extended versions.

4.2 Nonlinear Power Flow problem

The mathematical equations for the power flow problem are given by:

$$S_i = V_i I_i^* \tag{4.4}$$

$$= V_i \sum_{j=1}^{N_b} Y_{ij}^* V_j^* \tag{4.5}$$

where is S_i is the injected complex power at bus i and * represents the complex conjugate of a complex variable. The single-phase power flow problem (4.5) is extended to the three-phase power flow problem using equation (3.7) as:

$$S_i^p = V_i^p (I_i^p)^* = V_i^p \sum_{k=1}^{N_b} \sum_{q=a,b,c} (Y_{ik}^{pq})^* (V_k^q)^*. \tag{4.6}$$

Mathematically, the power flow problem comes down to solving a nonlinear system of equations where all variables are given in complex numbers. Since the power flow problem is nonlinear, we can rewrite equation (4.5) as $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ as given in (2.18). Furthermore, it is possible to formulate $\mathbf{F}(\mathbf{x})$ as **power** or **current**-mismatch functions and to designate the unknown bus voltages as the problem variables $\mathbf{x}$.

4.2.1 The power-mismatch formulation

The power flow problem (4.5) is reformulated as the power-mismatch function $\mathbf{F}(\mathbf{x})$ as follows:

$$\begin{aligned} \mathbf{F}_i(\mathbf{x}) = \Delta S_i(\mathbf{x}) &= S_i^{sp} - S_i(\mathbf{x}) \\ &= S_i^{sp} - V_i \sum_{k=1}^{N_b} Y_{ik}^* V_k^* \end{aligned} \tag{4.7}$$

where $S_i^{sp} = P_i^{sp} + \imath Q_i^{sp}$ is the specified complex power at bus i. In general, S_i^{sp} is computed as:

$$S_i^{sp} = S_i^G - S_i^L \tag{4.8}$$

where S_i^G is the specified complex power generation, whereas S_i^L is the specified complex power load at bus i. Here, S_i^L can be modeled as one of the load models described in Section 3.3. The complex power-mismatch function (4.7) can be further separated into real equations and variables using **polar** or **Cartesian** coordinates. Table 4.2 displays the vector of unknowns $\mathbf{x}$ given in different coordinates.

Table 4.2: *Unknown vector* $\mathbf{x}$ *in different coordinates.* V*: Complex voltage,* $|V|$*: voltage magnitude,* δ*: voltage angle,* V^r*: real part of* V*,* V^m*: imaginary part of* V*.*

Coordinates	**Unknown vector x**
Polar ($V_i = \|V_i\| e^{\imath \delta_i}$)	$\left[\delta_1, \cdots, \delta_{N_b}, \|V_1\|, \cdots, \|V_{N_b}\|\right]^T$
Cartesian ($V_i = V_i^r + \imath V_i^m$)	$\left[V_1^m, \cdots, V_{N_b}^m, V_1^r, \cdots, V_{N_b}^r\right]^T$
Complex form (V_i)	$\left[V_1, \cdots, V_{N_b}\right]^T$

Power-mismatch function in polar coordinates (PP)

The complex power-mismatch function (4.7) is rewritten in terms of real numbers using the polar coordinates as:

$$\mathbf{F}_i(\mathbf{x}) = \begin{bmatrix} \Delta P_i(\mathbf{x}) \\ \Delta Q_i(\mathbf{x}) \end{bmatrix} \tag{4.9}$$

where

$$\begin{aligned}\Delta P_i(\mathbf{x}) &= P_i^{sp} - \sum_{k=1}^{N_b} |V_i||V_k|(G_{ik}\cos\delta_k + B_{ik}\sin\delta_k) \\ \Delta Q_i(\mathbf{x}) &= Q_i^{sp} - \sum_{k=1}^{N_b} |V_i||V_k|(G_{ik}\sin\delta_k - B_{ik}\cos\delta_k)\end{aligned} \tag{4.10}$$

where G and B are conductance and susceptance of admittance as $Y = G + \imath B$.

Power-mismatch function in Cartesian coordinates (PC)

If we use the Cartesian coordinates instead of the polar coordinates then the active power-mismatch $\Delta P_i(\mathbf{x})$ and the reactive power-mismatch $\Delta Q_i(\mathbf{x})$ in (4.9) are computed as:

$$\begin{aligned}\Delta P_i(\mathbf{x}) &= P_i^{sp} - \sum_{k=1}^{N_b} \Big(V_i^r(G_{ik}V_k^r - B_{ik}V_k^m) + V_i^m(B_{ik}V_k^r + G_{ik}V_k^m) \Big) \\ \Delta Q_i(\mathbf{x}) &= Q_i^{sp} - \sum_{k=1}^{N_b} \Big(V_i^m(G_{ik}V_k^r - B_{ik}V_k^m) - V_i^r(B_{ik}V_k^r + G_{ik}V_k^m) \Big)\end{aligned} \tag{4.11}$$

4.2.2 The current-mismatch formulation

The power flow problem (4.5) is reformulated as the current-mismatch function $\mathbf{F}(\mathbf{x})$ as follows:

$$\begin{aligned}\mathbf{F}_i(\mathbf{x}) = \Delta I_i(\mathbf{x}) &= I_i^{sp} - I_i(\mathbf{x}) \\ &= \Big(\frac{S_i^{sp}}{V_i}\Big)^* - \sum_{k=1}^{N_b} Y_{ik}V_k\end{aligned} \tag{4.12}$$

where I_i^{sp} is the specified complex current at bus i. Similar to the complex power-mismatch function (4.7), the complex current-mismatch function (4.12) can be further separated into real equations and variables using **polar** and **Cartesian** coordinates.

Current-mismatch function in polar coordinates (CP)

When the polar coordinates is used to the complex current-mismatch function (4.12), we obtain the following mismatch function $\mathbf{F}(\mathbf{x})$ with real $\Delta I^r(\mathbf{x})$ and imaginary $\Delta I^m(\mathbf{x})$ parts of the complex current-mismatch function as:

$$\mathbf{F}_i(\mathbf{x}) = \begin{bmatrix}\Delta I_i^r(\mathbf{x}) \\ \Delta I_i^m(\mathbf{x})\end{bmatrix} \tag{4.13}$$

where

$$\begin{aligned}\Delta I_i^r(\mathbf{x}) &= \frac{P_i^{sp}\cos\delta_i + Q_i^{sp}\sin\delta_i}{|V_i|} - \sum_{k=1}^{N_b} |V_k|(G_{ik}\cos\delta_k - B_{ik}\sin\delta_k) \\ \Delta I_i^m(\mathbf{x}) &= \frac{P_i^{sp}\sin\delta_i - Q_i^{sp}\cos\delta_i}{|V_i|} - \sum_{k=1}^{N_b} |V_k|(G_{ik}\sin\delta_k + B_{ik}\cos\delta_k)\end{aligned} \tag{4.14}$$

Current-mismatch function in Cartesian coordinates (CC)

In case of the Cartesian coordinates, $\Delta I^r(\mathbf{x})$ and $\Delta I^m(\mathbf{x})$ in (4.13) are given as:

$$\begin{aligned}\Delta I_i^r(\mathbf{x}) &= \frac{P_i^{sp}V_i^r + Q_i^{sp}V_i^m}{(V_i^r)^2 + (V_i^m)^2} - \sum_{k=1}^{N_b}(G_{ik}V_k^r - B_{ik}V_k^m) \\ \Delta I_i^m(\mathbf{x}) &= \frac{P_i^{sp}V_i^m - Q_i^{sp}V_i^r}{(V_i^r)^2 + (V_i^m)^2} - \sum_{k=1}^{N_b}(G_{ik}V_k^m + B_{ik}V_k^r)\end{aligned} \tag{4.15}$$

The relation between the current-mismatch function (4.12) and the power-mismatch function (4.7) is given as:

$$\Delta I_i = \Big(\frac{\Delta S_i}{V_i}\Big)^* \qquad \text{(complex)} \tag{4.16}$$

$$= \frac{\cos\delta_i\Delta P_i + \sin\delta_i\Delta Q_i}{|V_i|} + \imath\frac{\sin\delta_i\Delta P_i - \cos\delta_i\Delta Q_i}{|V_i|} \qquad \text{(polar)} \tag{4.17}$$

$$= \frac{V_i^r\Delta P_i + V_i^m\Delta Q_i}{|V_i|^2} + \imath\frac{V_i^m\Delta P_i - V_i^r\Delta Q_i}{|V_i|^2} \qquad \text{(Cartesian).} \tag{4.18}$$

All six formulations described above using two mismatch functions (power or current) and three coordinates (polar, Cartesian and complex form) are equivalent for a load (PQ) bus where active P and reactive Q powers are specified and voltage magnitude $|V|$ and angle δ are to be determined. However, for the generator modeled as a PV bus, these formulations are not equivalent due to the fact that voltage magnitude $|V|$ is specified instead of reactive power Q. Therefore, each power flow solution method handles the PV bus differently depending on the formulation that is used. Furthermore, it is possible that one formulation can result in better performance than others for the same solution technique. In practice, the power flow formulation using the polar coordinates (4.10) is mainly chosen in most of the power flow solution techniques because with this formulation PV buses are modeled relatively easy compared to other formulations.

4.3 Newton Power Flow methods

In this section, we discuss all six versions of the Newton power flow method using a common framework and introduce new developments to improve the performance of all versions besides the most used version using the power-mismatch in polar coordinates [1]. We did significant improvements in Cartesian power-mismatch, polar current-mismatch and Cartesian current-mismatch versions of the Newton power flow method as you can see in the next sections. In versions using Cartesian coordinates, equations (4.2) and (4.3) are used for PV buses instead of the voltage-magnitude-squared mismatch equation (4.1). In case of versions using the current-mismatch functions regardless of the choice of the coordinates, the reactive power Q is considered as a dependent variable for each PV bus. Thus, we compute the correction ΔQ at each iteration and update Q using the computed corrections. In the Cartesian power-mismatch variant, the order of the system is decreased to $(2N - N_g - 2)$

whereas [92] uses a system with the order $(2N-2)$. The complex current-mismatch and complex power-mismatch versions are developed only for PQ buses.

4.3.1 Polar power-mismatch version (NR-p-pol [1])

The Jacobian matrix equation (2.20) derived from the power-mismatch function in polar coordinates is given in the partitioned form for convenience of presentation:

$$-\left[\begin{array}{c|c} J^{11} & J^{12} \\ \hline J^{21} & J^{22} \end{array}\right]\left[\begin{array}{c} \Delta\delta \\ \Delta|V| \end{array}\right]=\left[\begin{array}{c} \Delta P \\ \Delta Q \end{array}\right] \tag{4.19}$$

where all sub-matrices are computed as $J^{11}=\frac{\partial\Delta P}{\partial\delta}$, $J^{12}=\frac{\partial\Delta P}{\partial|V|}$, $J^{21}=\frac{\partial\Delta Q}{\partial\delta}$ and $J^{22}=\frac{\partial\Delta Q}{\partial|V|}$. The Jacobian matrix equation (4.19) has to be modified for all PV buses since the voltage magnitude $|V_j|$ is specified instead of the reactive power Q_j at each PV bus j. Therefore, Q_j^{sp} cannot be computed and ΔQ_j cannot be formulated for each PV bus j. All partial derivatives of it with respect to voltage magnitude $|V_i|$ and angle δ_i cannot be taken. Similarly, $\Delta|V_j|$ does need to be computed for PV bus j since $|V_j|$ is now known. Therefore, we eliminate all the $\frac{\partial\Delta P_i}{\partial|V_j|}$, $\frac{\partial\Delta Q_i}{\partial|V_j|}$, $\frac{\partial\Delta Q_j}{\partial\delta_i}$ and $\frac{\partial\Delta Q_j}{\partial|V_i|}$ from the Jacobian matrix $J(\mathbf{x})$, $\Delta|V_j|$ from the correction vector $\Delta\mathbf{x}$ and ΔQ_j from the power mismatch vector $\mathbf{F}(\mathbf{x})$ for each PV bus j. The order of the resulting Jacobian matrix equation is $(2N-N_g-2)$.

4.3.2 Cartesian power-mismatch version (NR-p-car)

The Jacobian matrix equation (2.20) is defined using the power-mismatch function in Cartesian coordinates as:

$$-\left[\begin{array}{c|c} J^{11} & J^{12} \\ \hline J^{21} & J^{22} \end{array}\right]\left[\begin{array}{c} \Delta V^m \\ \Delta V^r \end{array}\right]=\left[\begin{array}{c} \Delta P \\ \Delta Q \end{array}\right] \tag{4.20}$$

where all sub-matrices are given as $J^{11}=\frac{\partial\Delta P}{\partial V^m}$, $J^{12}=\frac{\partial\Delta P}{\partial V^r}$, $J^{21}=\frac{\partial\Delta Q}{\partial V^m}$ and $J^{22}=\frac{\partial\Delta Q}{\partial V^r}$. The Jacobian matrix equation (4.20) has to be modified for all PV buses for the same reason as we saw in 4.3.1. In this version, the reactive power-mismatch ΔQ_j cannot be formulated for each PV bus j and therefore all partial derivatives $\frac{\partial\Delta Q_j}{\partial V_k^m}$ and $\frac{\partial\Delta Q_j}{\partial V_k^r}$ cannot be taken.

In paper [92], the reactive power mismatch ΔQ is replaced by a voltage-magnitude-squared mismatch equation (4.1) for all PV buses and therefore all partial derivatives $\frac{\partial\Delta Q_j}{\partial V_k^m}$ and $\frac{\partial\Delta Q_j}{\partial V_k^r}$ are also replaced by $\frac{\partial\Delta|V_j|^2}{\partial V_k^m}$ and $\frac{\partial\Delta|V_j|^2}{\partial V_k^r}$ respectively. Moreover, the order of the Jacobian matrix equation remains $(2N-2)$ and it is concluded in [92] that the method is slightly less reliable and less rapid in convergence than the polar power-mismatch version 4.3.1.

In this thesis, we develop a new approach that improves the performance of this version. In our approach, the reactive power-mismatch ΔQ_j is removed from the power-mismatch vector $\mathbf{F}(\mathbf{x})$ for all PV buses and therefore all partial derivatives $\frac{\partial\Delta Q_j}{\partial V_k^m}$ and $\frac{\partial\Delta Q_j}{\partial V_k^r}$ are also eliminated from the Jacobian matrix $J(\mathbf{x})$. As a result of the elimination, the Jacobian matrix becomes a rectangular matrix. In order to make

the Jacobian matrix square, we use the equation (4.2) with $\Delta|V_j| = 0$ since $|V_j|$ is now specified for each PV bus j. This gives us the relation between the corrections ΔV_j^r and ΔV_j^m as:

$$\Delta V_j^r = -\frac{V_j^m}{V_j^r}\Delta V_j^m. \tag{4.21}$$

Using equation (4.21), the column of the Jacobian matrix with respect to the derivatives $\frac{\partial \Delta P_i}{\partial V_j^r}$ and $\frac{\partial \Delta Q_i}{\partial V_j^r}$ is added to the column with respect to the derivatives $\frac{\partial \Delta P_i}{\partial V_j^m}$ and $\frac{\partial \Delta Q_i}{\partial V_j^m}$ as follows:

$$\frac{\partial \Delta P_i}{\partial V_j^m}\Delta V_j^m = \Big(\frac{\partial \Delta P_i}{\partial V_j^m} - \frac{V_j^m}{V_j^r}\frac{\partial \Delta P_i}{\partial V_j^r}\Big)\Delta V_j^m \tag{4.22}$$

$$\frac{\partial \Delta Q_i}{\partial V_j^m}\Delta V_j^m = \Big(\frac{\partial \Delta Q_i}{\partial V_j^m} - \frac{V_j^m}{V_j^r}\frac{\partial \Delta Q_i}{\partial V_j^r}\Big)\Delta V_j^m. \tag{4.23}$$

Now the correction ΔV_j^r can be eliminated from the correction vector $\Delta\mathbf{x}$ for each PV bus j and therefore the order of the Jacobian matrix equation (4.20) is $(2N - N_g - 2)$.

4.3.3 Complex power-mismatch version (NR-p-com)

The Jacobian matrix equation (2.20) is computed using the power-mismatch function in complex form as:

$$-\begin{bmatrix} J \end{bmatrix}\begin{bmatrix} \Delta V \end{bmatrix} = \begin{bmatrix} \Delta S \end{bmatrix} \tag{4.24}$$

where the Jacobian matrix $J = \frac{\partial \Delta S}{\partial V}$ is obtained by taking the first order partial derivatives of the complex power-mismatch functions with respect to the complex voltage V. The Jacobian matrix equation (4.24) holds for all PQ buses but not for all PV buses because the complex power mismatch ΔS cannot be formulated for all PV buses. Therefore, this version can be applied to solve the power flow problem on networks with only a slack bus and PQ buses.

4.3.4 Polar current-mismatch version (NR-c-pol)

The Jacobian matrix equation (2.20) is derived from the current-mismatch function in polar coordinates as:

$$-\left[\begin{array}{c|c} J^{11} & J^{12} \\ \hline J^{21} & J^{22} \end{array}\right]\begin{bmatrix} \Delta\delta \\ \Delta|V| \end{bmatrix} = \begin{bmatrix} \Delta I^r \\ \Delta I^m \end{bmatrix} \tag{4.25}$$

where all sub-matrices are computed as $J^{11} = \frac{\partial \Delta I^r}{\partial \delta}$, $J^{12} = \frac{\partial \Delta I^r}{\partial |V|}$, $J^{21} = \frac{\partial \Delta I^m}{\partial \delta}$ and $J^{22} = \frac{\partial \Delta I^m}{\partial |V|}$. Same as the polar power-mismatch version 4.3.1, $\Delta|V_j|$ needs to be computed for each PV bus j since $|V_j|$ is now known. Therefore, we eliminate all the $\frac{\partial \Delta I_i^r}{\partial |V_j|}$ and $\frac{\partial \Delta I_i^m}{\partial |V_j|}$ from the Jacobian matrix $J(\mathbf{x})$ and $\Delta|V_j|$ from the correction vector $\Delta\mathbf{x}$ for each PV bus j. As a result of the elimination, the Jacobian matrix becomes a rectangular matrix.

In paper [93], each PQ bus is represented by the real ΔI^r and imaginary ΔI^m current-mismatch functions. A PV bus is represented by the active power-mismatch

ΔP and the voltage-magnitude-squared mismatch equation (4.1). Thus, the order of the Jacobian matrix equation is $(2N-2)$ and it is concluded in [32] that these versions perform less satisfactorily than the power-mismatch versions.

In our approach, the reactive power Q_j is chosen as a dependent variable as $|V|$ and δ for each PV bus j because we use the current-mismatch formulation directly. Since Q_j is an unknown variable, all first order partial derivatives $\frac{\partial \Delta I_i^r}{\partial Q_j}$ and $\frac{\partial \Delta I_i^m}{\partial Q_j}$ have to be computed as given in Table 4.3:

Table 4.3: *The partial derivatives of the current-mismatch function in polar coordinates with respect to the reactive power Q.*

$J_{ij} = \frac{\partial \mathbf{F}_i(\mathbf{x})}{\partial x_j}$		
	$i \neq j$	$\frac{\partial \Delta I_i^r(\mathbf{x})}{\partial Q_j} = 0$ $\frac{\partial \Delta I_i^m(\mathbf{x})}{\partial Q_j} = 0$
	$i = j$	$\frac{\partial \Delta I_j^r(\mathbf{x})}{\partial Q_j} = \frac{\sin \delta_j}{\lvert V_j \rvert^{sp}}$ $\frac{\partial \Delta I_j^r(\mathbf{x})}{\partial Q_j} = -\frac{\cos \delta_j}{\lvert V_j \rvert^{sp}}$

Now we add the derivatives $\frac{\partial \Delta I_i^r}{\partial Q_j}$ and $\frac{\partial \Delta I_i^m}{\partial Q_j}$ into the Jacobian matrix $J(\mathbf{x})$ and the correction ΔQ_j into the correction vector $\Delta \mathbf{x}$ for each PV bus j. As a result, the Jacobian matrix becomes a square again. The initial reactive power Q_j^0 at each PV bus j is computed as follows:

$$Q_j^0 = \sum_{k=1}^{N_b} |V_j||V_k|(G_{jk} \sin \delta_{jk} - B_{jk} \cos \delta_{jk}). \tag{4.26}$$

In each Newton iteration, the correction ΔQ_j is computed and the reactive power Q_j is updated using the computed correction.

4.3.5 Cartesian current-mismatch version (NR-c-car)

The Jacobian matrix equation (2.20) is computed using the current-mismatch function in Cartesian coordinates as:

$$-\left[\begin{array}{c|c} J^{11} & J^{12} \\ \hline J^{21} & J^{22} \end{array}\right] \begin{bmatrix} \Delta V^m \\ \Delta V^r \end{bmatrix} = \begin{bmatrix} \Delta I^r \\ \Delta I^m \end{bmatrix} \tag{4.27}$$

where all sub-matrices are given as $J^{11} = \frac{\partial \Delta I^r}{\partial V^m}$, $J^{12} = \frac{\partial \Delta I^r}{\partial V^r}$, $J^{21} = \frac{\partial \Delta I^m}{\partial V^m}$ and $J^{22} = \frac{\partial \Delta I^m}{\partial V^r}$.

In paper [40], the real ΔI^r and imaginary ΔI^m current-mismatch functions are expressed in terms of the real ΔP and reactive ΔQ power-mismatch functions. Then the reactive power-mismatch ΔQ is considered as a dependent variable for each PV bus and computed at each Newton iteration. Minor attempts were made to speed up the solution method using a partly constant approximation of the Jacobian during the iterations, but the results were not encouraging [40].

In our approach, the reactive power Q_j is chosen as a dependent variable for each PV bus j as polar current-mismatch version 4.3.4. Since Q_j is an unknown variable,

all the first order partial derivatives $\frac{\partial \Delta I_i^r}{\partial Q_j}$ and $\frac{\partial \Delta I_i^m}{\partial Q_j}$ have to be computed as shown in Table 4.4:

Table 4.4: *The partial derivatives of the current-mismatch function in Cartesian coordinates with respect to the reactive power Q.*

$J_{ij} = \frac{\partial \mathbf{F}_i(\mathbf{x})}{\partial x_j}$	$i \neq j$	$\frac{\partial \Delta I_i^r(\mathbf{x})}{\partial Q_j} = 0$ $\frac{\partial \Delta I_i^m(\mathbf{x})}{\partial Q_j} = 0$
	$i = j$	$\frac{\partial \Delta I_j^r(\mathbf{x})}{\partial Q_j} = \frac{V_j^m}{(V_j^r)^2+(V_j^m)^2}$ $\frac{\partial \Delta I_j^r(\mathbf{x})}{\partial Q_j} = \frac{-V_j^r}{(V_j^r)^2+(V_j^m)^2}$

Now we add the derivatives $\frac{\partial \Delta I_i^r}{\partial Q_j}$ and $\frac{\partial \Delta I_i^m}{\partial Q_j}$ into the Jacobian matrix $J(\mathbf{x})$ and the correction ΔQ_j into the correction vector $\Delta \mathbf{x}$ for each PV bus j. After the addition, the Jacobian matrix becomes a rectangular matrix. In order to make the Jacobian matrix square, we add the column of the Jacobian matrix with respect to the derivatives $\frac{\partial \Delta I_i^r}{\partial V_j^r}$ and $\frac{\partial \Delta I_i^m}{\partial V_j^r}$ to the column with respect to the derivatives $\frac{\partial \Delta I_i^r}{\partial V_j^m}$ and $\frac{\partial \Delta I_i^m}{\partial V_j^m}$ using (4.21) as follows:

$$\frac{\partial \Delta I_i^r}{\partial V_j^m}\Delta V_j^m = \Big(\frac{\partial \Delta I_i^r}{\partial V_j^m} - \frac{V_j^m}{V_j^r}\frac{\partial \Delta I_i^r}{\partial V_j^r}\Big)\Delta V_j^m \tag{4.28}$$

$$\frac{\partial \Delta I_i^m}{\partial V_j^m}\Delta V_j^m = \Big(\frac{\partial \Delta I_i^m}{\partial V_j^m} - \frac{V_j^m}{V_j^r}\frac{\partial \Delta I_i^m}{\partial V_j^r}\Big)\Delta V_j^m. \tag{4.29}$$

Then the correction ΔV_j^r can be eliminated from the correction vector $\Delta \mathbf{x}$ for each PV bus j. The initial reactive power Q_j^0 at a PV bus j is computed as follows:

$$Q_j^0 = \sum_{k=1}^{N_b} \Big(V_j^m(G_{jk}V_k^r - B_{jk}V_k^m) - V_j^r(B_{jk}V_k^r + G_{jk}V_k^m)\Big). \tag{4.30}$$

In each Newton iteration, the correction ΔQ_j is computed and the reactive power Q_j is updated using the computed correction.

4.3.6 Complex current-mismatch version (NR-c-com)

The Jacobian matrix equation (2.20) is calculated using the current-mismatch function in complex form as:

$$-\left[\ J\ \right]\left[\ \Delta V\ \right] = \left[\ \Delta I\ \right] \tag{4.31}$$

where the Jacobian matrix $J = \frac{\partial \Delta S}{\partial V}$ is obtained by taking the first order partial derivatives of the complex current-mismatch functions with respect to the complex voltage V. Same as the complex power-mismatch version 4.3.3, this version is applicable for the power flow problem on networks with only a slack bus and PQ buses.

Table 4.5: *The partial derivatives of power-mismatch function in different coordinates.*

Coordinates		$\mathbf{J_{ik}} = \frac{\partial \mathbf{F_i(x)}}{\partial \mathbf{x_k}}$
Polar	$i \neq k$	$\frac{\partial \Delta P_i(\mathbf{x})}{\partial \lvert V_k \rvert} = -\lvert V_i \rvert (G_{ik}\cos\delta_{ik} + B_{ik}\sin\delta_{ik})$ $\frac{\partial \Delta Q_i(\mathbf{x})}{\partial \lvert V_k \rvert} = -\lvert V_i \rvert (G_{ik}\sin\delta_{ik} - B_{ik}\cos\delta_{ik})$ $\frac{\partial \Delta P_i(\mathbf{x})}{\partial \delta_k} = -\lvert V_i \rvert \lvert V_k \rvert (G_{ik}\sin\delta_{ik} - B_{ik}\cos\delta_{ik})$ $\frac{\partial \Delta Q_i(\mathbf{x})}{\partial \delta_k} = -\lvert V_i \rvert \lvert V_k \rvert (-G_{ik}\cos\delta_{ik} - B_{ik}\sin\delta_{ik})$
	$i = k$	$\frac{\partial \Delta P_i(\mathbf{x})}{\partial \lvert V_i \rvert} = -\Big(2\lvert V_i \rvert G_{ii} + \sum_{i\neq k} \lvert V_k \rvert (G_{ik}\cos\delta_{ik} + B_{ik}\sin\delta_{ik})\Big)$ $\frac{\partial \Delta Q_i(\mathbf{x})}{\partial \lvert V_i \rvert} = -\Big(-2\lvert V_i \rvert B_{ii} + \sum_{i\neq k} \lvert V_k \rvert (G_{ik}\sin\delta_{ik} - B_{ik}\cos\delta_{ik})\Big)$ $\frac{\partial \Delta P_i(\mathbf{x})}{\partial \delta_i} = -\sum_{i\neq k} \lvert V_i \rvert \lvert V_k \rvert (-G_{ik}\sin\delta_{ik} + B_{ik}\cos\delta_{ik})$ $\frac{\partial \Delta Q_i(\mathbf{x})}{\partial \delta_i} = -\sum_{i\neq k} \lvert V_i \rvert \lvert V_k \rvert (G_{ik}\cos\delta_{ik} + B_{ik}\sin\delta_{ik})$
Cartesian	$i \neq k$	$\frac{\partial \Delta P_i(\mathbf{x})}{\partial V_k^r} = -\Big(V_i^r G_{ik} + V_i^m B_{ik}\Big)$ $\frac{\partial \Delta Q_i(\mathbf{x})}{\partial V_k^r} = -\Big(V_i^m G_{ik} - V_i^r B_{ik}\Big)$ $\frac{\partial \Delta P_i(\mathbf{x})}{\partial V_k^m} = -\Big(V_i^m G_{ik} - V_i^r B_{ik}\Big)$ $\frac{\partial \Delta Q_i(\mathbf{x})}{\partial V_k^m} = V_i^r G_{ik} + V_i^m B_{ik}$
	$i = k$	$\frac{\partial \Delta P_i(\mathbf{x})}{\partial V_i^r} = -\Big(V_i^r G_{ii} + V_i^m B_{ii} + \sum_{k=1}^{N_b}(G_{ik}V_k^r - B_{ik}V_k^m)\Big)$ $\frac{\partial \Delta Q_i(\mathbf{x})}{\partial V_i^r} = -\Big(V_i^m G_{ii} - V_i^r B_{ii} + \sum_{k=1}^{N_b}(B_{ik}V_k^r + G_{ik}V_k^m)\Big)$ $\frac{\partial \Delta P_i(\mathbf{x})}{\partial V_i^m} = -\Big(V_i^m G_{iu} - V_i^r B_{ii} + \sum_{k=1}^{N_b}(B_{ik}V_k^r + G_{ik}V_k^m)\Big)$ $\frac{\partial \Delta Q_i(\mathbf{x})}{\partial V_i^m} = V_i^r G_{ii} + V_i^m B_{ii} - \sum_{k=1}^{N_b}(G_{ik}V_k^r - B_{ik}V_k^m)$
Complex form	$i \neq k$	$\frac{\partial \Delta S_i(\mathbf{x})}{\partial V_k} = -V_i Y_{ik}^*$
	$i = k$	$\frac{\partial \Delta S_i(\mathbf{x})}{\partial V_i} = -\Big(V_i Y_{ii}^* + I^*\Big)$

4.3.7 First order partial derivatives of mismatch functions

In all six variants of the Newton power flow method, the first order partial derivatives of the mismatch function $J = \frac{\partial \mathbf{F(x)}}{\partial \mathbf{x}}$ must be computed with respect to unknown voltages. Table 4.5 and 4.6 show the partial derivatives of both power and current mismatch function in different coordinates. For all derivatives using the complex matrix notation, we refer to [13, 14].

4.3.8 Corrections

In each iteration of the Newton power flow methods, the unknown voltages $\mathbf{x}$ are updated using the computed corrections $\Delta\mathbf{x}$ as $\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}^k$. Table 4.7 gives voltage corrections in different coordinates.

Table 4.6: *The partial derivatives of current-mismatch function in different coordinates.*

Coordinates	$\mathbf{J_{ik}} = \frac{\partial \mathbf{F_i(x)}}{\partial \mathbf{x_k}}$	
Polar	$i \neq k$	$\frac{\partial \Delta I_i^r(\mathbf{x})}{\partial \lvert V_k \rvert} = -(G_{ik}\cos\delta_k - B_{ik}\sin\delta_k)$ $\frac{\partial \Delta I_i^m(\mathbf{x})}{\partial \lvert V_k \rvert} = -(G_{ik}\sin\delta_k + B_{ik}\cos\delta_k)$ $\frac{\partial \Delta I_i^r(\mathbf{x})}{\partial \delta_k} = \lvert V_k \rvert(G_{ik}\sin\delta_k + B_{ik}\cos\delta_k)$ $\frac{\partial \Delta I_i^m(\mathbf{x})}{\partial \delta_k} = -\lvert V_k \rvert(G_{ik}\cos\delta_k - B_{ik}\sin\delta_k)$
	$i = k$	$\frac{\partial \Delta I_i^r(\mathbf{x})}{\partial \lvert V_i \rvert} = -(G_{ii}\cos\delta_i - B_{ii}\sin\delta_i) - \frac{P_i^{sp}\cos\delta_i + Q_i^{sp}\sin\delta_i}{\lvert V_i \rvert^2}$ $\frac{\partial \Delta I_i^m(\mathbf{x})}{\partial \lvert V_i \rvert} = -(G_{ii}\sin\delta_i + B_{ii}\cos\delta_i) - \frac{P_i^{sp}\sin\delta_i - Q_i^{sp}\cos\delta_i}{\lvert V_i \rvert^2}$ $\frac{\partial \Delta I_i^r(\mathbf{x})}{\partial \delta_i} = \lvert V_i \rvert(G_{ii}\sin\delta_i + B_{ii}\cos\delta_i) - \frac{P_i^{sp}\sin\delta_i - Q_i^{sp}\cos\delta_i}{\lvert V_i \rvert}$ $\frac{\partial \Delta I_i^m(\mathbf{x})}{\partial \delta_i} = -\lvert V_i \rvert(G_{ii}\cos\delta_i - B_{ii}\sin\delta_i) + \frac{P_i^{sp}\cos\delta_i + Q_i^{sp}\sin\delta_i}{\lvert V_i \rvert}$
Cartesian	$i \neq k$	$\frac{\partial \Delta I_i^r(\mathbf{x})}{\partial V_k^r} = -G_{ik}$ $\frac{\partial \Delta I_i^m(\mathbf{x})}{\partial V_k^r} = B_{ik}$ $\frac{\partial \Delta I_i^r(\mathbf{x})}{\partial V_k^m} = B_{ik}$ $\frac{\partial \Delta I_i^m(\mathbf{x})}{\partial V_k^m} = -G_{ik}$
	$i = k$	$\frac{\partial \Delta I_i^r(\mathbf{x})}{\partial V_i^r} = -G_{ii} - \frac{P_i^{sp}((V_i^r)^2 - (V_i^m)^2) + 2V_i^r V_i^m Q_i^{sp}}{\lvert V_i \rvert^4}$ $\frac{\partial \Delta I_i^m(\mathbf{x})}{\partial V_i^r} = -B_{ii} + \frac{Q_i^{sp}((V_i^r)^2 - (V_i^m)^2) - 2V_i^r V_i^m P_i^{sp}}{\lvert V_i \rvert^4}$ $\frac{\partial \Delta I_i^r(\mathbf{x})}{\partial V_i^m} = B_{ii} + \frac{Q_i^{sp}((V_i^r)^2 - (V_i^m)^2) - 2V_i^r V_i^m P_i^{sp}}{\lvert V_i \rvert^4}$ $\frac{\partial \Delta I_i^m(\mathbf{x})}{\partial V_i^m} = -G_{ii} + \frac{P_i^{sp}((V_i^r)^2 - (V_i^m)^2) + 2V_i^r V_i^m Q_i^{sp}}{\lvert V_i \rvert^4}$
Complex form	$i \neq k$	$\frac{\partial \Delta I_i(\mathbf{x})}{\partial V_k} = -Y_{ik}$
	$i = k$	$\frac{\partial \Delta I_i(\mathbf{x})}{\partial V_i} = -\left(\frac{S_i^{sp}}{V_i^2} + Y_{ii}\right)$

Table 4.7: *Bus voltage corrections in different coordinates.*

Coordinates	**Bus type**	$\mathbf{x}^{h+1} := \mathbf{x}^h + \Delta\mathbf{x}^h$
Polar	PQ and PV	$V_i^{(h+1)} = \lvert V_i \rvert^{(h+1)} e^{\imath\delta_i^{(h+1)}}$
		$\lvert V \rvert_i^{(h+1)} = \lvert V \rvert_i^{(h)} + \Delta\lvert V_i \rvert^{(h)}$ $\delta_i^{(h+1)} = \delta_i^{(h)} + \Delta\delta_i^{(h)}$
Cartesian	PQ and PV	$V_i^{(h+1)} = (V_i^r)^{(h+1)} + \imath(V_i^m)^{(h+1)}$
		$(V_i^r)^{(h+1)} = (V_i^r)^{(h)} + (\Delta V_i^r)^{(h)}$ $(V_i^m)^{(h+1)} = (V_i^m)^{(h)} + (\Delta V_i^m)^{(h)}$
	PQ	$V_i^{(h+1)} = \lvert V_i \rvert^{(h+1)} e^{\imath\delta_i^{(h+1)}}$
		$\Delta\lvert V_j \rvert = \frac{V_j^r}{\lvert V_j \rvert}\Delta V_j^r + \frac{V_j^m}{\lvert V_j \rvert}\Delta V_j^m$ $\Delta\delta_j = \frac{V_j^r}{\lvert V_j \rvert^2}\Delta V_j^m - \frac{V_j^m}{\lvert V_j \rvert^2}\Delta V_j^r$
	PV	$\Delta\delta_j = \frac{\Delta V_j^m}{V_j^r}$
Complex	PQ (NR-p-com)	$V_i^{(h+1)} = V_i^{(h)} + (\Delta V_i^{(h)})^*$
	PQ (NR-c-com)	$V_i^{(h+1)} = V_i^{(h)} + \Delta V_i^{(h)}$

4.4 Numerical results

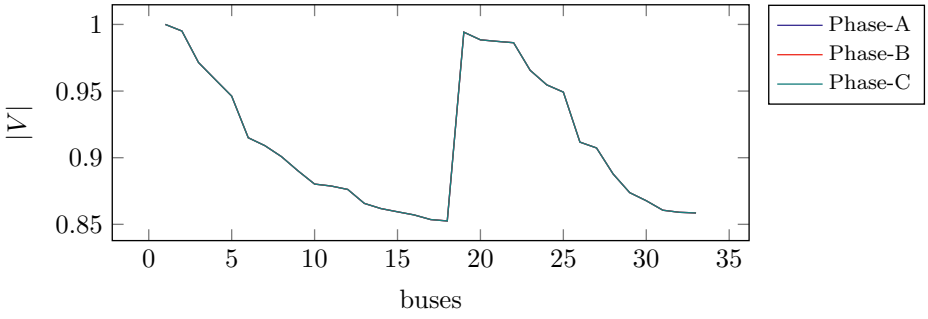
In this section, the newly developed/improved versions of the Newton power flow method (NR-p-car, NR-c-pol, NR-c-car, and NR-c-com) discussed in section 4.3, are compared to the existing versions of the Newton power flow method (NR-p-pol [1], NR-p-car [92] and NR-c-car [40]) in order to test the convergence ability and scalability of all variants. Furthermore, all newly developed versions of the Newton power flow method are extended to three-phase power flow problems with unbalanced distribution networks. Depending on the properties of a given network, one version can work better than the other version. Therefore, we use different load models, transformer connections, loading conditions, and R/X ratios in order to analyze the convergence ability and scalability of all variants of the Newton power flow method. The backward–forward sweep-based algorithm (BFS [65]) is implemented for comparisons with variants of the Newton power flow method.

Two balanced distribution networks (Dcase33 [94] and Dcase69 [95]) and four balanced transmission networks taken from Matpower [96] (Tcase1354, Tcase2737, Tcase9241 and Tcase13659) are used for single-phase nonlinear power flow computations. We apply two unbalanced IEEE test networks (UDcase13 [97] and UDcase37 [97]) for three-phase nonlinear power flow computations.

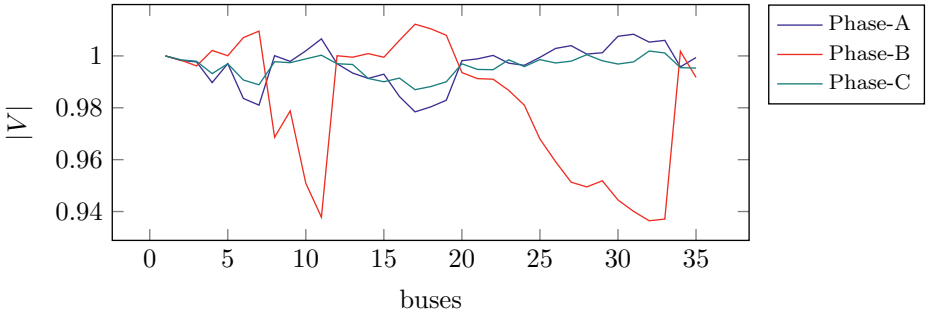
All methods are implemented in Matlab. The relative convergence tolerance is set to 10^{-5} and the maximum number of iterations is set to 10. All experiments are performed on an Intel computer i5-4690 3.5 GHz CPU with four cores and 64 Gb memory, running a Debian 64-bit Linux 8.7 distribution.

4.4.1 Comparison between single-phase and three-phases

For a balanced network such as a transmission grid, all electrical quantities at each phase are expected to be the same. Therefore, balanced networks are modeled with single-phase power flow problems (4.5). However, for unbalanced networks such as a distribution grid, three-phase power flow problems (4.6) have to be computed instead of (4.5). In Figure 4.1, we show computed voltage magnitudes of the balanced distribution network Dcase33 and the unbalanced distribution network UDcase37 for the comparison reason. As it is clear from the figure, computed voltage magnitudes are the same at each phase for Dcase33 whereas UDcase37 has different voltage magnitudes at each phase of the same bus. Moreover, a single admittance y between two buses in the balanced network becomes a 3×3 matrix in the unbalanced network as you can see in equation (3.6). Therefore, the computation work of three-phase power flow computations is much larger compared to single-phase power flow computations as size of problem increases. Figure 4.2 and 4.3 display the sparsity structure of admittance and Jacobian matrices for Dcase33 and UDcase37.

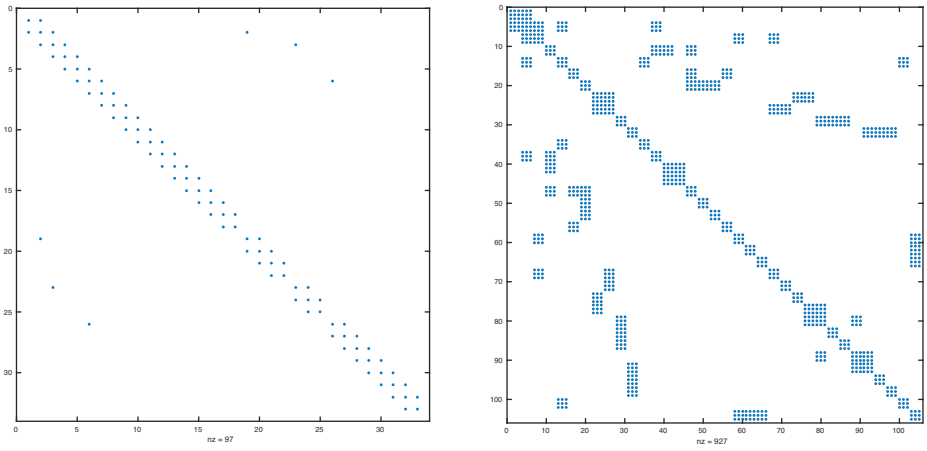


(a) $|V|$ *for Dcase33*



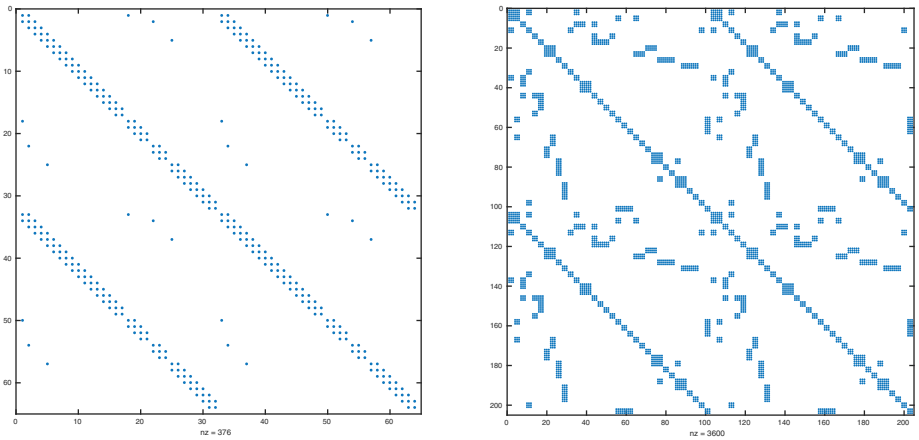
(b) $|V|$ *for UDcase37*

Figure 4.1: *Computed voltage magnitudes* $|V|$ *of Dcase33 and UDcase37.*



(a) Y *for Dcase33* (b) Y *for UDcase37*

Figure 4.2: *Sparsity of the admittance matrix* Y *for Dcase33 and UDcase37.*



(a) *J for Dcase33* (b) *J for UDcase37*

Figure 4.3: *Sparsity of the Jacobian matrix J for Dcase33 and UDcase37.*

4.4.2 Single-phase nonlinear power flow computations

Transmission networks:

Since the NR-p-com and NR-c-com versions are developed for only PQ buses, these variants are not applied to transmission power flow problems including PV buses. Tables 4.8 and Table 4.9 show the convergence results of all Newton power flow variants for transmission networks.

Table 4.8: *Numerical results of nonlinear power flow computations on small balanced transmission networks: Tcase1354 and Tcase2737.*

Methods	**Test Cases**					
	Tcase1354			**Tcase2737**		
	Iter	Time(s)	$\|\|F(\vec{x})\|\|_\infty$	Iter	Time(s)	$\|\|F(\vec{x})\|\|_\infty$
NR-p-pol [1]	3	0.028	6.267×10^{-6}	4	0.0640	1.535×10^{-8}
NR-p-car	3	0.026	1.579×10^{-6}	4	0.0634	2.350×10^{-6}
NR-p-car [92]	3	0.029	2.248×10^{-6}	5	0.0777	2.851×10^{-6}
NR-c-pol	3	0.031	8.300×10^{-10}	4	0.0700	6.173×10^{-7}
NR-c-car	3	0.030	6.144×10^{-10}	4	0.0649	8.678×10^{-7}
NR-c-car [40]	5	0.050	9.996×10^{-6}	5	0.0838	7.984×10^{-7}

For smaller transmission networks Tcase1354 and Tcase2737, all versions result in the same behavior except NR-c-car developed in [40] which requires extra one itera-

tion. As you can see in Table 4.9, the variant NR-c-car [40] diverges for TCase9241, whereas other versions converge.

Table 4.9: *Numerical results of nonlinear power flow computations on large balanced transmission networks: Tcase9241 and Tcase13659.*

Methods	**Test Cases**					
	Tcase9241			**Tcase13659**		
	Iter	Time(s)	$\|\|F(\vec{x})\|\|_\infty$	Iter	Time(s)	$\|\|F(\vec{x})\|\|_\infty$
NR-p-pol [1]	6	0.355	2.129×10^{-9}	5	0.389	2.289×10^{-9}
NR-p-car	5	0.291	2.102×10^{-8}	6	0.468	7.983×10^{-12}
NR-p-car [92]	5	0.318	2.074×10^{-6}	10	0.889	1.401×10^{148}
NR-c-pol	3	0.197	6.474×10^{-7}	4	0.363	3.436×10^{-9}
NR-c-car	3	0.199	1.943×10^{-6}	4	0.362	8.617×10^{-9}
NR-c-car [40]	10	0.659	0.002	10	0.903	1.148

Furthermore, NR-c-pol and NR-c-car versions developed in this thesis converge after only three iterations whereas other versions (NR-p-pol [1], NR-p-car and NR-p-car [92]) need five to six iterations. For the largest transmission network TCase13659, both variants NR-p-car [92] and NR-c-car [40] diverge whereas all variants (NR-p-car, NR-c-pol and NR-c-car) developed in this thesis and NR-p-pol [1] find the solution. Additionally, these four converged versions have a quadratic convergence as shown in Figure 4.4. Moreover, NR-c-pol and NR-c-car variants converge faster than the most famous variant NR-p-pol [1] in terms of iterations for both large transmission networks. Thus, we can conclude that NR-c-pol and NR-c-car variants are more preferable for large transmission power flow problems.

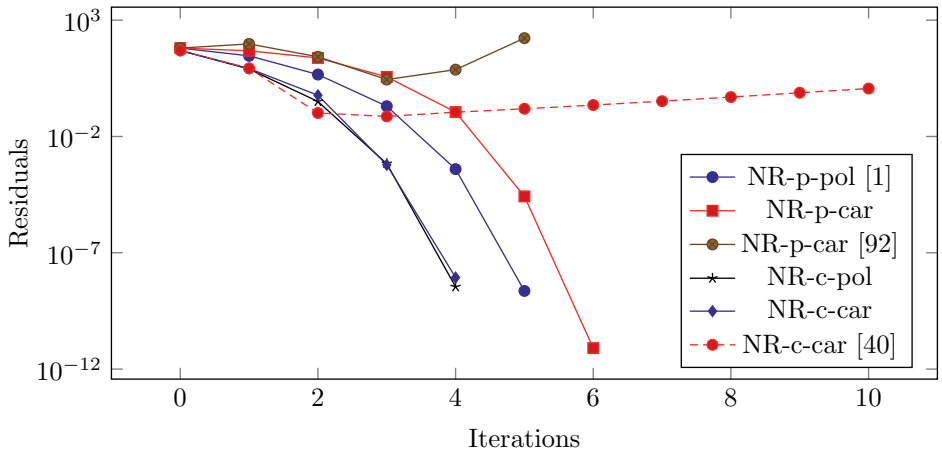


Figure 4.4: *Convergence of all Newton power flow versions for the large transmission network Tcase13659.*

Distribution networks:

The numerical results of all Newton power flow variants and the BFS algorithm are given in Table 4.10 for two balanced distribution network (Dcase33 and Dcase69).

Table 4.10: *Numerical results of nonlinear power flow computations on two balanced distribution networks: Dcase33 and Dcase69*

Methods	**Test Cases**					
	Dcase33			**Dcase69**		
	Iter	Time(s)	$\|\|F(\vec{x})\|\|_\infty$	Iter	Time(s)	$\|\|F(\vec{x})\|\|_\infty$
NR-p-pol [1]	3	0.0123	7.467×10^{-6}	4	0.0131	5.587×10^{-9}
NR-p-car	3	0.0067	1.043×10^{-6}	3	0.0069	8.177×10^{-6}
NR-p-car [92]	3	0.0072	1.089×10^{-6}	3	0.0081	8.094×10^{-6}
NR-p-com	6	0.0058	6.461×10^{-6}	7	0.0060	4.013×10^{-6}
NR-c-pol	3	0.0087	1.429×10^{-9}	3	0.0090	8.522×10^{-9}
NR-c-car	3	0.0073	1.395×10^{-9}	3	0.0077	1.950×10^{-8}
NR-c-car [40]	3	0.0111	1.396×10^{-9}	3	0.0124	1.947×10^{-8}
NR-c-com	7	0.0068	5.379×10^{-6}	10	0.0084	2.769×10^{-6}
BFS [65]	7	0.0102	1.045×10^{-6}	7	0.0104	7.777×10^{-6}

From Table 4.10, we observe that NR-c-pol and NR-c-car versions have the best performances in terms of a number of iterations and the residual norm of the mismatch function. Although NR-p-pol [1], NR-p-car and NR-p-car [92] versions converged after the same number of iterations, the value of the residual norm is larger than for the NR-c-pol and NR-c-car versions. This means that if we set the tolerance to 10^{-7}, these versions will need extra iterations to converge, whereas NR-c-pol and NR-c-car versions still converge after three iterations. We also see that NR-p-com, NR-c-com and BFS [65] methods need more iterations and have a linear convergence compared to other versions which have a quadratic convergence. These three methods solve the power flow problem in complex form, whereas other versions of the Newton power flow method reformulate the problem into real equations using Cartesian and polar coordinates. Overall, variants NR-c-pol and NR-c-car developed in this thesis perform the best for both distribution networks in terms of both a number of iterations and the residual norm.

Figure 4.5 compares the computed voltage magnitudes of Dcase69 using all Newton power flow variants with the well-known result of the existing method [70]. As we can see, all results of the proposed solution methods match the well-known result well with accuracy of 10^{-5}. In order to analyze the convergence ability and scalability of all Newton power flow variants, we compute the balanced distribution network Dcase69 with different load models, loading conditions, and R/X ratios. Different loading conditions are considered by multiplying each bus's power S by a constant k as $S = k * S$ where k is chosen from $[1; 1.5; 2]$. Similarly, different R/X ratios are obtained by multiplying each branch resistance by a constant k as $Z = k * R + \imath X$ with k is equal to one of $[1; 1.5; 2.5]$. Finally, the performance of the solution methods

is evaluated for a 'constant power' and a 'constant polynomial' load models defined in Section 3.3.

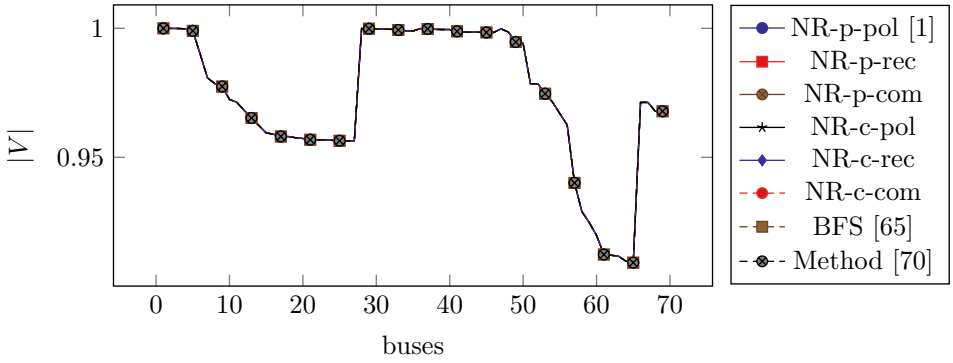


Figure 4.5: *Comparison of the computed voltage magnitude $|V|$ of Dcase69 using the Newton power flow variants with the well-known result [70].*

Figures 4.6 and Figure 4.7 show convergence results of all Newton power flow variants and the BFS algorithm for the balanced distribution network Dcase69 with different loading conditions and R/X ratios. We see that NR-p-com, NR-c-com, and BFS [65] methods are more sensitive to the change of loading conditions and R/X ratios compared to other versions that use real variables and values. Moreover, it is clear that NR-c-pol and NR-c-car variants are the robust variants of all. Figure 4.8 displays convergence results of all solution methods for Dcase69 having two different load models. Furthermore, a 'constant power' load model provides faster convergences to all solution methods than a 'constant polynomial' load model as we can see it from the figure.

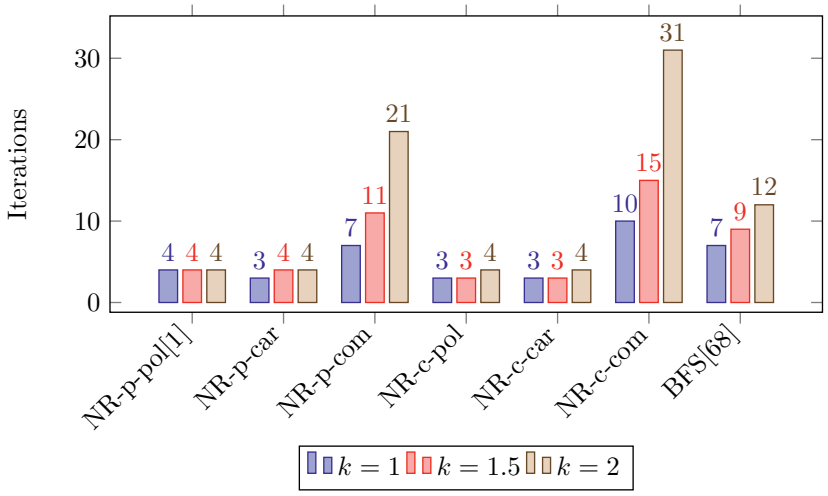


Figure 4.6: *Convergence results of all Newton power flow variants and the BFS algorithm for Dcase69 with different loading conditions ($S = k * S$).*

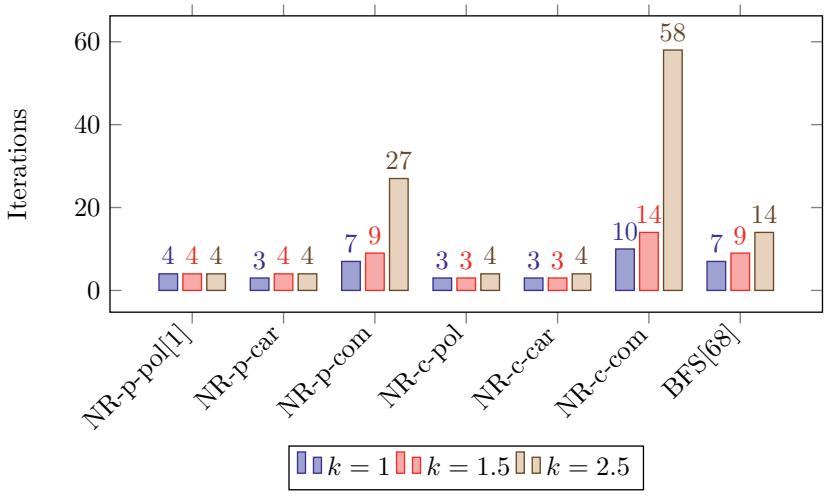


Figure 4.7: *Convergence results of all Newton power flow variants and the BFS algorithm for Dcase69 with different R/X ratios ($Z = k * R + \imath X$).*

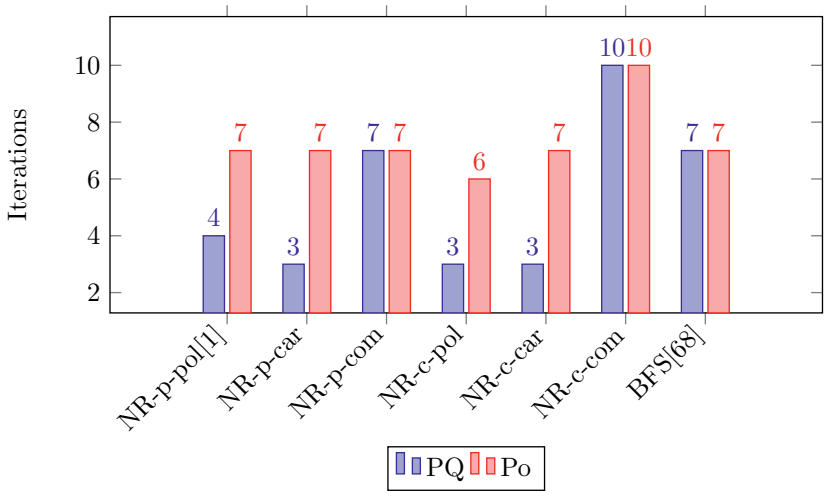


Figure 4.8: *Convergence results of all Newton power flow variants and the BFS algorithm for Dcase69 with different load models (constant power (PQ) and constant polynomial (Po)).*

From Table 4.10 and Figures 4.5 - 4.8, we can conclude that NR-c-pol and NR-c-car versions developed in this thesis are more suitable for balanced distribution networks than other versions (NR-p-pol [1], NR-p-car, NR-p-car [92] and NR-c-car [40]). Furthermore, NR-p-com and NR-c-com versions, as well as BFS [65] are the least preferable methods for balanced distribution networks in terms of convergence and robustness.

4.4.3 Three-phase nonlinear power flow computations

Unbalanced distribution networks:

For both unbalanced distribution networks UDcase13 and UDcase37, regulators are removed and all three-phase loads are chosen to be connected in a grounded Wye configuration as defined in Section 3.3. For UDcase13, the transformer is connected in Wye-G, whereas UDcase37 has the delta–delta transformer connection as defined in Section 3.6. The BFS method [65] is not implemented for three-phase power flow problems since it is not explained in sufficient detail how the three-phase transformer is handled for this method. Table 4.11 shows the convergence result of all solution methods for UDcase13 and UDcase37. From the table, we see that all methods converge after the same number of iterations except NR-p-com and NR-c-com versions. However, NR-c-pol and NR-c-car versions have better performance in terms of both the number of iterations and the residual norm of the mismatch function, as we had the same results for balanced distribution networks. Again, NR-p-com and NR-c-com versions need more iterations to converge compared to other versions.

Table 4.11: *Numerical results of nonlinear power flow computations on two unbalanced distribution networks: UDcase13 and UDcase37.*

Methods	**Test Cases**					
	UDcase13			**UDcase37**		
	Iter	Time(s)	$\|\|F(\vec{x})\|\|_\infty$	Iter	Time(s)	$\|\|F(\vec{x})\|\|_\infty$
NR-p-pol [1]	3	0.0116	1.557×10^{-9}	2	0.0134	3.415×10^{-7}
NR-p-car	3	0.0067	6.701×10^{-9}	2	0.0069	1.162×10^{-7}
NR-p-com	5	0.0055	5.095×10^{-7}	3	0.0055	5.339×10^{-7}
NR-c-pol	3	0.0087	6.997×10^{-11}	2	0.0094	3.975×10^{-8}
NR-c-car	3	0.0073	8.149×10^{-11}	2	0.0079	4.033×10^{-8}
NR-c-com	5	0.0067	3.558×10^{-7}	3	0.0065	7.498×10^{-7}

Again, the convergence ability and scalability of all Newton power flow variants are tested by solving the unbalanced distribution network UDcase13 with different load models, loading conditions, and R/X ratios. Different loading conditions are considered by multiplying each bus's power S by a constant k as $S = k * S$ where k is chosen from $[1; 10; 20]$. Similarly, different R/X ratios are obtained by multiplying each branch resistance by a constant k as $Z = k * R + \imath X$ with k is equal to one of $[1; 10; 20]$. A 'constant power' and a 'constant polynomial' load models are used for UDcase13. Convergence results of all solution methods for the unbalanced distribution network UDcase13 with different loading conditions and R/X ratios are shown in Figures 4.9 and 4.10, respectively. As in single-phase cases, NR-p-com and NR-c-com versions are more sensitive to the change of loading conditions and R/X ratios compared to other versions. Moreover, NR-c-pol and NR-c-car versions are more stable and therefore they can be applied to any unbalanced distribution networks with high R/X ratios or extreme loading conditions. All methods result in better performances with a 'constant power' (PQ) load model than a 'constant polynomial' load model as

shown in Figure 4.11. Therefore, we can conclude that versions using the current mismatch functions (NR-c-pol and NR-c-car) are more suitable than versions using the power-mismatch functions (NR-p-pol [1] and NR-p-car) for unbalanced distribution networks.

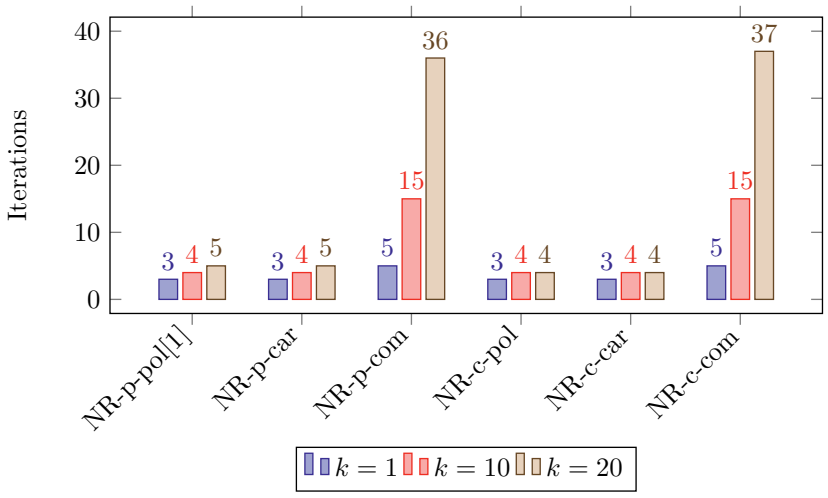


Figure 4.9: *Convergence results of all Newton power flow variants for UDcase13 with different loading conditions ($S = k * S$).*

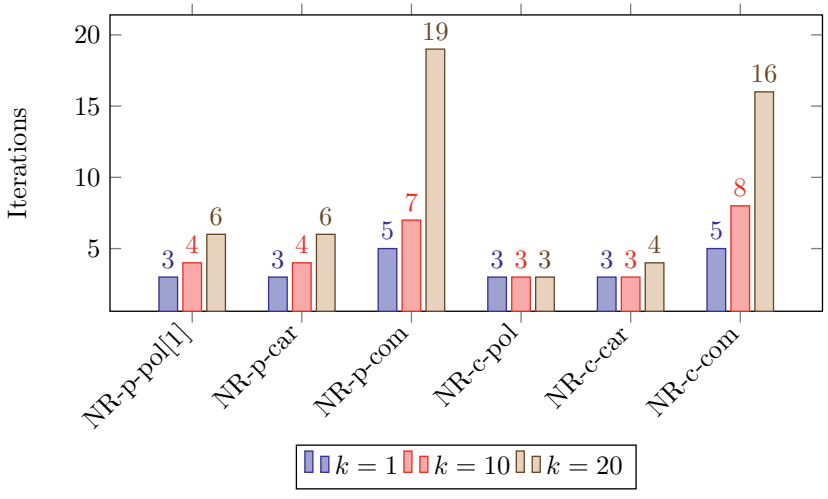


Figure 4.10: *Convergence results of all Newton power flow variants for UDcase13 with various R/X ratios ($Z = k * R + \imath X$).*

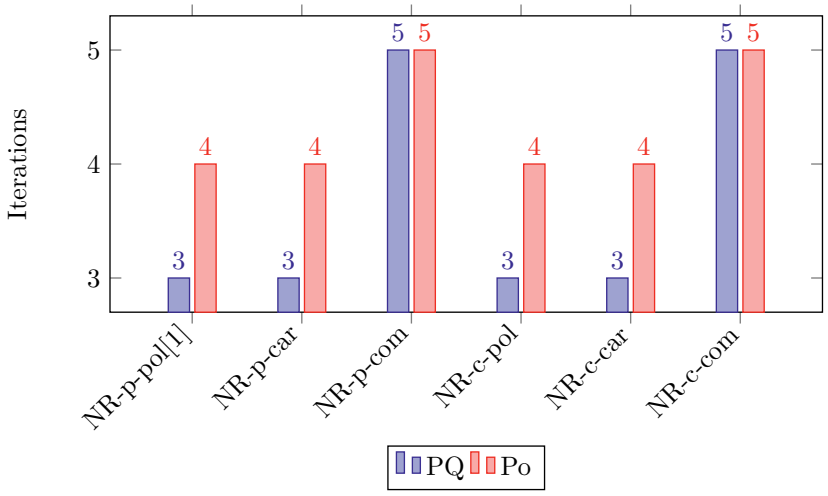


Figure 4.11: *Convergence results of all Newton power flow variants for UDcase13 with different load models (constant power (PQ) and constant polynomial (Po)).*

4.5 Conclusion

We formulate and analyze the Newton based power flow methods that are used for the power flow computation on distribution and transmission networks. For the various methods, we consider two different mismatch formulations: the current and power balance equations and three different coordinate systems: Cartesian, polar, and complex form. This leads to six different versions of the Newton power flow method. Studying these versions in a common framework enables us to analyze and compare all variants in a unified way. The existing variants of the Newton power flow method developed in [1, 40, 92] are implemented and compared with the new versions of the Newton power flow method (NR-p-car, NR-c-pol, NR-c-car, and NR-c-com) developed in this thesis. In case of the polar and Cartesian current-mismatch versions, the reactive power Q is chosen as a dependent variable for each PV bus. Thus, we compute the correction ΔQ at each iteration and update Q using the computed corrections. Equations (4.2) and (4.3) are used instead of the voltage-magnitude-squared mismatch equation (4.1) in versions using Cartesian coordinates. The order of the Jacobian matrix equation is $(2N - N_g - 2)$ for the versions using the power-mismatch function whereas versions using the current-mismatch function have $(2N - 2)$ linear equations.

Furthermore, all Newton power flow variants are extended to three-phase power flow problems on unbalanced distribution networks. The backward–forward sweep-based algorithm (BFS [65]) is implemented for comparisons on distribution networks. Various mathematical models of the load, three-phase load connection, and three-phase transformer connection are studied and applied in the numerical experiments. As a result of the numerical experiment, NR-c-pol and the NR-c-car that are developed in this thesis perform the best for both balanced and unbalanced networks. We also investigate which version can be applied to what kind of a power network by comparing all versions for distribution networks with different loading conditions,

R/X ratios, and load models. We observe that NR-c-pol and NR-c-car versions are more stable to the change of loading conditions and R/X ratios for both balanced and unbalanced networks, whereas the performance of other methods is highly sensitive to them. Therefore, we conclude that these two versions are the fastest and the most robust versions of all Newton power flow variants that can be applied to single or three-phase power flow problems in any balanced or unbalanced networks. All newly developed versions of the Newton power flow method have been implemented and included in the current version of Matpower (Matpower 7.0) which is a Matlab package for solving power flow and optimal power flow problems.

Chapter 5

Linear Power Flow Computations

In this chapter, we propose a linear formulation of the original nonlinear power flow problem. The theoretical background is introduced for the linearization of the nonlinear power flow problem. Our linear formulation leads to a simple linear matrix equation that can be solved directly. With any good approximation, the direct method can give a reasonable solution that is close to the original solution. Moreover, in order to obtain more accurate results, we develop the iterative linear power flow method solving the linear power flow equation in every iteration. Accuracy and efficiency of both direct and iterative linear approaches are validated by comparing them with the conventional Newton power flow algorithm on various transmission and distribution networks. We compare the performance of our linear power flow methods with nonlinear power flow methods in terms of CPU time and relative difference. The convergence of the iterative linear power flow method is studied in details.

5.1 Introduction

Traditionally, the power flow problem is formulated as a nonlinear system of equations. Thus, iterative type of methods such as the Gauss-Seidel (G-S), Newton power flow (N-R) and Fast Decoupled Load Flow (FDLF) [1, 30, 31] are widely used to solve the so-called Nonlinear Power Flow (NPF) problem for transmission networks. Many methods [40, 56, 64, 98] have been developed on distribution power flow analysis, and most of them are based on the Backward-Forward Sweep (BFS) algorithm. Several reviews on distribution power flow solution methods can be found in [85–87].

This chapter is based on:

B. Sereeter, A. Markensteijn, M. E. Kootte, C. Vuik, and C. Witteveen, "A novel linearized power flow approach for transmission and distribution networks," *IEEE PES Transaction on Power Systems [Under review]*, Dec 2019.

Numerous researchers developed the methods that linearize the NPF equations using some approximations and assumptions in order to obtain the Linear Power Flow (LPF) equations. After the linearization, the resulting LPF equations can be solved by direct solvers. In a linear approach, the actual model of the power system is altered and therefore, the final solution of LPF computations is different from the outcome of NPF computations. However, it is essential that both LPF and NPF computations result in similar or close solutions within the given thresholds. Therefore, LPF computations are generally faster than NPF computations and are more suitable to be applied on extensive networks with millions of cables for real-time simulation.

The best-known example of the LPF problem is the DC load flow [2] where linear relations are determined between the active power injections P and the voltage angles δ, and the reactive power injections Q and the deviations of the unknown voltage magnitudes $\Delta|V|$. Furthermore, the linear power flow formulation is obtained based on a voltage-dependent (ZI) load model and some numerical approximations on the imaginary part of the nodal voltages in [99]. Another linear power flow model based on Taylor's series expansion was proposed in [100]. A direct method taking advantage of the special structure of distribution systems is also developed in [65]. There are other linear power flow methods [100–107].

In this thesis, we propose a fast LPF algorithm that takes advantage of the mathematical model and physical knowledge of power systems. In order to improve the accuracy of our direct linear power flow method, we develop the iterative method solving the linear power flow in every iteration.

5.2 Linear Power Flow problem

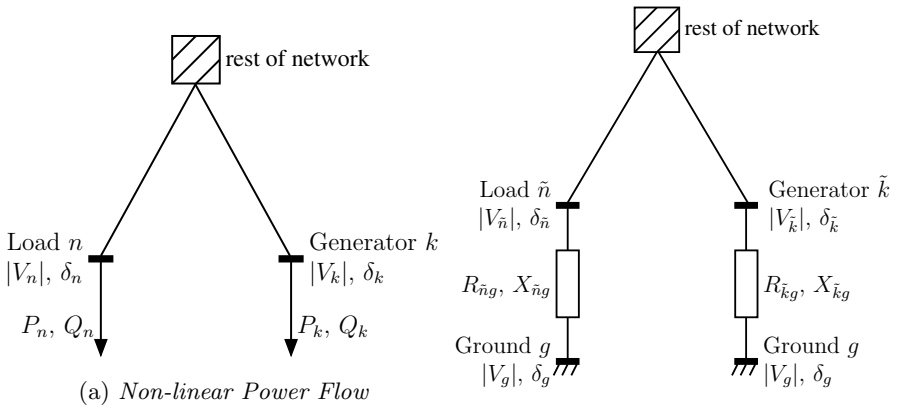
In this linear approach, we do not solve the actual nonlinear power flow equations (4.5) and instead we use the equation (3.4) defined by Kirchoff's Current Law to approximate the solution of equations (4.5):

$$I = YV. \tag{5.1}$$

It is impossible to compute voltage V from equation (5.1), because current I is also unknown as it depends on V. Therefore, we apply some assumptions to equation (5.1). First we find all load buses n with nonzero loads in the network and connect these buses to the artificial ground buses g. These additional ground buses g are then included in the network now as can be seen in Figure 5.1. The injected power of nonzero load n and generator k is shifted to the new connection, such that the buses $\tilde{n}$ and $\tilde{k}$ have zero injected power and current. The connection between the artificial ground bus and the nonzero load or generator bus, is modelled as a short transmission line. Using i to denote the load bus n or the generator bus k, we have:

$$\begin{aligned} P_{\tilde{\imath}g} &= G_{\tilde{\imath}g}|V_{\tilde{\imath}}|^2 - |V_{\tilde{\imath}}||V_g|\left(G_{\tilde{\imath}g}\cos\delta_{\tilde{\imath}g} + B_{\tilde{\imath}g}\sin\delta_{\tilde{\imath}g}\right) \\ Q_{\tilde{\imath}g} &= -B_{\tilde{\imath}g}|V_{\tilde{\imath}}|^2 - |V_{\tilde{\imath}}||V_g|\left(G_{\tilde{\imath}g}\sin\delta_{\tilde{\imath}g} - B_{\tilde{\imath}g}\cos\delta_{\tilde{\imath}g}\right) \end{aligned} \tag{5.2}$$

Here, G_{ij} and B_{ij} are the conductance and susceptance for a line between bus i and bus j, and $\delta_{ij} := \delta_i - \delta_j$ is the voltage angle difference. We assume that bus $\tilde{\imath}$ acts



(a) *Non-linear Power Flow* (b) *Linear Power Flow*

Figure 5.1: *Network representation of a load bus n and a generator bus k for standard NPF (a) and for LPF (b).*

the same as bus i, as seen from the rest of the network. That is, we assume

$$\begin{aligned} |V_{\tilde{\imath}}| &= |V_i|, \qquad \delta_{\tilde{\imath}} = \delta_i, \\ P_{\tilde{\imath}g} &= P_i, \qquad Q_{\tilde{\imath}g} = Q_i. \end{aligned} \tag{5.3}$$

Substituting these assumptions in (5.2) and setting $|V_g| = 0$ for the artificial ground buses, the conductance and susceptance for the additional lines are given by:

$$\begin{aligned} G_{\tilde{\imath}g} &= \frac{P_i}{|V_i|^2}, \\ B_{\tilde{\imath}g} &= \frac{-Q_i}{|V_i|^2}. \end{aligned} \tag{5.4}$$

Resistance $R_{\tilde{\imath}g}$ and reactance $X_{\tilde{\imath}g}$ for the additional lines are computed by substituting (5.4) into $Z = \frac{1}{Y} = \frac{G}{|Y|^2} - \frac{B}{|Y|^2}$:

$$\begin{aligned} R_{\tilde{\imath}g} &= \frac{|V_i|^2 P_i}{P_i^2 + Q_i^2}, \\ X_{\tilde{\imath}g} &= \frac{|V_i|^2 Q_i}{P_i^2 + Q_i^2}. \end{aligned} \tag{5.5}$$

For a nonzero load node n, the injected active P_n and reactive power Q_n are known, while the voltage magnitude $|V_k|$ and injected active power P_k are specified for a generator node k. Denoting unknown variables by $\hat{[\cdot]}$, the resistance R_{ng} and reactance X_{ng} of the additional branches for nonzero load buses n become:

$$R_{ng} = \frac{|\hat{V}_n|^2 P_n}{P_n^2 + Q_n^2}, \tag{5.6}$$

$$X_{ng} = \frac{|\hat{V}_n|^2 Q_n}{P_n^2 + Q_n^2}, \tag{5.7}$$

and the resistance R_{kg} and reactance X_{kg} of the additional branches for generator buses k are given:

$$R_{kg} = \frac{P_k}{P_k^2 + \hat{Q}_k^2}, \tag{5.8}$$

$$X_{kg} = \frac{\hat{Q}_k}{P_k^2 + \hat{Q}_k^2}, \tag{5.9}$$

where $|V_k| = 1$. As a result of the artificial ground buses and additional lines, the number of buses and branches in the network increases by the number of generator buses and nonzero load buses in the original network. Using the resistance and reactance (5.6)-(5.9), we can build the new admittance matrix $\mathbf{Y}$ including the additional branches. Thus, we obtain the following linear power flow equation:

$$\mathbf{I} = \mathbf{YV} \tag{5.10}$$

with

$$\mathbf{I} = \begin{bmatrix} I_g \\ I \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} Y_{gg} & Y_{ig}^T \\ Y_{ig} & Y \end{bmatrix}, \mathbf{V} = \begin{bmatrix} V_g \\ V \end{bmatrix}. \tag{5.11}$$

Here I, V and Y are original current, voltage and admittance matrix respectively whereas I_g, V_g and Y_{gg} are current, voltage and admittance matrix respectively with respect to additional ground buses. Note that equation (5.10) still cannot be solved directly because not all elements are known in neither vector $\mathbf{I}$ and $\mathbf{V}$. Since V_g is the voltage of ground buses, it can be specified as equal to zero. Furthermore, we know that all swing buses V_{ref} in the original network are also specified and are equal to nominal voltage levels. Therefore, we can order equation (5.10) in such a way that all swing buses V_{ref} and ground buses V_g are placed in V_1 and all unknown voltages of the remaining buses are placed in $\hat{V}_2$ as:

$$\begin{bmatrix} \hat{I}_1 \\ \hat{I}_2 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{21}^T \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ \hat{V}_2 \end{bmatrix}. \tag{5.12}$$

Due to the shift of the injected power from the original load and generator nodes to the additional ground buses, KCL dictates that $\sum I_{ij} = 0$ for every bus i in $\hat{V}_2$. Therefore, $I_2 = 0$ and the power flow equations become:

$$\begin{bmatrix} \hat{I}_1 \\ 0 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{21}^T \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ \hat{V}_2 \end{bmatrix}. \tag{5.13}$$

The second row of equation (5.13) is a linear system of equations for the unknown $\hat{V}_2$ since V_1 is known and $\hat{I}_2 = 0$. Furthermore, the original unknown voltages V can be assembled as $V = \begin{bmatrix} V_{\text{ref}} \\ \hat{V}_2 \end{bmatrix}$.

5.3 Linear Power Flow methods

This section explains two solution approaches solving the linear power flow problem (5.13).

5.3.1 Direct approach

For the additional branches connecting nonzero load buses to ground buses, resistance R_{ng} and reactance X_{ng} depend on the unknown voltage magnitudes $|\hat{V}_n|$ of the nonzero load buses. Numerically, $|\hat{V}_n|$ is between 0 and 1 as power flow computations are done in per unit normalization. Similarly, resistance R_{kg} and reactance X_{kg} connecting generator buses to ground buses are dependent on unknown reactive power $\hat{Q}_k$ of the generator buses. It is possible to predetermine $\hat{Q}_k$ using a power factor and the specified active power P_k of the generator buses. If suitable values are chosen for the resistance R and reactance X of the additional branches (5.6) - (5.9), $\hat{V}_2$ is close to the original solution of equation (4.5). In that case, we can solve $\hat{V}_2$ directly from the second row of (5.13) as:

$$Y_{22}\hat{V}_2 = -Y_{21}V_1, \tag{5.14}$$

$$Y_{22}\hat{V}_2 = b, \tag{5.15}$$

$$\hat{V}_2 = Y_{22}^{-1}b. \tag{5.16}$$

5.3.2 Iterative approach

If suitable values are hard to choose for the resistance R and reactance X of the additional branches, we can still find the solution $\hat{V}_2$ of (5.13) by solving equation (5.16) iteratively. The iteration process of the iterative LPF method is given in Algorithm 5. In this approach, we assume that all generator buses are modeled as PQ buses. Handling generator buses as PV buses is still under research (see Appendix A). If generator buses are modeled as PQ buses, we use equations (5.8)-(5.9) with $\hat{Q}_k = Q_k$ where Q_k is the predetermined reactive power. In this case, R_{kg} and X_{kg} become a constant. Furthermore, this algorithm starts with an initial value for $|\hat{V}_n|$ that has to be updated in every iteration. It is more practical to set $|\hat{V}_n| = 1$. In our approach, we update $|\hat{V}_n^h|$ by replacing it by using $\hat{V}_2^h$ computed in equation (5.16) as $|\hat{V}_n^{h+1}| := |\hat{V}_2^h(S > 0)|$. Moreover, we simply replace new iterate $|\hat{V}_n^{h+1}|$ by the computed voltage magnitudes of nonzero load buses of previous iteration.

Algorithm 5 LPF method for PQ buses

1: Set iteration counter to zero $h := 0$
2: Give initial $|\hat{V}_n^0|$ for all nonzero load buses n with $S > 0$ (between 0.5 and 1)
3: Compute initial R_{ng}^0 and X_{ng}^0 using equations (5.6)-(5.7)
4: Compute Y including additional branches
5: Segment Y into Y_{11}, Y_{21} and Y_{22}, and compute b from equation (5.14)
6: **while** not converged
7: Solve equation (5.16) for $\hat{V}_2^h$
8: Replace iterate $|\hat{V}_n^{h+1}| := |\hat{V}_2^h(S > 0)|$
9: Compute R_{ng}^{h+1} and X_{ng}^{h+1} using equations (5.6)-(5.7) with $|\hat{V}_n^{h+1}|$
10: Update elements of Y_{22} w.r.t R_{ng}^{h+1} and X_{ng}^{h+1}
11: $h := h + 1$
12: **end while**

The iteration process stops when the infinity norm of $\Delta|V_n| = |\hat{V}_n^{h+1}| - |\hat{V}_n^h|$ is

smaller than some tolerance as $||\Delta|\hat{V}_n|\,||_\infty \leq 10^{-5}$. In this approach, it is unnecessary to rebuild the admittance matrix Y in every iteration. Instead, we build Y once and update only some elements of matrix Y_{22} w.r.t buses with nonzero loads using new iterate $|\hat{V}_n^{h+1}|$. The following steps show how we can update matrix Y_{22} without rebuilding it again:

1: Let ll be the vector of locations of nonzero load buses ($S > 0$) for Y_{22}.
2: Compute the error $e = y_{ng}^{h+1} - y_{ng}^{h}$, where

$$y_{ng}^{h} = \frac{1}{Z_{ng}^{h}} = \frac{1}{R_{ng}^{h} + \imath X_{ng}^{h}}$$

3: $Y_{22}(ll, ll) = Y_{22}(ll, ll) + diag(e)$, where $diag(e)$ diagonal matrix with e on its main diagonal.

This modification is done in step 10 of Algorithm 5.

5.4 Numerical results

In this section, the accuracy and efficiency of our linear approach are validated by comparing it with the NPF computation on various transmission and distribution networks. We compare the above direct and iterative LPF approaches to the Newton power flow algorithm discussed in section 4.3. We use five balanced transmission and distribution test cases from Matpower that are given in Table 5.1. Both direct and iterative LPF methods are implemented in Matlab. The relative convergence tolerance is set to 10^{-5} for both the Newton power flow (NPF) method and the iterative LPF algorithm. The maximum number of iterations is set to 10 and 100 for NPF and iterative LPF methods respectively. For the numerical experiments, all computations are done on Intel computer i5-6500 3.2 GHz CPU with four cores and 64 GB memory.

Table 5.1: *Description of used test cases.*

Systems	**Buses**	**Generators**	**Branches**
Transmission networks			
Tcase9	9	3	9
Tcase30	30	6	41
Tcase57	57	7	80
Tcase89	89	12	210
Tcase118	118	54	186
Distribution networks			
Dcase22	22	1	21
Dcase33 [94]	33	1	32
Dcase69 [95]	69	1	68
Dcase85	85	1	84
Dcase141	141	1	140

5.4.1 Direct approach

This subsection compares the direct LPF approach developed in section 5.3.1 to the NPF algorithm in terms of accuracy and speed. According to the theory when the actual solutions are given to $|\hat{V}_n|$ and $\hat{Q}_k$ in (5.6)-(5.9) then equation (5.16) leads to the exact solution of (4.5). To prove this, we use the solution of NPF computations for $|\hat{V}_n|$ and $\hat{Q}_k$, and solve the LPF problem on some test cases. Table 5.2 shows the numerical results of both NPF and LPF computations for the CPU time and the relative difference $\frac{||V^N - V^L||_2}{||V^N||_2}$ where V^N and V^L are the computed voltages of NPF and LPF computations respectively. The CPU time also includes data processing time.

Table 5.2: *The CPU time and the relative difference between NPF and direct LPF ($|\hat{V}_n| = |V_n^N|$ & $\hat{Q}_k = Q_k^N$) computations.*

Test cases	**LPF**	**NPF**	$\frac{\textbf{Time (NPF)}}{\textbf{Time (LPF)}}$	$\frac{\|\mathbf{V^N - V^L}\|_2}{\|\mathbf{V^N}\|_2}$
	Time(s)	Time(s) & Iter		
Transmission networks				
Tcase89	0.0032	0.0233 & 4 it	7.25	8.88×10^{-11}
Tcase118	0.0029	0.0206 & 3 it	7.04	3.06×10^{-7}
Distribution networks				
Dcase85	0.0027	0.0204 & 3 it	7.61	4.65×10^{-8}
Dcase141	0.0026	0.0206 & 3 it	7.76	2.36×10^{-10}

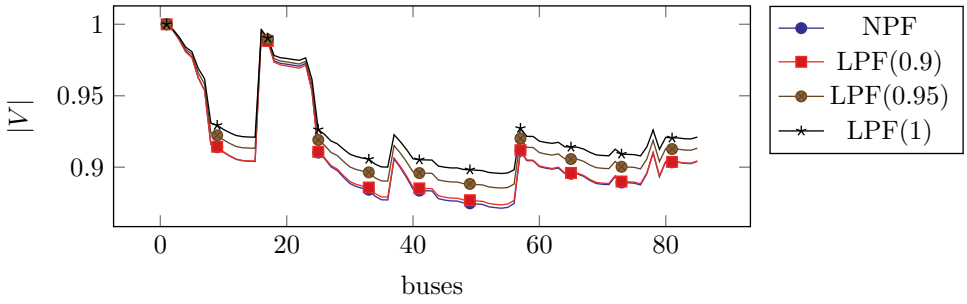
Table 5.2 proves that our LPF method becomes the exact linear reformulation of the original nonlinear power flow problem (4.5) when the exact voltage magnitude $|\hat{V}_n|$ is known for nonzero load buses and actual reactive power $\hat{Q}_k$ is specified for generator buses. Unfortunately, $|\hat{V}_n|$ of nonzero load buses and $\hat{Q}_k$ of generator buses modeled as PV buses are unknown until we do the power flow computation. However, it is possible to approximate $|\hat{V}_n|$ and $\hat{Q}_k$ precisely using the practical knowledge and mathematical model. Since the power flow computation is performed in p.u normalization, we have $0 < |\hat{V}_n| < 1$. A more suitable estimate can be made based on practical knowledge of the grid. Similarly, we can predetermine $\hat{Q}_k$ by modeling a generator bus as a load bus by using a power factor and the specified active power P_k of the generator buses.

Table 5.3 shows the relative difference between NPF and LPF computations when $|\hat{V}_n|$ is set to the same value for all nonzero load buses n, and $\hat{Q}_k$ is predetermined for all generator buses k by $\hat{Q}_k = Q_k^N - \epsilon$ where Q_k^N is the reactive power computed in the NPF computation, and ϵ is small constant. We can observe that the LPF solution is close enough to the NPF solution for both cases even though $|\hat{V}_n|$ is chosen to be same for all nonzero load buses n. Numerically, the difference is not small, but it is already sufficient for engineering application. For test case Tcase89, a flat start $|\hat{V}_n| = 1.0$ gives more accurate results whereas $|\hat{V}_n| = 0.9$ is the better choice for Dcase85. Moreover, this relative difference can be further improved by not choosing the same value for each $|\hat{V}_n|$.

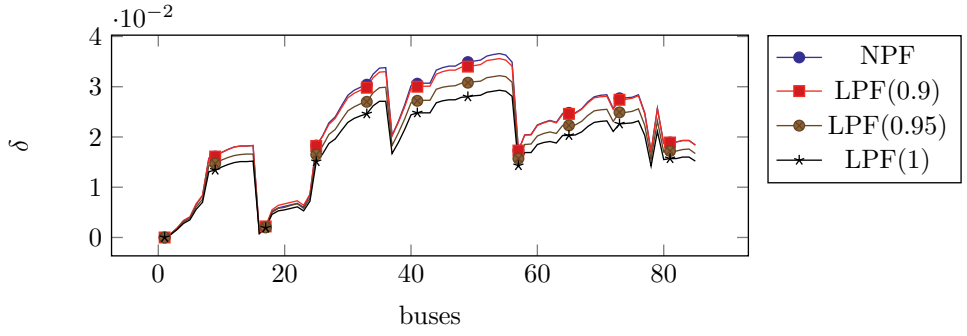
We display the voltage profile of test case Dcase85 in Figure 5.2 to compare the NPF results with the LPF results obtained by using different inputs for $|\hat{V}_n|$.

Table 5.3: *Relative difference between NPF and direct LPF ($|\hat{V}_n| = \{0.9; 0.95; 1.0\}$ and $\hat{Q}_k = Q_k^N - \epsilon$).*

Test cases	**Relative difference** $\frac{\|\mathbf{V^N} - \mathbf{V^L}\|_2}{\|\mathbf{V^N}\|_2}$		
	$\|\hat{V}_n\| = 0.9$	$\|\hat{V}_n\| = 0.95$	$\|\hat{V}_n\| = 1$
Tcase89	9.01×10^{-2}	4.04×10^{-2}	6.58×10^{-3}
Dcase85	1.36×10^{-3}	1.33×10^{-2}	2.02×10^{-2}



(a) *Computed voltage magnitudes $|V|$*



(b) *Computed voltage angles $|\delta|$*

Figure 5.2: *Voltage profile of the test case DCase85 for various $|\hat{V}_n|$.*

From the figure, it is more clear that this LPF method can be as accurate as NPF methods if the input $|\hat{V}_n|$ is chosen correctly. In addition, as we have seen in Table 5.2, our direct LPF approach is around seven times faster than the NPF computation. Our results show that this direct LPF method can be as accurate as classical NPF methods and additionally, it is much faster than NPF computations. Thus, this direct linear power flow approach can be a very powerful tool for electrical grid operators to control the very large networks in real-time.

In practice, a flat start 1.0 is used as an initial guess for Newton-based power flow methods. However, it is known that the Newton process has a local quadratic convergence meaning that if the initial iterate is far from the solution, then it diverges sometimes. Since our LPF method is much faster than NPF algorithms and provides acceptable voltage profile for a flat start $|\hat{V}_n| := 1.0$, we can perform the LPF com-

putation first with some inputs for $|\hat{V}_n|$ and use the result of it as an initial guess for NPF methods. Table 5.4 presents the result of NPF computations with the initial guess V^0 that is given as a flat start 1.0 or the result of the direct LPF computation with $|\hat{V}_n| = 0.95$. According to the table, the convergence of the NPF computation is improved by one to two iterations on all test cases when the result of the LPF computation is used as an initial guess. These improvements happen because the result of direct LPF computations is already very close to the solution of the original power flow problem. With this good initial guess, any NPF method converges very fast. This can be another application of our direct LPF method in power flow simulations.

Table 5.4: *Number of iteration for NPF computations using different initial guesses.*

Test cases	**Iterations**	
	NPF with $V^0 = 1.0$	NPF with $V^0 = \text{LPF}(\|\hat{V}_n\| = 0.95)$
Transmission networks		
Tcase9	3	2
Tcase30	3	2
Tcase57	3	2
Tcase89	4	2 ($|\hat{V}_n| = 0.99$)
Tcase118	3	2
Distribution networks		
Dcase22	2	1
Dcase33	3	2
Dcase69	4	2
Dcase85	3	2
Dcase141	3	1

5.4.2 Iterative approach

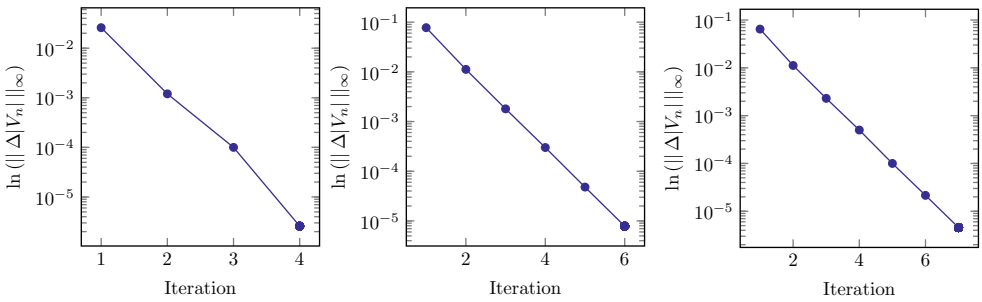
In this subsection, we compare the iterative LPF approach developed in section 5.3.2 to the NPF algorithm in terms of accuracy and speed. Algorithm 5 is used for the LPF computation. We study only distribution network cases in this subsection since the iteration process is still under research for generator buses modeled as PV buses. If generator buses are modeled as PQ buses, then algorithm 5 can be used to any transmission or distribution network.

In Table 5.5, we show the numerical results of NPF and iterative LPF computations for the CPU time and the relative difference. Both NPF and LPF algorithms start with a flat start $V^0 = 1.0$. As shown in the table, the LPF computation is still five to six times faster than the NPF computation even though the LPF method needs more iterations than the NPF algorithm. Additionally, the relative difference $\frac{||V^N - V^L||_2}{||V^N||_2}$ is very small for all test cases.

Table 5.5: *The CPU time and the relative difference between NPF and iterative LPF.*

Test cases	**NPF**($V^0 = 1.0$)		**LPF**($\|\hat{V}_n^0\| = 1$)		$\frac{\textbf{Time (NPF)}}{\textbf{Time (LPF)}}$	$\frac{\|\mathbf{V^N} - \mathbf{V^L}\|_2}{\|\mathbf{V^N}\|_2}$
	Iter	Time(s)	Iter	Time(s)		
Dcase22	2	0.0201	4	0.0030	6.72	2.27×10^{-7}
Dcase33	3	0.0194	6	0.0033	5.96	4.36×10^{-7}
Dcase69	4	0.0205	6	0.0036	5.76	5.76×10^{-7}
Dcase85	3	0.0218	7	0.0040	5.52	1.70×10^{-6}
Dcase141	3	0.0237	6	0.0043	5.50	1.34×10^{-7}

In Figure 5.3, the scaled residual norm $\ln(||\,\Delta|V_n|\,||_\infty)$ is shown for various test cases. Our iterative LPF method has a linear convergence as you can see in the figure.



(a) *Dcase22* (b) *Dcase69* (c) *Dcase85*

Figure 5.3: *Convergence of the iterative LPF method on various test cases.*

Moreover, we can conclude that this iterative LPF method has the same accuracy as NPF algorithms, and in addition, it is much faster than NPF computations. Therefore, this method can be used for power flow simulations on any transmission or distribution networks if generator buses are modeled as load buses.

5.5 Conclusion

In this chapter, we propose a linear formulation of the original nonlinear power flow problem. The theoretical background is introduced for the linearization of the nonlinear power flow problem. We develop the direct 5.3.1 and iterative 5.3.2 methods for the linear power flow problem (5.13). Accuracy and efficiency of both direct and iterative linear approaches are validated by comparing them with the conventional Newton power flow algorithm on various transmission and distribution networks. We compare the performance of our LPF methods with NPF methods in terms of CPU time and relative difference. We observe that our direct LPF approach is around seven times faster than the NPF computation. In addition, the direct LPF method can be as accurate as NPF methods if the input $|\hat{V}_n|$ is given with reasonable initial value. We have also demonstrated that the direct LPF method can be used to determine the initial guess of the NPF computation. Furthermore, the iterative LPF method has the same accuracy as NPF algorithms, and it is five to six times faster than NPF

computations. Overall, we can conclude that electrical grid operators can use both direct and iterative LPF approaches for real-time power system simulation of very large networks.

Chapter 6

Application of NA techniques on MV/LV network simulations

In this chapter, we propose a fast LPF algorithm improved with Numerical Analysis (NA) techniques to solve the Large LPF (LLPF) problem with 27 million buses simulating both the entire LV and MV networks in a single simulation. The direct LPF algorithm developed in section 5.3.1 is used as the main solution technique in this chapter. The entire LV and MV networks of Alliander DNO are used in our simulations. For the application of NA techniques, we consider the LLPF problem with first only real components and then complex components. In the numerical analysis, reordering technique (RCM) and a couple direct solvers (Cholesky, IC, LU, and ILU) and various Krylov subspace methods (CG, PCG, GMRES, and BiCGSTAB) are chosen and applied to the LLPF problem with both real and complex components.

6.1 Introduction

Due to the volatility of renewable energy sources (RES), Distribution Network Operators (DNOs) have a great need for faster power flow calculations for simulating different scenarios for network design. DNOs traditionally treat LV and MV networks as two separate entities where both voltage levels have their own set of assumptions

This chapter is based on:

B. Sereeter, W. van Westering, C. Vuik, and C. Witteveen, "Linear power flow method improved with numerical analysis techniques applied to a very large network," *Energies*, vol. 12, no. 21, p. 4078, 2019,

M. E. Kootte, B. Sereeter, J. E. Romate, and C. Vuik, "Comparison of numerical methods to solve the steady-state integrated transmission-distribution power flow problem," *IEEE PES Innovative Smart Grid Technologies Europe [Under review]*, Oct 2020.

and design policies. However, simulating both the LV and MV networks in a single power flow computation can result in more effective grid management and a new grid design paradigm [108]. On the other hand, this will increase the size of the power flow problem from a couple of thousand nodes to a couple of million nodes as well as the complexity of the power flow problem.

As we discussed in chapter 4, conventional power flow solution methods diverge sometimes [34] when they are applied to the distribution power flow problem due to some special features of the distribution network, such as radial or weakly meshed structure, high R/X ratio, line's length and unbalanced loads. Besides, these conventional methods are iterative type of methods which require lots of derivative calculations and matrix reconstructions in every iteration. Therefore, for very large MV/LV network simulation, the linear power flow model discussed in chapter 5 is more applicable than the nonlinear power flow models described in chapter 4.

It has been shown that iterative linear solvers can result in faster performances over sparse direct solvers for very large power flow problems [50, 109, 110]. In other words, the computational time of NPF computations can be improved by studying the properties of the linear system solved in every iteration and applying Numerical Analysis (NA) techniques such as different reordering schemes, various direct solvers and numerous Krylov subspace methods on them.

Even though we have seen that the LPF computation is much faster than the NPF computation as we have seen in Chapter 5, the LPF approach can still perform slower than we expect due to the large size of the grid if we do not optimize the algorithm numerically. In this chapter, we show how to improve the computational time of the LPF method by applying NA techniques. The direct LPF algorithm developed in section 5.3.1 is used as the main solution technique in this chapter.

6.2 Case study of large Dutch power grid (LLPF)

In order to demonstrate the impact of integrally simulating the MV/LV grid, a case study has been assembled. The case study focuses on voltage problems. Since the voltage end points are the same for both the MV/LV and LV simulations, the results are easy to compare. The goal of the model is to support large-scale investment policy decisions such as: 'how many transformers will be overloaded the next 30 years?' or 'In which area of the country should more engineers be recruited for cable replacement?'. The model was created for techniques such as time series analysis and agent based modeling which all require evaluating many different load configurations. Several real world MV/LV networks have been studied in literature, some of which have in the order of 100,000 buses [111, 112]. However, the networks from these studies are still several orders smaller than the network of this study which has over 24 million buses.

6.2.1 Data and assumptions

The network of Alliander DNO is used in our linear power flow computations, which contains both the LV and MV distribution networks and consists of approximately 80,000 km of cable serving over three million customers as shown in Figure 6.1. It covers over $1/3^{\text{rd}}$ of the total Dutch power grid. The MV network of Alliander DNO

consists of 100,000 cable segments whereas the LV network consists of over 24 million buses, three million customers (load nodes), several thousands of generators and around 250 substations. Data sets consist of all cable segments, connectivity, and impedance. The connectivity, voltage ratio and impedance of all transformers are used in the power flow computation. The network is mostly radial, but some LV networks are strongly meshed and can serve over 100,000 customers.

In line with Alliander DNO policy, a voltage problem is defined as a voltage drop of over 9% in the MV/LV network from the secondary side of the substation transformer to the customer, taking into account that both networks operate on a different voltage level. If only the LV network is considered, the allowed voltage drop is 4.5% from the secondary side of the distribution transformer to the customer. In the MV/LV simulation the nominal voltage (V_{ref}) is assumed to be 10.5 kV and in the LV simulation the nominal voltage (V_{ref}) is assumed to be 400 V.

Given that the linear model is only valid for relatively low voltage drops, it is important to note that voltage drops of 4.5% on the LV network and over 9% on the MV/LV network are always specified as a 'voltage problem' by the DNO. The exact height of the voltage problem is of less importance, as it needs to be solved anyway. The goal of the case study is therefore only to find the problems and not necessarily determining the problem severity.

The current presence of decentralized renewable energy generation is relatively low, around 4% of the total power generation. Generator buses are also modeled as load buses using the constant impedance model. While this is not accurate for the few buses controlling reactive power, it is in line with current Alliander DNO modeling practices.

The network is modeled as a single-phase balanced network, because no data is available of which customer is connected to which phase. While this is a best-case assumption, it is still a good starting point for finding voltage problems in the network and is only an issue for small LV networks with a little number of customers where the law of large numbers does not apply.

To run the linear power flow computation, all three million end users have been given a load of 1.1 kVA with a power factor of 0.95. The power consumption of 1.1 kW is the design peak power for regular households for LV grids containing over forty households.[†]

6.2.2 Solving in terms of only real numbers

It is possible to solve the linear power flow problem (5.16) in terms of only real numbers in order to ease the calculation or if your software does not support the combination of the complex variables and sparse matrices like the *R* programming language [108]. The following sections demonstrate the solution process of the linear power flow problem in terms of real numbers.

[†]While more detailed data is available within Alliander, it could not be used for publication purposes because of privacy issues. However, the 1.1 kVA assumption yields comparable results on locations with a sufficient number of customers.

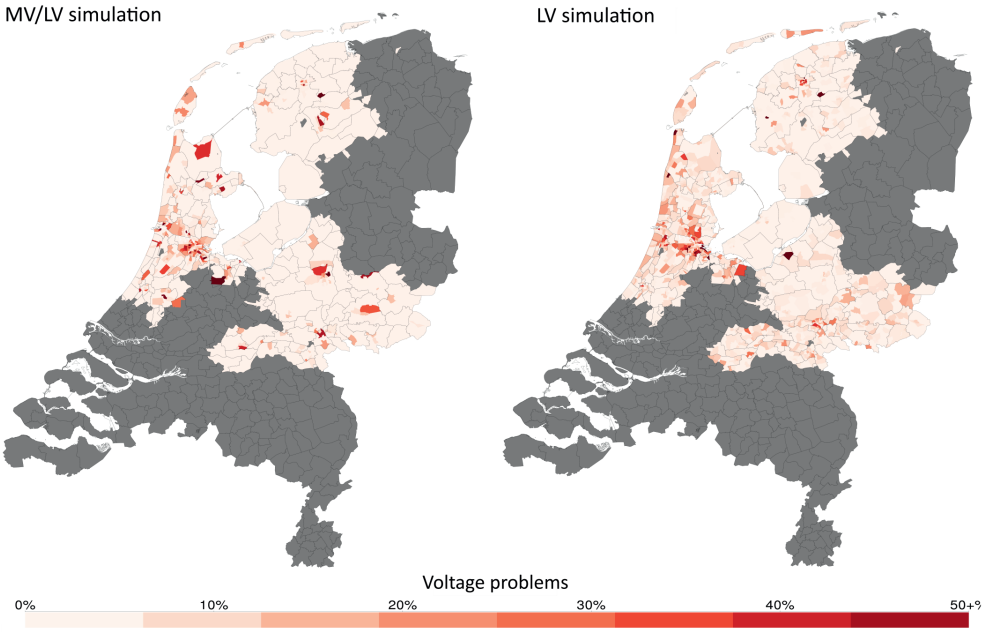


Figure 6.1: *The geographical distribution of simulated voltage problems in the Alliander service area on a postal code level. The area depicted is the entirety of The Netherlands of which Alliander services the non-grey area.*

Neglecting imaginary parts

In LV networks customers use or produce very little reactive power on average. Additionally, the reactive power and reactance are generally an order lower than the active power and resistance of the network respectively. If one simulates the LV grid only and starts from the secondary side of the distribution transformer, it is generally sufficient to only simulate the real part of the network [113]. Thus, if we neglect the impact of imaginary parts then we exclude the equivalent reactance X_{ng}^{eq} and the equivalent resistance R_{ng}^{eq} becomes as:

$$R_{ng}^{\text{eq}} = \frac{V_{\text{ref}}^2}{P_n}. \tag{6.1}$$

Furthermore, we know that $Y_{22} = G_{22} + \imath B_{22}$ and $b = b^r + \imath b^m$ in equation (5.16). By neglecting all imaginary parts from equation (5.16), we obtain the following linear equations as follows:

$$|V_2| = G_{22}^{-1} b^r, \tag{6.2}$$

where $|V_2|$ is voltage magnitudes, G_{22} is the conductance of admittance matrix Y_{22} and $b^r = -G_{21}|V_1|$. This makes the power flow computation roughly 50% faster and it might be worth the modeling error introduced by this assumption.

Reformulating equations with complex numbers

Matrix equation (5.16) is given as:

$$V_2^r + \imath V_2^m = (G_{22} + \imath B_{22})^{-1}(b^r + \imath b^m). \tag{6.3}$$

Equation (6.3) can be reformulated into into the following matrix equation:

$$\begin{bmatrix} V_2^r \\ V_2^m \end{bmatrix} = \begin{bmatrix} G_{22} & -B_{22} \\ B_{22} & G_{22} \end{bmatrix}^{-1} \begin{bmatrix} b^r \\ b^m \end{bmatrix}. \tag{6.4}$$

After the computation, original V_2 is calculated as $V_2 = V_2^r + \imath V_2^m$ using the computed real V_2^r and imaginary parts V_2^m in (6.4). In this case, we double the size of the equations but can avoid the complex numbering.

6.2.3 Simulation results

For this large power flow simulations on LV and MV/LV networks, we use the LPF approach described in Section 5.3. The LLPF problem with complex components (5.16) is solved in 58 seconds on a single processor core using the *R* programming language. If only active power is considered, the problem is solved in 29 seconds. All linear algebra is implemented using the Matrix package which is a C wrapper for the Blas and Lapack matrix computation libraries.

The resulting geographical distribution of voltage problems is displayed in Figure 6.1. Moreover, the absolute number of voltage problems in the MV/LV simulation is 150 thousand, 5% of the total number of customers. The absolute number of voltage problems in the LV simulation is 180 thousand, 6% of the total number of customers. While these percentages are low, they are still very significant as voltage problems can be quite costly to solve.

While the number of voltage problems are in the same order of magnitude between simulations, the locations of the problems are vastly different. The voltage problems overlapped only 20% between the two simulations as can also be observed in Figure 6.1. The lack of overlap has a severe implication, namely that searching for congestion by only simulating LV networks yields the wrong voltage problem locations. Therefore, it is clear that an identical load configuration will result in a very different layout of voltage problems if the MV/LV network is simulated integrally or only the LV network is taken into account.

It can also be observed from Figure 6.1 that the problems in the MV/LV simulation are more concentrated compared to the LV simulation due to the fact that a LV network with high loads influences neighboring networks via the MV network. This is very useful information for a DNO, since it also implies that multiple LV voltage problems can be solved by tactically strengthening the MV network.

A subject for future research is a comparison of the calculated problems in this case study with reported problems reported to the DNO. This is not trivial as a good comparison data set is not available. Voltage problems are an emerging issue and currently only very few voltage problems are actually detected by the DNO. This problem is also not easily solved using smart meter data. The smart meter only saves voltage and consumption data from the past 10 days, which is very little information to

obtain a good peak consumption pattern. Furthermore, privacy rules and bandwidth limitations also do not allow for constant customer voltage monitoring.

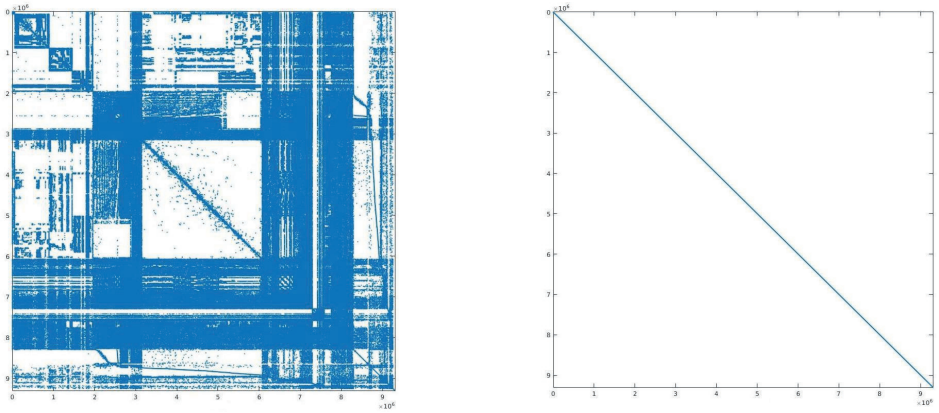
Between the two simulations paradigms (LV only or MV/LV integrally), the integral MV/LV network simulation gives a more accurate estimate of the voltage problem locations as differences in the MV network are taken into account. This conclusion calls for network design using integral MV/LV simulations; a new design paradigm for the DNOs.

6.3 Application of NA techniques

In this section, we apply the NA techniques on both LLPF problems with real (6.2) and complex components (5.16) in order to determine the iterative methods that are most suitable for our LLPF problems.

First, we consider the LLPF problem with real components (6.2) where the size of matrix G_{22} is $9,300,775 \times 9,300,775$ and the number of nonzeros is $27,867,547$. Due to the large dimension of the matrix, it is very costly to compute the inverse of the matrix G_{22}^{-1}. Therefore, we study the properties of the matrix G_{22} and seek the fastest way to solve equation (6.2).

By analyzing matrix G_{22}, we observe that G_{22} is a sparse and Symmetric and Positive Definite (SPD) matrix. Due to its SPD properties, we can use NA techniques that are developed for this type of matrices such as the Cholesky decomposition, Incomplete Cholesky (IC), and Conjugate Gradient (CG) iterative method on G_{22}. In addition, some reordering techniques such as Reverse Cuthill-McKee (RCM) and Approximate Minimum Degree (AMD) permutations could improve the properties of G_{22} as well. For large transmission networks, power flow solvers with AMD reordering performed the best [50]. In our study on large distribution networks, the RCM reordering algorithm results in better properties and bandwidths than AMD for the matrix G_{22}. Figure 6.2 shows the sparsity structure of G_{22} and reordered G_{22} using RCM. From the figure, it is clear that the sparseness properties of the matrix G_{22} are



(a) *Original* G_{22} (b) *Reordered* G_{22}

Figure 6.2: *Sparsity of matrix* G_{22} *and reordered* G_{22} *using RCM.*

improved by using RCM reordering.

For the direct solver, the Cholesky decomposition with RCM reordering could solve the linear system (6.2) fast. Since G_{22} is a SPD matrix, the best iterative method for matrix equation (6.2) is CG. Furthermore, the convergence rate of CG depends on eigenvalues λ_k of G_{22}. Table 6.1 shows the largest $\lambda_{\max}$ and smallest $\lambda_{\min}$ magnitude eigenvalues and the condition number $\kappa_2(\cdot) = \frac{\lambda_{\max}}{\lambda_{\min}}$ of G_{22} and preconditioned G_{22} as $M^{-1}G_{22}$. From the first row of Table 6.1, we see that the condition number of the

Table 6.1: *The largest and smallest magnitude eigenvalues and the condition numbers for matrix G_{22} and preconditioned G_{22}.*

Matrix	$\boldsymbol{\lambda_{\max}}$	$\boldsymbol{\lambda_{\min}}$	$\boldsymbol{\kappa_2(\cdot)}$
G_{22}	9.34×10^{8}	1.38×10^{-3}	6.73×10^{12}
$M^{-1}G_{22}$, $L = \text{Chol}$	1	1	1
$L = \text{IC}(10^{-5})$	1.70	0.01	123.43
$L = \text{IC}(10^{-6})$	1.28	0.27	4.72
$L = \text{IC}(10^{-7})$	1.15	0.83	1.38
$L = \text{IC}(10^{-8})$	1.002	0.994	1.008

matrix G_{22} is very large which means that G_{22} is ill-conditioned. Therefore, using CG without any preconditioner on the linear system (6.2) cannot improve the computational time since many iterations are required for CG. Thus, the Preconditioned Conjugate Gradient (PCG) method is a proper choice to use instead of CG. In PCG, we solve the transformed system as:

$$M^{-1}G_{22}|V_2| = M^{-1}b^r \tag{6.5}$$

where M is called a preconditioner and is a SPD matrix. The eigenvalues of $M^{-1}G_{22}$ should be clustered around one, resulting in a faster convergence for PCG. Generally, M is obtained as $M = LL'$ where L is a lower triangular matrix. We can compute L using Cholesky or Incomplete Cholesky decompositions on G_{22} or on reordered G_{22}. The eigenvalues of $M^{-1}G_{22}$ can be improved by choosing a right preconditioner M for G_{22}.

In the second row of Table 6.1, the Cholesky decomposition is used for L and results in eigenvalues equal to one for the preconditioned G_{22}. Therefore, PCG with the Choleksy decomposition is expected to converge after one iteration for equation (6.2). However, using the full Cholesky decomposition for L is computationally expensive and the solution time can be larger than using a direct method. In order to decrease the computation time of constructing the lower triangular matrix L, we can use the Incomplete Cholesky decomposition instead of the full Cholesky.

In rows 3 - 6 of Table 6.1, we see how the eigenvalues and condition number of $M^{-1}G_{22}$ are improved by changing the drop tolerance of IC. Moreover, we can conclude that preconditioner M using $\text{IC}(10^{-8})$ or IC with a drop tolerance smaller than 10^{-8} for L can be a good preconditioner for matrix G_{22} in terms of the computational time and number of iterations for PCG.

Let us consider the LLPF problem with complex components (5.16) and (6.4).

For simplicity, let us denote the matrix

$$\begin{bmatrix} G_{22} & -B_{22} \\ B_{22} & G_{22} \end{bmatrix}$$

in equation (6.4) by A. Matrices Y_{22} and A are not positive definite unlike G_{22}. Moreover, matrix Y_{22} is symmetric and matrix A is non-symmetric. Therefore, the Cholesky decomposition and CG are not suitable for these types of matrices. Instead, the LU decomposition, Generalized Minimal RESidual (GMRES) and Bi-Conjugate Gradient Stabilized (BiCGSTAB) methods are more convenient to use on matrices Y_{22} and A.

6.4 Numerical Results

In this section, we show how NA techniques can be used to improve the CPU time of the LLPF computation introduced in Section 6.2. For this purpose, all solution methods are re-implemented in Matlab. We consider both LLPF problems with complex numbers (5.16) and without imaginary parts (6.2). For the numerical experiments, all computations are done on Intel computer i5-6500 3.2 GHz CPU with four cores and 64 GB memory.

6.4.1 LLPF problem with real components

Table 6.2 shows the comparison between various linear solvers on equation (6.2) in terms of the CPU time, number of iterations and the number of non-zeros (NNZ). All results are averaged over 10 computations. For PCG, the maximum iteration and relative tolerance are set to 100 and 10^{-5} respectively.

Table 6.2: *Comparison between various NA techniques on the LLPF problem with real components* (6.2).

Algorithms	**Time** & **Iter**	$\frac{\|\|\boldsymbol{V_2^i}-\boldsymbol{V_2^d}\|\|_2}{\|\|\boldsymbol{V_2^d}\|\|_2}$	**NNZ**
$-G_{22}\backslash b^r$	14.32 sec	8.12×10^{-11}	$27,867,547$
$G_{22}\backslash b^r$	7.12 sec	0	
+ RCM	6.94 sec	6.69×10^{-12}	
Cholesky	152.2 sec	7.31×10^{-12}	$257,293,316$
+ RCM	5.01 sec	9.51×10^{-12}	$20,726,961$
PCG(IC(0)) + RCM	NA	NA	$18,584,161$
PCG(Cholesky) + RCM	6.24 sec & 1 it	9.51×10^{-12}	$20,726,961$
PCG(IC(10^{-5})) + RCM	6.65 sec & 4 it	0.007	$19,722,635$
PCG(IC(10^{-8})) + RCM	4.96 sec & 1 it	2.42×10^{-4}	$20,314,280$

The first and second rows of Table 6.2 are the results of direct solvers using Matlab's backslash \ operator (R2015a, MathWorks, Natick, MA, USA) without any additional techniques. It is necessary to mention that the CPU time of the first row doubles the CPU time of second row due to the positioning of the minus sign in equation (5.14). In addition, if we write the minus sign on the left side of the equation

(5.14), $-G_{22}$ is not a positive definite matrix which results in large computational time. Therefore, it is better to put the minus sign on the right side of equation (5.14) and to keep it inside the vector b.

For the direct solver, the Cholesky decomposition with RCM reordering results in the fastest computational time for matrix equation (6.2) as we can see from Table 6.2. Furthermore, as we expected, IC(10^{-8}) with RCM reordering is the best preconditioner for G_{22} that results in only one iteration in 4.96 seconds for PCG. However, when IC(10^{-8}) is used for the preconditioner, the relative difference between the direct and iterative solutions $\frac{||V_2^i - V_2^d||_2}{||V_2^d||_2}$ is high compared to other options. Therefore, we also solve the problem (6.2) with various tolerances for PCG and drop tolerances for IC. Numerical results are given in Table 6.3.

Table 6.3: *Numerical results of PCG with various tolerances for both IC and PCG.*

Time & **Iter** & $\frac{\|\boldsymbol{V_2^i - V_2^d}\|_2}{\|\boldsymbol{V_2^d}\|_2}$		**Relative tolerance for PCG**	
		10^{-7}	10^{-8}
Drop tolerance for IC	10^{-9}	4.96 sec & 1 it & 2.40×10^{-5}	4.96 sec & 1 it & 2.40×10^{-5}
	10^{-10}	4.96 sec & 1 it & 2.31×10^{-6}	4.96 sec & 1 it & 2.31×10^{-6}

From Table 6.3, we see that the relative difference $\frac{||V_2^i - V_2^d||_2}{||V_2^d||_2}$ can be improved by decreasing the drop tolerance (10^{-9}, 10^{-10}, $\cdots$) of IC for the preconditioner $M = LL'$ while keeping PCG still converge after 1 iteration. Additionally, applying IC gives us smaller NNZ compared to full Cholesky and direct solvers. Finally, the original computation time (14.32 sec) of LLPF problems with real components (6.2) is improved by 2.8 times (4.96 sec) using NA techniques.

6.4.2 LLPF problem with complex components

For iterative solvers GMRES and BiCGSTAB, the maximum iteration and relative tolerance are set to 20 and 10^{-6} respectively. Table 6.4 shows the comparison between various NA techniques on the LLPF problem (5.16) in terms of the CPU time, number of iterations and the relative difference between the direct and iterative solutions. In Alliander DNO, equation (6.4) is used to solve the LLPF problem because the R programming language does not support complex numbers. Furthermore, from the first and second rows of Table 6.4, we can see that using equation (5.16) to solve the LLPF problem with complex components is almost 2.5 times faster than using equation (6.4) when Matlab's backslash \ operator is used without any additional techniques. Therefore, we use equation (5.16) for further experiments.

The same RCM reordering is applied to matrix Y_{22} in order to improve the structure of the matrix. The best computational time (7.41 sec) is achieved by the direct solver LU decomposition on the reordered Y_{22} using RCM as can be seen from Table 6.4. For the iterative methods, the best computation time with the smallest relative difference is obtained by BiCGSTAB with ILU(10^{-14}) as a preconditioner and RCM reordering. However, the best CPU time of the iterative method is still larger

Table 6.4: *Comparison between numerous NA techniques on the LLPF problem with complex components* (5.16).

Algorithms	**Time** & **Iter**	$\frac{\|\|V_2^i - V_2^d\|\|_2}{\|\|V_2^d\|\|_2}$	**NNZ**
Eq. (6.4)	42.6 sec	0	$111,470,118$
Eq. (5.16): $Y_{22}\backslash b$	17.23 sec	3.03×10^{-11}	$27,867,547$
+ RCM	15.58 sec	1.90×10^{-11}	
LU + RCM	7.41 sec	5.84×10^{-11}	$32,284,123$
GMRES(ilu(0)) + RCM	177.86 sec & 20 it	0.3427	$27,867,547$
BiCGSTAB(ilu(0)) + RCM	56.21 sec & 20 it	0.2503	
GMRES(ilu(10^{-8})) + RCM	18.75 sec & 2 it	7.23×10^{-08}	$31,629,906$
GMRES(ilu(10^{-11})) + RCM	13.78 sec & 1 it	9.82×10^{-08}	$32,031,268$
GMRES(ilu(10^{-14})) + RCM	14.27 sec & 1 it	9.60×10^{-11}	$32,244,575$
BiCGSTAB(ilu(10^{-10})) + RCM	10.57 sec & 0.5 it	1.12×10^{-06}	$31,920,611$
BiCGSTAB(ilu(10^{-12})) + RCM	10.77 sec & 0.5 it	8.73×10^{-09}	$32,119,629$
BiCGSTAB(ilu(10^{-14})) + RCM	10.92 sec & 0.5 it	9.61×10^{-11}	$32,244,575$

than the best CPU time of the direct solver due to the fact that ILU, GMRES, and BiCGSTAB are not implemented in the optimal way in Matlab. Furthermore, both LU and ILU decompositions provide relatively similar NNZ for the LLPF problem with complex components.

As a result of the application of NA techniques, the original computation time (42.6 sec) of LLPF problems with complex components (5.16) is improved by 5.7 times (7.41 sec).

6.5 Conclusion

We propose a fast LPF method improved with NA techniques to solve very large power flow problems simulating both the entire LV and MV networks in a single simulation. The entire LV and MV networks of Alliander DNO are used in our linear power flow computations. In our research, it is shown that voltage problems can be identified more efficiently when MV and LV networks are integrally evaluated. Moreover, NA techniques are applied to the LLPF problem in order to improve the computation time by studying the properties of the linear system. In the numerical analysis, reordering technique (RCM), a couple of direct solvers (Cholesky, IC, LU, and ILU), and various Krylov subspace methods (CG, PCG, GMRES, and BiCGSTAB) are chosen and applied to the LLPF problem with both real and complex components. Finally, the original computation times of LLPF problems with real and complex components are reduced by 2.8 and 5.7 times respectively as a result of the application of NA methods.

The algorithms in this chapters are being applied within Alliander DNO. These applications include: large scale strategic modeling, automatic network design and automatic outage-recovery plans.

Chapter 7

Optimal Power Flow Computations

In this chapter, we study four equivalent mathematical formulations of the Optimal Power Flow (OPF) problem and their impacts on the performance of solution methods. We show how four mathematical formulations of the OPF problem can be obtained by rewriting equality constraints given as the power flow problem into four equivalent mathematical equations using power balance or current balance equations in polar or Cartesian coordinates while keeping the same physical formulation. In order to identify the formulation that results in the best convergence characteristics for the solution method, we apply MIPS (Matpower's Interior Point Method), KNITRO (Commercial software package for solving large scale nonlinear optimization problems), and FMINCON (Matlab's optimization solver) on various test cases using three different initial conditions. We compare all four formulations in terms of impact factors on the solution method such a number of nonzero elements in the Jacobian and Hessian matrices, the number of iterations and computational time on each iteration.

7.1 Introduction

The Optimal Power Flow (OPF) problem provides the optimal operational state of the electrical power system while satisfying system constraints and control limits. Many sub-classes of the OPF problem have been developed over the years using various objective functions, control variables and system constraints such as economic dispatch, security constrained OPF (SCOPF), unit commitment, loss minimization and

This chapter is based on:

B. Sereeter, C. Vuik, C. Witteveen, and P. Palensky, "Optimal power flow formulations and their impacts on the performance of solution methods," IEEE Power & Energy Society General Meeting, IEEE, Aug 2019.

probabilistic OPF (POPF) [114–116]. These OPF problems are physical formulations that are derived from the physical properties of actual power systems.

In the literature, many OPF solution techniques can be found and they can be divided into two main categories [116–119]:

- Deterministic (Classic)
  Gradient Methods [120, 121], Newton's Methods [122, 123], Simplex Methods [124], Sequential Linear Programming (SLP) [125, 126], Sequential Quadratic Programming (SQP) [127, 128], Interior Point Methods (IPM) [129–131],

- Non-Deterministic (Heuristic)
  Ant Colony Optimization (ACO) [132, 133], Artificial Neural Networks (ANN) [134, 135], Evolutionary Programming [136, 137], Particle Swarm Optimization (PSO) [138], Simulated Annealing (SA) techniques [139].

Despite the fact that deterministic methods are proven to be reliable for many types of OPF problems, these methods cannot provide the global optimality in general and cannot handle the discrete variables [114]. On the contrary, even though non-deterministic methods have some advantages of handling non-convexities and discrete variables, these methods are computationally expensive for large problems [140]. Therefore, there is no method that is the best for all OPF problems. Normally, the solution method is chosen depending on the type of the OPF problem.

In general, the original Nonlinear Power Flow (NPF) problem is used as the main equality constraints for the OPF problem. Moreover, the NPF problem is given in complex numbers and can be rewritten into four equivalent mathematical equations given in **real** numbers and variables using power balance or current balance equations in polar or Cartesian coordinates [8, 9], as we have seen in chapter 4. Therefore, we obtain four mathematical formulations of the OPF problem for a single physical formulation. These four formulations are equivalent since we just rewrite the mathematical equations for the equality constraints while keeping the same physical formulation. Due to the different mathematical equations, however, each formulation can result in different numerical and analytical properties for the OPF solution method.

In practice, researchers develop a new method or do the simulation based on only one (at most two) mathematical formulation of the OPF problem and compare the result with another method using the other formulation. The formulation having power balance equations in polar coordinates (known as Polar power-voltage) is mostly used in the literature. It is questionable how an OPF solution method performs if we change the chosen formulation to the other three mathematical formulations. When the OPF solver using one formulation does not converge, can the same method using another formulation converge? Which mathematical formulation results in the smallest computational time for each iteration of the solution method? Which formulation is more robust to the change of initial conditions? As far as we know no complete comparison exists between these four mathematical formulations of the OPF problem.

In [141, 142], **three** formulations (Polar Power-Voltage (PSV), Rectangular Power-Voltage (RSV) and Rectangular Current-Voltage (RIV)) are used to compare optimization software packages such as SNOPT, IPOPT, and KNITRO. Both papers suggest numerous strategies for choosing the initial condition. Both PSV and RIV formulations show the best performance in terms of CPU time in [142] whereas the

formulation using rectangular coordinates is preferred in [141]. Furthermore, formulations PSV and RSV in [141, 142] have the same nonlinear power balance equations in different coordinates used as equality constraints for the OPF problem. However, the RIV formulation used in both papers has the linear current balance equations where the injected complex current at buses is specified and not computed from specified complex power as given in [40, 93]. Thus, the RIV formulation is not equivalent to PSV and RSV formulations. Additionally, the formulation Polar Current-Voltage (PIV) is not considered in both papers. Therefore, the comparison in [141, 142] is not complete due to missing and inequivalent formulations.

In this thesis, we study all four equivalent mathematical formulations of the OPF problem and try to understand which formulation results in the best performance for OPF solution methods. We consider the OPF problem with minimization of active power generation costs as a cost function, power flow equations as equality constraints and squared apparent power limits as inequality constraints.

7.2 Optimal Power Flow problem

The general OPF problem can be written as follows:

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) \\ &\text{subject to} && g(\mathbf{x}) = 0, \\ &&& h(\mathbf{x}) \leq 0 \end{aligned} \tag{7.1}$$

where $\mathbf{x}$ is the optimization vector with the state and control variables, and $f(\mathbf{x})$ is the objective function to be minimized (maximized). The vector functions $g(\mathbf{x})$ and $h(\mathbf{x})$ represent equality and inequality constraints respectively.

7.2.1 Variables

In general, state variables include bus voltage magnitude $|V_i|$, bus voltage angle δ_i, branch power flow S_{ij}^L, generator active P_i^g and reactive Q_i^g power outputs, the real V_i^r and imaginary V_i^m parts of the complex voltage respectively. Control variables are generally chosen as active power generations, voltage magnitudes at generator buses, transformer tap settings, transformer phase shifters, generator voltage control settings, load shedding, shunt reactive devices, HVDC stations and Static Var Controllers [114]. Furthermore, we take two different optimization vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ depending on the chosen coordinates as:

- Polar coordinates:

$$\mathbf{x}_1 = \left[\Theta, \mathcal{V}, P^{\mathbf{g}}, Q^{\mathbf{g}}\right]^T \tag{7.2}$$

- Cartesian coordinates:

$$\mathbf{x}_2 = \left[U, W, P^{\mathbf{g}}, Q^{\mathbf{g}}\right]^T \tag{7.3}$$

where $\mathcal{V}$ and Θ are the vectors of voltage magnitudes $|V|$ and angles δ whereas U and W are the vectors of real V^r and imaginary V^m parts of the complex voltages respectively. Here, $P^{\mathbf{g}}$ and $Q^{\mathbf{g}}$ are denoted by the vectors of generator active and reactive power injections respectively.

7.2.2 Objective function

The most well-known objective functions are the economic dispatch (minimizing generation costs or system transmission losses, maximizing market surplus), environmental dispatch (minimizing emission) and maximizing power quality or power transfer capability [143]. In this thesis, we consider the objective function $f(\mathbf{x})$ as:

$$f(\mathbf{x}) = \sum_{i=1}^{N_g} \left(C_i^0 + C_i^1 P_i^g + C_i^2 (P_i^g)^2 \right) \tag{7.4}$$

where N_g is a number of generators in the network and C_i^0, C_i^1, C_i^2 are the positive coefficients of the polynomial cost functions. Moreover, the objective is to minimize the total cost for the active power generation in the system.

7.2.3 Equality constraints

Usually, the power flow equations (4.5) are used as equality constraints $g(\mathbf{x})$:

$$S_i = V_i \sum_{k=1}^{N_b} Y_{ik}^* V_k^* . \tag{7.5}$$

Moreover, the power flow problem (7.5) can be rewritten into four equivalent mathematical equations given in real numbers and variables using the power balance or current balance equations in polar or Cartesian coordinates as given in equations (4.10)-(4.11) and (4.14)-(4.15).

7.2.4 Inequality constraints

The inequality constraints are specified using the maximum and minimum limits for transmission lines, control, and state variables.

Branch flow limits

We consider inequality constraints $h(\mathbf{x})$ as squared branch flow limits for the apparent power:

$$h_{ij}(\mathbf{x}) = \begin{bmatrix} |S_{ij}^f(\mathbf{x})|^2 \\ |S_{ij}^t(\mathbf{x})|^2 \end{bmatrix} \leq \begin{bmatrix} (S_{ij}^{\max})^2 \\ (S_{ij}^{\max})^2 \end{bmatrix} \tag{7.6}$$

where $S_{ij}^f(\mathbf{x})$ and $S_{ij}^t(\mathbf{x})$ are the apparent power of branch flow *from* side and *to* side respectively, $S_{ij}^{\max}$ is the maximum branch flow limits between bus i and j. We denote a number of transmission lines in the network by N_l.

Variable limits

The following variable limits are considered in this thesis:

$$|V_i|^{\min} \leq |V_i| \leq |V_i|^{\max}, \tag{7.7}$$

$$(P_i^g)^{\min} \leq P_i^g \leq (P_i^g)^{\max}, \tag{7.8}$$

$$(Q_i^g)^{\min} \leq Q_i^g \leq (Q_i^g)^{\max}, \tag{7.9}$$

$$|V_i|^{\min} \leq \sqrt{(V_i^r)^2 + (V_i^m)^2} \leq |V_i|^{\max}. \tag{7.10}$$

7.2.5 Four equivalent formulations of the OPF problem

Combining (7.4) and (7.6) with one of (4.10)-(4.11) or (4.14)-(4.15) depending on the choice of the formulation and coordinates, we can obtain four equivalent mathematical formulations for a single physical formulation of the OPF problem (7.1). Table 7.1 shows the summary of all four formulations for the number of variables, equality, and inequality constraints.

Table 7.1: *Summary of all four formulations of OPF problem.*

	OPF formulations			
	PP	CP	PC	CC
Coordinates	Polar		Cartesian	
Variables	$\lvert V\rvert, \delta, P^g, Q^g$ $2N_b + 2N_g$		V^r, V^m, P^g, Q^g $2N_b + 2N_g$	
Nonlinear equality constraints	Power balance in Polar (4.10) $2N_b$	Current balance in Polar (4.14) $2N_b$	Power balance in Cartesian (4.11) $2N_b$	Current balance in Cartesian (4.15) $2N_b$
Nonlinear inequality constraints	Branch apparent power flow (7.6) $2N_l$			
			Variable limits (7.10) N_b	

7.3 Interior Point Method (MIPS)

We choose the Matpower's Interior Point Method (MIPS) using the Primal-Dual Interior Point algorithm described in Section 2.3.1 as the main solution method in this thesis. MIPS computes the linearized KKT conditions (2.29) iteratively until the stopping criteria is satisfied and the optimal solution of the OPF problem (7.1) is obtained. Furthermore, depending on the formulation of the OPF problem (PP, PC, CP, and CC), we obtain four different linear systems of equations (2.29) having different properties and values for the same physical formulation. For each variant, derivatives constructing the KKT conditions such as $g_{\mathbf{x}}$, $g_{\mathbf{xx}}$, $h_{\mathbf{x}}$, and $h_{\mathbf{xx}}$ require different mathematical equations and numerical calculations for the computation. Therefore,

we can expect different convergence characteristics of MIPS for each mathematical formulation of the OPF problem.

Originally, Matpower uses the PP formulation with (7.2) for the OPF computation and their derivatives for MIPS are given in Matpower's Technical Note 2 [144] using the complex matrix notation. We implement the other three formulations (PC, CP and CC) for MIPS in Matpower. Their derivatives are also computed using complex matrix notation and they are given below.

7.3.1 Derivatives of objective function $f(\mathbf{x})$

The objective function (7.4) is rewritten in the complex matrix notation as:

$$f(\mathbf{x}) = C^0\mathbf{e} + C^1 P^{\mathbf{g}} + C^2 (P^{\mathbf{g}})^2 \tag{7.11}$$

with C^0, C^1, $C^2 \in \mathbb{R}^{1\times N_g}$ are the vectors of positive coefficients given in (7.4) whereas $\mathbf{e} \in \mathbb{R}^{N_g\times 1}$ is the vector of all ones and $P^{\mathbf{g}} \in \mathbb{R}^{N_g\times 1}$ is the vector of generator active injections.

First order derivatives with respect to $\mathbf{x}_1$ and $\mathbf{x}_2$:

The first order derivatives of the objective function $f_{\mathbf{x}_1}$ and $f_{\mathbf{x}_2}$ are equal since the objective function only depends on P^g as:

$$\begin{aligned} f_{\mathbf{x}_1} = \frac{\partial f}{\partial \mathbf{x}_1} &= [f_\Theta \quad f_\mathcal{V} \quad f_{P^{\mathbf{g}}} \quad f_{Q^{\mathbf{g}}}] \\ &= [\mathbf{0} \quad \mathbf{0} \quad 2C^2[P^{\mathbf{g}}] + C^1 \quad \mathbf{0}] \\ &= [f_U \quad f_W \quad f_{P^{\mathbf{g}}} \quad f_{Q^{\mathbf{g}}}] = \frac{\partial f}{\partial \mathbf{x}_2} = f_{\mathbf{x}_2}, \end{aligned}$$

where $[P^{\mathbf{g}}]$ is denoted by a diagonal matrix with vector $P^{\mathbf{g}}$ on the main diagonal.

7.3.2 Derivatives of equality constraints $g(\mathbf{x})$

The power-mismatch (4.7) and current-mismatch (4.12) functions used as equality constraints are reformulated using the complex matrix notation as:

$$\Delta S(\mathbf{x}) = [V]Y^*V^* - S^{sp}, \tag{7.12}$$

$$\Delta I(\mathbf{x}) = YV - [S^{sp}]^*\Lambda^* \tag{7.13}$$

with $\Lambda = V^{-1}$ and S^{sp} is the vector the specified complex powers. For each formulation of the equality constraints, we obtain the following relations.

First order derivatives of the PP formulation:

$$g_{\mathbf{x}_1} = \begin{bmatrix} \mathrm{Re}\{\Delta S_{\mathbf{x}_1}\} \\ \mathrm{Im}\{\Delta S_{\mathbf{x}_1}\} \end{bmatrix}, \tag{7.14}$$

where

$$\begin{aligned}\Delta S_{\mathbf{x}_1} = \frac{\partial \Delta S}{\partial \mathbf{x}_1} &= [\Delta S_\Theta \quad \Delta S_\mathcal{V} \quad \Delta S_{P\mathbf{g}} \quad \Delta S_{Q\mathbf{g}}] \\ \Delta S_\Theta &= \imath[V]\Big([I^*] - Y^*[V^*]\Big) \\ \Delta S_\mathcal{V} &= [V]\Big([I^*] + Y^*[V^*]\Big)[\mathcal{V}]^{-1} \\ \Delta S_{P\mathbf{g}} &= -C_g \\ \Delta S_{Q\mathbf{g}} &= -\imath C_g.\end{aligned}$$

Here, $I = YV$ is the vector of complex currents and $C_g \in \mathbb{R}^{N_b \times N_g}$ is a generator connection matrix.

First order derivatives of the PC formulation:

$$g_{\mathbf{x}_2} = \begin{bmatrix}\mathrm{Re}\{\Delta S_{\mathbf{x}_2}\} \\ \mathrm{Im}\{\Delta S_{\mathbf{x}_2}\}\end{bmatrix}, \tag{7.15}$$

where

$$\begin{aligned}\Delta S_{\mathbf{x}_2} = \frac{\partial \Delta S}{\partial \mathbf{x}_2} &= [\Delta S_U \quad \Delta S_W \quad \Delta S_{P\mathbf{g}} \quad \Delta S_{Q\mathbf{g}}] \\ \Delta S_U &= [I^*] + [V]Y^* \\ \Delta S_W &= \imath\Big([I^*] - [V]Y^*\Big) \\ \Delta S_{P\mathbf{g}} &= -C_g \\ \Delta S_{Q\mathbf{g}} &= -\imath C_g.\end{aligned}$$

First order derivatives of the CP formulation:

$$g_{\mathbf{x}_1} = \begin{bmatrix}\mathrm{Re}\{\Delta I_{\mathbf{x}_1}\} \\ \mathrm{Im}\{\Delta I_{\mathbf{x}_1}\}\end{bmatrix}, \tag{7.16}$$

where

$$\begin{aligned}\Delta I_{\mathbf{x}_1} = \frac{\partial \Delta I}{\partial \mathbf{x}_1} &= [\Delta I_\Theta \quad \Delta I_\mathcal{V} \quad \Delta I_{P\mathbf{g}} \quad \Delta I_{Q\mathbf{g}}] \\ \Delta I_\Theta &= \imath\Big(Y[V] + [S^{sp}]^*[\Lambda^*]\Big) \\ \Delta I_\mathcal{V} &= Y[E] - [S^{sp}]^*[\mathcal{V}]^{-1}[\Lambda^*] \\ \Delta I_{P\mathbf{g}} &= -[\Lambda^*]C_g \\ \Delta I_{Q\mathbf{g}} &= \imath[\Lambda^*]C_g\end{aligned}$$

with $[E] = [V][\mathcal{V}]^{-1}$.

First order derivatives of the CC formulation:

$$g_{\mathbf{x}_2} = \begin{bmatrix}\mathrm{Re}\{\Delta I_{\mathbf{x}_2}\} \\ \mathrm{Im}\{\Delta I_{\mathbf{x}_2}\}\end{bmatrix}, \tag{7.17}$$

where

$$\begin{aligned}\Delta I_{\mathbf{x}_2} = \frac{\partial \Delta I}{\partial \mathbf{x}_2} &= [\Delta I_U \quad \Delta I_W \quad \Delta I_{P\mathbf{g}} \quad \Delta I_{Q\mathbf{g}}] \\ \Delta I_U &= Y - [S^{sp}]^*[\Lambda^*]^2 \\ \Delta I_W &= \imath\Big(Y + [S^{sp}]^*[\Lambda^*]^2\Big) \\ \Delta I_{P\mathbf{g}} &= -[\Lambda^*]C_g \\ \Delta I_{Q\mathbf{g}} &= \imath[\Lambda^*]C_g.\end{aligned}$$

7.3.3 Derivatives of inequality constraints $h(\mathbf{x})$

The inequality constraints (7.6) are rewritten using the complex matrix notation as:

$$h(\mathbf{x}) = \begin{bmatrix} h^f(\mathbf{x}) \\ h^t(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} [S^{f^*}]S^f - S^2_{\max} \\ [S^{t^*}]S^t - S^2_{\max} \end{bmatrix}, \tag{7.18}$$

where S^f and S^t are the vectors of the apparent power of branch flow *from* side and *to* side respectively, $S_{\max}$ is the vector of maximum branch flow limits. Due to the squared branch flow limits, the first order derivatives of inequality constraints are computed as:

$$\begin{aligned}h^f_{\mathbf{x}} &= [S^{f^*}]S^f_{\mathbf{x}} + [S^f]S^{f^*}_{\mathbf{x}} \\ &= 2\Big(\operatorname{Re}\{[S^f]\}\operatorname{Re}\{S^f_{\mathbf{x}}\} + \operatorname{Im}\{[S^f]\}\operatorname{Im}\{S^f_{\mathbf{x}}\}\Big).\end{aligned}$$

First order derivatives with respect to $\mathbf{x}_1$:

$$\begin{aligned}S^f_{\mathbf{x}_1} = \frac{\partial S^f}{\partial \mathbf{x}_1} &= [S^f_\Theta \quad S^f_\mathcal{V} \quad S^f_{P\mathbf{g}} \quad S^f_{Q\mathbf{g}}] \\ S^f_\Theta &= \imath\Big(\big[I^{f^*}\big]C_f[V] - [C_f V]Y^*_f[V^*]\Big) \\ S^f_\mathcal{V} &= \big[I^{f^*}\big]C_f[E] + [C_f V]Y^*_f[E^*] \\ S^f_{P\mathbf{g}} &= \mathbf{0} \\ S^f_{Q\mathbf{g}} &= \mathbf{0},\end{aligned}$$

where $C_f \in \mathbb{R}^{N_l \times N_b}$ is a branch connection matrix and N_l is a number of branches in the network.

First order derivatives with respect to $\mathbf{x}_2$:

$$
\begin{aligned}
S^f_{\mathbf{x}_2} = \frac{\partial S^f}{\partial \mathbf{x}_2} &= [S^f_U \quad S^f_W \quad S^f_{P\mathbf{g}} \quad S^f_{Q\mathbf{g}}] \\
S^f_U &= \Big[I^{f^*}\Big] C_f + [C_f V] Y^*_f \\
S^f_W &= \imath\Big(\big[I^{f^*}\big] C_f - [C_f V] Y^*_f\Big) \\
S^f_{P\mathbf{g}} &= \mathbf{0} \\
S^f_{Q\mathbf{g}} &= \mathbf{0}.
\end{aligned}
$$

The derivatives $h^t_{\mathbf{x}}$ are identical to $h^f_{\mathbf{x}}$ (i.e. replace all super-scripts f with t). For all second order derivatives $f_{\mathbf{xx}}$, $g_{\mathbf{xx}}$ and $h_{\mathbf{xx}}$, we refer to Appendix B.

From equations (7.14)-(7.17), we deserve that the first order derivatives of equality constraints $g_{\mathbf{x}}$ are computed using different mathematical equations for each formulation. In general, computing the derivatives of equality and inequality constraints in Cartesian coordinates is analytically easier than in polar coordinates. Therefore, less computational time can be expected from MIPS using PC and CC formulations due to less calculations for all derivatives at each iteration. However, when Cartesian coordinates are used, voltage magnitude limits (7.10) become nonlinear and we have to linearize it during the computation which requires extra work.

7.4 Numerical Results

In this section, we present the results of numerical experiments of all four mathematical formulations in order to verify the formulation resulting in the best performance for the OPF solution method. We implement all four mathematical formulations in Matpower and apply three optimization software packages such as MIPS, KNITRO, and FMINCON. In the numerical experiments, we use test cases from Matpower and IEEE PES Power Grid Library (PGLib) that are given in Table 7.2. The following impact factors on the solution method are considered for the comparison:

- number of nonzero elements (NNZ) in the Jacobian and Hessian matrices
- number of iterations for the solution method
- computational time for each iteration of the solution method.

Both feasibility and optimality tolerances are set to 10^{-6} and the number of iterations is limited by 450. The constant power load model is considered for all loads. The performance of the non-convex optimization problems such as OPF problems strongly depends on the choice of starting points. Therefore, we use three different initial conditions for all solution methods as given in Table 7.3. All experiments are performed on an Intel computer i5-4690 3.5 GHz CPU with four cores and 64 Gb memory, running a Debian 64-bit Linux 8.7 distribution.

Table 7.2: *Description of considered test cases.*

Systems	Buses	Generators	Branches	Abbr
Matpower-case89	89	12	210	c89
PGLib-case118	118	54	186	c118
Matpower-case300	300	69	411	c300
PGLib-case588	588	167	686	c588
PGLib-case2383	2383	327	2896	c2383
Matpower-case2736	2736	420	3504	c2736
Matpower-case3120	3120	505	3693	c3120

Table 7.3: *Three options for the initial condition.*

Options	Descriptions
IC-1	Interior point estimation (midpoint of their bounds)
IC-2	Use the current state in given test case
IC-3	Solve the power flow problem and use the resulting state

7.4.1 Number of nonzero elements

Table 7.4 shows the number of nonzero elements in the Jacobian and Hessian matrices that are recomputed at each iteration of MIPS. The best result is highlighted in bold. For the Jacobian matrix, there is no big difference between all four mathematical formulations. However, both formulations using the current balance equations (CP and CC) result in less nonzero entries for the Hessian matrix compared to PP and PC formulations. Especially, the CC formulation gives the smallest number of nonzero elements for the Hessian matrix on all test cases. Therefore, the CC formulation is the best choice for computing the Jacobian and Hessian matrices with respect to memory requirements. The IPM algorithm assembles the object function, equal-

Table 7.4: *Number of nonzero elements in the Jacobian and Hessian matrices after one iteration of MIPS.*

NNZ		Test cases					
		c118	c300	c588	c2383	c2736	c3120
Jacobian	PP	2048	**4611**	**7897**	33320	**37808**	**42677**
	PC	**2046**	4612	7959	33406	37826	42681
	CP	2152	4749	8143	34058	38365	43271
	CC	2118	4492	7947	**33212**	38316	43223
Hessian	PP	1904	4472	7750	32584	37044	41936
	PC	1670	3874	6594	27856	31578	35714
	CP	894	1687	2922	11596	12435	14063
	CC	**864**	**1492**	**2352**	**9940**	**10428**	**11660**

ity, and inequality constraints into the reduced and linearized Karush-Kuhn-Tucker (KKT) conditions and solves it at each iteration of the solution process. For each variant, derivatives of equality and inequality constraints constructing KKT conditions

require different mathematical equations and numerical calculations for the computation. Thus, we obtain four reduced and linearized KKT conditions having different properties for each mathematical formulation. Therefore, we can expect the different convergence characteristics for the solution method. Table 7.5 shows the condition number of the reduced and linearized KKT conditions for the test case c3120. We cannot prioritize the formulation over others as all formulations result in very high condition numbers due to the ill-conditioned nature of the problem.

Table 7.5: *Condition number of the reduced and linearized KKT conditions after one iteration of MIPS on test case c3120.*

ICs	**Condition numbers**			
	PP	PC	CP	CC
IC-1	$8.95 * 10^{12}$	$5.01 * 10^{13}$	$9.81 * 10^{13}$	$9.68 * 10^{13}$
IC-2	$1.57 * 10^{13}$	$1.21 * 10^{14}$	$1.92 * 10^{14}$	$1.99 * 10^{14}$
IC-3	$1.43 * 10^{13}$	$1.39 * 10^{14}$	$1.43 * 10^{13}$	$1.49 * 10^{14}$

7.4.2 Number of iterations

MIPS

In Table 7.6, we provide the number of iterations of MIPS using three different starting points on various test cases. From the table, we see that PP and CP formulations result in a faster convergence for MIPS compared to PC and CC formulations for most of the test cases. Between PP and CP formulations, MIPS using the CP formulation is slightly better. Regarding the initial conditions, IC-1 shows the robust performance for MIPS on all test cases. Both initial conditions IC-2 and IC-3 bring a Non-Convergence (NC) for two test cases (c89 and c2383). MIPS using the PP formulation diverge for both IC-2 and IC-3 on these two cases whereas CC and CP formulations deliver just one NC on those test cases. The PC formulation is the slowest variant but results in the robust convergence properties for MIPS on all scenarios. However, MIPS with the PC formulation is the slowest variant in terms of iterations. When a variant of MIPS using polar coordinates cannot converge to the optimal solution for some problems, another variant using Cartesian coordinates can be a good replacement.

KNITRO

Table 7.7 show the number of iterations of KNITRO using three different starting points. According to the table, KNITRO with the PP formulation is the fastest variant overall in terms of iterations. However, as we have seen in the previous section, the PP formulation also provides the bad performance for KNITRO using IC-2 on test cases c89 and c2636. Moreover, the other three variants of KNITRO perform better than KNITRO using PP on those test cases. Regarding the initial conditions, all four variants of KNITRO converge to the optimal solution for all three initial conditions. Moreover, KNITRO using IC-1 converges faster than KNITRO using IC-2 and IC-3 in terms of iterations.

Table 7.6: *Number of iterations for MIPS using three initial conditions on various test cases.*

ICs		**Iterations**						
		c89	c118	c300	c588	c2383	c2736	c3120
IC-1	PP	25	20	19	41	**33**	**29**	**43**
	PC	**18**	21	34	37	37	35	45
	CP	19	**19**	**18**	**35**	**33**	**29**	**43**
	CC	19	20	23	37	35	34	47
IC-2	PP	NC	20	**18**	41	**33**	28	108
	PC	**26**	21	31	37	37	34	54
	CP	30	**19**	**18**	**35**	**33**	**27**	**45**
	CC	NC	20	22	37	35	35	50
IC-3	PP	**14**	**22**	**16**	59	NC	27	**33**
	PC	15	24	38	**38**	43	32	36
	CP	**14**	**22**	17	68	NC	**26**	**33**
	CC	15	25	34	39	**42**	32	36

Table 7.7: *Number of iterations for KNITRO using three initial conditions on various test cases.*

ICs		**Iterations**						
		c89	c118	c300	c588	c2383	c2736	c3120
IC-1	PP	14	**11**	**10**	**21**	33	**20**	**27**
	PC	15	12	11	**21**	34	22	29
	CP	14	16	15	23	**32**	23	28
	CC	**13**	15	16	**21**	33	23	30
IC-2	PP	36	**11**	**11**	**21**	33	431	**28**
	PC	18	12	**11**	**21**	34	**21**	29
	CP	**15**	16	16	23	**32**	25	30
	CC	**15**	15	20	**21**	33	22	30
IC-3	PP	12	**15**	**13**	**25**	38	**20**	24
	PC	**11**	16	14	30	**32**	21	28
	CP	12	**15**	16	26	38	21	**23**
	CC	**11**	**15**	18	99	34	21	28

FMINCON

Matlab's optimization solver FMINCON has various choices for the solution algorithm. In this work, we use the algorithm-4 that applies Interior point with user-supplied Hessian. In Table 7.8, we display the number of iterations of FMINCON using three different starting points on various test cases. All four variants of FMINCON performs differently depending on the choice of the initial condition and the test case. Overall there is no formulation that is better than others. The PP formulation shows a bad performance for FMINCON on many test cases. Furthermore, PC and CC formulations which are the worst choice for MIPS and KNITRO, show the best

Table 7.8: *Number of iterations for FMINCON using three different initial conditions on various test cases.*

ICs		Iterations						
		c89	c118	c300	c588	c2383	c2736	c3120
IC-1	PP	36	**20**	**18**	63	105	46	**90**
	PC	34	24	20	**55**	106	**45**	100
	CP	**23**	31	29	91	96	50	104
	CC	28	27	20	70	**82**	57	114
IC-2	PP	121	**20**	20	63	105	NC	216
	PC	NC	24	**18**	**55**	106	**45**	**72**
	CP	61	31	37	91	96	156	NC
	CC	**54**	27	38	70	**82**	51	110
IC-3	PP	**15**	**24**	**19**	**69**	343	45	56
	PC	**15**	25	25	142	132	47	57
	CP	20	27	28	88	116	**43**	**47**
	CC	25	27	25	157	**109**	46	68

performance for FMINCON on many test cases.

7.4.3 CPU time on each iteration

In Figure 7.1, the computational time on each iteration $\left(\frac{\text{CPU time}}{\text{Number of iterations}}\right)$ is plotted for the comparison of all four formulations. From the figure, we discover that CP formulation shows the smallest computational time on each iteration for all three solvers. Additionally, all three solvers (MIPS, KNITRO and FMINCON) converge to the same objective value for all three initial conditions and four mathematical formulations on each test cases.

7.5 Conclusion

In this chapter, we study four equivalent mathematical formulations (PP, PC, CP, and CC) of the OPF problem and their computational impacts on the performance of the OPF solution methods. In order to identify the mathematical formulation resulting in the best computational properties for the OPF solution method, the numerical experiments are carried out using MIPS, KNITRO and FMINCON on various test cases of Matpower and IEEE PES Power Grid Library. All four mathematical formulations are compared in terms of the impact factors on the solution method such as the number nonzero elements in the Jacobian and Hessian matrices, number of iterations and computational time on each iteration.

For MIPS, the CP formulation shows the fastest convergence and the smallest number of nonzero elements in the Jacobian and Hessian matrices whereas the PP formulation delivered the best computational properties for KNITRO in terms of iterations. All four variants of FMINCON perform differently depending on the choice of the initial condition and the given test case. Overall there is no formulation that is better than others for FMINCON. However, PC and CC formulations which are the

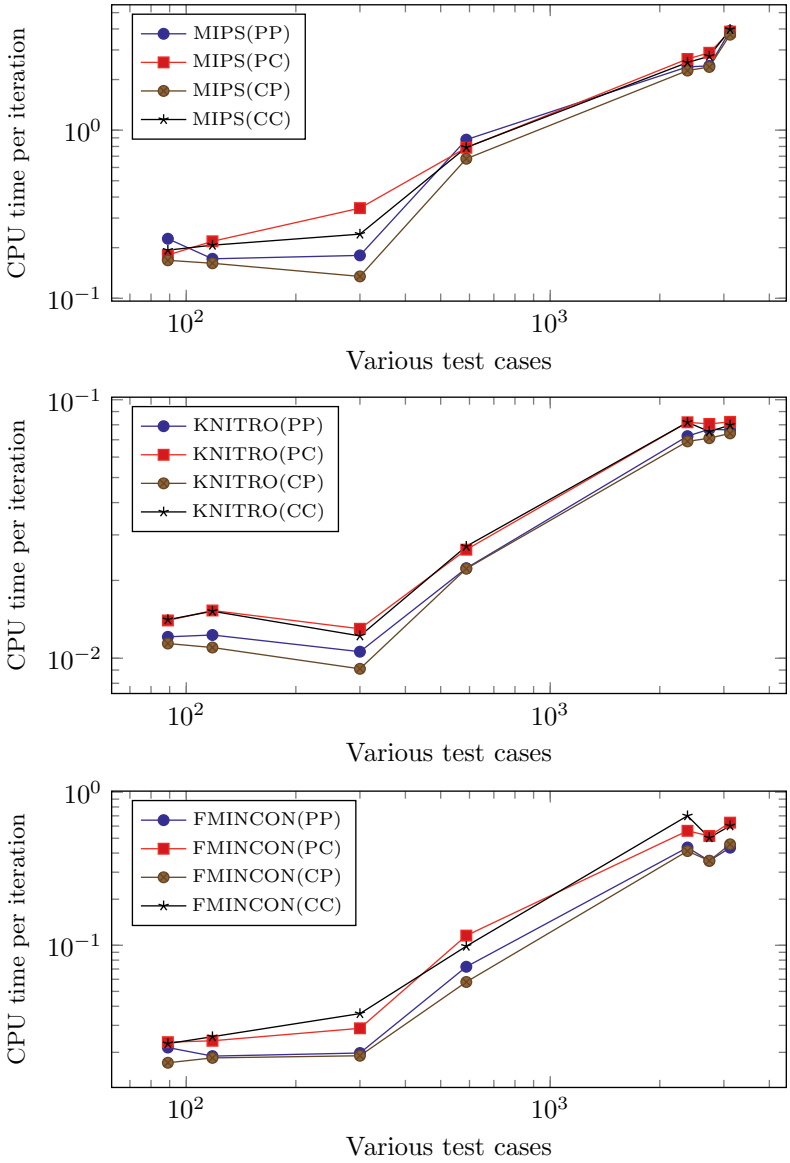


Figure 7.1: *Computational time spent on each iteration of all three solvers for IC-1 on various test cases.*

worst choice for MIPS and KNITRO, show the best performance for FMINCON on many test cases. In terms of computational time on each iteration, the CP formulation is the best choice for all three methods.

The numerical results show that the performance of the OPF solution method is not only dependent upon the choice of the solution method itself, but also upon the exact mathematical formulation used to specify the OPF problem. When the OPF

solution method using a certain formulation does not converge, one can obtain the optimal solution by just applying the other equivalent formulation while keeping the same algorithm. Three mathematical formulations (PC, CP, and CC) of the OPF problem are implemented in Matpower and included in the current release (*Matpower 7.0*). Additionally, two technical notes [13, 14] are written for Matpower to specify the first and second order derivatives of the equality and inequality constraints.

Chapter 8

Conclusions and Recommendations

8.1 Conclusions

Due to the modernization of the existing electrical grid, a large number of newly developed grid elements including smart meters, smart appliances, renewable energy resources, and storage devices are being integrated into the power system. Thus, the existing electrical grid is changing rapidly and becoming more and more complex to control for grid operators. Meantime, current simple mathematical models and computational methods are considered insufficient in a changing and complex electrical network.

In this thesis we develop new advanced mathematical formulations and algorithms for fast and robust power system simulations that are required for the changing electrical power system. Both balanced single-phase transmission and unbalanced three-phase distribution networks are considered for the development of advanced Power Flow (PF) and Optimal Power Flow (OPF) solution methods. These methods can be used for real-time monitoring and control of the smart grid infrastructure.

In order to perform power system simulations on any electrical power grid, we must have an adequate mathematical model that transforms the physical properties of the network into mathematical equations. In Chapter 3, we show how electrical power systems and its components such as transmission lines, loads, generators, and transformers are modeled in this work.

In Chapter 4, we formulate and analyze the Newton based power flow methods that are used for the power flow computation on distribution and transmission networks. For the various methods, we consider two different mismatch formulations: the current and power balance equations and three different coordinate systems: Cartesian, Polar, and Complex form. This leads to six different versions of the Newton power flow method. Studying these versions in a common framework enables us to analyze and compare all variants in a unified way. The existing variants of the Newton power flow method developed in [1, 40, 92] are implemented and compared with the new versions of the Newton power flow method developed in this thesis. In case of the polar and Cartesian current-mismatch versions, the reactive power Q is chosen as a dependent variable for each PV bus. Thus, we compute the correction ΔQ at each iteration

and update Q using the computed corrections. Furthermore, all Newton power flow variants are extended to three-phase power flow problems. The backward–forward sweep-based algorithm (BFS [65]) is implemented for comparisons on distribution networks. Various mathematical models of the load, three-phase load connection, and three-phase transformer connection are studied and applied in the numerical experiments. As a result of the numerical experiment, NR-c-pol and the NR-c-car that are developed in this thesis perform the best for both balanced and unbalanced networks. We also investigate which version can be applied to what kind of a power network by comparing all versions for distribution networks with different loading conditions, R/X ratios, and load models. We observe that NR-c-pol and NR-c-car versions are more stable to the change of loading conditions and R/X ratios for both balanced and unbalanced networks, whereas the performance of other methods is highly sensitive to them. Therefore, we conclude that these two versions are the fastest and the most robust versions of all Newton power flow variants. In addition, these two variants can be applied to single or three-phase power flow problems in any balanced or unbalanced networks. All newly developed versions of the Newton power flow method have been implemented and included in the current version of Matpower (*Matpower 7.0*) which is a Matlab package for solving power flow and optimal power flow problems.

In Chapter 5, we propose a linear formulation of the original nonlinear power flow problem. The theoretical background is introduced for the linearization of the nonlinear power flow problem. We develop the direct and iterative solution methods for the linear power flow problem. Accuracy and efficiency of both direct and iterative linear approaches are validated by comparing it with the conventional Newton power flow algorithm on various transmission and distribution networks. We compare the performance of our LPF methods with NPF methods developed in Chapter 4 in terms of CPU time and relative difference. We observe that our direct LPF approach is around seven times faster than the NPF computation. In addition, the direct LPF method can be as accurate as NPF methods if the input $|\hat{V}_n|$ is given with reasonable initial value. We have also demonstrated that the direct LPF method can be used to determine the initial condition of the NPF computation. Furthermore, the iterative LPF method has the same accuracy as NPF algorithms, and it is five to six times faster than NPF computations. Overall, we can conclude that electrical grid operators can use both direct and iterative LPF approaches for real-time power system simulation of very large networks.

Chapter 6 proposes a fast LPF method improved with NA techniques to solve very large power flow problems simulating both the entire LV and MV networks in a single simulation. The direct LPF algorithm developed in section 5.3.1 is used as the main solution technique in this chapter. The entire LV and MV networks of Alliander DNO are used in our linear power flow computations. In our research, it is shown that voltage problems can be identified more efficiently when MV and LV networks are integrally evaluated. Moreover, NA techniques are applied to the LLPF problem in order to improve the computation time by studying the properties of the linear system. In the numerical analysis, reordering technique (RCM), numerous direct solvers (Cholesky, IC, LU, and ILU), and various Krylov subspace methods (CG, PCG, GMRES, and BiCGSTAB) are chosen and applied to the LLPF problem with both real and complex components. Finally, the original computation times of

LLPF problems with real and complex components are reduced by 2.8 times and 5.7 times respectively as a result of the application of NA techniques. Hence, we confirm that our LPF algorithm improved with NA techniques is very fast and user friendly for power flow computations on a large distribution network. The algorithms in this chapters are being applied within *Alliander DNO*. These applications include: large scale strategic modeling, automatic network design and automatic outage-recovery plans.

In Chapter 7, we study four equivalent mathematical formulations (PP, PC, CP, and CC) of the OPF problem and their computational impacts on the performance of the OPF solution methods. In order to identify the mathematical formulation resulting in the best computational properties for the OPF solution method, the numerical experiments are carried out using MIPS, KNITRO and FMINCON on various test cases of Matpower and IEEE PES Power Grid Library. All four mathematical formulations are compared in terms of the impact factors on the solution method such as the number nonzero elements in the Jacobian and Hessian matrices, number of iterations and computational time on each iteration. For MIPS, the CP formulation shows the fastest convergence and the smallest number of nonzero elements in the Jacobian and Hessian matrices whereas the PP formulation delivered the best computational properties for KNITRO in terms of iterations. All four variants of FMINCON perform differently depending on the choice of the initial condition and the given test case. Overall there is no formulation that is better than others for FMINCON. However, PC and CC formulations which are the worst choice for MIPS and KNITRO, show the best performance for FMINCON on many test cases. In terms of computational time on each iteration, the CP formulation is the best choice for all three methods. The numerical results show that the performance of the OPF solution method is not only dependent upon the choice of the solution method itself, but also upon the exact mathematical formulation used to specify the OPF problem. When the OPF solution method using a certain formulation does not converge, one can obtain the optimal solution by just applying the other equivalent formulation while keeping the same algorithm. Three mathematical formulations (PC, CP, and CC) of the OPF problem are implemented in Matpower and included in the current release (*Matpower 7.0*). Additionally, two technical notes [13, 14] are written for Matpower to specify the first and second order derivatives of the equality and inequality constraints.

Finally, our Newton power flow variants (NR-c-pol and the NR-c-car) are proven to be faster and more robust than the conventional Newton power flow methods. Numerical results support that these variants can be applied for the power flow computation on any complex network such as unbalanced distribution networks with new grid elements. We show that our LPF algorithms are very fast and user friendly for power flow computations on a large distribution network. Thus, the LPF algorithms are being applied within *Alliander DNO*. This thesis confirms that the performance of any OPF solution method can be improved by changing the mathematical formulation used to specify the OPF problem while keeping the same algorithm. Mathematical formulations and computational methods based on this thesis are implemented in *Matpower 7.0* for future research and practical use.

8.2 Recommendations

We address the steady-state analysis of the power system using new mathematical formulations and algorithms developed in this thesis for chosen PF and OPF problems, and solution methods. It is possible to apply these mathematical formulations and algorithms to the dynamic analysis of the power system or different PF solution techniques or various OPF problems using different solution techniques. We present below some recommendations for possible research directions.

Mathematical formulations of the Power Flow problem

In Chapter 4, we study six mathematical formulations of the power flow problem. A general framework is given for applying the Newton-Raphson method to solve nonlinear power flow problems, using power and current-mismatch functions in polar, Cartesian coordinates and complex form. We develop a new approach to handle generator (PV) buses for each variant that results in better converge for the Newton-Rapson method applied to different mathematical formulations of the power flow problem. But there are other conventional power flow solution methods such as Gauss–Seidel (GS) and Fast Decoupled Load Flow (FDLF) [30, 31] using only the power mismatch formulation in polar coordinates. Thus, a similar framework can be obtained for GS and FDLF methods using all six mathematical formulations together with equations (4.2)-(4.3).

Robustness of the Newton Power Flow variants using Cartesian coordinates

For the Newton power flow variants using Cartesian coordinates, we can use equations (4.2)-(4.3) with $\Delta|V_j| = 0$ for each PV bus j to make the Jacobian matrix square. This gives us the relation between the corrections ΔV_j^r and ΔV_j^m as:

$$\Delta V_j^r = -\frac{V_j^m}{V_j^r}\Delta V_j^m, \tag{8.1}$$

$$\Delta V_j^m = -\frac{V_j^r}{V_j^m}\Delta V_j^r. \tag{8.2}$$

In this thesis, only ΔV_j^r is chosen to be eliminated in every Newton iteration using equation (8.1) for all PV buses as it already results in strong robustness for the Newton Power Flow variants using Cartesian coordinates. However, the robustness of these variants can be further improved by using both (8.1) and (8.2) alternately. In order to do that, one has to implement the code that eliminates ΔV_j^r using (8.1) or ΔV_r^m using (8.2) in every Newton iteration depending on which one of V^r and V^m is close to zero.

Iterative LPF method handling PV buses

In Chapter 5, the iterative LPF method was developed for the LPF problem (5.13) assuming all generator buses are modeled as PQ buses (negative constant power loads). We extended the method to handle the generator buses as PV buses. In our

LPF approach, we update the unknown reactive power $\hat{Q}_k^{h+1}$ as $\hat{Q}_k^{h+1} := \hat{Q}_k^h + \Delta P$ for all generator buses k using the active power mismatch ΔP. However, this process was not the most optimal way to update $\hat{Q}_k^h$ as we see in our numerical experiments presented in Appendix A. Thus, it remains an open question to develop the optimal way to update the reactive power $\hat{Q}_k^{h+1}$ in every iteration of Algorithm 6.

OPF problem with equality constraints as LPF equations

We consider the OPF problem with minimization of active power generation costs as a cost function, nonlinear power flow equations as equality constraints and squared apparent power limits as inequality constraints in Chapter 7. We study all equivalent mathematical formulations of the OPF problem in order to identify the formulation that results in the best convergence characteristics for the solution method. If the computational time is more important than the accuracy, then linear power flow equations developed in Chapter 5 can be used as equality constraints instead of nonlinear power flow equations for the OPF problem. In addition, all four equivalent formulations could be applied to different OPF problems using other deterministic or heuristic optimization methods that are not considered in this thesis.

Application of Numerical Analysis techniques on Newton power flow variants

In Chapter 6, we apply the numerical analysis (NA) techniques to improve the computational time of the large linear power flow (LLPF) problem with 27 million nonzeros simulating both the entire LV and MV networks. The original computation times of LLPF problems with real and complex components are reduced by 2.8 and 5.7 times respectively as a result of the application of NA methods. Also, similar NA techniques are used in [29] to develop a Newton-Krylov that is much faster than traditional solvers for large nonlinear power flow problems determined by using a power mismatch function in polar coordinates. The same numerical analysis application is required to speed up the computational time of new variants of the Newton power flow method developed in Chapter 4.

Dynamic analysis of the power system

In practice, the power mismatch formulation is used for the steady-state and dynamic analysis of the power system. In this thesis, we have seen that the current mismatch formulation results in faster and more robust performances for steady-state analysis than the power mismatch formulation overall. Thus, an application of the current mismatch formulation in the dynamic analysis of the power system brings an open question subject to future research.

The variants of the Newton power flow method developed in Chapter 4 and equivalent mathematical formulations of the OPF problem developed in Chapter 7 are included in *Matpower 7.0*. The functions based on Chapter 4 and Chapter 7 can be found at the following Matpower's github account:

URL	`https://github.com/MATPOWER/matpower/blob/master/lib/`
Chapter 4	`newtonpf_I_cart.m` `newtonpf_I_polar.m` `newtonpf_S_cart.m` `newtonpf_I_hybrid.m` `newtonpf_S_hybrid.m`
Chapter 7	`dImis_dV.m` `d2Imis_dV2.m` `d2Imis_dVdSg.m` `dSbr_dV.m` `d2Sbr_dV2.m` `dSbus_dV.m` `opf_branch_ang_fcn.m` `opf_branch_ang_hess.m` `opf_current_balance_fcn.m` `opf_current_balance_hess.m` `opf_power_balance_fcn.m` `opf_power_balance_hess.m` `opf_veq_fcn.m` `opf_veq_hess.m` `opf_vlim_fcn.m` `opf_vlim_hess.m` `opf_vref_fcn.m` `opf_vref_hess.m`

Appendix $\mathcal{A}$

Numerical results for the LPF method handling PV buses

In Chapter 5, the iterative LPF method was developed for the LPF problem (5.13) assuming all generator buses are modeled as PQ buses (negative constant power loads). This appendix extends the algorithm to handle the generator buses as PV buses. The iteration process of the iterative LPF method handling PV buses is given in Algorithm 6.

Algorithm 6 LPF method for both PQ and PV buses

Set iteration counter to zero $h := 0$
Give initial $|\hat{V}_n^0|$ for all nonzero load buses n with $S > 0$ (between 0.5 and 1)
Give initial $\hat{Q}_k^0$ for all generator buses k
Compute initial R_{ng}^0 and X_{ng}^0 using equations (5.6)-(5.7)
Compute initial R_{kg}^0 and X_{kg}^0 using equations (5.8)-(5.9)
Compute Y including additional branches
Segment Y into Y_{11}, Y_{21} and Y_{22}, and compute b from equation (5.14)
while not converged
    Solve equation (5.16) for $\hat{V}_2^h$
PQ buses:
    Replace iterate $|\hat{V}_n^{h+1}| := |\hat{V}_2^h(S > 0)|$
    Compute R_{ng}^{h+1} and X_{ng}^{h+1} using equations (5.6)-(5.7) with $|\hat{V}_n^{h+1}|$
    Update elements of Y_{22} w.r.t R_{ng}^{h+1} and X_{ng}^{h+1}
PV buses:
    Compute complex power S^h using computed V^h as $S^h = V^h(YV^h)^*$
    Compute ΔP for all generator buses k as $\Delta P = P_k + \mathrm{Re}\{S_k^h\}$
    Update iterate $\hat{Q}_k^{h+1} := \hat{Q}_k^h + \Delta P$
    Compute R_{kg}^{h+1} and X_{kg}^{h+1} using equations (5.8)-(5.9) with $\hat{Q}_k^{h+1}$
    Update elements of Y_{22} w.r.t R_{kg}^{h+1} and X_{kg}^{h+1}
    $h := h + 1$
end while

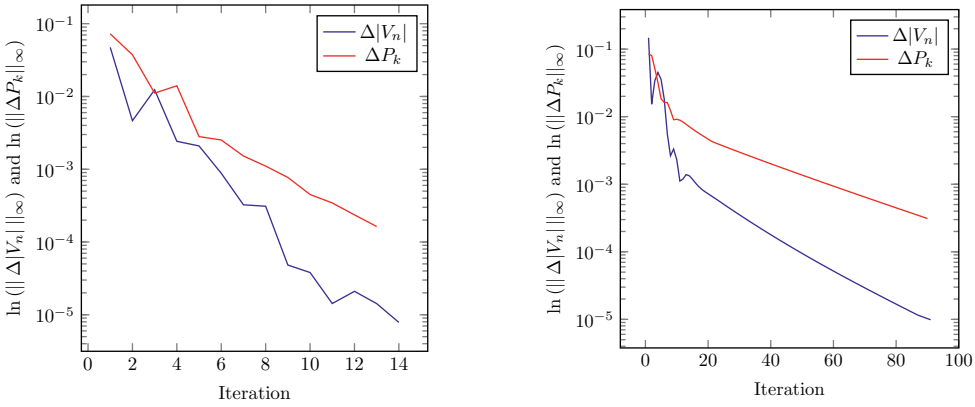
In this approach, we start with an initial reactive power $\hat{Q}_k^0$ for all generator buses k and compute R_{kg}^{h+1} and X_{kg}^{h+1} using equations (5.8)-(5.9) with $\hat{Q}_k^{h+1}$ in every iteration. The most challenging part is to properly update $\hat{Q}_k^{h+1}$ using other computed parameters such as $\hat{V}_2^h$. In our approach, we update $\hat{Q}_k^{h+1}$ as $\hat{Q}_k^{h+1} := \hat{Q}_k^h + \Delta P$ using the active power mismatch ΔP that is computed as $\Delta P = P_k + \text{Re}\{S_k^h\}$. However, this was not the most optimal way to update $\hat{Q}_k^h$ as we see in our numerical experiments below.

Algorithm 6 and the Newton power flow method developed in Chapter 4 are used for the power flow computations. Table A.1 shows the numerical results of both NPF and LPF computations for the CPU time and the relative difference $\frac{||V^N - V^L||_2}{||V^N||_2}$. In this experiments, we use two transmission networks with a couple of generators. From the table, we can see that Algorithm 6 finds the solution that is very accurate to the solution of the NPF computation. However, the number of iteration is very large.

Table A.1: *The CPU time and the relative difference between NPF and iterative LPF.*

Test cases	**NPF(** $V^0 = 1.0$**)**		**LPF(**$\|\hat{V}_n^0\| = 1$, $\hat{Q}_k^0 = C$**)**		$\frac{\textbf{Time (NPF)}}{\textbf{Time (LPF)}}$	$\frac{\|\|V^N - V^L\|\|_2}{\|\|V^N\|\|_2}$
	Iter	Time(s)	Iter	Time(s)		
Tcase9	3	0.0226	14	0.0053	4.22	3.18×10^{-5}
Tcase30	3	0.0209	91	0.0216	0.96	1.73×10^{-4}

In Figure A.1, we show the scaled residual norms $\ln(||\Delta|V_n|\,||_\infty)$ and $\ln(||\Delta P_k||_\infty)$ for two test cases. According to the graph, we conclude that Algorithm 6 has unstable convergence and additionally, $\hat{Q}_k^{h+1} := \hat{Q}_k^h + \Delta P$ is not the right way to update the reactive power $\hat{Q}_k^{h+1}$ for generator buses.



(a) *Tcase9* (b) *Tcase30*

Figure A.1: *Convergence of the iterative LPF method on two test cases.*

Appendix $\mathcal{B}$

Derivatives $f_{\mathbf{xx}}$, $g_{\mathbf{xx}}$ and $h_{\mathbf{xx}}$

The second order derivatives of the objective function $f(x)$, equality $g(x)$ and inequality $h(x)$ constraints are given this appendix. For intermediate calculations of all derivatives, we refer to [13, 14, 144].

B.1 Derivatives of objective function $f(\mathbf{x})$

Second order derivatives:

$$\begin{aligned} f_{\mathbf{x}_1\mathbf{x}_1} &= \frac{\partial}{\partial \mathbf{x}_1}(f_{\mathbf{x}_1}^T) \\ &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & f_{P\mathbf{g}P\mathbf{g}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \\ &= \frac{\partial}{\partial \mathbf{x}_2}(f_{\mathbf{x}_2}^T) = f_{\mathbf{x}_2\mathbf{x}_2} \end{aligned}$$

with

$$f_{P\mathbf{g}P\mathbf{g}} = 2[C^2].$$

B.2 Derivatives of equality constraints $g(\mathbf{x})$

Second order derivatives of the PP formulation:

$$\begin{aligned} g_{\mathbf{x}_1\mathbf{x}_1}(\lambda) &= \frac{\partial}{\partial \mathbf{x}_1}(g_{\mathbf{x}_1}^T\lambda) \\ &= \begin{bmatrix} g_{\Theta\Theta}(\lambda) & g_{\Theta\mathcal{V}}(\lambda) & \mathbf{0} & \mathbf{0} \\ g_{\mathcal{V}\Theta}(\lambda) & g_{\mathcal{V}\mathcal{V}}(\lambda) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \end{aligned}$$

with

$$
\begin{aligned}
g_{\Theta\Theta}(\lambda) &= \frac{\partial}{\partial\Theta}(g_\Theta^T\lambda) \\
&= [V^*]\Big(Y^{*^T}[V][\lambda] - \big[Y^{*^T}[V]\lambda\big]\Big) + [\lambda][V](Y^*[V^*] - [I^*]) \\
g_{\mathcal{V}\Theta}(\lambda) &= \frac{\partial}{\partial\Theta}(g_\mathcal{V}^T\lambda) \\
&= \imath[\mathcal{V}]^{-1}\left([V^*]\Big(Y^{*^T}[V][\lambda] - \big[Y^{*^T}[V]\lambda\big]\Big) - [\lambda][V]\Big(Y^*[V^*] - [I^*]\Big)\right) \\
g_{\Theta\mathcal{V}}(\lambda) &= \frac{\partial}{\partial\mathcal{V}}(g_\Theta^T\lambda) \\
&= \imath\left(\Big([\lambda][V]Y^* - \big[Y^{*^T}[V]\lambda\big]\Big)[V^*] - \Big([V^*]Y^{*^T} - [I^*]\Big)[\lambda][V]\right)[\mathcal{V}]^{-1} = g_{\mathcal{V}\Theta}^T(\lambda) \\
g_{\mathcal{V}\mathcal{V}}(\lambda) &= \frac{\partial}{\partial\mathcal{V}}(g_\mathcal{V}^T\lambda) \\
&= [\mathcal{V}]^{-1}\Big([\lambda][V]Y^*[V^*] + [V^*]Y^{*^T}[V][\lambda]\Big)[\mathcal{V}]^{-1}.
\end{aligned}
$$

Second order derivatives of the PC formulation:

$$
\begin{aligned}
g_{\mathbf{x}_2\mathbf{x}_2}(\lambda) &= \frac{\partial}{\partial\mathbf{x}_2}(g_{\mathbf{x}_2}^T\lambda) \\
&= \begin{bmatrix} g_{UU}(\lambda) & g_{UW}(\lambda) & \mathbf{0} & \mathbf{0} \\ g_{WU}(\lambda) & g_{WW}(\lambda) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}
\end{aligned}
$$

with

$$
\begin{aligned}
g_{WW}(\lambda) &= \frac{\partial}{\partial W}(g_W^T\lambda) \\
&= [\lambda]Y^* + Y^{*^T}[\lambda] \\
g_{WU}(\lambda) &= \frac{\partial}{\partial U}(g_W^T\lambda) \\
&= \imath\Big([\lambda]Y^* - Y^{*^T}[\lambda]\Big) \\
g_{UU}(\lambda) &= \frac{\partial}{\partial U}(g_U^T\lambda) \\
&= [\lambda]Y^* + Y^{*^T}[\lambda] \\
g_{UW}(\lambda) &= \frac{\partial}{\partial W}(g_U^T\lambda) \\
&= -\imath\Big([\lambda]Y^* - Y^{*^T}[\lambda]\Big) = g_{WU}^T(\lambda).
\end{aligned}
$$

Second order derivatives of the CP formulation:

$$
\begin{aligned}
g_{\mathbf{x}_1\mathbf{x}_1}(\lambda) &= \frac{\partial}{\partial \mathbf{x}_1}(g_{\mathbf{x}_1}^T\lambda) \\
&= \begin{bmatrix} g_{\Theta\Theta}(\lambda) & g_{\Theta\mathcal{V}}(\lambda) & g_{\Theta P^{\mathbf{g}}}(\lambda) & g_{\Theta Q^{\mathbf{g}}}(\lambda) \\ g_{\mathcal{V}\Theta}(\lambda) & g_{\mathcal{V}\mathcal{V}}(\lambda) & g_{\mathcal{V}P^{\mathbf{g}}}(\lambda) & g_{\mathcal{V}Q^{\mathbf{g}}}(\lambda) \\ g_{P^{\mathbf{g}}\Theta}(\lambda) & g_{P^{\mathbf{g}}\mathcal{V}}(\lambda) & \mathbf{0} & \mathbf{0} \\ g_{Q^{\mathbf{g}}\Theta}(\lambda) & g_{Q^{\mathbf{g}}\mathcal{V}}(\lambda) & \mathbf{0} & \mathbf{0} \end{bmatrix}
\end{aligned}
$$

with

$$
\begin{aligned}
g_{P^{\mathbf{g}}\Theta}(\lambda) &= \frac{\partial}{\partial \Theta}(g_{P^{\mathbf{g}}}^T\lambda) \\
&= -\imath C_g^T[\lambda][\Lambda^*] \\
g_{P^{\mathbf{g}}\mathcal{V}}(\lambda) &= \frac{\partial}{\partial \mathcal{V}}(g_{P^{\mathbf{g}}}^T\lambda) \\
&= C_g^T[\lambda][\mathcal{V}]^{-1}[\Lambda^*] \\
g_{Q^{\mathbf{g}}\Theta}(\lambda) &= \frac{\partial}{\partial \Theta}(g_{Q^{\mathbf{g}}}^T\lambda) \\
&= -C_g^T[\lambda][\Lambda^*] \\
g_{Q^{\mathbf{g}}\mathcal{V}}(\lambda) &= \frac{\partial}{\partial \mathcal{V}}(g_{Q^{\mathbf{g}}}^T\lambda) \\
&= -\imath C_g^T[\lambda][\Lambda^*][\mathcal{V}]^{-1} \\
g_{\Theta P^{\mathbf{g}}}(\lambda) &= \frac{\partial}{\partial P^{\mathbf{g}}}(g_{\Theta}^T\lambda) \\
&= -\imath\big[[\Lambda^*]\lambda\big]C_g = g_{P^{\mathbf{g}}\Theta}^T(\lambda) \\
g_{\Theta Q^{\mathbf{g}}}(\lambda) &= \frac{\partial}{\partial Q^{\mathbf{g}}}(g_{\Theta}^T\lambda) \\
&= -\big[[\Lambda^*]\lambda\big]C_g = g_{Q^{\mathbf{g}}\Theta}^T(\lambda) \\
g_{\mathcal{V}P^{\mathbf{g}}}(\lambda) &= \frac{\partial}{\partial P^{\mathbf{g}}}(g_{\mathcal{V}}^T\lambda) \\
&= \big[[\mathcal{V}]^{-1}[\Lambda^*]\lambda\big]C_g = g_{P^{\mathbf{g}}\mathcal{V}}^T(\lambda) \\
g_{\mathcal{V}Q^{\mathbf{g}}}(\lambda) &= \frac{\partial}{\partial Q^{\mathbf{g}}}(g_{\mathcal{V}}^T\lambda) \\
&= -\imath\big[[\mathcal{V}]^{-1}[\Lambda^*]\lambda\big]C_g = g_{Q^{\mathbf{g}}\mathcal{V}}^T(\lambda)
\end{aligned}
$$

$$
\begin{aligned}
g_{\Theta\Theta}(\lambda) &= \frac{\partial}{\partial\Theta}(g_{\Theta}^{T}\lambda) \\
&= -\big[Y^{T}\lambda\big][V] + [S^{sp^{*}}][\lambda][\Lambda^{*}] \\
g_{\Theta\mathcal{V}}(\lambda) &= \frac{\partial}{\partial\mathcal{V}}(g_{\Theta}^{T}\lambda) \\
&= \imath\Big(\big[Y^{T}\lambda\big][E] + [S^{sp^{*}}][\lambda][\mathcal{V}]^{-1}[\Lambda^{*}]\Big) \\
g_{\mathcal{V}\Theta}(\lambda) &= \frac{\partial}{\partial\Theta}(g_{\mathcal{V}}^{T}\lambda) \\
&= \imath\Big(\big[Y^{T}\lambda\big][E] + [S^{sp^{*}}][\mathcal{V}]^{-1}[\lambda][\Lambda^{*}]\Big) \\
g_{\mathcal{V}\mathcal{V}}(\lambda) &= \frac{\partial}{\partial\mathcal{V}}(g_{\mathcal{V}}^{T}\lambda) \\
&= -2[S^{sp^{*}}][\Lambda^{*}][\lambda][\mathcal{V}]^{-2}.
\end{aligned}
$$

Second order derivatives of the CC formulation:

$$
\begin{aligned}
g_{\mathbf{x}_2\mathbf{x}_2}(\lambda) &= \frac{\partial}{\partial\mathbf{x}_2}(g_{\mathbf{x}_2}^{T}\lambda) \\
&= \begin{bmatrix}
g_{WW}(\lambda) & g_{WU}(\lambda) & g_{WP^{\mathbf{g}}}(\lambda) & g_{WQ^{\mathbf{g}}}(\lambda) \\
g_{UW}(\lambda) & g_{UU}(\lambda) & g_{UP^{\mathbf{g}}}(\lambda) & g_{UQ^{\mathbf{g}}}(\lambda) \\
g_{P^{\mathbf{g}}W}(\lambda) & g_{P^{\mathbf{g}}U}(\lambda) & \mathbf{0} & \mathbf{0} \\
g_{Q^{\mathbf{g}}W}(\lambda) & g_{Q^{\mathbf{g}}U}(\lambda) & \mathbf{0} & \mathbf{0}
\end{bmatrix}
\end{aligned}
$$

with

$$
\begin{aligned}
g_{UP^{\mathbf{g}}}(\lambda) &= \frac{\partial}{\partial P^{\mathbf{g}}}(g_{U}^{T}\lambda) \\
&= \big[[\Lambda^{*}]^{2}\lambda\big]C_g = g_{P^{\mathbf{g}}U}^{T}(\lambda) \\
g_{UQ^{\mathbf{g}}}(\lambda) &= \frac{\partial}{\partial Q^{\mathbf{g}}}(g_{U}^{T}\lambda) \\
&= -\imath\big[[\Lambda^{*}]^{2}\lambda\big]C_g = g_{Q^{\mathbf{g}}\mathcal{V}}^{T}(\lambda) \\
g_{WW}(\lambda) &= \frac{\partial}{\partial W}(g_{W}^{T}\lambda) \\
&= 2[S^{sp^{*}}][\lambda][\Lambda^{*}]^{3} \\
g_{WU}(\lambda) &= \frac{\partial}{\partial U}(g_{W}^{T}\lambda) \\
&= 2\imath[S^{sp^{*}}][\lambda][\Lambda^{*}]^{3} \\
g_{UU}(\lambda) &= \frac{\partial}{\partial U}(g_{U}^{T}\lambda) \\
&= -2[S^{sp^{*}}][\lambda][\Lambda^{*}]^{3} \\
g_{UW}(\lambda) &= \frac{\partial}{\partial W}(g_{U}^{T}\lambda) \\
&= 2\imath[S^{sp^{*}}][\lambda][\Lambda^{*}]^{3}
\end{aligned}
$$

$$
\begin{aligned}
g_{P^{\mathbf{g}}W}(\lambda) &= \frac{\partial}{\partial W}(g_{P^{\mathbf{g}}}^T\lambda) \\
&= -\imath C_g^T[\lambda][\Lambda^*]^2 \\
g_{P^{\mathbf{g}}U}(\lambda) &= \frac{\partial}{\partial U}(g_{P^{\mathbf{g}}}^T\lambda) \\
&= C_g^T[\lambda][\Lambda^*]^2 \\
g_{Q^{\mathbf{g}}W}(\lambda) &= \frac{\partial}{\partial W}(g_{Q^{\mathbf{g}}}^T\lambda) \\
&= -C_g^T[\lambda][\Lambda^*]^2 \\
g_{Q^{\mathbf{g}}U}(\lambda) &= \frac{\partial}{\partial U}(g_{Q^{\mathbf{g}}}^T\lambda) \\
&= -\imath C_g^T[\lambda][\Lambda^*]^2 \\
g_{WP^{\mathbf{g}}}(\lambda) &= \frac{\partial}{\partial P^{\mathbf{g}}}(g_W^T\lambda) \\
&= -\imath\big[[\Lambda^*]^2\lambda\big]C_g = g_{P^{\mathbf{g}}\Theta}^T(\lambda) \\
g_{WQ^{\mathbf{g}}}(\lambda) &= \frac{\partial}{\partial Q^{\mathbf{g}}}(g_W^T\lambda) \\
&= -\big[[\Lambda^*]^2\lambda\big]C_g = g_{Q^{\mathbf{g}}\Theta}^T(\lambda).
\end{aligned}
$$

B.3 Derivatives of inequality constraints $h(\mathbf{x})$

Second order derivatives w.r.t $\mathbf{x}_1$:

$$
\begin{aligned}
h^f_{\mathbf{x}_1\mathbf{x}_1}(\mu) &= \frac{\partial}{\partial \mathbf{x}_1}(h_{\mathbf{x}_1}^{f^T}\mu) \\
&= 2\,\mathrm{Re}\left\{S^f_{\mathbf{x}_1\mathbf{x}_1}\big([S^{f^*}]\mu\big) + S_{\mathbf{x}_1}^{f^T}[\mu]S_{\mathbf{x}_1}^{f^*}\right\}.
\end{aligned}
$$

$$
\begin{aligned}
S^f_{\mathbf{x}_1\mathbf{x}_1}(\mu) &= \frac{\partial}{\partial \mathbf{x}_1}(S_{\mathbf{x}_1}^{f^T}\mu) \\
&= \begin{bmatrix}
g_{\Theta\Theta}(\mu) & g_{\Theta\mathcal{V}}(\mu) & \mathbf{0} & \mathbf{0} \\
g_{\mathcal{V}\Theta}(\mu) & g_{\mathcal{V}\mathcal{V}}(\mu) & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}
\end{bmatrix}
\end{aligned}
$$

with

$$
\begin{aligned}
S^f_{\Theta\Theta}(\mu) &= \frac{\partial}{\partial\Theta}(S^{f^T}_{\Theta}\,\mu) \\
&= [V^*]Y^{*^T}_f[\mu]C_f[V] + [V]C^T_f[\mu]Y^*_f[V^*] - \Big[Y^{*^T}_f[\mu]C_fV\Big][V^*] - \big[C^T_f[\mu]Y^*_fV^*\big][V] \\
S^f_{\mathcal{V}\Theta}(\mu) &= \frac{\partial}{\partial\Theta}(S^{f^T}_{\mathcal{V}}\,\mu) \\
&= \imath[\mathcal{V}]^{-1}\Bigg([V^*]Y^{*^T}_f[\mu]C_f[V] - [V]C^T_f[\mu]Y^*_f[V^*] - \Big[Y^{*^T}_f[\mu]C_fV\Big][V^*] + \Big[C^T_f[\mu]Y^*_fV^*\Big][V]\Bigg) \\
S^f_{\Theta\mathcal{V}}(\mu) &= \frac{\partial}{\partial\mathcal{V}}(S^{f^T}_{\Theta}\,\mu) \\
&= \imath\Bigg([V]C^T_f[\mu]Y^*_f[V^*] - [V^*]Y^{*^T}_f[\mu]C_f[V] - \Big[Y^{*^T}_f[\mu]C_fV\Big][V^*] + \big[C^T_f[\mu]Y^*_fV^*\big][V]\Bigg)[\mathcal{V}]^{-1} \\
&= S^{f^T}_{\mathcal{V}\Theta}(\mu) \\
S^f_{\mathcal{V}\mathcal{V}}(\mu) &= \frac{\partial}{\partial\mathcal{V}}(S^{f^T}_{\mathcal{V}}\,\mu) \\
&= [\mathcal{V}]^{-1}\Big([V^*]Y^*_f[\mu]C_f[V] + [V]C^T_f[\mu]Y^*_f[V^*]\Big)[\mathcal{V}]^{-1}.
\end{aligned}
$$

Second order derivatives w.r.t $\mathbf{x}_2$:

$$
\begin{aligned}
S^f_{\mathbf{x}_2\mathbf{x}_2}(\mu) &= \frac{\partial}{\partial\mathbf{x}_2}(S^{f^T}_{\mathbf{x}_2}\,\mu) \\
&= \begin{bmatrix} S^f_{UU}(\mu) & S^f_{UW}(\mu) & \mathbf{0} & \mathbf{0} \\ S^f_{WU}(\mu) & S^f_{WW}(\mu) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}
\end{aligned}
$$

with

$$
\begin{aligned}
S^f_{UU}(\mu) &= \frac{\partial}{\partial U}(S^{f^T}_{U}\,\mu) \\
&= C^T_f[\mu]Y^*_f + Y^{*^T}_f[\mu]C_f \\
S^f_{WU}(\mu) &= \frac{\partial}{\partial U}(S^{f^T}_{W}\,\mu) \\
&= \imath\Big(C^T_f[\mu]Y^*_f - Y^{*^T}_f[\mu]C_f\Big) \\
S^f_{UW}(\mu) &= \frac{\partial}{\partial W}(S^{f^T}_{U}\,\mu) \\
&= -\imath\Big(C^T_f[\mu]Y^*_f - Y^{*^T}_f[\mu]C_f\Big) = S^{f^T}_{WU}(\mu) \\
S^f_{WW}(\mu) &= \frac{\partial}{\partial W}(S^{f^T}_{W}\,\mu) \\
&= C^T_f[\mu]Y^*_f + Y^{*^T}_f[\mu]C_f.
\end{aligned}
$$

The derivatives $h^t_{\mathbf{xx}}$ are identical to $h^f_{\mathbf{xx}}$ (i.e. replace all super-scripts f with t).

Bibliography

[1] W. F. Tinney and C. E. Hart, "Power flow solution by Newton's method," *IEEE Transactions on Power Apparatus and systems*, no. 11, pp. 1449–1460, 1967.

[2] P. Schavemaker and L. van der Sluis, *Electrical Power System Essentials*. Wiley, 2008.

[3] W. Commons, "Electricity grid schematic english." `https://en.wikipedia.org/wiki/File:Electricity_Grid_Schematic_English.svg`, 2010.

[4] A. Keyhani and M. Marwali, *Smart Power Grids 2011*. Power Systems, Springer Berlin Heidelberg, 2012.

[5] K. Budka, J. Deshpande, and M. Thottan, *Communication Networks for Smart Grids: Making Smart Grid Real*. Computer Communications and Networks, Springer-Verlag, 2014.

[6] M. Thomas and J. McDonald, *Power System SCADA and Smart Grids*. Taylor & Francis, 2015.

[7] Wikipedia, "Smart grid — wikipedia, the free encyclopedia." `https://en.wikipedia.org/w/index.php?title=Smart_grid&oldid=694384742`, 2015. [Online; accessed 17-December-2015].

[8] B. Sereeter, K. Vuik, and C. Witteveen, "Newton power flow methods for unbalanced three-phase distribution networks," *Energies*, vol. 10, no. 10, p. 1658, 2017.

[9] B. Sereeter, C. Vuik, and C. Witteveen, "On a comparison of Newton-Raphson solvers for power flow problems," *Journal of Computational and Applied Mathematics*, vol. 360, pp. 157–169, Nov 2019.

[10] B. Sereeter, A. Markensteijn, M. E. Kootte, C. Vuik, and C. Witteveen, "A novel linearized power flow approach for transmission and distribution networks," *IEEE PES Transaction on Power Systems [Under review]*, Dec 2019.

[11] B. Sereeter, W. van Westering, C. Vuik, and C. Witteveen, "Linear power flow method improved with numerical analysis techniques applied to a very large network," *Energies*, vol. 12, no. 21, p. 4078, 2019.

[12] M. E. Kootte, B. Sereeter, J. E. Romate, and C. Vuik, "Comparison of numerical methods to solve the steady-state integrated transmission-distribution power flow problem," *IEEE PES Innovative Smart Grid Technologies Europe [Under review]*, Oct 2020.

[13] B. Sereeter and R. D. Zimmerman, "*Addendum* to AC power flows and their derivatives using complex matrix notation: Nodal Current Balance," tech. rep., April 2018. MATPOWER Technical Note 3, Available: `https://matpower.org/docs/TN3-More-OPF-Derivatives.pdf`.

[14] B. Sereeter and R. D. Zimmerman, "AC power flows and their derivatives using complex matrix notation and Cartesian coordinate voltages," tech. rep., April 2018. MATPOWER Technical Note 4, Available: `https://matpower.org/docs/TN4-OPF-Derivatives-Cartesian.pdf`.

[15] B. Sereeter, C. Vuik, C. Witteveen, and P. Palensky, "Optimal power flow formulations and their impacts on the performance of solution methods," IEEE Power & Energy Society General Meeting, IEEE, Aug 2019.

[16] Y. Saad, *Iterative methods for sparse linear systems*, vol. 82. siam, 2003.

[17] G. H. Golub and C. F. Van Loan, *Matrix computations*, vol. 3. JHU press, 2012.

[18] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, vol. 49. NBS Washington, DC, 1952.

[19] H. A. Van der Vorst, "Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems," *SIAM Journal on scientific and Statistical Computing*, vol. 13, no. 2, pp. 631–644, 1992.

[20] P. Sonneveld, "CGS, a fast Lanczos-type solver for nonsymmetric linear systems," *SIAM journal on scientific and statistical computing*, vol. 10, no. 1, pp. 36–52, 1989.

[21] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on scientific and statistical computing*, vol. 7, no. 3, pp. 856–869, 1986.

[22] W. E. Arnoldi, "The principle of minimized iterations in the solution of the matrix eigenvalue problem," *Quarterly of applied mathematics*, vol. 9, no. 1, pp. 17–29, 1951.

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

[24] S. Granville, "Optimal reactive dispatch through interior point methods," *IEEE Transactions on power systems*, vol. 9, no. 1, pp. 136–146, 1994.

[25] K. Lindén and I. Segerqvist, "Modelling of load devices and studying load/system characteristics," *Licentiate thesis*, 1992.

[26] J. Subrahmanyam and C. Radhakrishna, "A simple approach of three phase distribution system modeling for power flow calculations," *International Journal of Energy and Power Engineering*, vol. 3, p. 3, 2010.

[27] EPRI, "Engineering guide for integration of distributed generation and storage into power distribution systems," tech. rep., December 2000.

[28] T.-H. Chen, M.-S. Chen, T. Inoue, P. Kotas, and E. A. Chebli, "Three-phase cogenerator and transformer models for distribution system analysis," *IEEE Transactions on Power Delivery*, vol. 6, no. 4, pp. 1671–1681, 1991.

[29] R. Idema, D. J. Lahaye, *et al.*, "Computational methods in power system analysis," 2014.

[30] W. D. Stevenson, "Elements of power system analysis," 1975.

[31] B. Stott and O. Alsaç, "Fast decoupled load flow," *IEEE transactions on power apparatus and systems*, no. 3, pp. 859–869, 1974.

[32] B. Stott, "Review of load-flow calculation methods," *Proceedings of the IEEE*, vol. 62, no. 7, pp. 916–929, 1974.

[33] S. Iwamoto and Y. Tamura, "A load flow calculation method for ill-conditioned power systems," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-100, pp. 1736–1743, April 1981.

[34] S. C. Tripathy, G. D. Prasad, O. P. Malik, and G. S. Hope, "Load-flow solutions for ill-conditioned power systems by a Newton-like method," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-101, pp. 3648–3657, Oct 1982.

[35] S. S. K. Reddy, S. Suresh and S. V. J. Kumar, "Load flow solution for ill-conditioned power systems using Runge-Kutta and Iwamoto methods with facts devices," *Journal of Theoretical & Applied Information Technology*, vol. 5, no. 6, 2009.

[36] H.-D. Chiang, "A decoupled load flow method for distribution power networks: Algorithms, analysis and convergence study," *International Journal of Electrical Power and Energy Systems*, vol. 13, no. 3, pp. 130 – 138, 1991.

[37] T.-H. Chen, M.-S. Chen, K.-J. Hwang, P. Kotas, and E. A. Chebli, "Distribution system power flow analysis-a rigid approach," *Power Delivery, IEEE Transactions on*, vol. 6, no. 3, pp. 1146–1152, 1991.

[38] F. Zhang and C. S. Cheng, "A modified Newton method for radial distribution system power flow analysis," *IEEE Transactions on Power Systems*, vol. 12, pp. 389–397, Feb 1997.

[39] H. L. Nguyen, "Newton-Raphson method in complex form," *Power Systems, IEEE Transactions on*, vol. 12, no. 3, pp. 1355–1359, 1997.

[40] V. M. da Costa, N. Martins, and J. L. R. Pereira, "Developments in the Newton-Raphson power flow formulation based on current injections," *IEEE Transactions on Power Systems*, vol. 14, pp. 1320–1326, Nov 1999.

[41] P. A. N. Garcia, J. L. R. Pereira, S. Carneiro, V. M. da Costa, and N. Martins, "Three-phase power flow calculations using the current injection method," *IEEE Transactions on Power Systems*, vol. 15, pp. 508–514, May 2000.

[42] W.-M. Lin and J.-H. Teng, "Three-phase distribution network fast-decoupled power flow solutions," *International Journal of Electrical Power and Energy Systems*, vol. 22, no. 5, pp. 375 – 380, 2000.

[43] J.-H. Teng and C.-Y. Chang, "A novel and fast three-phase load flow for unbalanced radial distribution systems," *Power Systems, IEEE Transactions on*, vol. 17, no. 4, pp. 1238–1244, 2002.

[44] J.-H. Teng, "A modified Gauss-Seidel algorithm of three-phase power flow analysis in distribution networks," *International Journal of Electrical Power and Energy Systems*, vol. 24, no. 2, pp. 97 – 102, 2002.

[45] J. Vieira, W. Freitas, and A. Morelato, "Phase-decoupled method for three-phase power-flow analysis of unbalanced distribution systems," *IEE Proceedings-Generation, Transmission and Distribution*, vol. 151, no. 5, pp. 568–574, 2004.

[46] T.-H. Chen and N.-C. Yang, "Loop frame of reference based three-phase power flow for unbalanced radial distribution systems," *Electric power systems research*, vol. 80, no. 7, pp. 799–806, 2010.

[47] H. Sun, D. Nikovski, T. Ohno, T. Takano, and Y. Kojima, "A fast and robust load flow method for distribution systems with distributed generations," *Energy Procedia*, vol. 12, pp. 236–244, 2011.

[48] P. J. Lagace, "Power flow methods for improving convergence," in *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*, pp. 1387–1392, IEEE, 2012.

[49] M. M. A. Abdelaziz, H. E. Farag, E. F. El-Saadany, and Y. A.-R. I. Mohamed, "A novel and generalized three-phase power flow algorithm for islanded microgrids using a Newton trust region method," *IEEE transactions on power systems*, vol. 28, no. 1, pp. 190–201, 2012.

[50] R. Idema, D. J. Lahaye, C. Vuik, and L. van der Sluis, "Scalable Newton-Krylov solver for very large power flow problems," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 390–396, 2012.

[51] C. Li, S. K. Chaudhary, J. C. Vasquez, and J. M. Guerrero, "Power flow analysis algorithm for islanded LV microgrids including distributed generator units with droop control and virtual impedance loop," in *Applied Power Electronics Conference and Exposition (APEC), 2014 Twenty-Ninth Annual IEEE*, pp. 3181–3185, IEEE, 2014.

[52] L. Wang, C. Chen, and T. Shen, "Improvement of power flow calculation with optimization factor based on current injection method," *Discrete Dynamics in Nature and Society*, vol. 2014, 2014.

[53] F. Milano, "Analogy and convergence of Levenberg's and Lyapunov-based methods for power flow analysis," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1663–1664, 2016.

[54] F. Mumtaz, M. Syed, M. Al Hosani, and H. Zeineldin, "A novel approach to solve power flow for islanded microgrids using modified Newton-Raphson with droop control of DG," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 2, pp. 493–503, 2016.

[55] R. Pourbagher and S. Y. Derakhshandeh, "A powerful method for solving the power flow problem in the ill-conditioned systems," *International Journal of Electrical Power & Energy Systems*, vol. 94, pp. 88–96, 2018.

[56] C. S. Cheng and D. Shirmohammadi, "A three-phase power flow method for real-time distribution system analysis," *Power Systems, IEEE Transactions on*, vol. 10, no. 2, pp. 671–679, 1995.

[57] D. Thukaram, H. W. Banda, and J. Jerome, "A robust three phase power flow algorithm for radial distribution systems," *Electric Power Systems Research*, vol. 50, no. 3, pp. 227–236, 1999.

[58] R. Ranjan, B. Venkatesh, A. Chaturvedi, and D. Das, "Power flow solution of three-phase unbalanced radial distribution network," *Electric Power Components and Systems*, vol. 32, no. 4, pp. 421–433, 2004.

[59] S. Khushalani, J. M. Solanki, and N. N. Schulz, "Development of three-phase unbalanced power flow using PV and PQ models for distributed generation and study of the impact of DG models," *Power Systems, IEEE Transactions on*, vol. 22, no. 3, pp. 1019–1025, 2007.

[60] S. Moghaddas-Tafreshi and E. Mashhour, "Distributed generation modeling for power flow studies and a three-phase unbalanced power flow solution for radial distribution systems considering distributed generation," *Electric Power Systems Research*, vol. 79, no. 4, pp. 680–686, 2009.

[61] D. Shirmohammadi, H. Hong, A. Semlyen, and G. Luo, "A compensation-based power flow method for weakly meshed distribution and transmission networks," *Power Systems, IEEE Transactions on*, vol. 3, no. 2, pp. 753–762, 1988.

[62] G. X. Luo and A. Semlyen, "Efficient load flow for large weakly meshed networks," *IEEE Transactions on Power Systems*, vol. 5, pp. 1309–1316, Nov 1990.

[63] M. H. Haque, "Efficient load flow method for distribution systems with radial or mesh configuration," *IEE Proceedings - Generation, Transmission and Distribution*, vol. 143, pp. 33–38, Jan 1996.

[64] M. Haque, "A general load flow method for distribution systems," *Electric Power Systems Research*, vol. 54, no. 1, pp. 47–54, 2000.

[65] J.-H. Teng, "A direct approach for distribution system load flow solutions," *Power Delivery, IEEE Transactions on*, vol. 18, no. 3, pp. 882–887, 2003.

[66] D. Rajicic, R. Ackovski, and R. Taleski, "Voltage correction power flow," *IEEE Transactions on Power Delivery*, vol. 9, no. 2, pp. 1056–1062, 1994.

[67] Y. Zhu and K. Tomsovic, "Adaptive power flow method for distribution systems with dispersed generation," *Power Delivery, IEEE Transactions on*, vol. 17, no. 3, pp. 822–827, 2002.

[68] J.-H. Teng, "Modelling distributed generations in three-phase distribution load flow," *Generation, Transmission & Distribution, IET*, vol. 2, no. 3, pp. 330–340, 2008.

[69] A. Augugliaro, L. Dusonchet, S. Favuzza, M. Ippolito, and E. R. Sanseverino, "A new backward/forward method for solving radial distribution networks with PV nodes," *Electric Power Systems Research*, vol. 78, no. 3, pp. 330–336, 2008.

[70] S. Ghosh and D. Das, "Method for load-flow solution of radial distribution networks," *IEE Proceedings - Generation, Transmission and Distribution*, vol. 146, pp. 641–648, Nov 1999.

[71] C. L. Samuel Mok, S. Elangovan and M. Salama, "A new approach for power flow analysis of balanced radial distribution systems," *Electric Machines & Power Systems*, vol. 28, no. 4, pp. 325–340, 2000.

[72] P. Aravindhababu, S. Ganapathy, and K. Nayar, "A novel technique for the analysis of radial distribution systems," *International journal of electrical power & energy systems*, vol. 23, no. 3, pp. 167–171, 2001.

[73] J. Liu, M. Salama, and R. Mansour, "An efficient power flow algorithm for distribution systems with polynomial load," *International Journal of Electrical Engineering Education*, vol. 39, no. 4, pp. 371–386, 2002.

[74] R. Cespedes, "New method for the analysis of distribution networks," *IEEE Transactions on Power Delivery*, vol. 5, no. 1, pp. 391–396, 1990.

[75] G. Jasmon and L. Lee, "Distribution network reduction for voltage stability analysis and loadflow calculations," *International Journal of Electrical Power & Energy Systems*, vol. 13, no. 1, pp. 9–13, 1991.

[76] D. Das, H. Nagi, and D. Kothari, "Novel method for solving radial distribution networks," in *Generation, Transmission and Distribution, IEE Proceedings-*, vol. 141, pp. 291–298, IET, 1994.

[77] D. Das, D. Kothari, and A. Kalam, "Simple and efficient method for load flow solution of radial distribution networks," *International Journal of Electrical Power & Energy Systems*, vol. 17, no. 5, pp. 335–346, 1995.

[78] M. Haque, "Load flow solution of distribution systems with voltage dependent load models," *Electric Power Systems Research*, vol. 36, no. 3, pp. 151–156, 1996.

[79] M. Afsari, S. Singh, G. Raju, and G. Rao, "A fast power flow solution of radial distribution networks," *Electric Power Components and Systems*, vol. 30, no. 10, pp. 1065–1074, 2002.

[80] S. Mekhamer, S. Soliman, M. Moustafa, and M. El-Hawary, "Load flow solution of radial distribution feeders: A new contribution," *International journal of electrical power & energy systems*, vol. 24, no. 9, pp. 701–707, 2002.

[81] R. Ranjan and DAS, "Simple and efficient computer algorithm to solve radial distribution networks," *Electric power components and systems*, vol. 31, no. 1, pp. 95–107, 2003.

[82] U. Eminoglu and M. H. Hocaoglu, "A new power flow method for radial distribution systems including voltage dependent load models," *Electric power systems research*, vol. 76, no. 1, pp. 106–114, 2005.

[83] S. Satyanarayana, T. Ramana, S. Sivanagaraju, and G. Rao, "An efficient load flow solution for radial distribution network including voltage dependent load models," *Electric Power Components and Systems*, vol. 35, no. 5, pp. 539–551, 2007.

[84] M. S. Srinivas, "Distribution load flows: A brief review," in *Power Engineering Society Winter Meeting, 2000. IEEE*, vol. 2, pp. 942–945 vol.2, 2000.

[85] J. A. Martinez and J. Mahseredjian, "Load flow calculations in distribution systems with distributed resources. A review," in *Power and Energy Society General Meeting, 2011 IEEE*, pp. 1–8, IEEE, 2011.

[86] K. Balamurugan and D. Srinivasan, "Review of power flow studies on distribution network with distributed generation," in *Power Electronics and Drive Systems (PEDS), 2011 IEEE Ninth International Conference on*, pp. 411–417, IEEE, 2011.

[87] U. Eminoglu and M. H. Hocaoglu, "Distribution systems forward/backward sweep-based power flow algorithms: A review and comparison study," *Electric Power Components and Systems*, vol. 37, no. 1, pp. 91–110, 2008.

[88] A. J. Flueck and H.-D. Chiang, "Solving the nonlinear power flow equations with an inexact Newton method using GMRES," *IEEE Transactions on Power Systems*, vol. 13, no. 2, pp. 267–273, 1998.

[89] V. Da Costa, J. Pereira, and N. Martins, "An augmented Newton-Raphson power flow formulation based on current injections," *International journal of electrical power & energy systems*, vol. 23, no. 4, pp. 305–312, 2001.

[90] F. De Leon and A. Semlyen, "Iterative solvers in the Newton power flow problem: Preconditioners, inexact solutions, and partial Jacobian updates," *IEE Proceedings-Generation, Transmission and Distribution*, vol. 149, no. 4, pp. 479–484, 2002.

[91] U. Thongkrajay, N. Poolsawat, T. Ratniyomchai, and T. Kulworawanichpong, "Alternative Newton-Raphson power flow calculation in unbalanced three-phase power distribution systems," in *Proceedings of the 5th WSEAS international conference on Applications of electrical engineering*, pp. 24–29, World Scientific and Engineering Academy and Society (WSEAS), 2006.

[92] J. E. Van Ness and J. H. Griffin, "Elimination methods for load-flow studies," *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, vol. 80, no. 3, pp. 299–302, 1961.

[93] H. W. Dommel, W. F. Tinney, and W. L. Powell, "Further developments in Newton's method for power system applications," *IEEE Winter Power Meeting, Conference Paper*, pp. CP 161–PWR New York, January 1970.

[94] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Transactions on Power Delivery*, vol. 4, no. 2, pp. 1401–1407, 1989.

[95] M. E. Baran and F. F. Wu, "Optimal capacitor placement on radial distribution systems," *IEEE Transactions on power Delivery*, vol. 4, no. 1, pp. 725–734, 1989.

[96] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on power systems*, vol. 26, no. 1, pp. 12–19, 2011.

[97] W. H. Kersting, "Radial distribution test feeders," in *Power Engineering Society Winter Meeting, 2001. IEEE*, vol. 2, pp. 908–912, IEEE, 2001.

[98] G. Chang, S. Chu, and H. Wang, "An improved backward/forward sweep load flow algorithm for radial distribution systems," *IEEE Transactions on power systems*, vol. 22, no. 2, pp. 882–884, 2007.

[99] J. R. Martí, H. Ahmadi, and L. Bashualdo, "Linear power-flow formulation based on a voltage-dependent load model," *IEEE Transactions on Power Delivery*, vol. 28, no. 3, pp. 1682–1690, 2013.

[100] O. D. Montoya, L. Grisales-Noreña, D. González-Montoya, C. Ramos-Paja, and A. Garces, "Linear power flow formulation for low-voltage DC power grids," *Electric Power Systems Research*, vol. 163, pp. 375–381, 2018.

[101] A. Garces, "A linear three-phase load flow for power distribution systems," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 827–828, 2015.

[102] S. Bolognani and S. Zampieri, "On the existence and linear approximation of the power flow solution in power distribution networks," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 163–172, 2015.

[103] H. Ahmadi, J. R. Martı, and A. von Meier, "A linear power flow formulation for three-phase distribution systems," *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 5012–5021, 2016.

[104] S. V. Dhople, S. S. Guggilam, and Y. C. Chen, "Linear approximations to AC power flow in rectangular coordinates," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 211–217, IEEE, 2015.

[105] Y. Wang, N. Zhang, H. Li, J. Yang, and C. Kang, "Linear three-phase power flow for unbalanced active distribution networks with PV nodes," *CSEE Journal of Power and Energy Systems*, vol. 3, no. 3, pp. 321–324, 2017.

[106] K. Liu, C. Wang, W. Wang, Y. Chen, and H. Wu, "Linear power flow calculation of distribution networks with distributed generation," *IEEE Access*, vol. 7, pp. 44686–44695, 2019.

[107] O. Montoya, W. Gil-González, and L. Grisales-Noreña, "Linear–based Newton–Raphson approximation for power flow solution in DC power grids," in *2018 IEEE 9th Power, Instrumentation and Measurement Meeting (EPIM)*, pp. 1–6, IEEE, 2018.

[108] W. van Westering, B. Droste, and H. Hellendoorn, "Combined medium voltage and low voltage simulation to accurately determine the location of voltage problems in large grids," in *CIRED 27nd International Conference on Electricity Distribution*, 2019.

[109] Y.-S. Zhang and H.-D. Chiang, "Fast Newton-FGMRES solver for large-scale power flow study," *IEEE Transactions on Power Systems*, vol. 25, no. 2, pp. 769–776, 2010.

[110] D. Lahaye and K. Vuik, "Globalized Newton–Krylov–Schwarz AC load flow methods for future power systems," in *Intelligent Integrated Energy Systems*, pp. 79–98, Springer, 2019.

[111] A. B. Birchfield, T. Xu, and T. J. Overbye, "Power flow convergence and reactive power planning in the creation of large synthetic grids," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6667–6674, 2018.

[112] R. Bolgaryn, A. Scheidler, and M. Braun, "Combined planning of medium and low voltage grids," *13th IEEE PowerTech Conference*, 2019.

[113] W. van Westering and H. Hellendoorn, "Low voltage power grid congestion reduction using a community battery: Design principles, control and experimental validation," *International Journal of Electrical Power & Energy Systems*, vol. 114, p. 105349, 2020.

[114] S. Frank, I. Steponavice, and S. Rebennack, "Optimal power flow: A bibliographic survey I," *Energy Systems*, vol. 3, no. 3, pp. 221–258, 2012.

[115] P. Panciatici, M. C. Campi, S. Garatti, S. Low, D. K. Molzahn, A. Sun, and L. Wehenkel, "Advanced optimization methods for power systems," in *Power Systems Computation Conference (PSCC), 2014*, pp. 1–18, IEEE, 2014.

[116] B. Bhandari, K.-T. Lee, G.-Y. Lee, Y.-M. Cho, and S.-H. Ahn, "Optimization of hybrid renewable energy power systems: A review," *International journal of precision engineering and manufacturing-green technology*, vol. 2, no. 1, pp. 99–112, 2015.

[117] B. Mary, P. Richard, and C. Anya, "Optimal Power Flow and Formulations," *Energy Regulatory*, vol. 1, no. 1, pp. 1–36, 2012.

[118] Y. Xiang, J. Liu, F. Li, Y. Liu, Y. Liu, R. Xu, Y. Su, and L. Ding, "Optimal active distribution network planning: A review," *Electric Power Components and Systems*, vol. 44, no. 10, pp. 1075–1094, 2016.

[119] R. Amarnath and N. Ramana, "State of art in optimal power flow solution methodologies," *Theoretical and Applied Information Technology*, vol. 30, pp. 128–154, 2011.

[120] F. J. Nogales, F. J. Prieto, and A. J. Conejo, "A decomposition methodology applied to the multi-area optimal power flow problem," *Annals of operations research*, vol. 120, no. 1-4, pp. 99–116, 2003.

[121] C. Y. David, J. E. Fagan, B. Foote, and A. A. Aly, "An optimal load flow study by the generalized reduced gradient approach," *Electric Power Systems Research*, vol. 10, no. 1, pp. 47–53, 1986.

[122] D. I. Sun, B. Ashley, B. Brewer, A. Hughes, and W. F. Tinney, "Optimal power flow by Newton approach," *IEEE Transactions on Power Apparatus and systems*, no. 10, pp. 2864–2880, 1984.

[123] E. Housos and G. Irisarri, "A sparse variable metric optimization method applied to the solution of power system problems," *IEEE Transactions on Power Apparatus and Systems*, no. 1, pp. 195–202, 1982.

[124] V. Klee and G. J. Minty, "How good is the simplex algorithm," tech. rep., Department of mathematics, Washington university, 1970.

[125] D. S. Kirschen and H. P. Van Meeteren, "MW/voltage control in a linear programming based optimal power flow," *IEEE Transactions on Power Systems*, vol. 3, no. 2, pp. 481–489, 1988.

[126] K. Zehar and S. Sayah, "Optimal power flow with environmental constraint using a fast successive linear programming algorithm: Application to the algerian power system," *Energy Conversion and Management*, vol. 49, no. 11, pp. 3362–3366, 2008.

[127] C. Lehmkoster, "Security constrained optimal power flow for an economical operation of FACTS-devices in liberalized energy markets," *IEEE transactions on Power Delivery*, vol. 17, no. 2, pp. 603–608, 2002.

[128] X. Lin, A. David, and C. Yu, "Reactive power optimisation with voltage stability consideration in power market systems," *IEE Proceedings-Generation, Transmission and Distribution*, vol. 150, no. 3, pp. 305–310, 2003.

[129] J. A. Momoh and J. Zhu, "Improved interior point method for OPF problems," *IEEE Transactions on Power Systems*, vol. 14, no. 3, pp. 1114–1120, 1999.

[130] W. Qiu, A. J. Flueck, and F. Tu, "A new parallel algorithm for security constrained optimal power flow with a nonlinear interior point method," in *Power Engineering Society General Meeting, 2005. IEEE*, pp. 447–453, IEEE, 2005.

[131] R. A. Jabr, "A primal-dual interior-point method to solve the optimal power flow dispatching problem," *Optimization and Engineering*, vol. 4, no. 4, pp. 309–336, 2003.

[132] A. Maniezzo, "Distributed optimization by ant colonies," in *Toward a practice of autonomous systems: proceedings of the First European Conference on Artificial Life*, p. 134, Mit Press, 1992.

[133] B. Gasbaoui and B. Allaoua, "Ant colony optimization applied on combinatorial problem for optimal power flow solution," *Leonardo Journal of Sciences*, vol. 14, pp. 1–17, 2009.

[134] T. Nguyen, "Neural network load-flow," *IEE Proceedings-generation, transmission and distribution*, vol. 142, no. 1, pp. 51–58, 1995.

[135] B. Das and P. K. Verma, "Artificial neural network-based optimal capacitor switching in a distribution system," *Electric Power Systems Research*, vol. 60, no. 2, pp. 55–62, 2001.

[136] Q. H. Wu and J. Ma, "Power system optimal reactive power dispatch using evolutionary programming," *IEEE Transactions on power systems*, vol. 10, no. 3, pp. 1243–1249, 1995.

[137] T. Bouktir, L. Slimani, and M. Belkacemi, "A genetic algorithm for solving the optimal power flow problem," *Leonardo Journal of Sciences*, vol. 4, pp. 44–58, 2004.

[138] H. Yoshida, K. Kawata, Y. Fukuyama, S. Takayama, and Y. Nakanishi, "A particle swarm optimization for reactive power and voltage control considering voltage security assessment," *IEEE Transactions on power systems*, vol. 15, no. 4, pp. 1232–1239, 2000.

[139] C. Roa-Sepulveda and B. Pavez-Lazo, "A solution to the optimal power flow using simulated annealing," *International journal of electrical power & energy systems*, vol. 25, no. 1, pp. 47–57, 2003.

[140] S. Frank, I. Steponavice, and S. Rebennack, "Optimal power flow: A bibliographic survey II," *Energy Systems*, vol. 3, no. 3, pp. 259–289, 2012.

[141] A. Castillo and R. P. O'Neill, "Computational performance of solution techniques applied to the ACOPF," *Published online at http://www.ferc.gov/industries/electric/indus-act/market-planning/opf-papers/acopf-5-computational-testing.pdf*, 2013.

[142] B. Park, L. Tang, M. C. Ferris, and C. L. DeMarco, "Examination of three different ACOPF formulations with generator capability curves," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2913–2923, 2017.

[143] J. A. Momoh, *Electric power system applications of optimization*. CRC press, 2008.

[144] R. D. Zimmerman, "AC power flows, generalized OPF costs and their derivatives using complex matrix notation," 2010.

Curriculum Vitae



Baljinnyam Sereeter was born on the 13th of December, 1990, in Zavkhan, Mongolia. He received his Bachelor degree in Applied Mathematics at the National University of Mongolia, Ulaanbaatar, Mongolia, in 2011. Baljinnyam worked as a software engineer at Infosoft Co., LTD in Ulaanbaatar, Mongolia until he received the full scholarship for double degree master program "Computer Simulation for Science and Engineering (COSSE)" from *Erasmus Mundus*. He joined the COSSE master's programme in 2012 and obtained the double MSc degree in Scientific Computing from Technical University of Berlin, Berlin, Germany and in Applied Mathematics from Technical University of Delft, Delft, The Netherlands, in 2014. From 2015-2019, Baljinnyam worked as a PhD researcher at the Numerical Analysis research group of the Delft Institute of Applied Mathematics. He was hired to work on Project C of the "Smart Energy Systems in the Built Environment (SES-BE)" program. His research focused on mathematical formulations and algorithms for fast and robust power system simulations. As a PhD candidate, he won the first prize at the PowerWeb PhD Poster contest in 2018 and received the SIAM student travel award at SIAM conference on Computational Science and Engineering (CSE19) in 2019. His research paper was selected in the best paper session at IEEE Power & Energy Society General Meeting in 2019. He is also a code contributor of Matpower which is free and open-source tools for electric power system simulation and optimization.

`http://baljaass.github.io/`

List of Scientific Activities

Journal Publications

- B. Sereeter, C. Vuik, and C. Witteveen, "Newton power flow methods for unbalanced three-phase distribution networks," *Energies*, vol. 10, no. 10, p. 1658, 2017.
- B. Sereeter, C. Vuik, and C. Witteveen, "On a comparison of Newton-Raphson solvers for power flow problems," *Journal of Computational and Applied Mathematics*, vol. 360, pp. 157–169, Nov 2019.
- B. Sereeter, W. van Westering, C. Vuik, and C. Witteveen. "Linear power flow method improved with numerical analysis techniques applied to a very large network," *Energies*, vol. 12, no. 21, p. 4078, 2019.
- B. Sereeter, A. Markensteijn, M. E. Kootte, C. Vuik, and C. Witteveen, "A novel linearized power flow approach for transmission and distribution networks" *IEEE PES Transaction on Power Systems*, Dec 2019. [Under review]

Peer-Reviewed Conference Proceedings

- B. Sereeter, C. Vuik, C. Witteveen, and P. Palensky, "Optimal power flow formulations and their impacts on the performance of solution methods," IEEE Power & Energy Society General Meeting, IEEE, Aug 2019. [Presented in best paper session]
- M. E. Kootte, B. Sereeter, J. E. Romate, and C. Vuik, "Comparison of numerical methods to solve the steady-state integrated transmission-distribution power flow problem," *IEEE PES Innovative Smart Grid Technologies Europe*, Oct 2020.[Under review]

Technical Reports

- B. Sereeter and R. D. Zimmerman, "*Addendum* to AC power flows and their derivatives using complex matrix notation: Nodal Current Balance," tech. rep., April 2018. MATPOWER Technical Note 3, Available: `https://matpower.org/docs/TN3-More-OPF-Derivatives.pdf`
- B. Sereeter and R. D. Zimmerman, "AC power flows and their derivatives using complex matrix notation and Cartesian coordinate voltages," tech. rep., April 2018. MATPOWER Technical Note 4, Available: `https://matpower.org/docs/TN4-OPF-Derivatives-Cartesian.pdf`.

Oral Presentations

- *Algorithms for Distributed Monitoring, Prediction and Control of Smart Grids*, Presented at: Centrum Wiskunde & Informatica, Amsterdam, The Netherlands, 2015-2019.
- *The Newton power flow method for unbalanced three-phase distribution networks*, Presented at: International conference Energy-Open, Enschede, The Netherlands, May 19, 2017.
- *Interior Point Methods for Optimal Power Flow Problems*, Presented at: Max-Plank Institute (MPI), Magdeburg, Germany, May 22, 2018.
- *Algorithms for Distributed Monitoring, Prediction and Control of Smart Grids*, Presented at: Mongolian Young Leadership Program, Ulaanbaatar, Mongolia, January 26, 2019.
- *Four mathematical formulations of the (Optimal) Power Flow problem and their impacts on the performance of solution methods*, Presented at: SIAM Conference on Computational Science and Engineering (CSE19), Spokane, United States, February 27, 2019. [SIAM Student Travel Award]
- *Fast and robust mathematical formulations of the Optimal Power Flow problem*, Presented at: IEEE Power & Energy Society (PES) General Meeting, Atlanta, United States, August 4-8, 2019.
- *Mathematical formulations and algorithms for fast and robust power system simulations*, Presented at: European Commission (Directorate General Joint Research Centre), Petten, The Netherlands, November 20, 2019.

Scientific Posters

- 40th-43rd Woudschoten conference on Scientific Computing, Dutch-Flemish Scientific Computing Society, Zeist, The Netherlands, 2015-2018.
- Computational Sciences for Future Energy conference, Utrecht, The Netherlands, October 11, 2016.
- PowerWeb conference, Delft, The Netherlands, June 2016 & 2018.
- PowerWeb PhD Poster contest, Delft, The Netherlands, November 14, 2018. [First place]
- PhD Energy Poster Event, Delft, The Netherlands, November 2, 2017.
- SES-BE annual meeting, Eindhoven, The Netherlands, June 2015-2019

Teaching experience at TU Delft

- AESB1310: Mathematics 3 (2016-2017)
- TB131A: Analyse en Differentiaalvergelijkingen (2018-2019)
- TB132A: Lineaire Algebra en Differentiaalvergelijkin (2019)

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor and promotor Kees Vuik for giving me the opportunity to do my PhD research work in his group at TU Delft. I am truly thankful for your support, guidance and encouragement. I have learned a lot from you. I am sincerely grateful to my second promotor Kees Witteveen who gave me many valuable comments, feedbacks and suggestions on my research work. Your insights and comments were very important to improve my research work. I am also thankful for NWO for funding this research project. The author greatly appreciates all committee members for reading my thesis and accepting the invitation.

I want to acknowledge Robert van Amerongen, Domenico Lahaye, and Milos Cvetkovic who took their time to answer my questions regarding the power system modelling and simulation in my first year of PhD. I highly appreciate Ray Zimmerman who allowed me to implement my methods in his power system simulation package Matpower. I want to thank Deborah Dongor for taking care of all the administrative tasks and Kees Lemmens for his support on any issues related to IT and Linux. It was nice meeting you Neil, Kristof, Martin, Fred, Matthias and Duncan.

I had successful collaborations with very smart PhD students: Werner van Westering, Anne Markensteijn, and Maria Kootte. I enjoyed working with you guys and am truly satisfied with the results we got. I am deeply grateful to Anne for translating the summary and propositions in my thesis into Dutch. I appreciate my cousin Batdorj Tsedev for making the cover picture of the dissertation.

I want to thank all my office mates: Reinaldo, Behrouz and Fahim for sharing your experiences with me and advising me on all PhD related matters. My special gratitude goes to Reinaldo who helped me whenever I need some support.

I am grateful to Kees Vuik and Berna Torun for being very supportive and generous when we initiate the Krylov Tigers. Thanks to you, DCSE decided to sponsor the team with the jersey. I want to thank my teammates for all the fun and laughter we shared: Gaby, Luis, Roel, Manuel, Shuaiqiang, Reinaldo, Dave, Mohamed, Soumya, Slawek, Keshav, Nirmal, and Tomas. It was fun playing with you guys.

I want to express my appreciation to all my former and current colleagues: Alice, Xiujie, Merel, Mousa, Vandana, Prajakta, Jochen, Menel, Jiao, Lisa, Thomas, Joost, Owen, Thea, Lin, Hugo and Luyu.

Last but not least, I would like to extend my deepest gratitude to my wife, Munkhzul, for her love, support, and patience. She also gave me the biggest gift in the world, our son Gerelt-Tulga. I want to thank my parents Sereeter Suren and Lkham-Osor Vandan for their endless love and also believing in me all the time, and parents-in-law Zorigt Mandshir and Lkhagvabayar Batjargal for taking care of my son while I was busy with my PhD work. I am lucky to have Sukhbat Sereeter as my little brother.

Baljinnyam Sereeter
Delft, March 18, 2020


$S_i^P = V_i^P \sum_{k=1}^{N_b} \sum_{q=a,b,c} (Y_{ik}^{Pq})^* (V_k^q)^*$
$\begin{bmatrix} \hat{I}_1 \\ 0 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{21}^T \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ \hat{V}_2 \end{bmatrix}$
$S_i = V_i \sum_{j=1}^{N_b} Y_{ij}^* V_j^*$
TUDelft